Lecture 6-c: Forecasting, Prediction and Model Selection

Brooks (4th edition): Chapter 5

1

(for private use, not to be posted/shared online)

Review - Functional Form: Structural Change

• We test if an event at that time T_{SB} affected our model, creating a "before" and an "after" in the parameters: That is,

$$y_i = \beta_0^1 + \beta_1^1 X_{1,i} + \beta_2^1 X_{2,i} + \beta_3^1 X_{3,i} + \varepsilon_i \qquad \text{for } i \leq T_{SB}$$
$$y_i = \beta_0^2 + \beta_1^2 X_{1,i} + \beta_2^2 X_{2,i} + \beta_3^2 X_{3,i} + \varepsilon_i \qquad \text{for } i > T_{SB}$$

The event caused *structural change* in the model.

• A Chow test, an F-test, tests if one model applies to both regimes:

$$y_i = \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \beta_3 X_{3,i} + \varepsilon_i \qquad \text{for all } i$$

• We test H_0 (No *structural change*): $\beta_0^1 = \beta_0^2 = \beta_0$
$\beta_1^1 = \beta_1^2 = \beta_1$
$\beta_2^1 = \beta_2^2 = \beta_2$
$\beta_3^1 = \beta_3^2 = \beta_3$

H_1 (*structural change*): For at least one k (= 0, 1, 2, 3): $\beta_k^1 \neq \beta_k^2$ 2

Review - Functional Form: Structural Change

• We structure the Chow test to test H_0 (No *structural change*), as usual.

• Steps for Chow (Structural Change) Test:

(1) Run OLS with all the data, with no distinction between regimes. (Restricted or pooled model). Keep RSS_R.

(2) Run two separate OLS, one for each regime (Unrestricted model):

Before Date T_{SB}. Keep RSS_1.

After Date T_{SB}. Keep RSS_2. $\Rightarrow RSS_U = RSS_1 + RSS_2$.

(3) Run a standard F-test (testing Restricted vs. Unrestricted models):

$$F = \frac{(RSS_R - RSS_U)/(k_U - k_R)}{(RSS_U)/(T - k_U)} = \frac{(RSS_R - [RSS_1 + RSS_2])/k}{(RSS_1 + RSS_2)/(T - 2k)}$$

3

Review - Functional Form: Structural Change

• Before, when we presented the Chow test, we used the F-distribution, which will be appropriate under (**A5**).

• In general, we rely on the asymptotic distribution –i.e., we do not rely on (**A5**). It is common to approximate the distribution of the Chow test, under H_0 and assuming a large number of observations pre- and post-break, with

$$J * F \xrightarrow{d} \chi^2_J \qquad \text{(sometimes written as } F \xrightarrow{d} \chi^2_J/J\text{).}$$

4

Functional Form: Structural Change

Example: 3 Factor Fama-French Model for SLB

Q: Did the financial crisis (Sep 2008, T_{SB} = **429**) affect the structure of the FF Model? Sample: January 1973 – December 2023 (T = 611).

Pooled RSS = **3.5290**

Jan 1973 – Sep 2008 RSS = RSS_1 = **2.0010** (T = 428)

Oct 2008 – Dec 2023 RSS = RSS_2 = **1.1213** (T = 183)

$$F = \frac{[RSS_R - (RSS_1 + RSS_2)]/J}{(RSS_1 + RSS_2)/(T-k)} = \frac{[3.5290 - (2.0010 + 1.1213)]/4}{(2.0010 + 1.1213)/(611 - 2*4)} = \mathbf{19.6356}$$

⇒ Since $F_{4,611,.05}$ = **2.39**, we reject H_0

	Constant	Mkt – rf	SMB	HML	RSS	T
1973-2020	-0.0073*	1.2138*	0.0123	0.4182*	**3.5290**	611
1973-2001	0.0013	0.9038*	-0.2394*	-0.3477*	**2.0010**	428
2002 – 2023	-0.0141*	1.3129*	0.3703	1.1496*	**1.1213**	183

5

Functional Form: Structural Change

Example (continuation): The R package *sctrucchange* estimates the Chow test. (As usual, you need to install package first.)

```
>x_slb <- SFX_da$SLB
>lr_slb <- log(x_slb[-1]/x_slb[-T])
>slb_x <- lr_slb - RF
>library(sctrucchange)
> t_s <- 428
> sctest(slb_x ~ Mkt_RF + SMB + HML, type = "Chow", point = t_s)

        Chow test

data:  slb_x ~ Mkt_RF + SMB + HML
F = 19.636, p-value = 3.331e-15
```

6

Review - Functional Form: Structural Change

Example: We test if the Oct 1973 oil shock in quarterly GDP growth rates had an structural change on the GDP growth rate model.

We model the GDP growth rate with an **AR(1) model**, that is, GDP growth rate depends only on its own lagged growth rate:

$$y_t = \beta_0 + \beta_1 \, y_{t-1} + \varepsilon_t$$

```
GDP_da <- read.csv("http://www.bauer.uh.edu/rsusmel/4397/GDP_q.csv", head=TRUE,
sep=",")
x_date <- GDP_da$DATE
x_gdp <- GDP_da$GDP
x_dummy <- GDP_da$D73
T <- length(x_gdp)
t_s <- 108                          # T_SB = Oct 1973

lr_gdp <- log(x_gdp[-1]/x_gdp[-T])
T <- length(lr_gdp)
lr_gdp0 <- lr_gdp[-1]
lr_gdp1 <- lr_gdp[-T]
t_s <- t_s -1                       # Adjust t_s (we lost the first observation)
```

7

Review - Functional Form: Structural Change

Example (continuation):

```
y <- lr_gdp0
x1 <-  lr_gdp1
T <- length(y)
x0 <- matrix(1,T,1)
x <- cbind(x0,x1)
k <- ncol(x)

# Restricted Model (Pooling all data)
fit_ar1 <- lm(lr_gdp0 ~ lr_gdp1)          # Fitting AR(1) (Restricted) Model
e_R <- fit_ar1$residuals                  # regression residuals, e
RSS_R <- sum(e_R^2)                       # RSS Restricted

> summary(fit_ar1)

Coefficients:
             Estimate   Std. Error  t value   Pr(>|t|)
(Intercept)  0.011411   0.001109    10.291    < 2e-16 ***
lr_gdp1      0.263353   0.055141    4.776     2.78e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01242 on 306 degrees of freedom
```

8

Review - Functional Form: Structural Change

Example (continuation):

```
# Unrestricted Model (Two regimes)

y_1 <- y[1:t_s]
x_u1 <- x[1:t_s,]
fit_ar1_1 <- lm(y_1 ~ x_u1 - 1)             # AR(1) Regime 1
e1 <- fit_ar1_1$residuals                   # Regime 1 regression residuals, e
RSS1 <- sum(e1^2)                           # RSS1: RSS Regime 1

kk = t_s+1                                  # Starting date for Regime 2
y_2 <- y[kk:T]
x_u2 <- x[kk:T,]
fit_ar1_2 <- lm(y_2 ~ x_u2 - 1)             # AR(1) Regime 2
e2 <- fit_ar1_2$residuals                   # Regime 2 regression residuals, e
RSS2 <- sum(e2^2)                           #  RSS2: RSS Regime 2

F <- ((RSS_R - (RSS1+RSS2))/k)/((RSS1+RSS2)/(T - 2*k))
> F
[1] 4.3382
p_val <- 1 - pf(F, df1 = 2, df2 = T - 2*k)  # p-value of F_test
> p_val
[1] 0.01388
```

⇒ small p-values: Reject H_0 (No *structural change*).

3 9

Review - Modeling Structural Change

- Under H_0 (No *structural change*), we pool the data into one model. That is, the parameters are the same under both regimes. We fit the same model for all t, for example:

$$y_t = \beta_0 + \beta_1\, y_{t-1} + \varepsilon_t$$

- If the Chow test rejects H_0, we need to reformulate the model. A typical reformulation includes a dummy variable ($D_{SB,t}$). For example, with vector $\boldsymbol{x}_t$ of explanatory variables:

$$y_t = \beta_0 + \boldsymbol{\beta}_1'\boldsymbol{x}_t + \beta_2\, D_{SB,t} + \boldsymbol{\gamma}_1'\boldsymbol{x}_t\, D_{SB,t} + \varepsilon_t$$

where

$D_{SB,t}$ = 1 if observation t occurred after T_{SB}
= 0 otherwise.

10

Review - Modeling Structural Change

Example: We are interested in introduce in the AR(1) model for GDP growth rates ($\boldsymbol{x}_t = y_{t-1}$), the effect of the Oct 1973 oil shock. We include a dummy variable in the AR(1) model, say $\mathbf{D}_{73}$:

$$D_{73,t} = 1 \text{ if observation } t \text{ occurred after October 1973}$$
$$= 0 \text{ otherwise.}$$

Then, $$y_t = \beta_0 + \beta_1\, y_{t-1} + \gamma_0\, D_{73,t} + \gamma_1\, y_{t-1} * D_{73,t} + \varepsilon_t$$

In the model, the oil shock affects the constant and the slopes.

	Constant	Slopes:
Before oil shock ($\mathbf{D}_{73} = 0$):	β_0	β_1
After oil shock ($\mathbf{D}_{73} = 1$) :	$\beta_0 + \gamma_0$	$\beta_1 + \gamma_1$

• We estimate the above model and perform an F-test to test if H_0 (No *structural change*): $\gamma_0 = 0$ & $\gamma_1 = 0$.

11

0.003779 0.002349 1.609

Review - Modeling Structural Change

Example: We add an Oct 1973 dummy in the **AR(1) GDP model**.

```
T1 <- T - t_s                        # Number of Observations after SB
D73_0 <- rep(0,t_s)                  # Dummy_t = 0 if t <= t_s
D73_1 <- rep(1,T1)                   # Dummy_t = 1 of t > t_s
D73 <- c(D73_0,D73_1)                # SB Dummy variable t_s <- 108
lr_gdp1_D73 <- lr_gdp1 * D73         # interactive dummy (effect on slope)
fit_ar1_d_2 <- lm(lr_gdp0 ~ lr_gdp1 + D73 + lr_gdp1_D73)
summary(fit_ar1_d_2)

Coefficients:
              Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)   0.008953  0.001935    4.627    5.5e-06 ***
lr_gdp1       0.467457  0.090623    5.158    4.5e-07 ***
D73           0.003779  0.002349    1.609    0.1086        ⇒ no significant effect on constant
lr_gdp1_D73  -0.326809  0.113592   -2.877    0.0043 **     ⇒ significant effect of oil shock on slope.
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Conclusion: After the oil shock the slope significantly changed from 0.467457 to **0.140648** (= 0.467457 + (-0.326809)).

12

Review - Modeling Structural Change

Example (continuation): Using the car package, we test the joint hypothesis H_0 (No *structural change*): $\gamma_0 = 0$ & $\gamma_1 = 0$.

```
> library(car)
> linearHypothesis(fit_ar1_d_2, c("D73 = 0","lr_gdp1_D73 = 0"), test="F")
Linear hypothesis test

Hypothesis:
D73 = 0
lr_gdp1_D73 = 0

Model 1: restricted model
Model 2: lr_gdp0 ~ lr_gdp1 + D73 + lr_gdp1_D73

  Res.Df      RSS Df Sum of Sq      F  Pr(>F)
1    306 0.047201
2    304 0.045806  2 0.0013957 4.6316 0.01044 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

13

Review - Functional Form: Structural Change

• It is also possible to do a Wald test to test H_0, using only the unrestricted estimators. Steps:

1) Run two separate OLS, one for each regime (Unrestricted model):

Before Date T_{SB}. Keep $\mathbf{b}_1$ & Var[$\mathbf{b}_1$]

After Date T_{SB}. Keep $\mathbf{b}_2$ & Var[$\mathbf{b}_2$]

2) Compute the Wald test:

$$W = (\mathbf{b}_1 - \mathbf{b}_2)' \{\text{Var}[\mathbf{b}_1] + \text{Var}[\mathbf{b}_2]\}^{-1} (\mathbf{b}_1 - \mathbf{b}_2),$$

where $\text{Var}[\mathbf{b}_1 - \mathbf{b}_2] = \text{Var}[\mathbf{b}_1] + \text{Var}[\mathbf{b}_2]$, assuming $\text{Cov}(\mathbf{b}_1, \mathbf{b}_2) = 0$.

Under H_0 (& if the number of observations pre- and post-break are large), the Wald test follows: $W \xrightarrow{d} \chi^2_J$

14

Review – Chow Test with Unknown Break

• Problem with the (classic) Chow test: We condition on a specific date for the structural break (say, October 73), which in general is unknown.

• An easy solution? Run a Chow test for all possible dates, select the date, τ, that maximizes the Chow test.

$$Rolling\ Chow\ Test = \max_{\tau \epsilon \{\tau_{min}, \dots, \tau_{max}\}} F_T(\tau)$$

• We cannot run the Chow test for all dates. We need enough observations to run OLS on both sides of the potential breaking points. Then, we start to check for a breaking point at date τ_{min}, and we finish to check at date τ_{max}.

• This is called, "trimming" the data. Usually, we set τ_{min} and τ_{max} by leaving a percentage, π, of the initial of observations and final observations. Usually, π = 10% or 15%. 3

Review – Chow Test with Unknown Break

• This test was proposed by Quandt (1958):

$$QLR_T = \max_{\tau \epsilon \{\tau_{min}, \dots, \tau_{max}\}} F_T(\tau)$$

• It is also possible to run the Wald test version of the Chow test for all possible dates, again, selecting the date that maximizes

$$QLR_T = \max_{\tau \epsilon \{\tau_{min}, \dots, \tau_{max}\}} W_T(\tau)$$

• The first QLR_T is called the **SupF** test, the second the **SupW**.

Technical Problem: With this approach, the technical conditions under which the asymptotic distribution is derived are not met in this setting.

• Andrews (1993) showed that under appropriate conditions, the QLR statistic, also known as Sup-test (F, W, LR) statistic, has a *non-standard limiting distribution* ("*non-standard*" = no existing table; needs a new one).

Review – Chow Test with Unknown Break

• The QLR statistic, also known as **Sup-test** (F, W, LR) statistic, has a *non-standard limiting distribution.*

• The distribution depends on the number of parameters of the model, k, which are tested for stability, trimming values, π_0, which only affect the distribution through $\lambda = (1-\pi_0)^2/\pi_0^2$.

• Andrews (1993) tabulated the non-standard distribution of the SupW for different k, α, and trimming values (π_0).

<u>Note</u>: It is usual to test the **SupF**, using the critical values of **SupW**, by dividing the **SupW** critical values by k. Andrews (2003) issued a slightly corrected Table. In the next slide, we present the SupF critical values for π_0 = 15%, taken from Stock and Watson (2011).

3

Review – Chow Test with Unknown Break

Critical values of the QLR test distribution, from Stock and Watson (2011), setting π_0 = 15%; for α = 1%, 5%, 10%; and different k $(= q)$.

TABLE 14.6 Critical Values of the QLR Statistic with 15% Trimming

Number of Restrictions (q)	10%	5%	1%
1	7.12	8.68	12.16
2	5.00	5.86	7.78
3	4.09	4.71	6.02
4	3.59	4.09	5.12
5	3.26	3.66	4.53
6	3.02	3.37	4.12
7	2.84	3.15	3.82
8	2.69	2.98	3.57
9	2.58	2.84	3.38
10	2.48	2.71	3.23

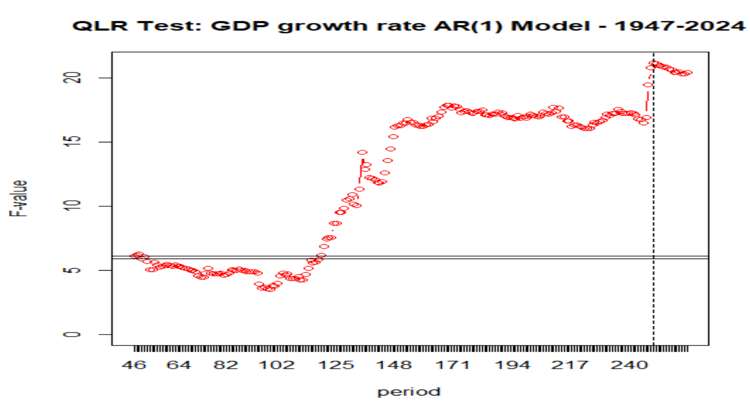
Review – Chow Test with Unknown Break

Example (continuation): We search for breaking points for GDP growth rate in AR(1) model. We use package *desk*. (You can use library *strucchange*, but it runs the SupW (F= SupW/2), see Andrews (1993.))

```
library(desk)
pie <- .15
T0 <- round(T * pie)
T1 <- round(T *(1-pie))
my.qlr <- qlr.test(lr_gdp0 ~ lr_gdp1, from = T0, to = T1, sig.level = 0.05, details = TRUE)
> my.qlr              # Print test results

QLR-Test for structural breaks at unknown date

Hypotheses:
                   H0:                        H1:
 No break in t = 46...262  Some break in t = 46...262

Test results:
 f.value  lower.cv  upper.cv  p.value  sig.level   H0
 21.1441      5.86     6.085  < 1e-04       0.05   rej.
```

Review – Chow Test with Unknown Break

Example (continuation): We search for breaking points for GDP growth rate in AR(1) model. Below, we plot all F-tests starting at T*15:



- Maximum F is **21.14** occurs in **Jan 2009** (observation #250). Then, $\widehat{QLR}$ = **21.14** > **5.86** ⇒ Reject H_0 at 5% level & break is not Oct 73!.

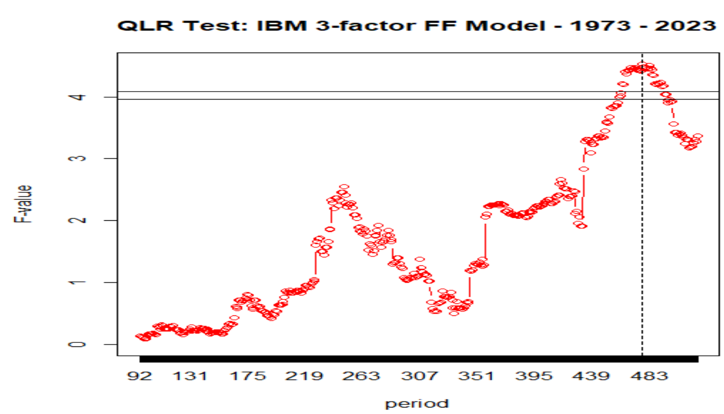
Review – Chow Test with Unknown Break

Example: We search for breaking points for IBM returns in the 3-factor FF model.

```
T<- length(ibm_x)
pie <- .15
T0 <- round(T * pie)
T1 <- round(T *(1-pie))
my.qlr <- qlr.test(fit_ibm_ff3, from = T0, to = T1, sig.level = 0.05, details = TRUE)
> my.qlr # Print test results

QLR-Test for structural breaks at unknown date
------------------------------------------------

Hypotheses:
                H0:                    H1:
 No break in t = 92...519  Some break in t = 92...519

Test results:
 f.value  lower.cv  upper.cv  p.value  sig.level   H0
 4.5302      3.96      4.09   0.0243      0.05     rej.
```

Review – Chow Test with Unknown Break

Example: We search for breaking points for IBM returns in the 3-factor FF model. Below, we plot all F-tests starting at T*15:



- Maximum F is **4.5302** occurs in **Sep 2012** (observation #477). Then, $\widehat{QLR}$ = **4.5302** > **4.09** $\Rightarrow$ Reject H_0 at 5% level.

Chow Test: Structural Change – Forecasting

- Ignoring structural change affects forecasts.

Example: We want to forecast GDP growth for the 2nd quarter of 2024 (2024: II) using the estimated AR(1) models. We have data from 1947:II to 2024:I. We know the GDP growth in 2024:I"

$y_{t=2024:I}$ (GDP growth 2024:I) = **0.01362407**

Under no structural change, the 2024:II forecasts is:

$\hat{y}_{t=2024:II}$ = 0.011411 + 0.263353 $y_{t=2024:I}$
= 0.011411 + 0.263353 * **0.01362407** = **0.014999**

Under structural change, the 2024:II forecasts is:

$\hat{y}_{t=2024:II}$ = (0.008953 + 0.003779) + (0.467457 - 0.326809) $y_{t=2024:I}$
= 0.012732 + 0.140648 * **0.01362407** = **0.014648**

3

Chow Test: Structural Change – Remarks

- The results are *conditional* on the breaking point –say, **October 73**, **Dec 2001**, or **January 2009**.

- The breaking point is usually unknown. It needs to be estimated.

- It can deal only with one structural break –i.e., two categories!

- The number of breaks is also unknown. They need to be estimated.

- Characteristics of the data (heteroscedasticity –for example, regimes in the variance- and unit roots (high persistence) complicate the test.

- Missing structural breaks in deterministic parameters (intercepts, trends, etc.) can be a cause of forecast failure.

- There are many modern tests that take care of these issues, but usually also with *non-standard* distributions.

3

Forecasting and Prediction

"There are two kind of forecasters: those who don't know and those who don't know they don't know"

John Kenneth Galbraith (1993)

- Objective: Forecast
- Distinction: Ex post vs. Ex ante forecasting
  - – Ex post: RHS data are observed
  - – Ex ante (true forecasting): RHS data must be forecasted

- Prediction and Forecast

Prediction: Explaining an outcome, which could be a future outcome.

Forecast: A particular prediction, focusing in a future outcome.

Example: Prediction: Given $\boldsymbol{x}^0$ $\Rightarrow$ predict y^0.

Forecast: Given $\boldsymbol{x}_{t+1}^0$ $\Rightarrow$ predict y_{t+1}.

Forecasting and Prediction: Types

- Two types of predictions:

- *In sample* (**prediction**): The expected value of $\boldsymbol{y}$ (in-sample), given the estimates of the parameters. The in-sample prediction are the fitted values, $\hat{\boldsymbol{y}}$.

- *Out of sample* (**forecasting**): The value of a future $\boldsymbol{y}$ that is not observed by the sample.

Notation & Terminology:

Let T be the forecast origin and l is the forecast horizon.

- Prediction for T made at T: $\hat{Y}_T$.
- Forecast for $T+l$ made at T: $\hat{Y}_{T+l}$, $\hat{Y}_{T+l|T}$, $\hat{Y}_T(l)$.
- $\hat{Y}_T(l)$: l-*step ahead* forecast = Forecasted value Y_{T+l} at time T.

Forecasting and Prediction: Information Set, I_T

• Any prediction or forecast needs an **information set**, I_T. This includes data, models and/or assumptions available at time *T*. The predictions and forecasts will be conditional on I_T.

For example, in-sample, $I_T = \{\boldsymbol{x}^0\}$ to predict $\boldsymbol{y}^0$ (= $\mathbf{b}\,\boldsymbol{x}^0$).
Or in time series, $I_T = \{\boldsymbol{x}^0_{T-1}, \boldsymbol{x}^0_{T-2}, ..., \boldsymbol{x}^0_{T-q}\}$ to predict y_{T+l}.

• Then, the forecast is just the conditional expectation of Y_{T+l}, given the observed sample:

$$\hat{Y}_{T+l} = E[Y_{T+l}\,|\,X_T, X_{T-1}, \dots, X_1]$$

Example: If $X_T = Y_T$, that is, we have an autoregressive model, using only the past of Y_T to predict Y_{T+l}, then, one-step ahead forecast is:

$$\hat{Y}_{T+1} = E[Y_{T+1}\,|\,Y_T, Y_{T-1}, \dots, Y_1]$$

Forecasting and Prediction: Conditional E[.]

• The conditional expectation of Y_{T+l} is, in general, based on a model, the experience of the forecaster or a combination of both.

Example: We base the conditional expectation on the 3-factor FF model:

$$\hat{Y}_{T+l} = \mathrm{E}[(\beta_0 + \beta_1\,(r_{m,T+l} - r_f) + \beta_2\,SMB_{T+l} + \beta_3\,HML_{T+l})\,|\,I_T]$$

• In the above equation, the forecast of Y_{T+l} also needs a forecast for the driving variables in the model. That is, we need a forecast for:

- $\mathrm{E}[(r_{m,T+l} - r_f)\ |\,I_T\,]$
- $\mathrm{E}[SMB_{T+l}\ |\,I_T]$
- $\mathrm{E}[HMM_{T+l}\,|\,I_T\,]$

• In general, we will need a model for $\hat{X}_{T+l}$. Things can get complicated very quickly.

Forecasting and Prediction: Forecasts are RV

• Keep in mind that the **forecasts** are a **random variable**. Technically speaking, they can be fully characterized by a pdf.

• In general, it is difficult to get the pdf for the forecast. In practice, we get a **point estimate** (the forecast) and a **C.I**. to gauge the uncertainty in the forecast.

• Q: What is a good forecast? We need metrics to evaluate the forecasting performance of different models.

• In general, the evaluation of forecasts relies on MSE.

Note: Later in this class, when we cover time series (Brooks, Chapter 6), we go deeper into forecasting.

Forecasting and Prediction: Variance-bias

• We start with general model (DGP):

(**A1**) DGP: $\boldsymbol{y} = f(\mathbf{X}, \theta) + \boldsymbol{\varepsilon}$.

• Given $\boldsymbol{x}^0$, we predict y^0, using the expectation: $\mathrm{E}[y|\mathbf{X}, \boldsymbol{x}^0] = f(\boldsymbol{x}^0, \theta)$

• We estimate $\mathrm{E}[y|\mathbf{X}, \boldsymbol{x}^0]$ with $\hat{y}^0 = f(\boldsymbol{x}^0, \hat{\theta})$.

• The realization y^0 is just: $y^0 = f(\boldsymbol{x}^0, \theta) + \varepsilon^0$

• With y^0 observed, we compute the prediction error,

$$e^0 = \hat{y}^0 - y^0 = f(\boldsymbol{x}^0, \hat{\theta}) - f(\boldsymbol{x}^0, \theta) - \varepsilon^0$$

• The associated expected squared prediction error can be written as:

$$\mathrm{E}[(e^0)^2] = \mathrm{E}[(\hat{y}^0 - y^0)^2] = \mathrm{Var}[\hat{y}^0] + [\mathrm{Bias}(\hat{y}^0)]^2 + \mathbf{Var}[\boldsymbol{\varepsilon}]$$

• We want to minimize this squared error, $\mathrm{E}[(e^0)^2]$.

Forecasting and Prediction: Variance-bias

• The associated expected squared prediction error can be written as:

$$E[(e^0)^2] = \text{Var}[\hat{y}^0] + [\text{Bias}(\hat{y}^0)]^2 + \textbf{Var}[\boldsymbol{\varepsilon}]$$

• We want to minimize this squared error. Note that there is nothing a forecaster can do regarding the last term, called the *irreducible error*.

• Then, all efforts are devoted to minimize the sum of a variance and a squared bias. This creates the *variance-bias trade-off* in forecasting.

• It is possible that a biased forecast can produce a lower MSE than an unbiased one. In this lecture, we based our forecasts on OLS estimates, which under CLM assumptions, produce unbiased forecasts.

Note: The variance-bias trade-off is always present in forecasting. In general, more flexible models have less bias and more variance. The key is to pick an "*optimal*" mix of both.

Prediction Intervals: Point Estimate

• Prediction: Given $\boldsymbol{x}^0 \Rightarrow$ predict y^0.

• Given the CLM, we have:

Expectation: $E[y|\mathbf{X}, \boldsymbol{x}^0] = \boldsymbol{\beta}' \boldsymbol{x}^0$;

Predictor: $\hat{y}^0 = \mathbf{b}' \boldsymbol{x}^0$

Realization: $y^0 = \boldsymbol{\beta}' \boldsymbol{x}^0 + \varepsilon^0$

Note: The predictor includes an estimate of ε^0:

$\hat{y}^0 = \mathbf{b}' \boldsymbol{x}^0$ + estimate of ε^0. (Estimate of ε^0=0, but with variance.)

• Associated with $\hat{y}^0$ (a point estimate), there is a forecast error, e^0:

$$e^0 = \hat{y}^0 - y^0 = \mathbf{b}'\boldsymbol{x}^0 - \boldsymbol{\beta}'\boldsymbol{x}^0 - \varepsilon^0 = (\mathbf{b} - \boldsymbol{\beta})'\boldsymbol{x}^0 - \varepsilon^0$$

and a variance

$$\Rightarrow \text{Var}[(\hat{y}^0 - y^0)|\boldsymbol{x}^0] = E[(\hat{y}^0 - y^0)'(\hat{y}^0 - y^0)|\boldsymbol{x}^0]$$

$$\text{Var}[e^0|\boldsymbol{x}^0] = \boldsymbol{x}^{0\prime}\, \text{Var}[(\mathbf{b} - \boldsymbol{\beta})|\boldsymbol{x}^0]\, \boldsymbol{x}^0 + \sigma^2$$

Prediction Intervals: Point Estimate

Example: We estimated the 3 Factor FF Model for IBM returns:

> summary(fit_ibm_ff3)

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.005089	0.002488	-2.046	0.0412 *
Mkt_RF	0.908299	0.056722	16.013	<2e-16 ***
SMB	-0.212460	0.084112	-2.526	0.0118 *
HML	-0.171500	0.084682	-2.025	0.0433 *

Suppose we are given $\boldsymbol{x}^0$ = [1.0000 -0.0189 -0.0142 -0.0027]

Then,

$\hat{y}^0$ = -0.005089 + 0.908299 * (-0.0189) - 0.212460 * (-0.0142)

\- - 0.171500 * (-0.0027) = **-0.01877582**

Suppose we observe $\mathbf{y^0}$ = **0.1555214**. Then, the forecast error is

$\hat{y}^0 - y^0$ = **-0.01877582** – **0.1555214** = -0.1742973

Prediction Intervals: Point Estimate

Example (continuation): In R:

```
b_ibm <- fit_ibm_ff3$coefficients                # regression coefficients, b
x_0 <- rbind(1.0000, -0.0189, -0.0142, -0.0027)   # x^0
 y_0 <- 0.1555214
y_f0 <-  t(b_ibm)%*% x_0
> y_f0
          [,1]
[1,] -0.01877582
ef_0 <- y_f0 - y_0
> ef_0
          [,1]
[1,] -0.1742973
```

Prediction Intervals: C.I.

- We estimate the uncertainty behind the forecast with the Var$[e^0]$.

Two cases:
(1) If $\boldsymbol{x}^0$ is given –i.e., constants. Then,

$$\text{Var}[e^0] = \boldsymbol{x}^{0\prime}\ \text{Var}[\mathbf{b}\,|\,\boldsymbol{x}^0]\ \boldsymbol{x}^0 + \sigma^2$$

$\Rightarrow$ Form a (1- α)% C.I. as usual.

$$[\hat{y}^0 \pm t_{T\text{-}k,1\text{-}\alpha/2} * \text{sqrt}(\text{Var}[e^0])]$$

Note: In out-of-sample forecasting, usually, $\boldsymbol{x}^0$ is unknown, it has to be estimated.

(2) If $\boldsymbol{x}^0$ has to be estimated, then we use a random variable. The C.I. becomes more complicated. A bootstrap can be used.

Prediction Intervals: C.I. and Forecast Variance

- Assuming $\boldsymbol{x}^0$ is known, the variance of the forecast error is

$$\sigma^2 + \boldsymbol{x}^{0\prime}\ \text{Var}[\mathbf{b}\,|\,\boldsymbol{x}^0]\ \boldsymbol{x}^0 = \sigma^2 + \sigma^2[\boldsymbol{x}^{0\prime}\ (\mathbf{X}'\mathbf{X})^{-1}\boldsymbol{x}^0]$$

If the model contains a constant term, this is

$$\text{Var}[e^0] = \sigma^2\left[1 + \frac{1}{N} + \sum_{j=1}^{K-1}\sum_{k=1}^{K-1}(x_j^0 - \bar{x}_j)(x_k^0 - \bar{x}_k)(\text{Z}'\text{M}^0\text{Z})^{jk}\right]$$

(where $\mathbf{Z}$ is $\mathbf{X}$ without $\mathbf{x}_1 = \mathbf{i}$). In terms squares and cross products of deviations from means.

Note: Large σ^2, small N, and large deviations from the means, decrease the precision of the forecasting error.

Interpretation: Forecast variance is smallest in the middle of our "experience" and increases as we move outside it.

Prediction Intervals: C.I. and Forecast Variance

• Then, the $(1-\alpha)\%$ C.I. is given by: $[\hat{y}^0 \pm t_{T\text{-}k,\alpha/2} * \text{sqrt}(\text{Var}[e^0])]$

• As $\boldsymbol{x}^0$ moves away from its mean, the C.I increases, this is known as the "*butterfly effect*."

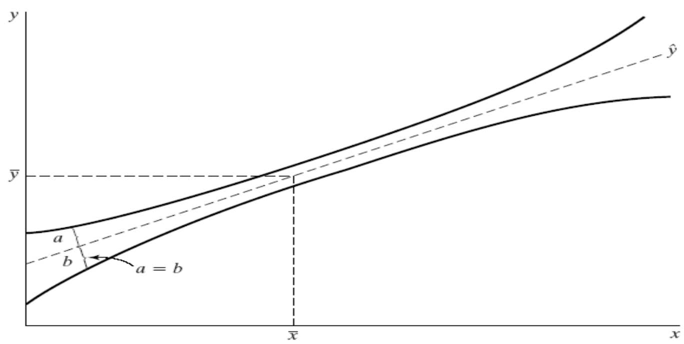


FIGURE 6.1 Prediction Intervals.

Prediction Intervals: C.I. and Forecast Variance

Example (continuation): We want to calculate the variance of the forecast error: for thee given $\boldsymbol{x}^0$ = [1.0000 -0.0189 -0.0142 -0.0027]
Recall we got $\hat{y}^0 = \mathbf{b}' \boldsymbol{x}^0$ = **-0.01877587**

Then,

Estimated $\text{Var}[e^0 | \boldsymbol{x}^0] = \boldsymbol{x}^{0\prime} \text{Var}[\mathbf{b} | \boldsymbol{x}^0]\, \boldsymbol{x}^0 + s^2$ = **0.003429632**

```
Var_b <- vcov(fit_ibm_ff3)
var_ef_0 <- t(x_0) %*% Var_b %*% x_0 + Sigma2
> var_ef_0
          [,1]
[1,] 0.003429632
> sqrt(var_ef_0)
          [,1]
[1,] 0.05856306
```

Check: What is the forecast error if $\boldsymbol{x}^0$ = colMeans(x)?

Prediction Intervals: C.I. and Forecast Variance

Example (continuation):

(1-alpha)% C.I. for prediction (alpha = .05)

```
CI_lb <- y_f0 – 1.96 * sqrt(var_ef_0)
> CI_lb
>[1] -0.1335594

CI_ub <- y_f0 + 1.96 * sqrt(var_ef_0)
>CI_ub
>[1] 0.09600778
```

That is, CI for prediction: [**-0.13356**; **0.09601**] with 95% confidence. A wide interval, which makes clear the uncertainty surrounding the point forecast: $\hat{y}^0$ = **-0.01877587**

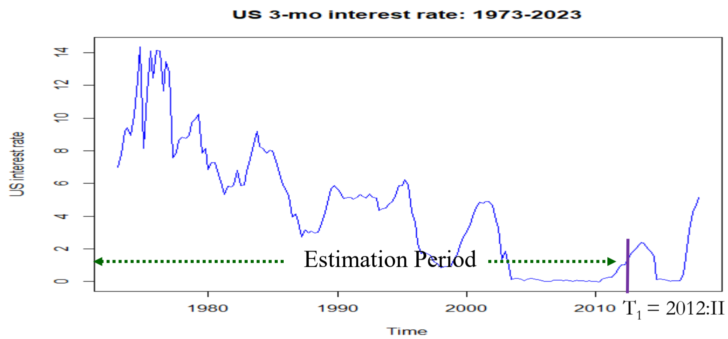
Forecasting and Prediction: Model Validation

• *Model validation* refers to establishing the statistical adequacy of the assumptions behind the model –i.e., (**A1**)-(**A5**) in this lecture. Predictive power or forecast accuracy can be used to do model validation.

• In the context of prediction and forecasting, model validation is done by fitting a model in-sample, but keeping a small part of the sample, the *hold-out-sample*, to check the accuracy of OOS forecasts.

• **Hold out sample**: We estimate the model using only a part of the sample (say, up to time T_1). The rest of the observations, the hold out sample, ($T - T_1$ observations) are used to check the predictive power of the model –i.e., the accuracy of predictions, by comparing $\hat{y}^0$ with actual y^0.

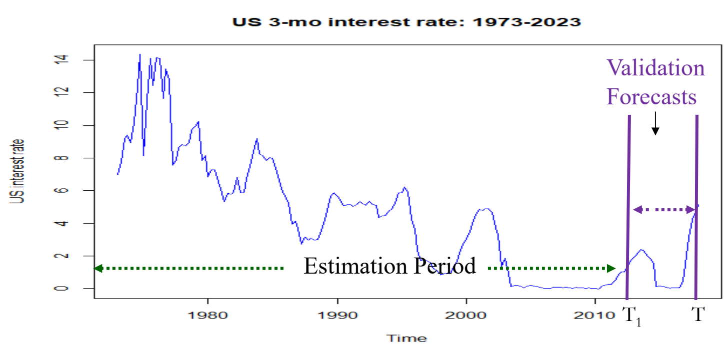
Review: Forecasting - Model Validation



• For model validation, we keep a small part of the sample for checking the forecasting skills (or accuracy) of the model. Steps:

Step 1. Estimate the model using all the observation up to T_1 (above from 1973:I to 2012:II). The period used is called "**estimation period** or **estimation sample**." (Get in-sample forecasts, $\hat{y}$.)

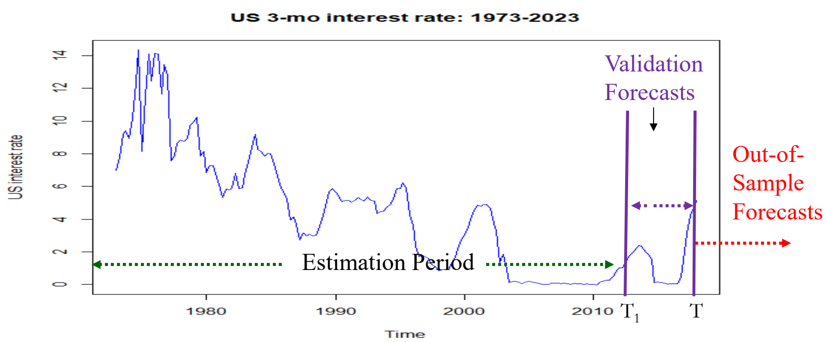
Review: Forecasting - Model Validation



Step 2. Keep a (short) part of the sample, $(T - T_1)$, to check the model's forecasting skills. Using the estimates from **Step 1**, we produce forecasts, $\hat{y}$, for the period $(T - T_1)$. Since in the period $(T - T_1)$ we know y, we can compute true MSE or MAE. This is the ***validation step***.

For example, we compute: $\text{MSE} = \frac{1}{(T-T_1)} \sum_{i=T_1+1}^{T} (\hat{y}_i - y_i)^2$

Review: Forecasting - Model Validation



Step 3. If happy with **Step 2**, we proceed to do true out-of-sample forecasts. In general, for the out-of-sample forecast, we re-estimate the model using all the sample –i.e., all T observations.

To evaluate the true OOS forecasts, we have to wait, say m periods, to compute an MSE : MSE = $\frac{1}{m}\sum_{i=T+1}^{m}(\hat{y}_i - y_i)^2$

Forecasting and Prediction: Model Validation

Details:

1) ***Estimation period***. Use the first T_1 observations to estimate the parameters of the model. Get in-sample forecasts, $\hat{y}$. In-sample evaluation of model (R^2, t- & F-tests) is usually performed here.

2) ***Validation period***. Use $(T - T_1)$ observations to check the model's forecasting skills. Given estimates in (1), OLS **b**, & using $\boldsymbol{x}^0$, get OSS $\hat{y}^0$ = $\mathbf{b}'\,\boldsymbol{x}^0$. Since y^0 is known, calculate true MSE or MAE. For example:

$$MSE = \frac{1}{(T-T_1)}\sum_{i=(T_1+1)}^{(T-T_1)}(\hat{y}_i^0 - y_i^0)^2$$

Note: It is common to set $(T - T_1)$ close to 10% of sample.

3) ***True OOS forecast period***. Re-estimate model Produce OSS $\hat{y}^0$, but since y^0 is not known now, it will take time to evaluate the true OOS forecasts.

Forecasting and Prediction: Model Validation

Note: In the **Machine Learning** literature, the terminology used for model validation is slightly different.

Step 1 is called "*training*," the data used (say, first T_1 observations) are called *training data/set*. In this step, we fit *(train)* the model, subject to the assumptions, for example, (**A1**)-(**A4**).

Step 2 has the same name, *validation (or "single-split" validation)*. This step can be used to "*tune (hyper-)parameters.*" In our CLM, we can "tune" the model for departures of (**A1**)-(**A4**), for example, by including more or different variables (**A1**) and re-estimating the model accordingly using "training data" alone. We choose the model with lower MSE or MAE.

Remark: The idea of this step is to **simulate** out-of-sample accuracy. But, the "tuned" parameters selected in Step 2 are fed back to Step 1.

Step 3 *tests* the true out-of-sample forecast accuracy of model selected by **Step 1** & **Step 2**. This last part of the sample is called "*testing sample*."

Forecasting and Prediction: Cross Validation

• **Step 2** is used as a testing ground of the model before performing OOS forecasting. There are many ways to approach the validation step.

• Instead of a single split, split the data in K parts. This is called *K-fold cross-validation*. For j = 1, 2, …, K, use all folds but fold j to estimate model; use fold j to check model's forecasting skills by computing MSE, MSE_j. The K-fold CV estimate is an average of each fold MSE's:

$$CV_K = \frac{1}{K}\sum_{j=1}^{K} MSE_j$$

Usual choices for K are 5 & 10. (These are arbitrary choices.)

Random and non-random splits of data can be used. The non-random splits are used for some special cases, such as qualitative data, to make sure the splits are "representative."

Forecasting and Prediction: Cross Validation

• Use a single observation for validation. This is called *leave-one-out cross-validation* (LOOCV). A special case of K-*fold cross-validation* with K = T. That is, use (T - 1) observations for estimation, and, then, use the observation left out, i = 1, …, T, to compute $MSE_{(-i)}$, which is just $(\hat{y}_{(-i)} - y_i)^2$, where $\hat{y}_{(-i)}$ is the prediction for observation i based on the full sample but observation i. Then, compute:

$$CV_n = \frac{1}{n}\sum_{i=1}^{n} MSE_{(-i)}$$

• Instead of just one, it is possible to leave p observations for validation. This is called *leave-p-out cross-validation* (LpOCV).

Remark: In time series, since the order of the data matters, cross validation is more complicated. In general, rolling windows are used.

Forecasting and Prediction: Cross Validation

Example: We do cross-validation on the 3-factor Fama-French Model for IBM returns with K = 5:

```
y <- ibm_x
ff_cv_data <- data.frame(Mkt_RF, SMB, HML)
###### CV: Cross-Validation K-fold Code Function ######
CV<- function(dats, n.folds){
 folds <- list()                   # flexible object for storing folds
 fold.size <- nrow(dats)/n.folds
 remain <- 1:nrow(dats)            # all obs are in
 for (i in 1:n.folds){
  select <- sample(remain, fold.size, replace = FALSE)    #randomly sample fold_size from remaining obs)
  folds[[i]] <- select             # store indices ( write a special statement for last fold if 'leftover points')
    if (i == n.folds){
    folds[[i]] <- remain
  }
remain <- setdiff(remain, select)    #update remaining indices to reflect what was taken out
remain
 }
```

Forecasting and Prediction: Cross Validation

Example (continuation):

```
 results <- matrix(0,1,n.folds)

 for (i in 1:n.folds){
  # fold i
  indis <- folds[[i]]                          #unpack into a vector
  estim <- dats[-indis, ]                      #split into estimation (train) & validation (test) sets
  test <- dats[indis, ]

  lm.model <- lm(y[-indis] ~ ., data = estim)  # OLS with estimation data
  pred <- predict(lm.model, newdata = test)    # predicted values for fold not used
  MSE <- mean((y[indis] - pred)^2)             # MSE (any other evaluation measure can be used)
  results[[i]]<- MSE                           # Accumulate MSE in vector
 }
 return(results)
}

CV_ff_5 <- CV(ff_cv_data, 5)
> mean(CV_ff_5)
[1] 0.00346262
```

Evaluation of Forecasts: Measures of Accuracy

- Popular measures of OOS forecast accuracy, after m forecasts:

Mean Absolute Error (MAE) $= \frac{1}{m}\sum_{i=T+1}^{T+m} |\hat{y}_i - y_i| = \frac{1}{m}\sum_{i=T+1}^{T+m} |e_i|$

Mean Squared Error (MSE) $= \frac{1}{m}\sum_{i=T+1}^{T+m} (\hat{y}_i - y_i)^2 = \frac{1}{m}\sum_{i=T+1}^{T+m} {e_i}^2$

Root Mean Square Error (RMSE) $= \sqrt{\frac{1}{m}\sum_{i=T+1}^{T+m} {e_i}^2}$

Mean Absolute Percentage Error (MAPE) $= \frac{1}{m}\sum_{i=T+1}^{T+m} |\frac{\hat{y}_i - y_i}{y_i} * 100|$

Theil's U statistics $= \dfrac{\sqrt{\frac{1}{m}\sum_{i=T+1}^{T+m} {e_i}^2}}{\sqrt{\frac{1}{T}\sum_{i=1}^{T} {y_i}^2}}$

Evaluation of Forecasts: Measures of Accuracy

• Theil's U statistics has the interpretation of an R^2. But, it is not restricted to be smaller than 1.

• An OOS R^2 can be computed as:

$$R^2_{OOS} = 1 - \frac{MSE_A}{MSE_N}$$

with $MSE_A = \sum_{t=1}^{m}(y_{t+\tau} - \hat{y}_{t+\tau})^2$

$MSE_N = \sum_{t=1}^{m}(y_{t+\tau} - \bar{y}_t)^2$

where τ is the forecasting horizon. (See Goyal and Welch (2008) for a well-known finance application.)

• Again, cross-validation measures can be used to evaluate forecasting performance.

Evaluation of Forecasts: Measures of Accuracy

Example: We want to check the forecast accuracy of the 3 FF Factor Model for IBM returns. We estimate the model using only 1973 to 2017 data (T=539), leaving **2018-2020** (***m*** = **30** observations) for validation of predictions.

```
T0 <- 1
T1 <- 539                          # End of Estimation Period (Dec 2017)
T2 <- T1+1                         # Start of Validation Period (Jan 2018)
y1 <- y[T0:T1]
x1 <- x[T0:T1,]
fit_ibm_2 <- lm(y1~ x1 - 1)        # Estimation Period Regression From T0 to T1
b1 <- fit_ibm_2$coefficients       # Extract OLS coefficients from regression
> summary(fit_ibm_2)
            Estimate Std. Error t value Pr(>|t|)
x1          -0.003848   0.002571  -1.497  0.13510
x1Mkt_RF     0.865579   0.059386  14.575  < 2e-16 ***
x1SMB       -0.224914   0.085505  -2.630  0.00877 **
x1HML       -0.230838   0.090251  -2.558  0.01081 *
```


Evaluation of Forecasts: Measures of Accuracy

Example (continuation): We condition on the observed data (no model to predict FF factors used) from 2018: Jan to 2020: Jun.

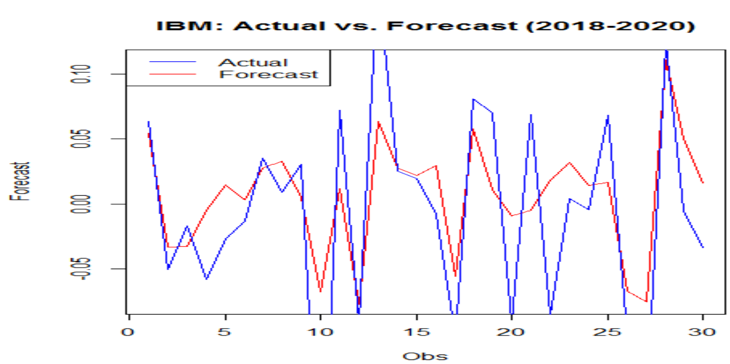
```
x_0 <- x[T2:T,]                          # Validation data
y_0 <- y[T2:T]                           # Validation data
y_f0 <-  x_0%*% b_ibm                    # Forecast
ef_0 <- y_f0 - y_0                       # Forecasat error
mse_ef_0 <- sum(ef_0^2)/nrow(x_0)        # MSE
> mse_ef_0
[1] 0.003703207
mae_ef_0 <- sum(abs(ef_0))/nrow(x_0)     # MAE
> mae_ef_0
[1] 0.04518326
```

That is, MSE = **0.003703207**
MAE = **0.04518326**

Evaluation of Forecasts: Measures of Accuracy

Example (continuation): Plot of actual IBM returns and forecasts.

```
plot(y_f0, type="l", col="red", main = "IBM: Actual vs. Forecast (2018-2020)",
    xlab = "Obs",  ylab = "Forecast")
lines(y_0, type = "l", col = "blue")
legend("topleft",  legend = c("Actual", "Forecast"),  col = c("blue", "red"),  lty = 1)
```



Evaluation of Forecasts: Measures of Accuracy

• So far, we have judged the model with the better (usually, lower) measure of accuracy as the better forecasting mode.

But, measures of accuracy are RV. Then, we cannot look at these measures and establish that Model 1 is "more accurate" than Model 2. Statistical error ("luck") can create problems.

• The most popular measure of accuracy is the MSE.

Q: How do we know the MSE for model 1 is significantly better than the MSE for model 2? We need a test for

H_0: MSE(1) = MSE(2)
H_1: MSE(1) ≠ MSE(2).

.

Evaluation of Forecasts: Testing Accuracy

• Suppose two competing forecasting procedures produce a vector of errors: $e^{(1)}$ & $e^{(2)}$. Then, if expected MSE is the criterion used, the procedure with the lower MSE will be judged superior.

• We want to test H_0: MSE(1) = MSE(2)
H_1: MSE(1) ≠ MSE(2).

<u>Assumptions</u>: forecast errors are unbiased, normal, and uncorrelated. If forecasts are unbiased, then MSE = Variance.

• Consider, the pair of RVs: $(e^{(1)} + e^{(2)})$ & $(e^{(1)} - e^{(2)})$. Now,

$$E[(e^{(1)} + e^{(2)})(e^{(1)} - e^{(2)})] = \sigma_1^2 - \sigma_2^2$$

• That is, we test H_0 by testing that the two RVs are not correlated!
Under H_0, $E[(e^{(1)} + e^{(2)})(e^{(1)} - e^{(2)})] = 0$.

Evaluation of Forecasts: Testing Accuracy

• Under H_0, $(e^{(1)} + e^{(2)})$ & $(e^{(1)} - e^{(2)})$ are not correlated –i.e., zero covariance. (See Morgan, Granger & Newbold (MGN, 1977).)

• There is a simpler way to do the MGN test. Steps:

1. Define $e^{(1)}$ & $e^{(2)}$, where $e^{(1)}$ is the error with the higher MSE. Let
   $z_t = e^{(1)} + e^{(2)}$ – $e^{(1)}$: the error with the higher MSE.
   $x_t = e^{(1)} - e^{(2)}$
2. Do a regression: $z_t = \beta\, x_t + \varepsilon_t$
3. Test H_0: $\beta = 0$ ⇒ a simple *t-test*.

The MGN test statistic is exactly the same as that for testing H_0: $\beta = 0$. This is the approach taken by Harvey, Leybourne & Newbold (1997).

• If the assumptions are violated, these tests have problems.

Evaluation of Forecasts: Testing Accuracy

Example: We produce IBM returns one-step-ahead forecasts for 2018-2020 using the 3-factor F-F Model for IBM returns:

$$(r_i - r_f)_t = \beta_0 + \beta_1\,(r_m - r_f)_t + \beta_2 SMB_t + \beta_3\, HML_t + \varepsilon_t$$

Taking expectations at time t+1, conditioning on time t information set, $I_t = \{(r_m - r_f)_t,\ SMB_t,\ HML_t\}$

$$E[(r_i - r_f)_{t+1} | I_t] = \beta_0 + \beta_1\, E[(r_m - r_f)_{t+1} | I_t] + \\ + \beta_2\, E[SMB_{t+1} | I_t] + \beta_3\, E[HML_{t+1} | I_t]$$

In order to produce forecast, we will make a naive assumption: The best forecast for the FF factors is the previous observation. Then,

$$E[(r_i - r_f)_{t+1} | I_t] = \beta_0 + \beta_1\,(r_m - r_f)_t + \beta_2 SMB_t + \beta_3 HML_t.$$

Now, replacing the β by the estimated **b**, we have our one-step-ahead forecasts. We produce one forecast at a time.

Evaluation of Forecasts: Testing Accuracy

Example: We compare the forecast accuracy relative to a random walk model for IBM excess returns. That is,

$$E[(r_i - r_f)_{t+1} | I_t] = (r_i - r_f)_t$$

Using R, we create the forecasting errors for both models and MSE:

```
T1 <- 539                              # End of Estimation Period (Dec 2017)
x_0f <- x[T1:(T-1),]                   # By assumption on the X, it starts at T1.
y_0 <- y[T2:T]                         # T2 = T1 + 1 (Jan 2018)
y_f0 <- x_0f %*% b1                    # b1 coefficients from fit_ibm_2
ef_0 <- y_f0 - y_0                     # e_t^(2)
mse_ef_0 <- sum(ef_0^2)/nrow(x_0)
> mse_ef_0                             # MSE(2)
[1] 0.01106811

ef_rw_0 <- y[T1:(T-1)] - y_0           # e_t^(1)
mse_ef_rw_0 <- sum(ef_rw_0^2)/nrow(x_0)
> mse_ef_rw_0                          # MSE(1)      <= (1) is the higher MSE.
[1] 0.02031009
```

Evaluation of Forecasts: Testing Accuracy

Example: Now, we create $z_t = e^{(1)} + e^{(2)}$, & $x_t = e^{(1)} - e^{(2)}$.
Then, regress: $z_t = \beta\, x_t + \varepsilon_t$ and test H_0: $\beta = 0$.

```
# Step 1. Define errors and z & x
z_mgn <- ef_rw_0 + ef_0
x_mgn <- ef_rw_0 - ef_0

# Step 2. Regress x on z
fit_mgn <- lm(z_mgn ~ x_mgn)
> summary(fit_mgn)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.05688    0.03512   1.619    0.117
x_mgn        2.77770    0.58332   4.762 5.32e-05 ***          ⇒ significant!

# Step 3. t-test on β
> coef(summary(fit_mgn))[, "t value"]
1.619      4.762
```

Conclusion: We reject that both MSEs are equal ⇒ MSE of RW is higher.

Evaluation of Forecasts: MSE/MAE?

• MSE and MAE are very popular criteria to judge the forecasting power of a model. However, they may not be the best measure for everybody.

• Richard Levich's textbook compares forecasting services to the freely available forward rate. He finds that forecasting services may have some ability to predict direction (appreciation or depreciation).

For some investors, the direction is what really matters, since direction determines potential profits, not the error.

Forecasting Application: MSE/MAE?

Example: Two forecasts: Forward Rate ($F_{t,T}$) and Forecasting Service (FS)

$F_{t,1\text{-month}}$ = .7335 USD/CAD

$E_{FS,t}[S_{t+1\text{-month}}]$= **.7342** USD/CAD. (Assume S_t = **.7330** USD/CAD).

(Investor's strategy: buy CAD forward if FS forecasts CAD appreciation.)

Based on the FS forecast, Ms. Sternin decides to buy CAD forward at $F_{t,1\text{-m}}$.

(A) Suppose that the CAD appreciates to S_{t+1} = .7390 USD/CAD.

MAE_{FS} = |.7390 - **.7342**| = .0052 USD/CAD.

Investor makes a profit of .7390 - .7335 = **USD .055** USD.

(B) Suppose that the CAD depreciates to S_{t+1} = .7315 USD/CAD.

MAE_{FS} = |.7315 - **.7342**| = .0027 USD/CAD. ⇒ smaller MAE!

Investor takes a loss of .7315 - .7335 = **USD -.0020**. ¶

