Time Series Analysis & Forecasting

Class 5

Arnab Bose, Ph.D.
MSc Analytics
University of Chicago

Time Series Features

1. Stationarity of a TS (refers to the statistical properties of the RVs of a TS).
2. Domain specific – e.g. number of times IoT sensor failed in some time period.
3. TS concepts – Reading Materials (Visualising Forecasting Algorithm Performance)
    1. Spectral Entropy F_1 - represents the relative contribution of different frequencies, lower means contains more signal and is more forecastable.
    2. Strength of Trend F_2 - compares the variances of the de-trended + de-seasonalized to the de-seasonalized TS.
    3. Strength of Seasonality F_3 - compares the variances of the de-trended + de-seasonalized to the de-trended TS.
    4. First order autocorrelation F_4 - compares the correlation of TS to its one-time shift and a higher absolute value indicates higher predictability of TS.
    5. Optimal Box-Cox transformation parameter F_6 - the value of λ that lowers variation in TS.

https://cran.r-project.org/web/packages/tsfeatures/vignettes/tsfeatures.html

R code – TS Features

```
install.packages("tsfeatures")
library("tsfeatures")

ts.plot(AirPassengers)
tsfeatures(AirPassengers)
entropy(AirPassengers)
hurst(AirPassengers) # fractal dimension D = 2 - hurst
stl_features(AirPassengers)
```

ARIMA Point Forecasting

- Expand ARIMA equation with Y_t on the left hand side and all other terms on the right
- Rewrite the equation replacing t with $t + l$, where l is the forecasting horizon
- On the right hand side of the equation, replace future observations by their forecasts, future errors by zero, and past errors by the corresponding residuals

ARIMA Forecast Updates

ARIMA Process $\phi(B)Y_t = \theta(B)e_t$

1. Directly in terms of the difference equation of previous Y's and current and previous white noise error e's

$$Y_{t+l} = \phi_1 Y_{t+l-1} + \ldots + \phi_{p+d} Y_{t+l-p-d} - \theta_1 e_{t+l-1} - \ldots - \theta_q e_{t+l-q} - e_{t+l}$$

2. Infinite weighted sum of current and previous shocks e's

$$Y_{t+l} = \sum_{j=0}^{\infty} \psi_j e_{t+l-j}$$

3. Infinite weighted sum of previous observations plus current shock e

$$Y_{t+l} = \sum_{j=0}^{\infty} \pi_j Y_{t+l-j} + e_{t+l}$$

Forecast for Non-zero Mean Models

Process with mean μ

AR(1) process

$$\hat{y}_t(l) \approx \mu \; for \; large \; l$$

MA(1) process

$$\hat{y}_t(l) = \mu \; for \; l > 1$$

Forecast Prediction Interval

- Take a linear regression

$$y_i = \beta_0 + \beta_t x_i$$

- Distance value = $\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum(x_i - \bar{x})^2}$

  where x_0 is a particular value of x_i

- Assume S to be the residual standard error

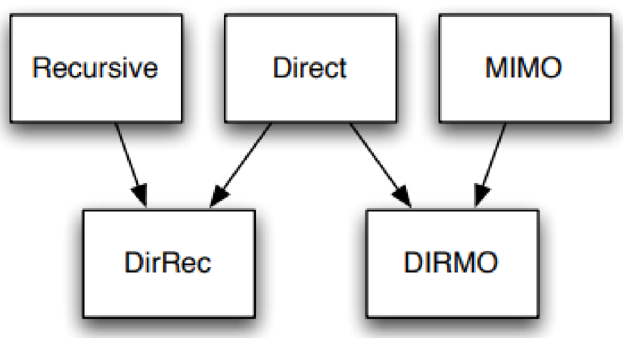
- Then $100(1 - \alpha)\%$ prediction interval for an individual value of y_i when $x_i = x_0$ is

$$\hat{y_i} \pm t * S * \sqrt{1 + Distance\ value}$$

  where t is the t-value at $^{\alpha}/_{2}$ level of significance for $n - 2$ DOF

Multi-step Forecasting Strategies

- Recursive – one trained model for 1-step ahead forecasts
- Direct – *h* models for *h* time horizons
- Direct + Recursive (DirRec) –*h* models with enlarging inputs
- Multi-Input Multi-Output (MIMO) – 1-step multiple output forecasts
- DIRect + miMO (DIRMO) – multiple output forecasts using *H* models with each model output of size *s*



In Reading: S. B. Taieba, G. Bontempia, A. Atiyac, A. Sorjamaa, *A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition*, https://arxiv.org/pdf/1108.3259.pdf, 2011.

Multi-step Forecasting Strategies

	Pros	Cons	Computational time needed
Recursive	Suitable for noise-free time series (e.g. chaotic)	Accumulation of errors	+
Direct	No accumulation of errors	Conditional independence assumption	+ + ++
DirRec	Trade-off between Direct and Recursive	Input set grows linearly with H	+ + + + +
MIMO	No conditional independence assumption	Reduced flexibility: same model structure for all the horizons	++
DIRMO	Trade-off between total dependence and total independence of forecasts	One additional parameter to estimate	+ + +

Table 3: A Summary of the Pros and Cons of the Different Multi-step Forecasting Strategies

Cross-Validation - General

1. Model validation technique
2. Divide a dataset into a training set and a validation set
3. K-fold cross validation => original data is divided into K equal size subsamples. Of the K, 1 subsample is retained as the validation set. Remaining K-1 are used as the training data
4. The above is repeated K times with each K subsample used exactly once as the validation data
5. The results from the validations will be combined (eg. average) to produce a single model estimation

Cross-Validation – Time Series

1. Take a rolling window for train, set forecast with h steps ahead
2. Calculate accuracy
3. Roll window forward by h steps and repeat
4. https://robjhyndman.com/hyndsight/tscv/
5. Challenges with nested CV: https://towardsdatascience.com/time-series-nested-cross-validation-76adba623eb9

R code – TS CV

```
data(gas)

auto.arima(gas)

far2 <- function(x, h){forecast(Arima(x, order=c(2,1,1), seasonal =
c(0,1,1)), h=h)}

e <- tsCV(gas, far2, h=3)

eDf <- as.data.frame(e)

matplot(eDf, type="l")
```

Bootstrapping

1. Any test or metric that relies on random sampling of the data with replacement

2. Expression “pulling oneself up by one’s bootstraps”

3. Advantage => simplicity

4. Disadvantage => assumption of independence of samples

5. The population is to the sample ***as*** the sample is to the bootstrap samples

6. Model-free resampling – block resampling with estimates for TS statistics

Block Bootstrapping

1. Apply Box-Cox decomposition together with STL and bootstrap remainder series
2. Divide remainder TS into overlapping blocks – there are multiple methodologies to select block size (such as CV with lowest error).
3. Shuffle the blocks.
4. Add the shuffled blocks together with the trend and seasonal components and reverse Box-Cos transformation

R code - Bootstrapping

```
library("forecast", lib.loc="~/R/win-library/3.3")
ts.plot(WWWusage)

bootst <- bld.mbb.bootstrap(WWWusage, 5)
bootstDf <- as.data.frame(bootst)
matplot(bootstDf, type="l")
```

Forecast Evaluation

- Forecast error at time t

$$e_t = y_t - \widehat{y_t}$$

- Absolute Deviation

$$|e_t| = |y_t - \widehat{y_t}|$$

- Mean Absolute Deviation

$$MAD = \frac{\sum_1^n |e_t|}{n}$$

- Mean Squared Error

$$MSE = \frac{\sum_1^n e_t^2}{n}$$

TS Forecast Evaluation

- Absolute Percentage Error

$$APE = \frac{|e_t|}{y_t} \times 100\%$$

- Mean Absolute Percentage Error

$$MAPE = \frac{\sum_1^n APE}{n}$$

- Symmetric Mean Absolute Percentage Error

$$sMAPE = \frac{1}{n}\sum_1^n \frac{|e_t|}{|y_t| + |\widehat{y_t}|}$$

- Mean Absolute Scaled Error (MASE) https://stats.stackexchange.com/questions/124365/interpretation-of-mean-absolute-scaled-error-mase

Improve TS Forecast Accuracy

Use an ensemble approach.

Refer to section 13.4 in Hyndman's book (FPP).

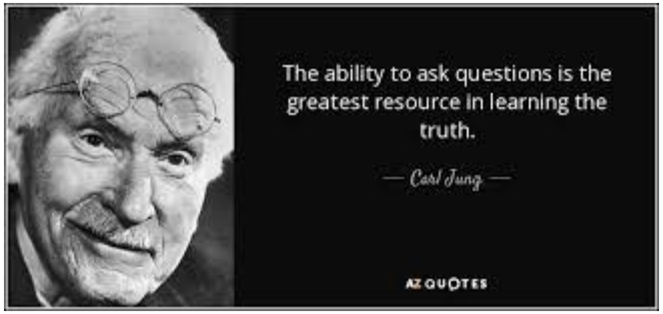
Textbook Chapters

Materials covered available in book chapters:

FPP: 5, 6, 12, 13

PTS: 9, 11

Thank You


The ability to ask questions is the greatest resource in learning the truth.
— Carl Jung —
AZ QUOTES