Time Series Analysis & Forecasting

Class 9

Arnab Bose, Ph.D.

MSc Analytics

University of Chicago

Cointegration

Regression of a non-stationary TS on another non-stationary TS

$$Y_t = \beta_0 + \beta_1 X_t + u_t$$

i.e. $$u_t = Y_t - \beta_0 - \beta_1 X_t$$

where Y_t, X_t ~ *I(d)*

u_t ~ *I(0)*

The linear combination cancels out the stochastic trends in X_t and Y_t.

X_t and Y_t are said to be cointegrated

To check for cointegration, verify that the residuals u_t are *I(0)* or stationary

Error Correction Model

From cointegration, X_t and Y_t have a long term equilibrium relationship.

But that may get off-balance in the short term

$$\Delta Y_t = \alpha_0 + \alpha_1 \Delta X_t + \alpha_2 u_{t-1} + \varepsilon_t$$

where Δ is the first difference operator

ε_t is the random error

u_{t-1} one-period lagged error from cointegration regression

Note that $\alpha_2 < 0$ to drive short term disequilibrium to equilibrium

Granger Causality Test

Two TS X_t and Y_t - determine which one is useful in forecasting the other.

Use this test to determine G-causality

H_0: X_t does not Granger-cause Y_t

2 regressions

$$Y_t = a_0 + a_1 Y_{t-1} + .. + a_m Y_{t-m} + residual$$

$$Y_t = a_0 + a_1 Y_{t-1} + .. + a_m Y_{t-m} + b_p X_{t-p} + .. + b_q X_{t-q}$$

Use F-test whose null hypotheses is no explanatory power is added by X_t

Note if TS are cointegrated, then there must be a G-causality between them – either one-way or in both directions

Granger Causality Test in R

The Granger Causality test fits a VAR model and tests the NULL Hypothesis $H0$ that x does NOT Granger cause y, i.e. that the coefficients of the lags of x are not significant.

$$\begin{bmatrix} y_t \\ x_t \end{bmatrix} = \begin{bmatrix} \alpha_{1,11} & \alpha_{1,12} \\ \alpha_{1,21} & \alpha_{1,22} \end{bmatrix} \begin{bmatrix} y_{t-1} \\ x_{t-1} \end{bmatrix} + \cdots + \begin{bmatrix} \alpha_{p,11} & \alpha_{p,12} \\ \alpha_{p,21} & \alpha_{p,22} \end{bmatrix} \begin{bmatrix} y_{t-p} \\ x_{t-p} \end{bmatrix} + \begin{bmatrix} a_{1t} \\ a_{2t} \end{bmatrix}$$

$$H0: \alpha_{p,12} = \alpha_{p,22} = 0$$

NOTE – the p-value comes from a Wald test. If it is lower than 0.05 significance level you can reject $H0$ and conclude that x Granger-causes y.

R code – Granger Causality Test

```
library("lmtest")
grangertest(ChickEgg[, 1], ChickEgg[, 2], order = 3)

grangertest(ChickEgg[, 2], ChickEgg[, 1], order = 3)
```

Machine Learning for Time Series

1. No supposition or formulation of the underlying process generating the TS unlike in statistical models - welcome to pattern recognition.
2. ML models are not "time aware" unlike Box-Jenkins models (e.g. ARMA, ARIMA, SARIMA) – no need to make stationary, but good to de-trend TS since ML models are challenged at extrapolating.
3. Transform the data (Box-Cox, difference).
4. Use time delay TS values.
5. Integer or dummy coded periodic time (month of year, week day, hour of day).
6. Use features to find patterns in TS.

TS Supervised Model – Ensembles

1. Wisdom of Crowds
    1. Independent models
    2. Using local information
    3. No intermodel communication
    4. Method to aggregate individual model outcomes
2. Voting
    1. Output of each model weighted into one response
3. Bagging
    1. Bootstrap aggregation
4. Random Forest
    1. Improvement on bagging by randomizing set of features for different trees
5. Boosting
    1. Reweight training data to find submodels that complement each other

TS Unsupervised Model – Clustering using Similarity in TS

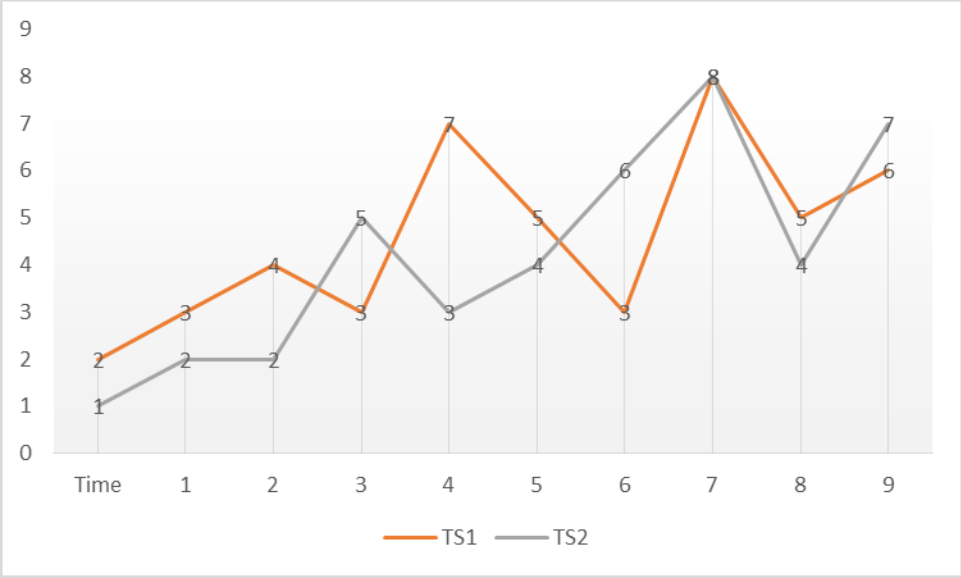
1. To use clustering in ML need a concept of "similar" TS.
2. Euclidean distances do not work with TS.
3. Use Dynamic Time Warping (DTW) for TS.

Dynamic Time Warping (DTW)

1. Every point in one TS has to be matched with atleast one point in the other TS
2. The first and last data points of each TS must be matched with their counterparts.
3. No going back in time – time moves forward such that a data point in one TS cannot match a data point in other TS that has already passed in the time axis.

DTW – TS Example

Time	1	2	3	4	5	6	7	8	9	10
TS1	2	3	4	3	7	5	3	8	5	6
TS2	1	2	2	5	3	4	6	8	4	7



DTW Matrix

7	24	17	13	14	9	9	11	9	11	11
4	19	13	10	11	9	7	8	12	10	12
8	17	12	10	11	6	8	11	9	12	12
6	11	7	6	7	5	6	9	9	10	10
4	7	4	4	4	6	7	7	10	10	11
3	5	3	4	3	7	8	6	11	9	10
5	4	3	3	5	6	6	6	7	7	8
2	1	2	4	4	8	10	4	9	10	11
2	1	2	4	3	7	7	3	8	7	8
1	1	2	3	2	6	4	2	5	4	5
	2	3	4	3	7	5	3	8	5	6

$$d(i,j) = \left|TS1_i - TS2_j\right|$$

$$DTW(i,j) = d(i,j) + \min[DTW(i-1,j), DTW(i,j-1), DTW(i-1,j-1)]$$

Warping Sequence and Time Normalized Distance

7	24	17	13	14	9	9	11	9	11	11
4	19	13	10	11	9	7	8	12	10	12
8	17	12	10	11	6	8	11	9	12	12
6	11	7	6	7	5	6	9	9	10	10
4	7	4	4	4	6	7	7	10	10	11
3	5	3	4	3	7	8	6	11	9	10
5	4	3	3	5	6	6	6	7	7	8
2	1	2	4	4	8	10	4	9	10	11
2	1	2	4	3	7	7	3	8	7	8
1	1	2	3	2	6	4	2	5	4	5
	2	3	4	3	7	5	3	8	5	6

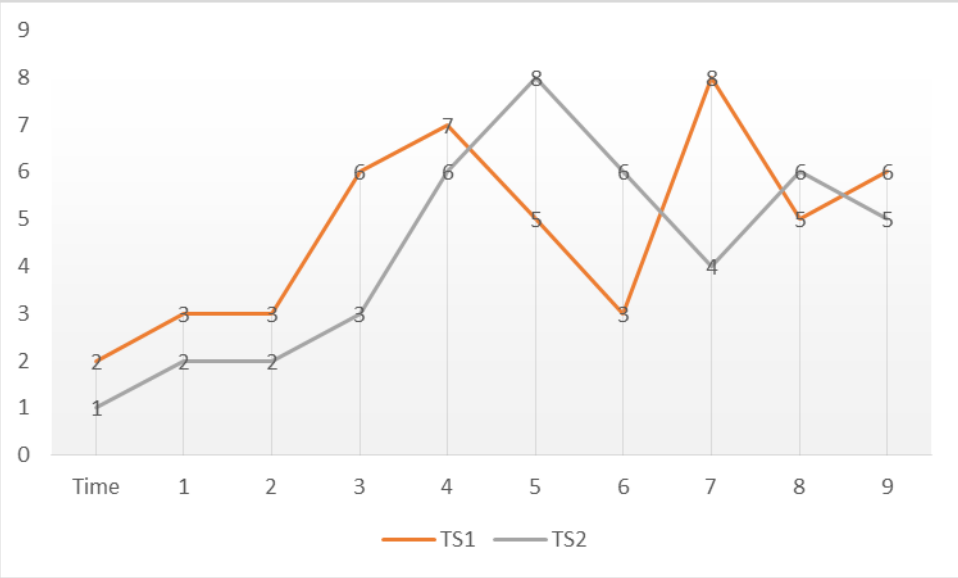
$$d = \frac{\sum_k d_i}{k} = \frac{79}{10} = 7.9$$

$$d = \frac{\sum_k d_i}{k} = \frac{64}{10} = 6.4$$

Brijnesh J. Jain and David Schultz, *Optimal Warping Paths are unique for almost every Pair of Time Series*, https://arxiv.org/pdf/1705.05681.pdf, accessed Feb 2021.

DTW – “More Similar” TS Example

Time	1	2	3	4	5	6	7	8	9	10
TS1	2	3	3	6	7	5	3	8	5	6
TS2	1	2	2	3	6	8	6	4	6	5



Warping Sequence and Time Normalized Distance

5	28	18	18	6	7	5	7	9	6	7
6	24	16	16	5	6	5	7	6	7	7
4	20	13	13	5	6	4	4	8	8	9
6	18	12	12	3	3	3	6	8	7	9
8	14	9	9	3	2	5	8	6	9	11
6	8	4	4	1	2	3	6	8	9	9
3	4	1	1	4	8	10	10	15	17	20
2	1	2	3	7	12	15	16	22	25	29
2	1	2	3	7	12	15	16	22	25	29
1	1	2	4	9	15	19	21	28	32	37
	2	3	3	6	7	5	3	8	5	6

$$d = \frac{\sum_k d_i}{k} = \frac{34}{10} = 3.4$$

https://dtw.r-forge.r-project.org/images/index.html

R code – DTW

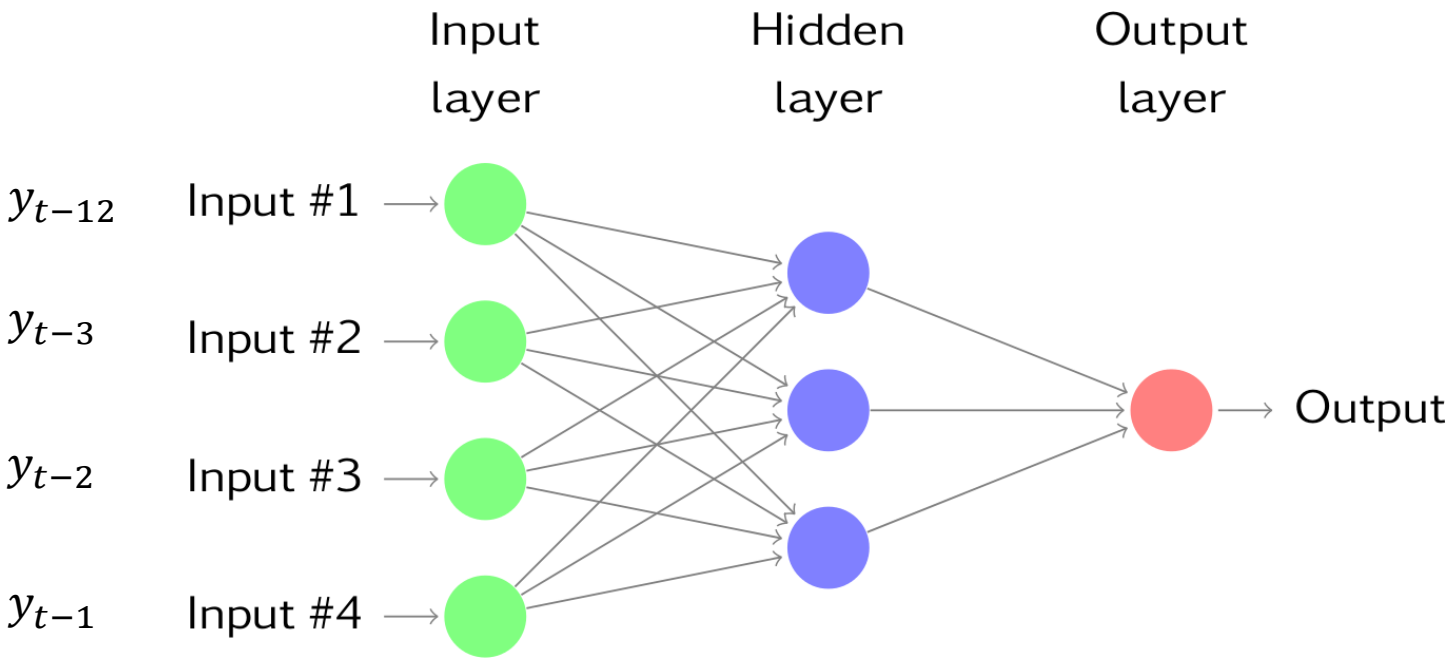
```
## A noisy sine wave as query
idx<-seq(0,6.28,len=100);
query<-sin(idx)+runif(100)/10;

## A cosine is for template;
template<-cos(idx)

# plot the 2 TS
plot(query, type = 'l')
lines(template, type = 'l', col = 'red')
```

R code – DTW

```
## Find the best match with the canonical recursion formula
library(dtw);
alignment<-dtw(query,template,keep=TRUE);

## Display the warping curve, i.e. the alignment curve
dtwPlotTwoWay(alignment)
plot(alignment,type="threeway")

## Align and plot with the Rabiner-Juang type VI-c unsmoothed recursion
plot(dtw(query,template,keep=TRUE,
     step=rabinerJuangStepPattern(6,"c")), type="twoway", offset=-2);

## See the recursion relation, as formula and diagram
rabinerJuangStepPattern(6,"c")
plot(rabinerJuangStepPattern(6,"c"))
```

Neural Networks for Time Series

- Fully connected Neural Network with 1 hidden layer
- NNAR $(3,1,3)_{12}$ with 3 lag inputs, 1 seasonal lag input and 3 nodes in the hidden layer
- The non-linearity is due to the sigmoid activation function in the hidden layer
- Packages: forecast (nnetar) and nnfor (elm and mlp)



https://otexts.org/fpp2/nnetar.html#fig:nnet2

R code – Neural Network Forecasting (nnfor)

```
library(nnfor)
?elm
fit <- elm(AirPassengers, hd=10)
print(fit)
plot(fit)
frc <- forecast(fit, h=36)
plot(frc)

?mlp
fit2 <- mlp(AirPassengers, hd = c(10,5))
plot(fit2)
frc2 <- forecast(fit2, h=36)
plot(frc2)
```

https://kourentzes.com/forecasting/2019/01/16/tutorial-for-the-nnfor-r-package/

Using NN for TS

1. Detrend TS
2. NN can model seasonalities, but you can use seasonal differencing or dummy variables as an input(s)
3. However, may be simpler to deseasonalize the data

Deep Learning – Recurrent Neural Network (RNN)

1. RNNs address the temporal relationship of their inputs by maintaining an internal state.
2. RNNs are biased towards learning patterns which occur in temporal order – i.e. they are less prone to learning random correlations which do not occur in temporal order.
3. In theory, RNNs are absolutely capable of handling "long-term dependencies." But in practice, RNNs suffer from vanishing gradient problem.
4. Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies*.

http://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed Nov, 2016.

GluonTS

1. Gluon is an open-source deep learning library from AWS and Microsoft and is part of Apache MXNet.
2. Gluon Time Series (GluonTS) is the Gluon toolkit for probabilistic time series modeling, focusing on deep learning-based models.
3. Works with Deep learning TS models such as DeepAR.

https://ts.gluon.ai/index.html

DeepAR: Deep Learning Autoregressive Recurrent Network

1. Developed by Amazon Research for forecasting with correlated multivariate time series.
2. Builds a global model using multiple related historical time series.
3. Probabilistic forecasting – estimate the statistical distribution of the time series during training, hence output is non-deterministic between runs.
4. Autoregressive – current TS value uses last step value as input.
5. Recurrent – network output in last step is used as input for current step.
6. The parameters of the model are the same for all time-series. During training, different time-series windows are sampled and the model is tuned to predict them. Note that the parameters are the same between time-series but the data is not. The model may learn things such as averaging values over past seasonality or trend for instance with the same parameters given that the input data is different. Akin to NLP translation models for instance: the parameters are fixed but the model can still parse different sentences.

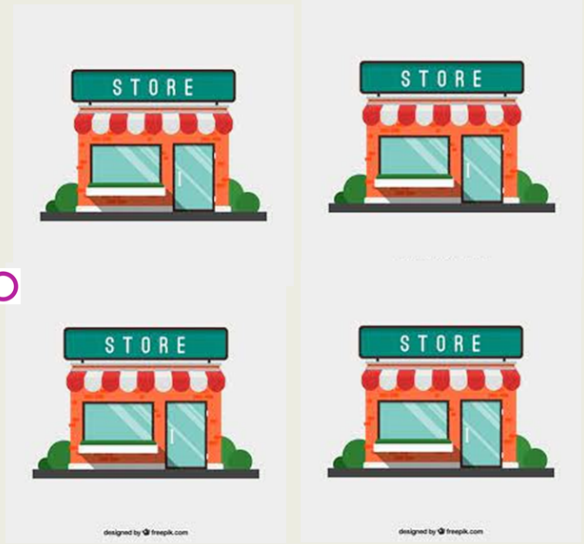
DeepAR Forecasting Algorithm, https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html

DeepAR: Characteristics

1. Requires minimal (manual) feature engineering – the model learns seasonal behavior and dependencies on given covariates across time series and minimal manual feature engineering is needed to capture complex, group-dependent behavior.
2. One algorithm for multiple TS – one algorithm to learn from multiple TS.
3. Performs Monte Carlo sampling – model makes probabilistic forecasts in the form of Monte Carlo samples that can be used to compute consistent quantile estimates for all sub-ranges in the prediction horizon.
4. Built-in item similarity – by learning from similar items, model provides forecasts for items with little or no history at all.
5. Variety of likelihood functions – model does not assume Gaussian noise, instead uses a wide range of likelihood functions adapting to the statistical properties of the data.
6. But takes time and intensive resources to train for large datasets – hyperparameters difficult to tune.

D. Salinas, V. Flunkert, J. Gasthaus, *DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks*, https://arxiv.org/pdf/1704.04110.pdf

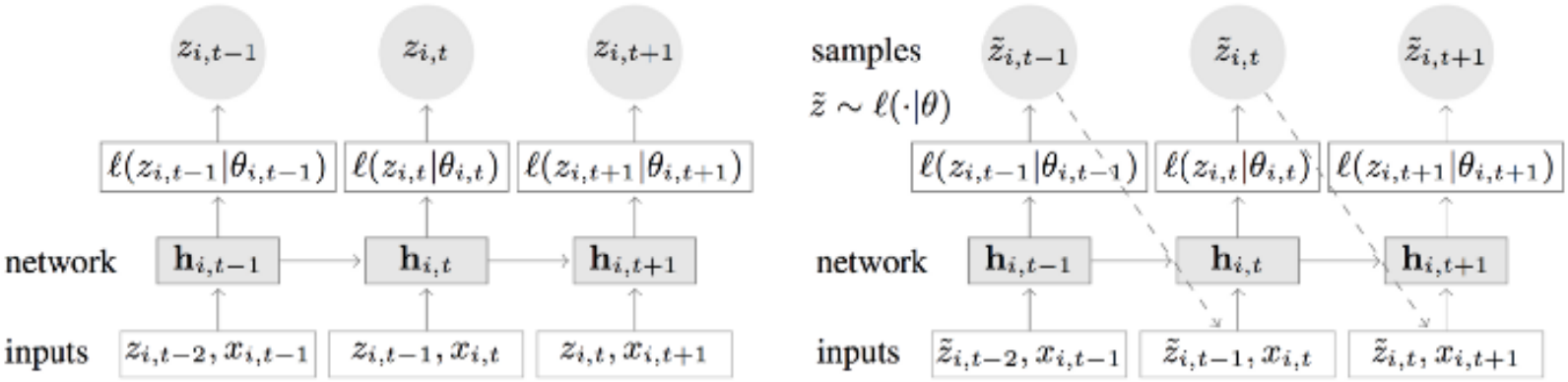
Retail Demand Forecasting



We understand retail nuances well enough to know that as items are sold across stores, their sales pattern may have varying magnitude and the distribution of magnitude can also be strongly skewed as seen above.

We apply this domain knowledge in developing our unique Deep Learning (DL) based solution that learns from this local variation in individual items/stores to build a global model. *"Its one model for the entire organization"*.

DeepAR Training + Forecasting



Training

The output of the network is $\boldsymbol{h}_{i,t} = h(\boldsymbol{h}_{i,t-1}, z_{i,t-1}, x_{i,t}, \Theta)$. That is used to learn the parameters $\theta_{i,t} = \theta(\boldsymbol{h}_{i,t}, \Theta)$ of the likelihood function $l(z|\theta)$.

Here $h(.), \theta(.)$ are functions with model parameters that comprise Θ and are learnt by maximizing the likelihood function $l(.)$.

Forecasting

To predict for future time steps, estimate $\tilde{z}_{i,t}$ is sampled from the likelihood function $l(.|\theta)$.

The estimates are fed recursively into the model forecasting with $h(.)$ for the next step.

Causal Inference

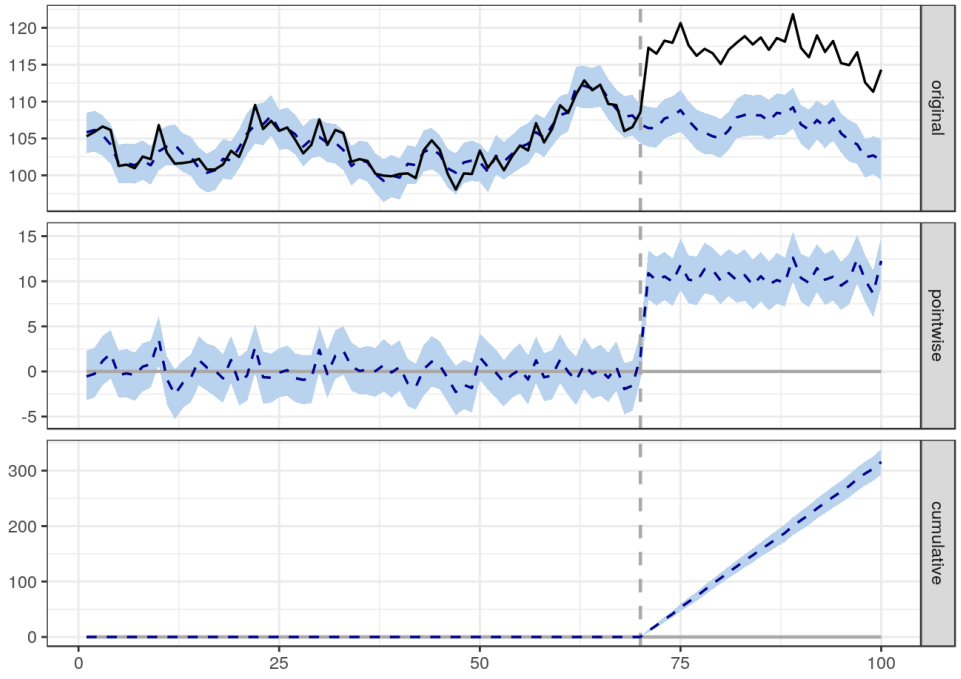
Intervention address “treatment” or outside influence historically.

Question – what would have happened with no treatment/outside influence?

In other words, understand the causal effect of the treatment – understand from the perspective of with and without the treatment, i.e. predicting counterfactual (what would have happened if there was no treatment?).

Assumptions:

1. there is a set control TS that is unaffected by the intervention – key to understand counterfactual
2. The relationship between covariates and treated TS remains stable post-treatment as it was in pre-treatment.



http://google.github.io/CausalImpact/CausalImpact.html

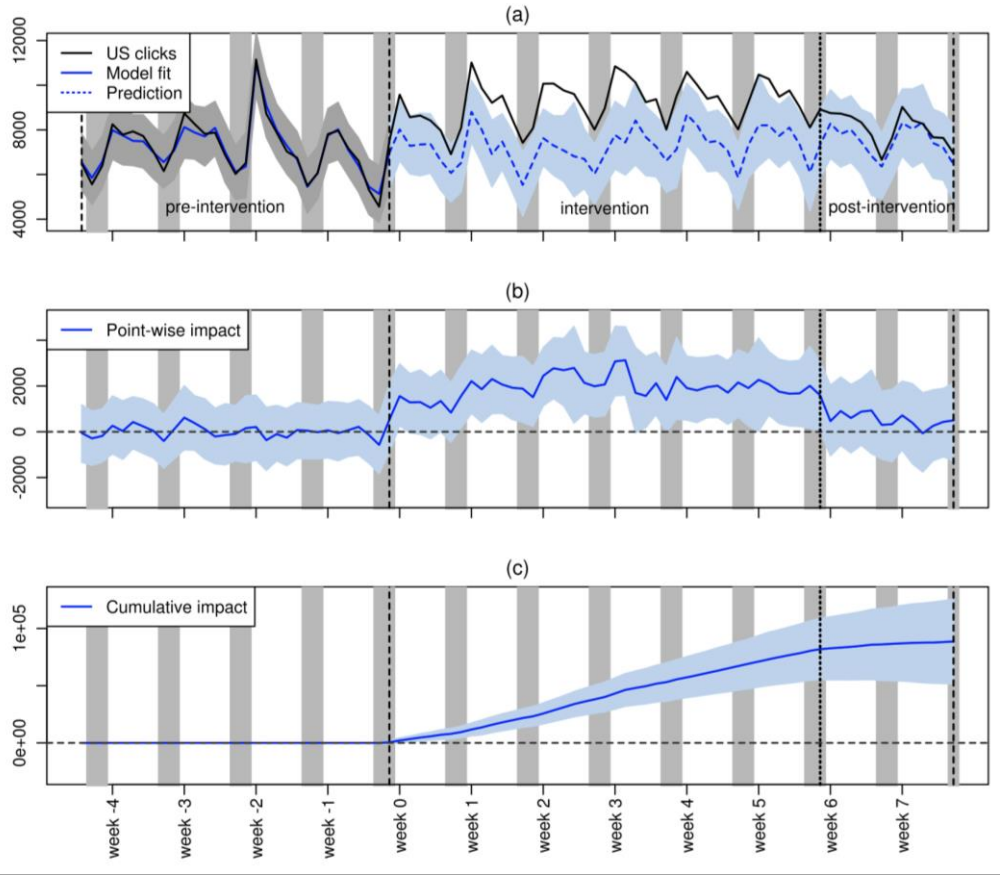
Causal Inference Output – Google Advertising Effect

The controls was given by the untreated Designated Market Areas (DMAs). The predictive model provided an excellent fit on the precampaign trajectory of clicks.

(a) Following the onset of the campaign, observations quickly began to diverge from counterfactual predictions: the actual number of clicks was consistently higher than what would have been expected in the absence of the campaign. The curves did not reconvene until one week after the end of the campaign.

(b) The incremental lift caused by the campaign. It peaked after about three weeks into the campaign, and faded away after about one week after the end of the campaign.

(c) The campaign led to a sustained cumulative increase in total clicks (as opposed to a mere shift of future clicks into the present or a pure cannibalization of organic clicks by paid clicks). Specifically, the overall effect amounted to 88,400 additional clicks in the targeted regions (posterior expectation; rounded to three significant digits), that is, an increase of 22%, with a central 95% credible interval of [13%, 30%].



Causal effect of Google online advertising on clicks in treated regions. (a) Time series of search-related visits to the advertiser's website (including both organic and paid clicks). (b) Pointwise (daily) incremental impact of the campaign on clicks. Shaded vertical bars indicate weekends. (c) Cumulative impact of the campaign on clicks.

https://storage.googleapis.com/pub-tools-public-publication-data/pdf/41854.pdf

Causal Inference – R example

```
library(TSA)
library(CausalImpact)
data("airmiles")
start = as.yearmon("1996-01-01")
end = as.yearmon("2004-12-31")
pre.period <- c(start, as.yearmon("2001-08-31"))
post.period <- c(as.yearmon("2001-09-01"), end)

impact_911 <- CausalImpact(airmiles, pre.period, post.period,
model.args = list(niter = 1000, nseasons = 12))

plot(impact_911)
summary(impact_911)
summary(impact_911, "report")
```

Causal Impact Analysis on VolksWagen Emissions Scandal

http://rstudio-pubs-static.s3.amazonaws.com/263637_c02e26f8546e49f09aef7eff090fbecf.html, accessed May 2021.

Hybrid Modeling – ARIMA + ANN

Practical applications benefit from a hybrid approach that incorporates ARIMA (Statistical) for the linear part L_t and ANN (Deep Learning) for the non-linear part N_t .

There are 2 types of hybrid models:

1. Additive

$$y_t = L_t + N_t$$

With the ANN modeling the error defined as $e_t = y_t - L_t$ and with non-linear mapping function

$$e_t = f(e_{t-1}, e_{t-2}, \dots, e_{t-n}) + \varepsilon_t$$

2. Multiplicative

$$y_t = L_t \times N_t$$

The ANN non-linear mapping function remains the same, but the error is defined as $e_t = y_t / L_t$

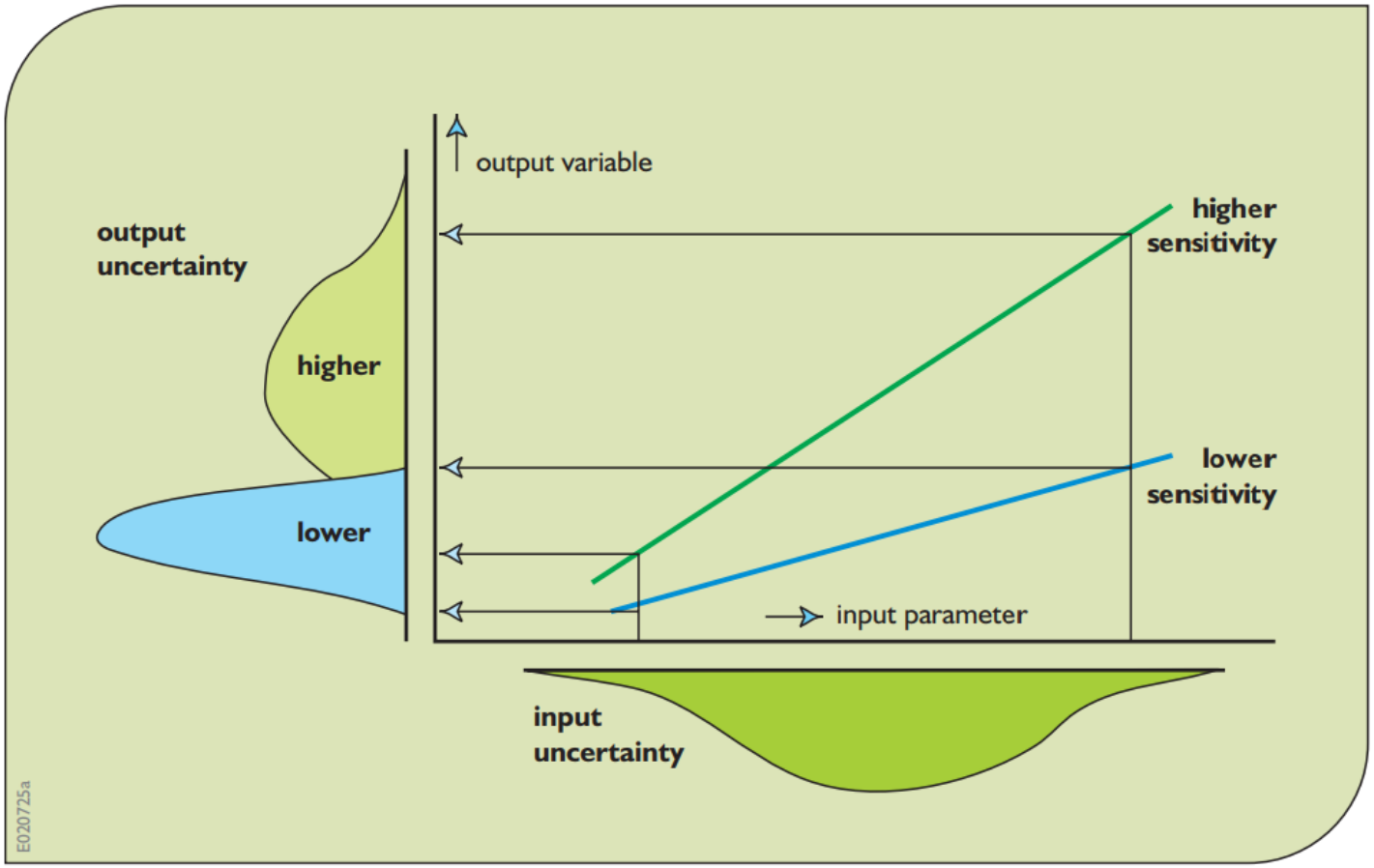
Model Uncertainty

- Sequential Stability
    - Take different sample sizes and calculate the % of times same model selected
- Perturbation Stability
    - Change dataset with new additional samples and selected model should remain the same
- NOTE – Model selection method does not guarantee any forecasting performance

Sensitivity Analysis

- Similar to risk, sensitivity assesses how the uncertainty affects area of interest for a particular use case.
- Measures the sensitivity of a model output to changes in model input values.
- Input/output scatterplots are a simple method for sensitivity analysis.
- To calculate model output sensitivity analysis to different input parameters, use Monte Carlo simulations.
- To focus on specific regions, use Monte Carlo Filtering.

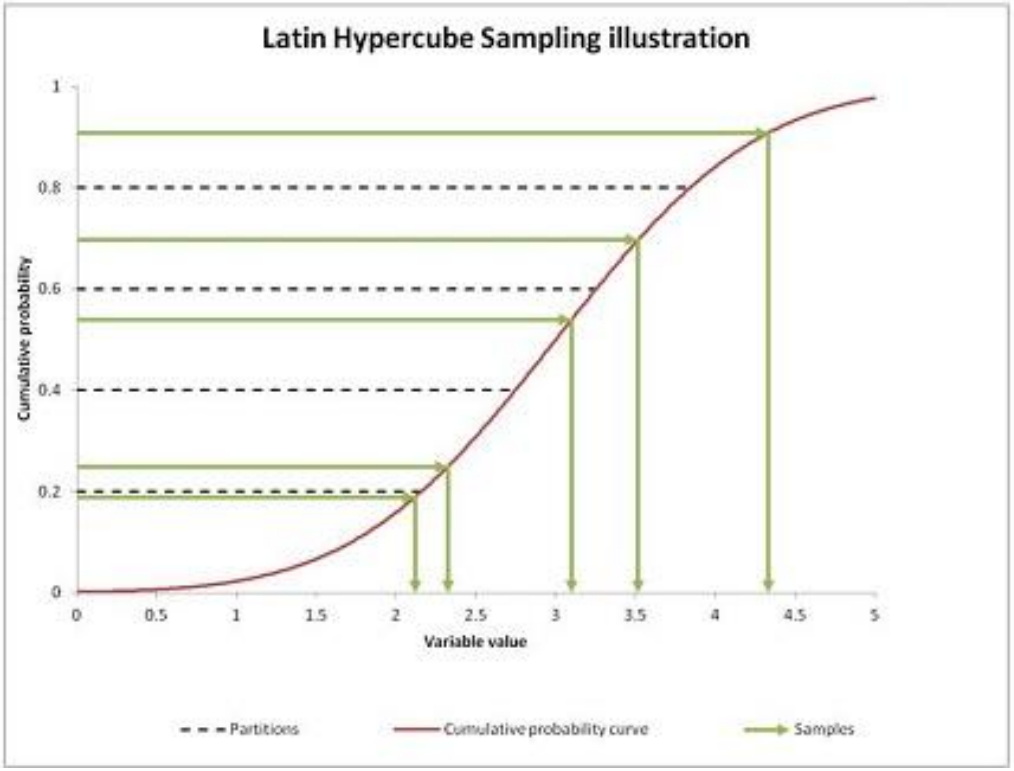
Schematic diagram for input param uncertainty and sensitivity



LAL, W. 1995. Sensitivity and uncertainty analysis of a regional model for the natural system of South Florida. West Palm Beach, Fla., South Florida Water Management District. Draft report, November.

https://ecommons.cornell.edu/bitstream/handle/1813/2804/09_chapter09.pdf;sequence=12

Latin Hypercube Sampling



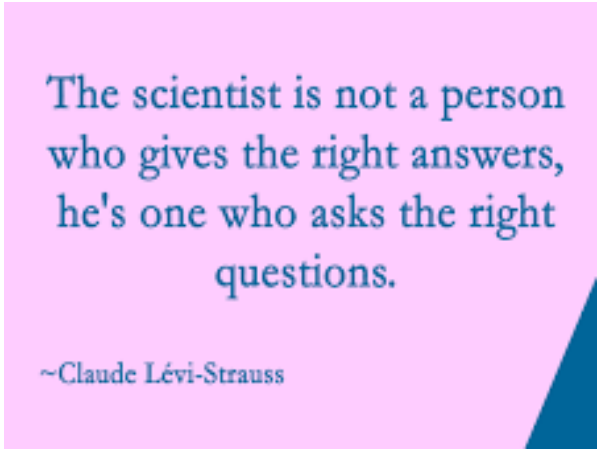
http://liprof.com/blog/the-pros-and-cons-of-latin-hypercube-sampling

Textbook Chapters

Materials covered available in book chapters:

PTS: 9, 10

Thank You


The scientist is not a person who gives the right answers, he's one who asks the right questions.
~Claude Lévi-Strauss