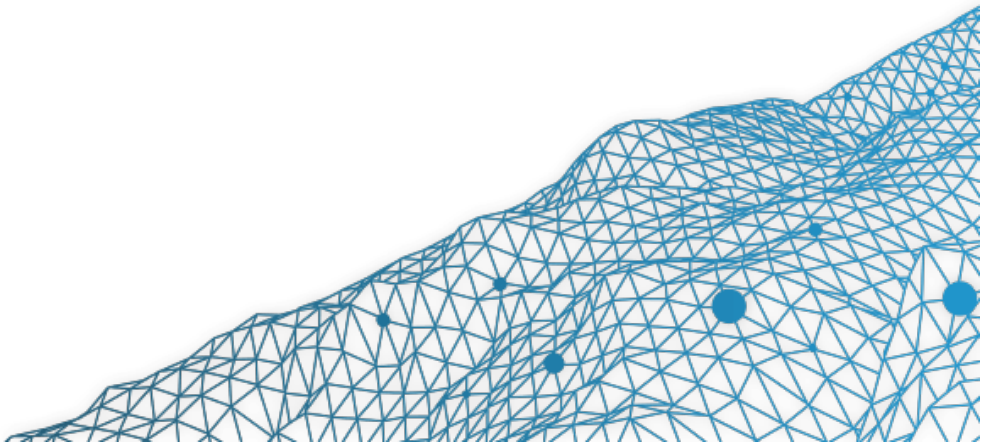
7 Times Series and Forecasting

Arthur Charpentier (Université du Québec à Montréal)

Machine Learning & Econometrics

SIDE Summer School - July 2019



Time Series

Time Series

A time series is a sequence of observations (y_t) ordered in time.

Write $y_t = s_t + u_t$, with systematic part s_t (signal / trend) and 'residual' term u_t

(u_t) is supposed to be a strictly stationary time series

(s_t) might be a 'linear' trend, plus a seasonal cycle

Buys-Ballot (1847, Les changements périodiques de température, dépendants de la nature du soleil et de la lune, mis en rapport avec le pronostic du temps, déduits d'observations néerlandaises de 1729 à 1846) - original probably in Dutch.

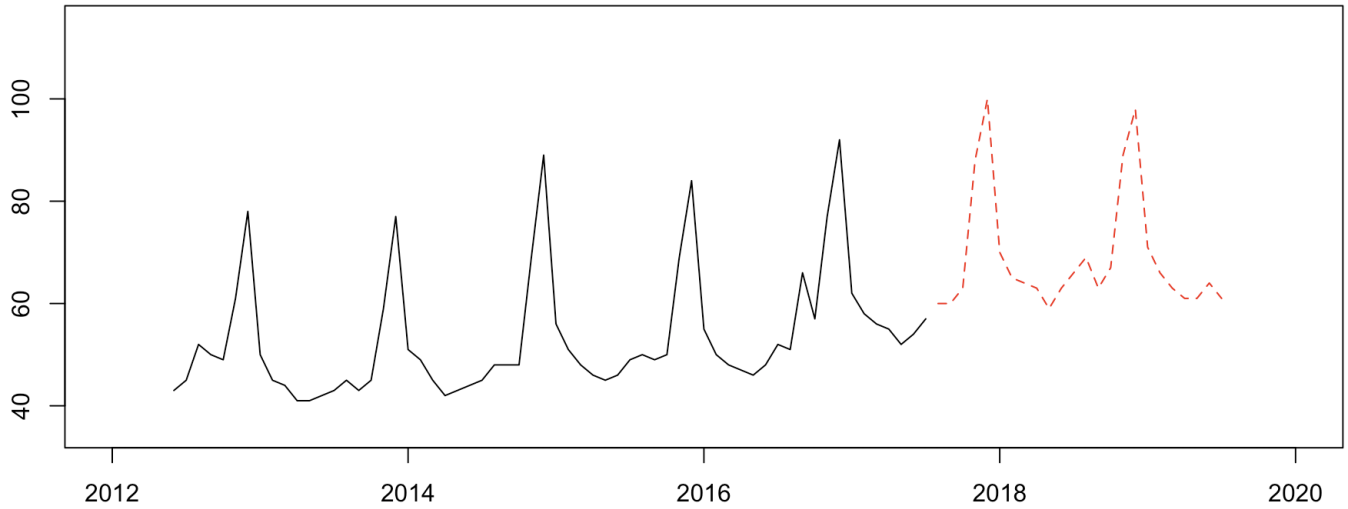
TABLEAU REPRÉSENTANT LA MARCHE DE LA TEMPÉRATURE PENDANT L'ANNÉE.

Date.	Temp.	Temp. calculée.	Différ.	Temp. calculée.	Différ.	Date.	Temp.	Temp. calculée.	Différ.	Temp. calculée.	Différ.
10 Janv.	32.58	32.58	0	32.58	0	17 Juill.	0.58	+ 0.24	+ 0.34	63.68	+ 0.66
15 «	+ 1.12	+ 0.78	+ 0.23	+ 0.88	+ 0.13	22 «	— 0.36	64.33	— 0.25	64.03	+ 0.05
20 «	0.69	34.14	+ 0.25	34.34	+ 0.05	27 «	+ 0.50	64.57	+ 0.01	64.37	+ 0.21
25 «	0.83	34.92	+ 0.30	35.22	0	1 Août	+ 0.26	64.83	+ 0.02	64.71	+ 0.13
30 «	1.26	35.70	+ 0.78	+ 0.59	+ 0.67	6 «	+ 0.22	65.06	0	65.06	0
4 Févr.	— 0.01	36.47	0	36.40	+ 0.07						
9 «	+ 0.39	+ 0.55	— 0.16	36.99	— 0.13	11 «	— 0.45	— 0.42	— 0.03	— 0.50	+ 0.05
14 «	+ 0.36	37.57	— 0.35	37.58	— 0.36	16 «	0.34	64.22	+ 0.05	64.05	+ 0.22
19 «	— 0.05	38.13	— 0.96	38.17	— 1.00	21 «	0.42	63.80	+ 0.05	63.55	+ 0.30
24 «	+ 1.59	38.68	+ 0.08	38.76	0	26 «	0.66	63.38	+ 0.19	63.04	+ 0.15
1 Mars	0.48	39.24	0	+ 0.69	— 0.25	1 Sept.	0.23	62.96	0	62.54	+ 0.42
6 «	0.25	+ 0.73	— 0.48	40.14	— 0.65	6 «	0.93	— 1.13	+ 0.20	62.03	0
11 «	0.11	40.69	— 1.10	40.83	— 1.23	11 «	1.12	60.71	+ 0.06	— 1.26	0
17 «	1.83	41.42	+ 0.03	41.52	— 0.09	16 «	0.89	59.59	+ 0.43	59.51	+ 0.51
22 «	0.79	42.15	+ 0.07	42.22	0	21 «	1.37	58.46	+ 0.19	58.25	+ 0.41
27 «	0.67	42.89	0	+ 1.29	— 0.62	26 «	1.32	57.33	0	56.99	+ 0.34
1 Avril	1.34	+ 1.36	— 0.02	44.80	— 0.57	1 Oct.	1.59	— 1.32	0.27	55.74	0
6 «	1.90	45.61	+ 0.52	46.09	+ 0.04	6 «	1.08	54.69	— 0.03	— 1.43	+ 0.35
11 «	1.23	46.99	+ 0.37	47.38	— 0.02	11 «	1.49	53.37	— 0.20	52.88	+ 0.29
16 «	0.89	48.36	— 0.11	48.67	— 0.42	16 «	1.23	52.05	— 0.12	51.46	+ 0.48
21 «	1.45	49.70	0	49.96	— 0.26	21 «	1.21	50.73	0	50.03	+ 0.70
26 «	1.56	+ 1.21	+ 0.33	51.26	0	26 «	2.13	— 1.63	— 0.50	48.60	0
2 Mai	1.56	52.13	+ 0.69	+ 1.12	+ 0.44	1 Nov.	1.65	47.44	— 0.49	— 1.41	— 0.24
7 «	1.50	53.34	+ 0.98	53.50	+ 0.82	6 «	1.71	45.82	— 0.58	45.77	— 0.53
12 «	0.29	54.56	— 0.01	54.62	— 0.07	11 «	1.04	44.20	0	44.36	+ 0.16
17 «	1.18	55.79	0	55.74	+ 0.05	16 «	0.95	— 0.76	+ 0.19	42.95	+ 0.30
22 «	1.28	+ 0.95	+ 0.34	56.87	+ 0.20	21 «	1.72	42.68	— 1.15	41.53	0
27 «	0.92	57.69	+ 0.30	57.99	0	26 «	0.58	41.92	— 0.97	— 0.70	+ 0.12
1 Juin	0.26	58.64	— 0.39	+ 0.66	— 0.40	1 Déc.	+ 0.20	41.15	0	40.13	+ 0.76
6 «	1.37	59.60	+ 0.02	59.32	+ 0.30	6 «	1.55	— 0.85	— 0.42	39.43	+ 0.27
11 «	0.93	60.55	0	59.98	+ 0.57	11 «	0.91	39.19	— 0.40	38.73	+ 0.06
16 «	0.57	+ 0.55	+ 0.02	60.65	+ 0.47	16 «	0.75	38.33	— 0.29	38.04	0
21 «	0.06	61.65	— 0.47	61.31	— 0.13	21 «	0.64	37.40	0	— 1.04	+ 0.40
26 «	0.77	62.20	— 0.25	61.97	— 0.02	26 «	1.61	— 1.14	— 0.47	35.96	— 0.17
2 Juill.	0.68	62.75	— 0.12	62.63	0	31 «	1.05	35.12	— 0.38	34.92	— 0.18
7 «	0.70	63.31	+ 0.02	+ 0.35	+ 0.35	5 Juin	1.44	33.98	— 0.68	33.88	— 0.58
12 «	0.53	63.86	0	63.33	+ 0.53	10 «	0.72	32.58	0	32.58	0

Time Series

Consider the general prediction ${}_t\widehat{y}_{t+h} = m(\text{information available at time } t)$

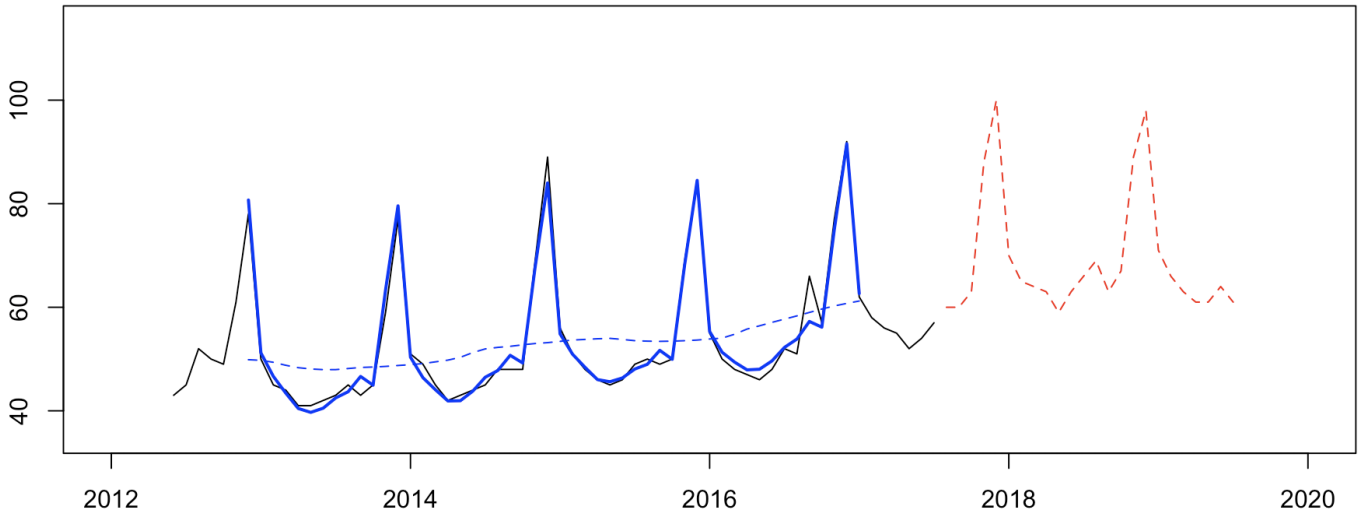
```
hp <- read.csv("http://freakonometrics.free.fr/multiTimeline.csv",
    skip=2)
T=86-24
trainY <- ts(hp[1:T,2], frequency= 12, start= c(2012, 6))
validY <- ts(hp[(T+1):nrow(hp),2], frequency= 12, start= c(2017, 8))
```



Time Series

In $y_t = s_t + u_t$, s_t can be a trend, plus a seasonal cycle

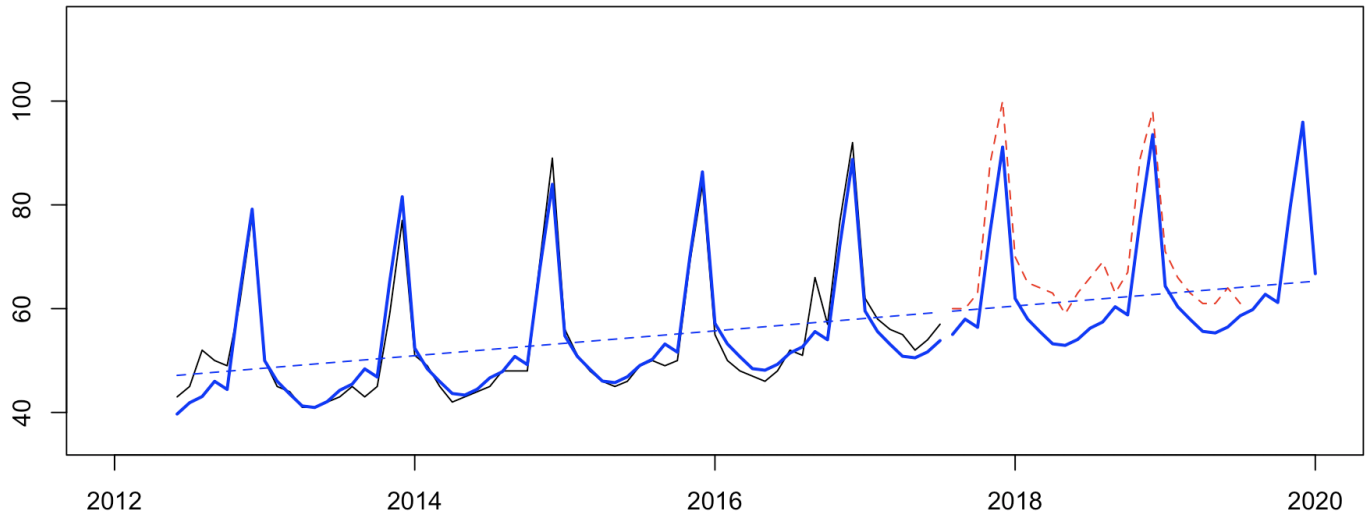
```
1 stats::decompose(trainY)
```



Time Series

The Buys-Ballot model is based on $y_t = s_t + u_t$, with

$$s_t = \beta_0 + \beta_1 t + \sum_{h=1}^{12} \gamma_{t \text{ mod } 12}$$



Time Series : Exponential Smoothing

“

from Hyndman *et al.* (2008, Forecasting with Exponential Smoothing)

Time Series : Exponential Smoothing

Exponential smoothing - Simple

From time series (y_t) define a smooth version

$$s_t = \alpha \cdot y_t + (1-\alpha) \cdot s_{t-1} = s_{t-1} + \alpha \cdot (y_t - s_{t-1})$$

for some $\alpha \in (0,1)$ and starting point $s_0 = y_1$ Forecast is ${}_t\widehat{y}_{t+h} = s_t$

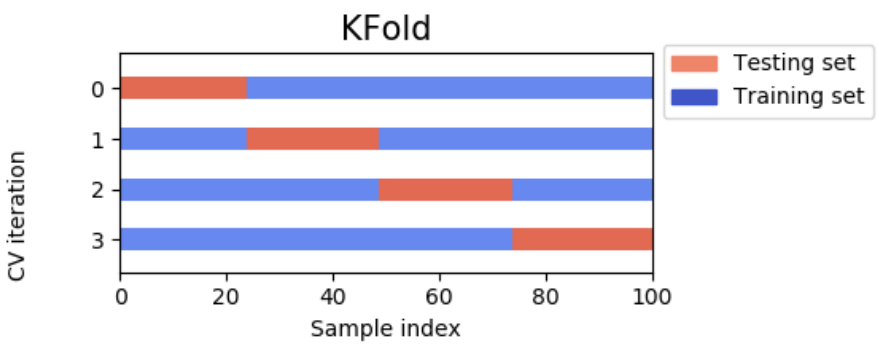
It is called exponential smoothing since

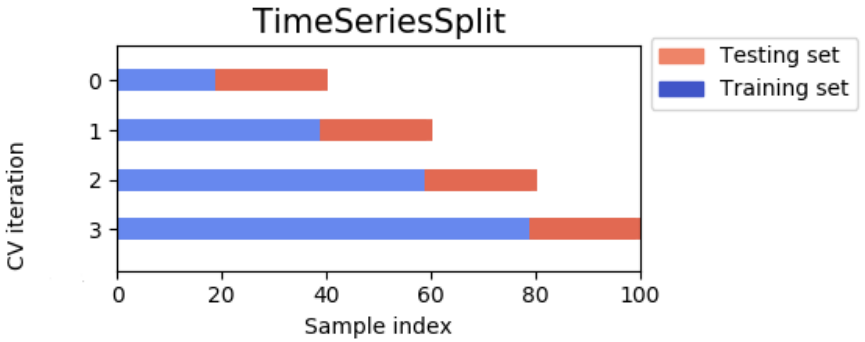
$$\begin{aligned} s_t &= \alpha y_t + (1-\alpha) s_{t-1} \\ &= \alpha y_t + \alpha(1-\alpha) y_{t-1} + (1-\alpha)^2 s_{t-2} \\ &= \alpha \left[y_t + (1-\alpha) y_{t-1} + (1-\alpha)^2 y_{t-2} + (1-\alpha)^3 y_{t-3} + \cdots + (1-\alpha)^{t-1} y_1 \right] + (1-\alpha)^t \end{aligned}$$

corresponding to exponentially weighted moving average

Need to adapt cross-validation techniques,

Time Series : Exponential Smoothing





$$\text{Optimal } \alpha \text{ ? } \alpha^\star \in \text{argmin}\left\{\sum_{t=2}^{T} \ell_2(y_t - {}_{t-1}\widehat{y}_t)\right\} \text{ (leave-one-out strategy)}$$

See Hyndman *et al.* (2008, Forecasting with Exponential Smoothing)

Exponential smoothing - Double

From time series (y_t) define a smooth version

$$\begin{cases} s_t = \alpha y_t + (1-\alpha)(s_{t-1} + b_{t-1}) s_t = \alpha y_t + (1-\alpha)(s_{t-1} + b_{t-1}) \\ b_t = \beta(s_t - s_{t-1}) + (1-\beta) b_{t-1} \, b_t = \beta(s_t - s_{t-1}) + (1-\beta) b_{t-1} \end{cases}$$

for some $\alpha \in (0,1)$, some trend $\beta \in (0,1)$ and starting points s_0 and b_0, $s_0 = y_0$ and $b_0 = y_1 - y_0$. Forecast is ${}_t\widehat{y}_{t+h} = s_t + h \cdot b_t$.

Time Series : Exponential Smoothing

Exponential smoothing - Seasonal with lag L (Holt-Winters)

From time series (y_t) define a smooth version

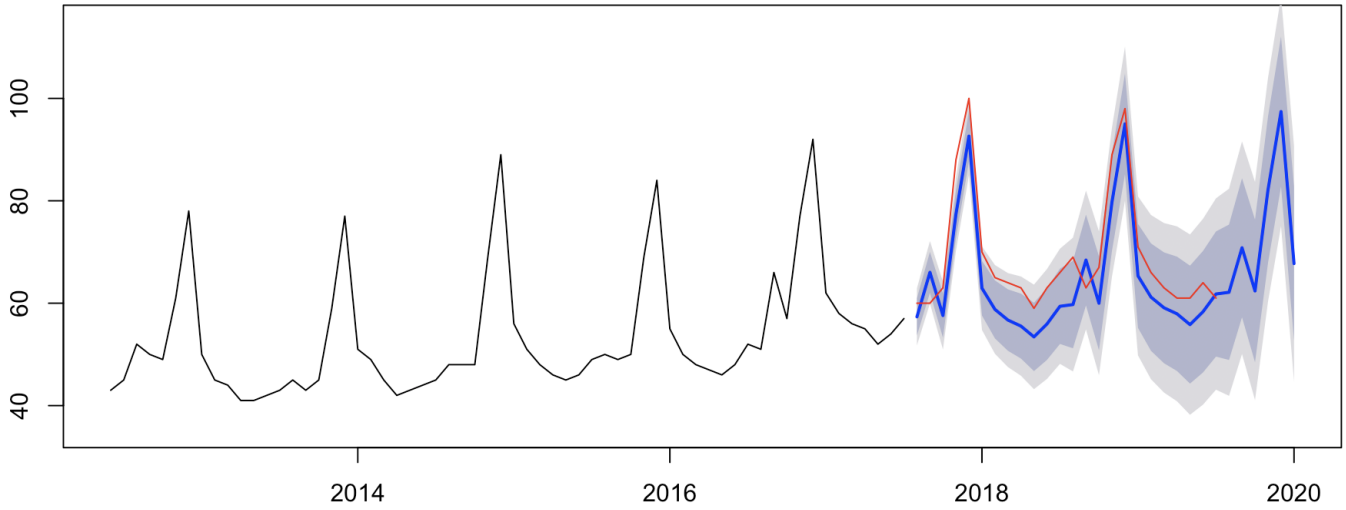
$$\begin{cases} s_t = \alpha\dfrac{x_t}{c_{t-L}} + (1-\alpha)(s_{t-1}+b_{t-1}) \\ b_t = \beta(s_t - s_{t-1}) + (1-\beta)b_{t-1} \\ c_t = \gamma\dfrac{y_t}{s_t} + (1-\gamma)c_{t-L} \end{cases}$$

for some $\alpha \in (0,1)$, some trend $\beta \in (0,1)$, some seasonal change smoothing factor, $\gamma \in (0,1)$ and starting points $s_0 = y_0$. Forecast is ${}_t\widehat{y}_{t+h} = (s_t + hb_t)c_{t-L+1+(h-1) \mod L}$.

See `stats::HoltWinters()`

Time Series : Exponential Smoothing

```
hw_fit <- stats::HoltWinters(trainY)
library(forecast)
plot(forecast(hw_fit, h=30))
lines(validY,col="red")
```

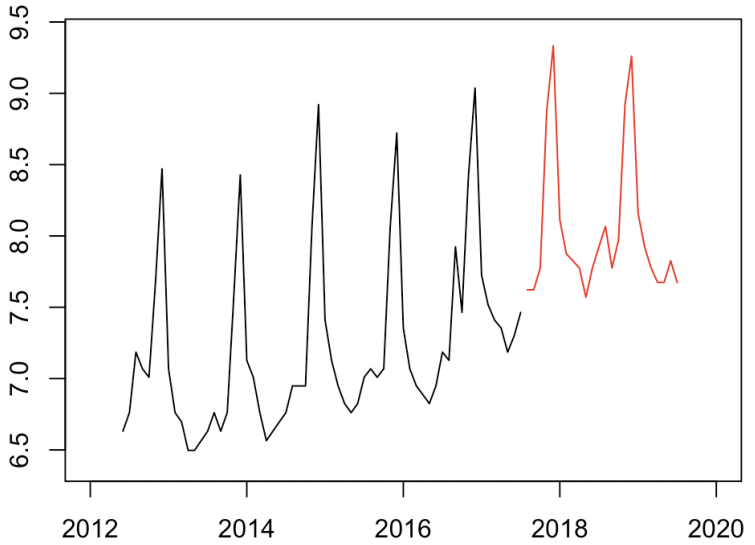


Time Series : State Space Models

See De Livera, Hyndman & Snyder (2011, Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing),based on Box-Cox transformation on y_t: $y_t^{(\lambda)} = \dfrac{y_t^\lambda - 1}{\lambda}$ if $\lambda \neq 0$ (otherwise $\log y_t$)

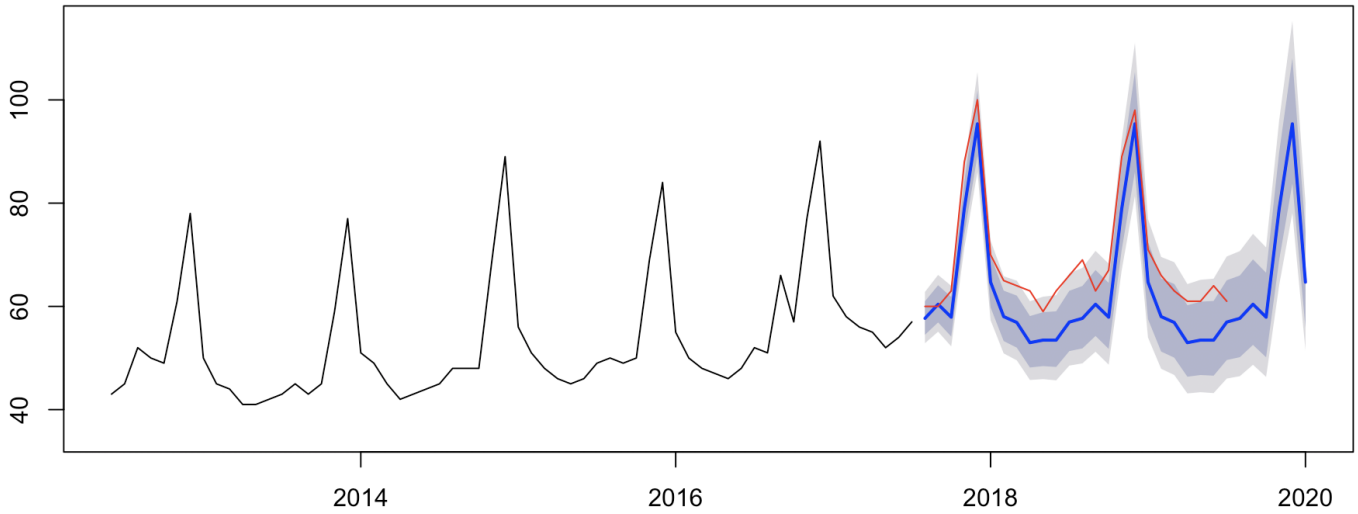
See `forecast::tbats`

```
1 library(forecast)
2 forecast::tbats(trainY)$lambda
3 [1] 0.2775889
```



Time Series : State Space Models

```
library(forecast)
tbats_fit <- tbats(trainY)
plot(forecast(tbats_fit, h=30))
lines(validY, col="red")
```



Time Series : State Space Models

Exponential smoothing state space model with Box-Cox transformation, ARMA errors, Trend and Seasonal components

$$y_t^{(\lambda)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{T} s_{t-m_i}^{(i)} + d_t \text{ where}$$

- (ℓ_t) is some local level, $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$
- (b_t) is some trend with damping, $b_t = \phi b_{t-1} + \alpha d_t$
- (d_t) is some ARMA process for the stationary component

$$d_t = \sum_{i=1}^{p} \varphi_i d_{t-i} + \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j}$$

- $(s_t^{(i)})$ is the i-th seasonal component

Time Series : State Space Models

Let $\boldsymbol{x}_t$ denote state variables (e.g. level, slope, seasonal).
Classical statistical approach: compute likelihood from errors $\varepsilon_1, \cdots, \varepsilon_T$
see `forecast::ets`

Innovations state space models

Let $\boldsymbol{x}_t = (s_t, b_t, \boldsymbol{c}_t)$ and suppose ε_t i.i.d. $\mathcal{N}(0, \sigma^2)$

State equation : $\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}) + g(\boldsymbol{x}_{t-1})\varepsilon_t$

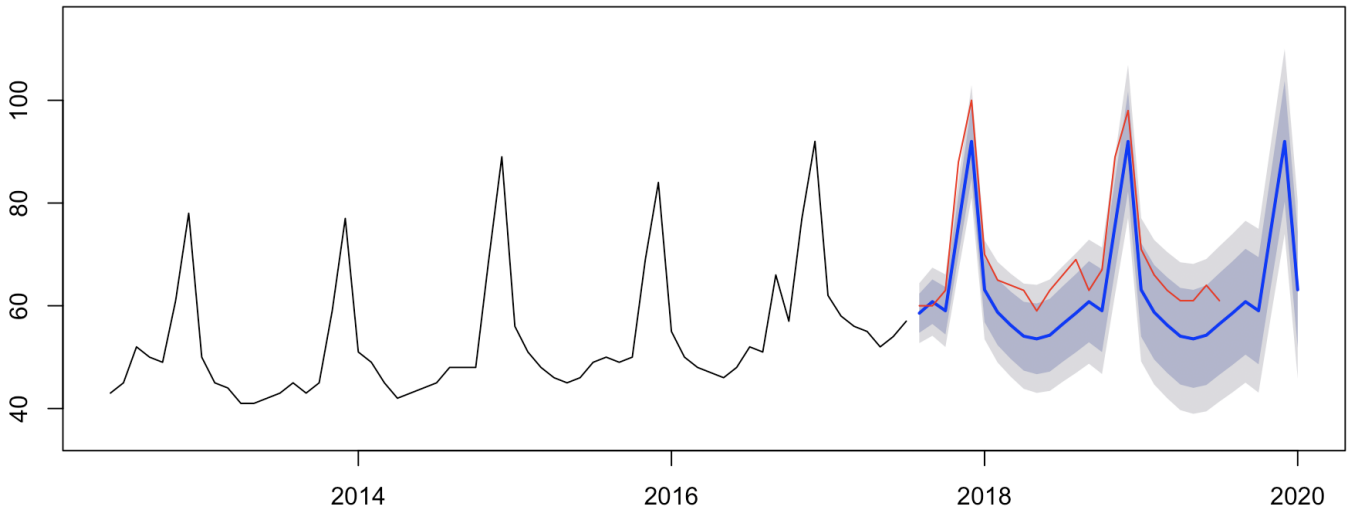
Observation equation : $y_t = \mu_t + \boldsymbol{e}_t = h(\boldsymbol{x}_{t-1}) + \sigma(\boldsymbol{x}_{t-1})\varepsilon_t$

Inference based on $\log \mathcal{L} = n \log \left(\sum_{t=1}^{T} \frac{\varepsilon_t^2}{\sigma(\boldsymbol{x}_{t-1})} \right) + 2 \sum_{t=1}^{T} \log |\sigma(\boldsymbol{x}_{t-1})|$

One can use time series cross-validation, valued on a rolling forecast origin

Time Series : State Space Models

```
1 library(forecast)
2 ets_fit <- forecast::ets(trainY)
3 plot(forecast(ets_fit, h=30))
```

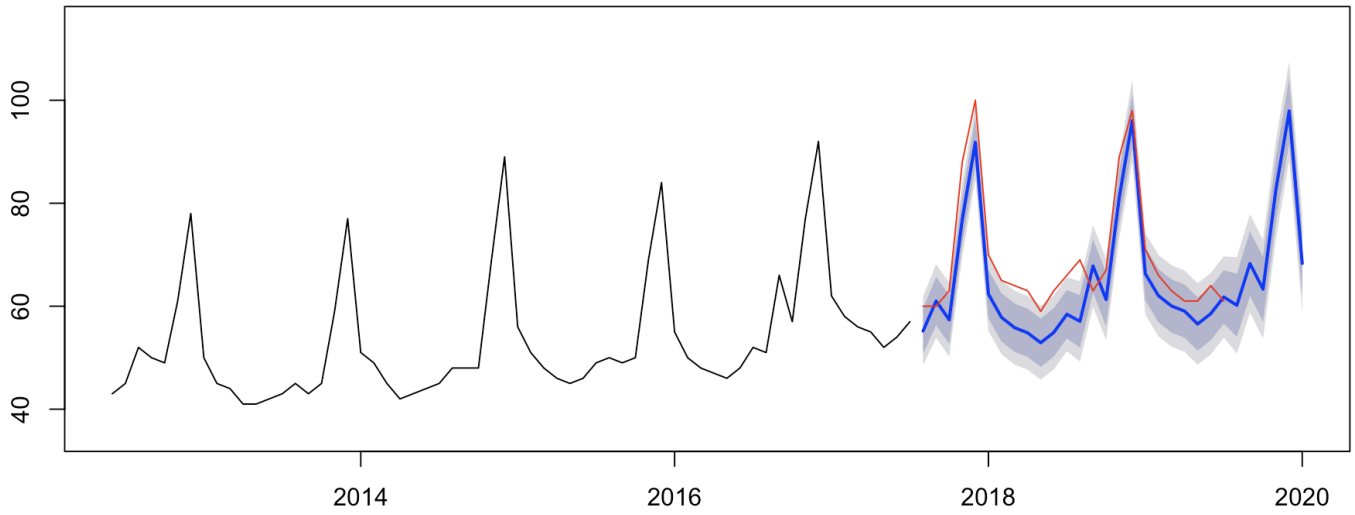


Time Series : Automatic ARIMA

Consider a general seasonal ARIMA process,

$$\Phi_s(L^s)\Phi(L)(1-L)^d(1-L^s)^{d_s}y_t = c + \Theta_s(L^s)\Theta(L)\varepsilon_t$$

See `forecast::autoarima(, include.drift=TRUE)` , "*Automatic algorithms will become more general - handling a wide variety of time series*" (Rob Hyndman)



Time Series : RNN (recurrent neural nets)

RNN : Recurrent neural network

Class of neural networks where connections between nodes form a directed graph along a temporal sequence.

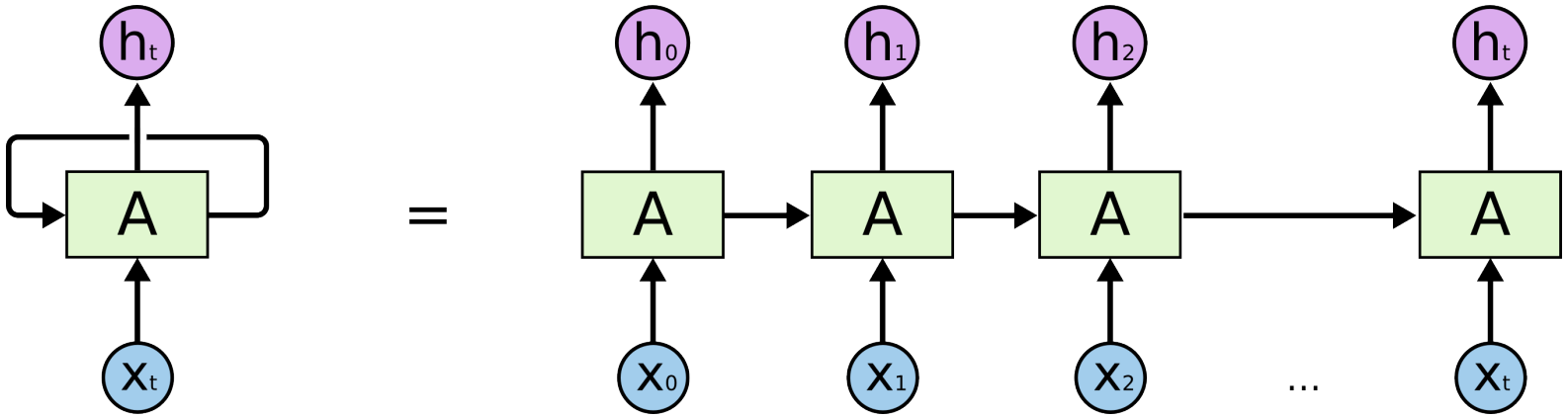
Recurrent neural networks are networks with loops, allowing information to persist.

Classical neural net $y_i = m(\boldsymbol{x}_i)$

Recurrent neural net $y_t = m(\boldsymbol{x}_t, y_{t-1}) = m(\boldsymbol{x}_t, m(\boldsymbol{x}_{t-1}, y_{t-2})) = \cdots$

Time Series : RNN (recurrent neural nets)

A is the neural net, h is the output (y) and x some covariates.

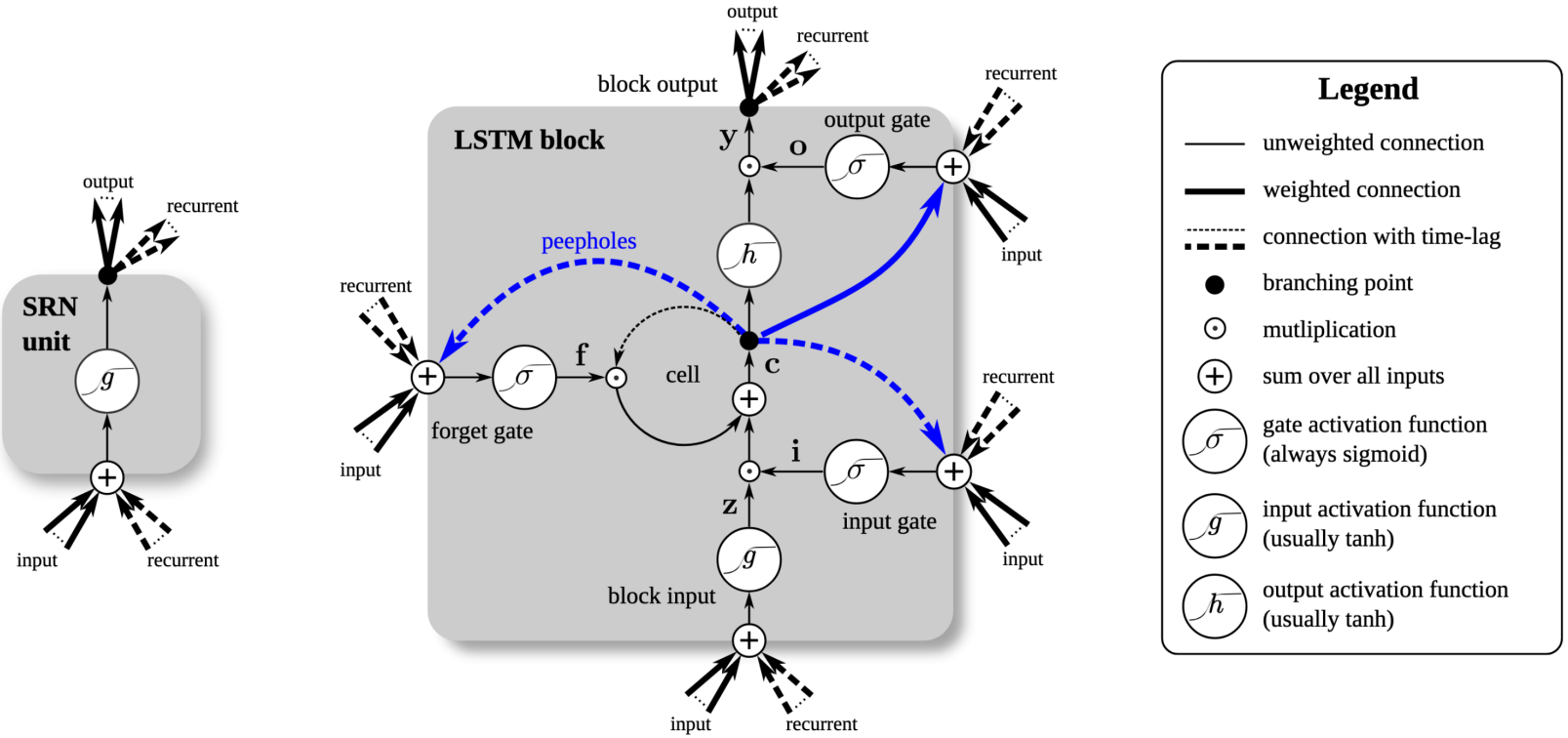


(source https://colah.github.io/)

See Sutskever (2017, Training Reccurent Neural Networks)

From recurrent networks to LSTM

Time Series : RNN and LTSM

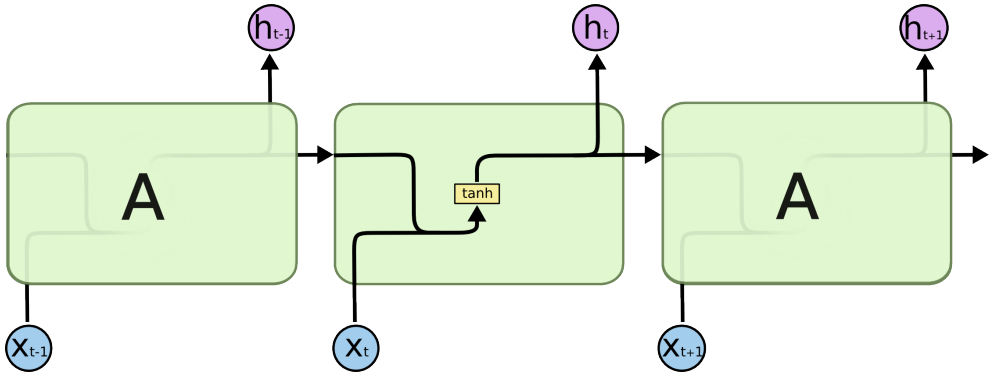


(source Greff *et al.* (2017, LSTM: A Search Space Odyssey))

see Hochreiter & Schmidhuber (1997, Long Short-Term Memory)

Time Series : RNN and LSTM

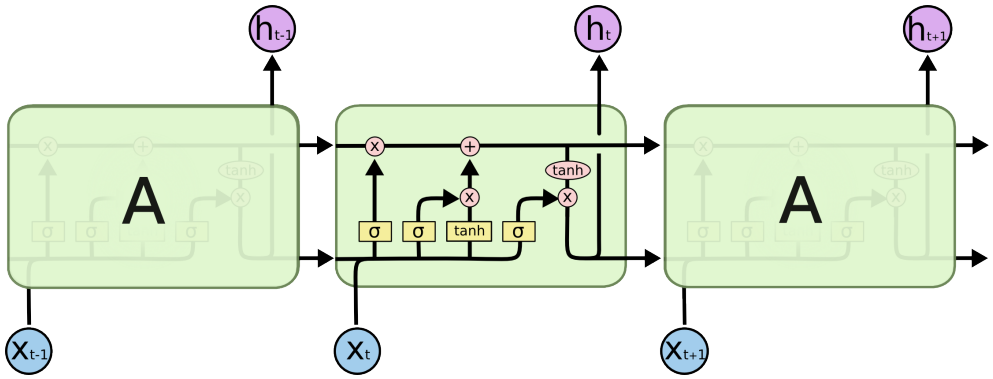
A classical RNN (with a single layer) would be



(source https://colah.github.io/)

"*In theory, RNNs are absolutely capable of handling such 'long-term dependencies'. A human could carefully pick parameters for them to solve toy problems of this form. Sadly, in practice, RNNs don't seem to be able to learn them*" see Benghio *et al.* (1994, Learning long-term dependencies with gradient descent is difficult)

Time Series : RNN and LSTM



"*RNNs can keep track of arbitrary long-term dependencies in the input sequences.The problem of "vanilla RNNs" is computational (or practical) in nature: when training a vanilla RNN using back-propagation, the gradients which are back-propagated can "vanish" (that is, they can tend to zero) "explode" (that is, they can tend to infinity), because of the computations involved in the process*" (from wikipedia)

Time Series : LSTM

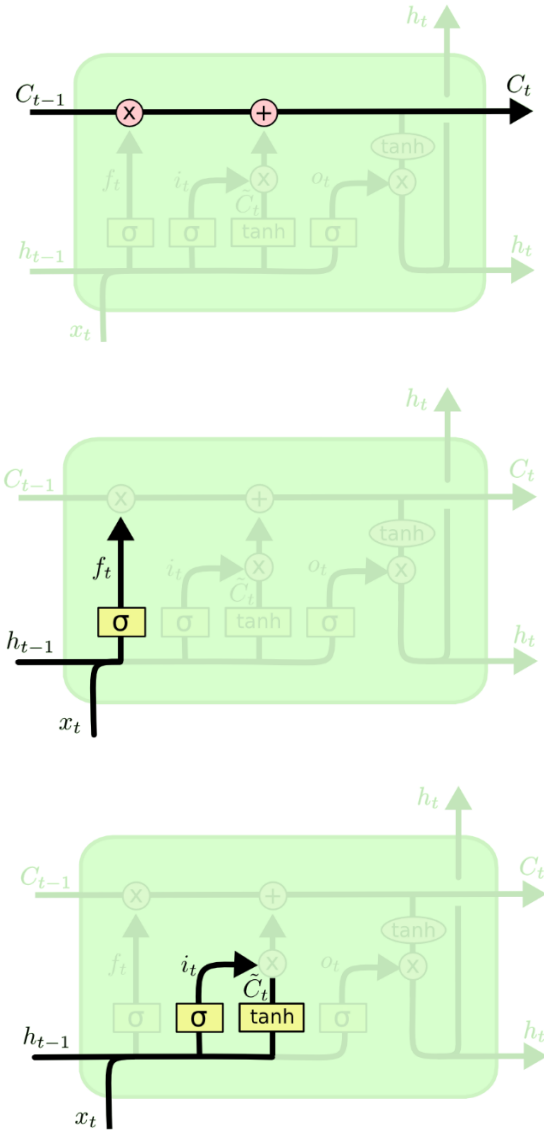
C is the long-term state

H is the short-term state

forget gate: $f_t = \text{sigmoid}(\boldsymbol{A}_f[h_{t-1}, x_t] + b_f)$

input gate: $i_t = \text{sigmoid}(\boldsymbol{A}_i[h_{t-1}, x_t] + b_i)$

new memory cell: $\tilde{c}_t = \tanh(\boldsymbol{A}_c[h_{t-1}, x_t] + b_c)$

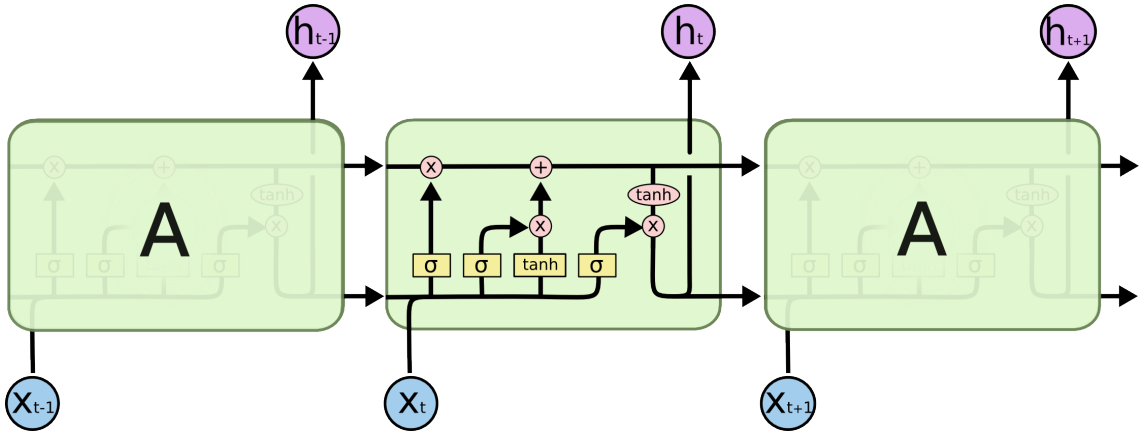


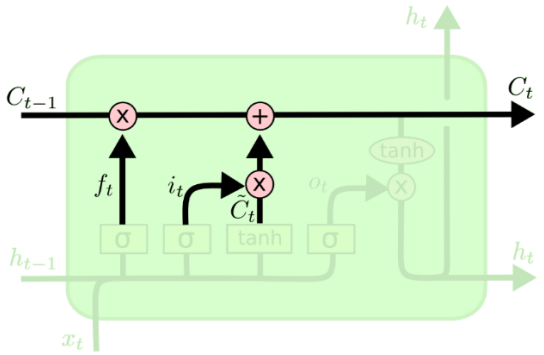
Time Series : LSTM

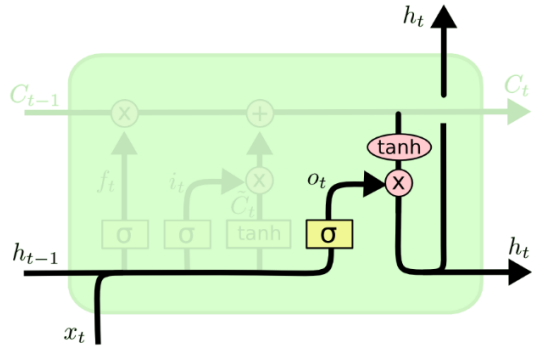
final memory cell: $c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t$

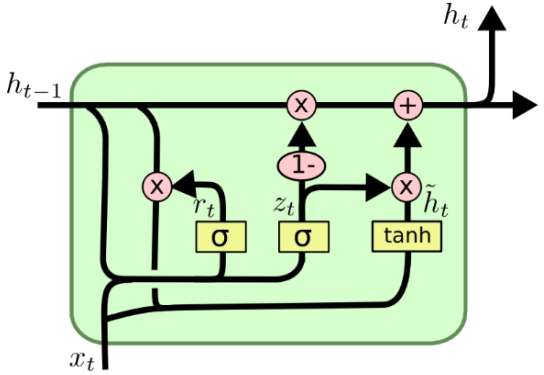
output gate: $o_t = \text{sigmoid}(\boldsymbol{A}_o[h_{t-1}, x_t] + b_o)$

$h_t = o_t \cdot \tanh(c_t)$









Elicitable Measures & Forecasting

"elicitable" means "*being a minimizer of a suitable expected score*", see Gneiting (2011) Making and evaluating point forecasts.

Elicitable function

T is an elicatable function if there exits a scoring function $S : \mathbb{R}\times\mathbb{R} \to [0,\infty)$

$$T(Y) = \underset{x\in\mathbb{R}}{\text{argmin}} \left\{ \int_{\mathbb{R}} S(x,y)dF(y) \right\} = \underset{x\in\mathbb{R}}{\text{argmin}} \left\{ \mathbb{E}\big[S(x,Y)\big] \text{ where } Y \sim F \right\}$$

Example: mean, $T(Y) = \mathbb{E}[Y]$ is elicited by $S(x,y) = \|x-y\|_{\ell_2}^2$

Example: median, $T(Y) = \text{median}[Y]$ is elicited by $S(x,y) = \|x-y\|_{\ell_1}$

Example: quantile, $T(Y) = Q_Y(\tau)$ is elicited by
$S(x,y) = \tau(y-x)_+ + (1-\tau)(y-x)_-$

Example: expectile, $T(Y) = E_Y(\tau)$ is elicited by
$S(x,y) = \tau(y-x)_+^2 + (1-\tau)(y-x)_-^2$

Forecasts and Predictions

Mathematical statistics is based on inference and testing, using probabilistic properties.

If we can reproduce past observations, it is supposed to proved good predictions.

Why not consider a collection of scenarios likely to occur on a given time horizon (drawing fro, a (predictive) probability distribution)

The closer forecast $\widehat{y}_t$ is to observed y_t, the better the model, either according to ℓ_1-norm - with $|\widehat{y}_t - y_t|$ - or to the ℓ_2-norm - with $(\widehat{y}_t - y_t)^2$.

If this is an interesting information about central tendency, it cannot be used to anticipate extremal events.

Forecasts and Predictions

More formally, we try to compare two very different objects : a function (the predictive probability distribution) and a real value number (the observed value).

Natural idea : introduce a score, as in Good (1952, Rational Decisions) or Winkler (1969, Scoring Rules and the Evaluation of Probability Assessors), used in meteorology by Murphy & Winkler (1987, A General Framework for Forecast Verification).

Let F denote the predictive distribution, expressing the uncertainty attributed to future values, conditional on the available information.

Probabilistic Forecasts

Notion of probabilistic forecasts, Gneiting & Raftery (2007 Strictly Proper Scoring Rules, Prediction, and Estimation).

In a general setting, we want to predict value taken by random variable Y.

Let F denote a cumulative distribution function.

Let $\mathcal{A}$ denote the information available when forecast is made.

F is the ideal forecast for Y given $\mathcal{A}$ if the law of $Y|\mathcal{A}$ has distribution F.

Suppose F continuous. Set $Z_F = F(Y)$, the probability integral transform of Y.

F is probabilistically calibrated if $Z_F \sim \mathcal{U}([0,1])$

F is marginally calibrated if $\mathbb{E}[F(y)] = \mathbb{P}[Y \leq y]$ for any $y \in \mathbb{R}$.

Probabilistic Forecasts

Observe that for a ideal forecast, $F(y) = \mathbb{P}[Y \leq y|\mathcal{A}]$, then

- $\mathbb{E}[F(y)] = \mathbb{E}[\mathbb{P}[Y \leq y|\mathcal{A}]] = \mathbb{P}[Y \leq y]$

This forecast is est marginally calibrated

- $\mathbb{P}[Z_F \leq z] = \mathbb{E}[\mathbb{P}[Z_F \leq z|\mathcal{A}]] = z$

This forecast is probabilistically calibrated

Suppose $\mu \sim \mathcal{N}(0,1)$. And that ideal forecast is $Y|\mu \sim \mathcal{N}(\mu, 1)$.

E.g. if $Y_t \sim \mathcal{N}(0,1)$ and $Y_{t+1} = y_t + \varepsilon_t \sim \mathcal{N}(y_t, 1)$.

One can consider $F = \mathcal{N}(0,2)$ as naïve forecast. This distribution is marginally calibrated, probabilistically calibrated and ideal.

One can consider F a mixture $\mathcal{N}(\mu, 2)$ and $\mathcal{N}(\mu \pm 1, 2)$ where "± 1" means $+1$ or -1 probability $1/2$, hesitating forecast. This distribution is probabilistically calibrated, but not marginally calibrated.

Probabilistic Forecasts

Indeed $\mathbb{P}[F(Y) \leq u] = u$,

$$\mathbb{P}[F(Y) \leq u] = \frac{\mathbb{P}[\Phi(Y) \leq u] + \mathbb{P}[\Phi(Y+1) \leq u]}{2} + \frac{\mathbb{P}[\Phi(Y) \leq u] + \mathbb{P}[\Phi(Y-1) \leq u]}{2}$$

One can consider $F = \mathcal{N}(-\mu, 1)$. This distribution is marginally calibrated, but not probabilistically calibrated.

In practice, we have a sequence (Y_t, F_t) of pairs, $(\boldsymbol{Y}, \boldsymbol{F})$.

The set of forecasts $\boldsymbol{F}$ is said to be performant if for all t, predictive distributions F_t are precise (sharpness) and well-calibrated.

Precision is related to the concentration of the predictive density around a central value (uncertainty degree).

Calibration is related to the coherence between predictive distribution F_t and observations y_t.

Probabilistic Forecasts

Calibration is poor if 80%-confidence intervals (implied from predictive distributions, i.e. $\left[F_t^{-1}(\alpha), F_t^{-1}(1-\alpha)\right]$) do not contain y_t's about 8 times out of 10.

To test marginal calibration, compare the empirical cumulative distribution function

$$\widehat{G}(y) = \lim_{n\to\infty} \frac{1}{n}\mathbf{1}_{Y_t\leq y}$$

and the average of predictive distributios

$$\overline{F}(y) = \lim_{n\to\infty} \frac{1}{n}\sum_{t=1}^{n} F_t(y)$$

To test probabilistic calibration, test if sample $\{F_t(Y_t)\}$ has a uniform distribution - PIT approach, see Dawid (1984, Present Position and Potential Developments: The Prequential Approach).

Probabilistic Forecasts

One can also consider a score $S(F, y)$ for all distribution F and all observation y.

The score is said to be proper if

$$\forall F, G, \mathbb{E}[S(G, Y)] \leq \mathbb{E}[S(F, Y)] \text{ where } Y \sim G.$$

In practice, this expected value is approximated using $\frac{1}{n}\sum_{t=1}^{n} S(F_t, Y_t)$

One classical rule is the logarithmic score $S(F, y) = -\log[F'(y)]$ if F is (abs.) continuous.

Another classical rule is the continuous ranked probability score (CRPS, see Hersbach (2000, Decomposition of the Continuous Ranked Probability Score for Ensemble Prediction Systems))

$$S(F, y) = \int_{-\infty}^{+\infty} (F(x) - \mathbf{1}_{x \geq y})^2 dx = \int_{-\infty}^{y} F(x)^2 + \int_{y}^{+\infty} (F(x) - 1)^2 dx$$

Probabilistic Forecasts

with empirical version

$$\widehat{S} = \frac{1}{n}\sum_{t=1}^{n} S(F_t, y_t) = \frac{1}{n}\sum_{t=1}^{n}\int_{-\infty}^{+\infty} (F_t(x) - \mathbf{1}_{x\geq y_t})^2 dx$$

studied in Murphy (1970, The ranked probability score and the probability score: a comparison).

This rule is proper since

$$\mathbb{E}[S(F,Y)] = \int_{-\infty}^{\infty} \mathbb{E}\big[F(x) - \mathbf{1}_{x\geq Y}\big]^2 dx$$

$$= \int_{-\infty}^{\infty} \big[[F(x) - G(x)]^2 + G(x)[1-G(x)]\big]^2 dx$$

is minimal when $F = G$.

Probabilistic Forecasts

If F corresponds to the $\mathcal{N}(\mu, \sigma^2)$ distribution

$$S(F, y) = \sigma \left[\frac{y-\mu}{\sigma} \left(2\Phi \left(\frac{y-\mu}{\sigma} \right) - 1 \right) + 2\frac{y-\mu}{\sigma} - \frac{1}{\sqrt{\pi}} \right]$$

Observe that

$$S(F, y) = \mathbb{E}\big|X - y\big| - \frac{1}{2}\mathbb{E}\big|X - X'\big| \text{ où } X, X' \sim F$$

(where X and X' are independent versions), cf Gneiting & Raftery (2007, Strictly Proper Scoring Rules, Prediction, and Estimation.

If we use for F the empirical cumulative distribution function $\widehat{F}_n(y) = \dfrac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{y_i \le y}$ then

$$S(\widehat{F}_n, y) = \frac{2}{n}\sum_{i=1}^{n} (y_{i:n} - y) \left(\mathbf{1}_{y_{i:n} \le y} - \frac{i - 1/2}{n} \right)$$

Probabilistic Forecasts

Consider a Gaussian $AR(p)$ time series,

$$Y_t = c + \varphi_1 Y_{t-1} + \cdots + \varphi_p Y_{t-p} + \varepsilon_t, \text{ with } \varepsilon_t \sim \mathcal{N}(0, \sigma^2)$$

then forecast with horizon 1 yields

$$F_t \sim \mathcal{N}({}_{t-1}\widehat{Y}_t, \sigma^2)$$

where ${}_{t-1}\widehat{Y}_t = c + \varphi_1 Y_{t-1} + \cdots + \varphi_p Y_{t-p}$.

Probabilistic Forecasts

Suppose that Y can be explained by covariates $\boldsymbol{x} = (x_1, \cdots, x_m)$. Consider some kernel based conditional density estimation

$$\widehat{p}(y|\boldsymbol{x}) = \frac{\widehat{p}(y, \boldsymbol{x})}{\widehat{p}(\boldsymbol{x})} = \frac{\sum_{i=1}^n K_h(y - y_i)K_h(\boldsymbol{x} - \boldsymbol{x}_i)}{\sum_{i=1}^n K_h(\boldsymbol{x} - \boldsymbol{x}_i)}$$

In the case of a linear model, there exists $\boldsymbol{\theta}$ such that $\widehat{p}(y|\boldsymbol{x}) = \widehat{p}(y|\boldsymbol{\theta}^\mathsf{T}\boldsymbol{x})$, and

$$\widehat{p}(y|\boldsymbol{\theta}^\mathsf{T}\boldsymbol{x} = s) = \frac{\sum_{i=1}^n K_h(y - y_i)K_h(s - \boldsymbol{\theta}^\mathsf{T}\boldsymbol{x}_i)}{\sum_{i=1}^n K_h(s - \boldsymbol{\theta}^\mathsf{T}\boldsymbol{x}_i)}$$

Parameter $\boldsymbol{\theta}$ can be estimated using a proxy of the log-likelihood

$$\widehat{\boldsymbol{\theta}} = \text{argmax}\left\{\sum_{i=1}^n \log \widehat{p}(y_i|\boldsymbol{\theta}^\mathsf{T}\boldsymbol{x}_i)\right\}$$

Time Series : Stacking

See Clemen (1989, Combining forecasts: A review and annotated bibliography)

See `opera::oracle(Y = Y, experts = X, loss.type ='square', model ='convex')`

(on Online Prediction by Expert Aggregation)

Natural Language Processing & Probabilistic Language Models

Idea : $\mathbb{P}[\text{today is Wednesday}] > \mathbb{P}[\text{today Wednesday is}]$

$\mathbb{P}[\text{today is Wednesday}] > \mathbb{P}[\text{today is Wendy}]$

Google

today is

- today is
- today is gonna be the day
- today is your birthday
- today is a good day to die
- today is a good day to have a good day

Google

tomorrow is

- tomorrow is
- tomorrow is another day
- tomorrow is not promised
- tomorrow is the future yesterday is history
- tomorrow is the first day of the rest of your life

E.g. try to predict the missing word I grew up in France, I speak fluent _____

Natural Language Processing & Probabilistic Language Models

Use of the chain rule

$$\mathbb{P}[A_1, A_2, \cdots, A_n] = \prod_{i=1}^{n} \mathbb{P}[A_i | A_1, A_2, \cdots, A_{i-1}]$$

$$\mathbb{P}(\text{the wine is so good})$$

$$= \mathbb{P}(\text{the}) \cdot \mathbb{P}(\text{wine}|\text{the}) \cdot \mathbb{P}(\text{is}|\text{the wine}) \cdot \mathbb{P}(\text{so}|\text{the wine is}) \cdot \mathbb{P}(\text{good}|\text{the wine is so})$$

Markov assumption & k-gram model

$$\mathbb{P}[A_1, A_2, \cdots, A_n] \sim \prod_{i=1}^{n} \mathbb{P}[A_i | A_{i-k}, \cdots, A_{i-1}]$$