LEC002 Demand Forecasting

VG441 SS2021

Cong Shi
Industrial & Operations Engineering
University of Michigan

Plan for forecasting

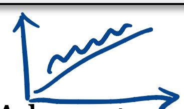
- basic time series model
  - exponential smoothing
  - ARIMA
- linear regression
- logistic regression
  - multinomial
- base diffusion model
  - predicting new products
- decision trees, gradient boosting, XGboost

VG441 TA

Mr. Shunyi Zhu

- TA office hours: Wednesday Night 8:00-9:30pm
- Location: 326D
- Background: research on optimization algorithms in crowdsourcing and energy-efficient network node deployment strategies.
- Hobbies: Crayon Shin-chan

Properties Time series



- ***Trend***: A long-term increase or decrease in the data. This can be seen as a slope (doesn't have to be linear) roughly going through the data.

length of 1 cycle

- ***Seasonality***: A time series is said to be seasonal when it is affected by seasonal factors (hour of day, week, month, year, etc.). Seasonality can be observed with nice cyclical patterns of fixed frequency. less predictable

- ***Cyclicity***: A cycle occurs when the data exhibits rises and falls that are not of a fixed frequency. These fluctuations are usually due to economic conditions, and are often related to the "business cycle". The duration of these fluctuations is usually at least 2 years.

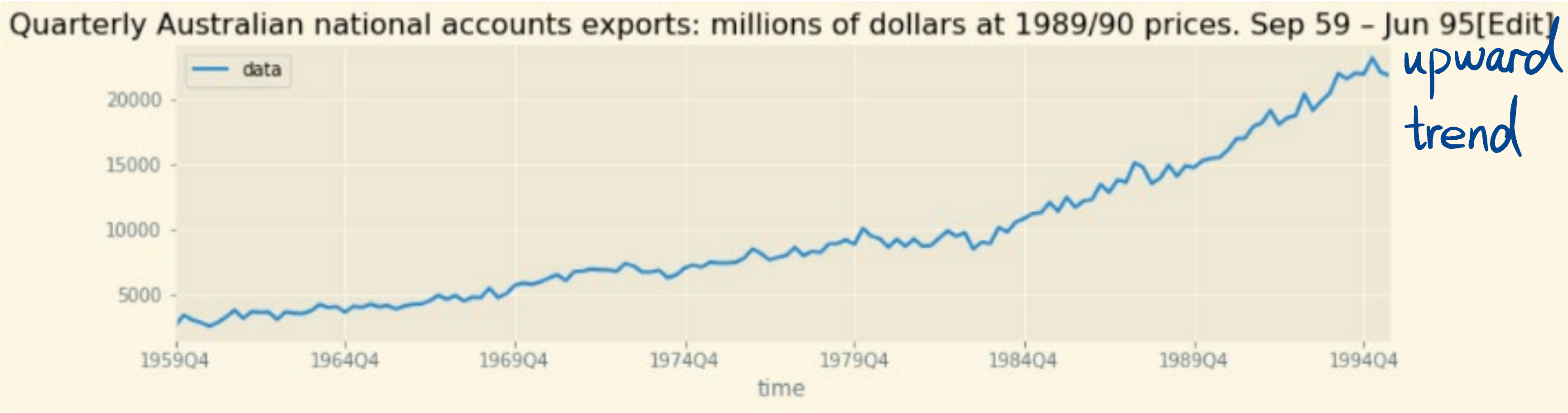
- ***Residuals***: Each time series can be decomposed in two parts:
  - A forecast, made up of one or several *forecasted* values
  - Residuals. They are the difference between an observation and its predicted value at each time step. Remember that

error

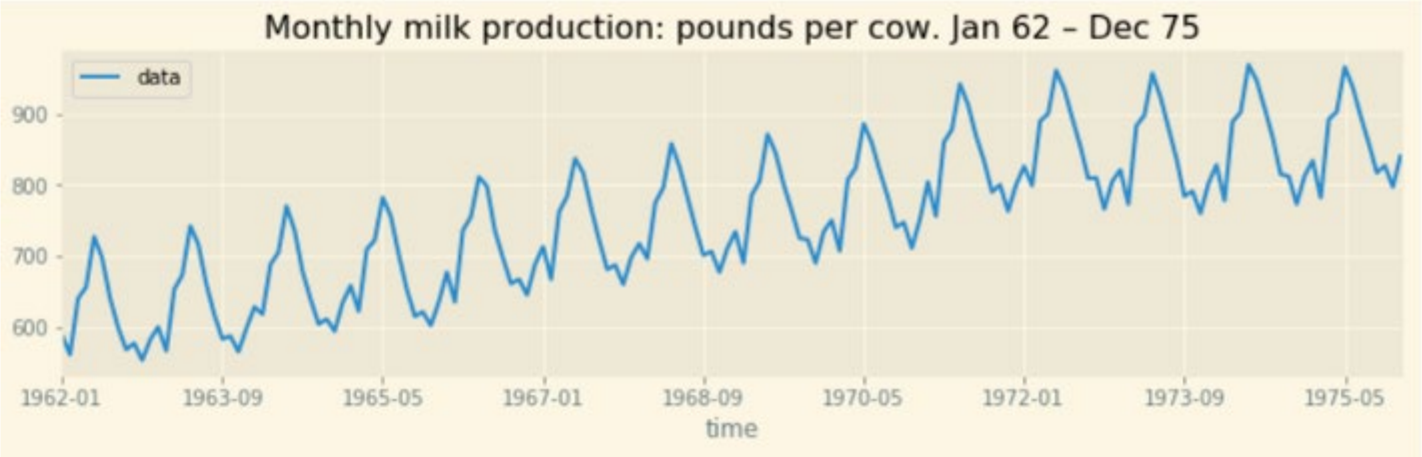
Value of series at time t = Predicted value at time t + Residual at time t

true value

Examples



upward trend



seasonal

Examples

cyclic

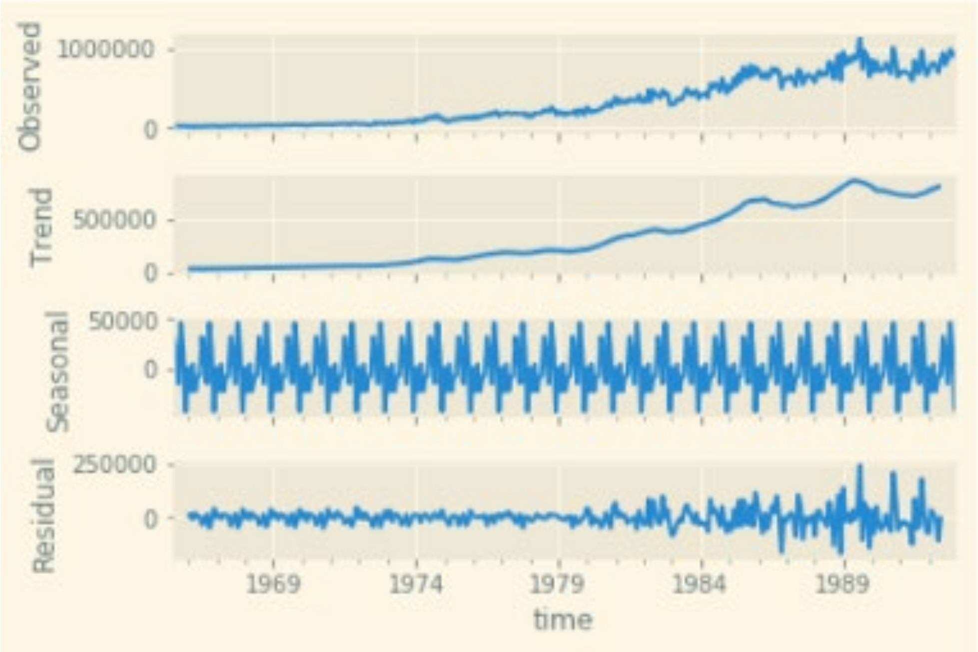
has no cycles, pretty much flat

Decomposition of Time Series

任何一个real time series都可以被分解为

Each time series can be thought as a mix between several parts :

- A trend (upward or downwards movement)
- A seasonal component
- Residuals we expect the stationary distribution residual



Simple Average

- Stationary model 不变的 $D_t = I + \epsilon_t$ demand base signal random noise (normal distribution) ~N(0, 6²)

- Static forecast $\hat{y} = \frac{\sum_{t=1}^{N} D_t}{N}$ simple moving average

- Derived based on minimizing MSE mean square error

residual

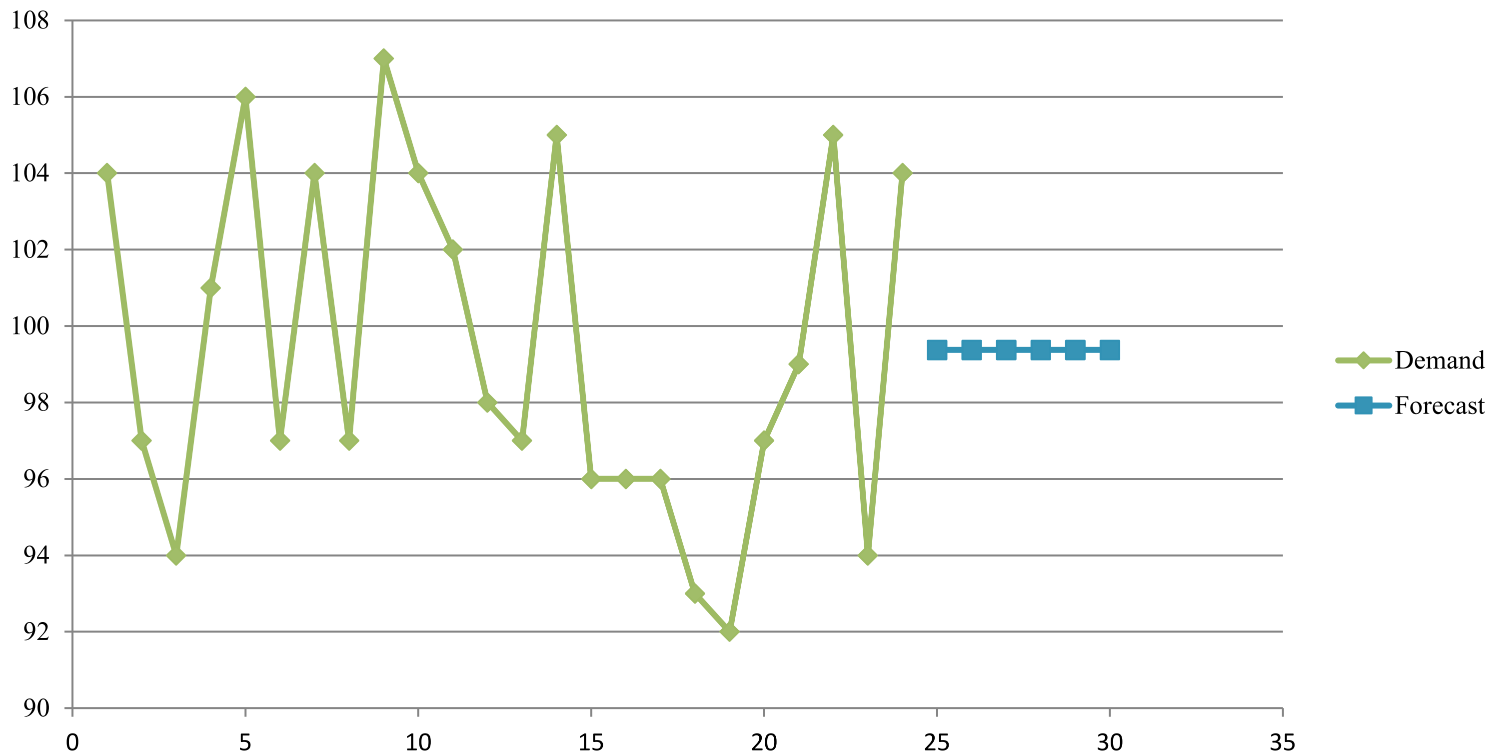
$$e_t = D_t - \hat{y}$$

true predicted

minimize the sum of residual square. (Σ residual²)

$$\frac{d\left(\sum_{t=1}^{N} e_t^2\right)}{d\hat{y}} = \frac{d\left[\sum_{t=1}^{N} (d_t - \hat{y})^2\right]}{d\hat{y}} = -2\sum_{t=1}^{N} (d_t - \hat{y}) = 0$$

Simple Average Model



Moving Average (MA)

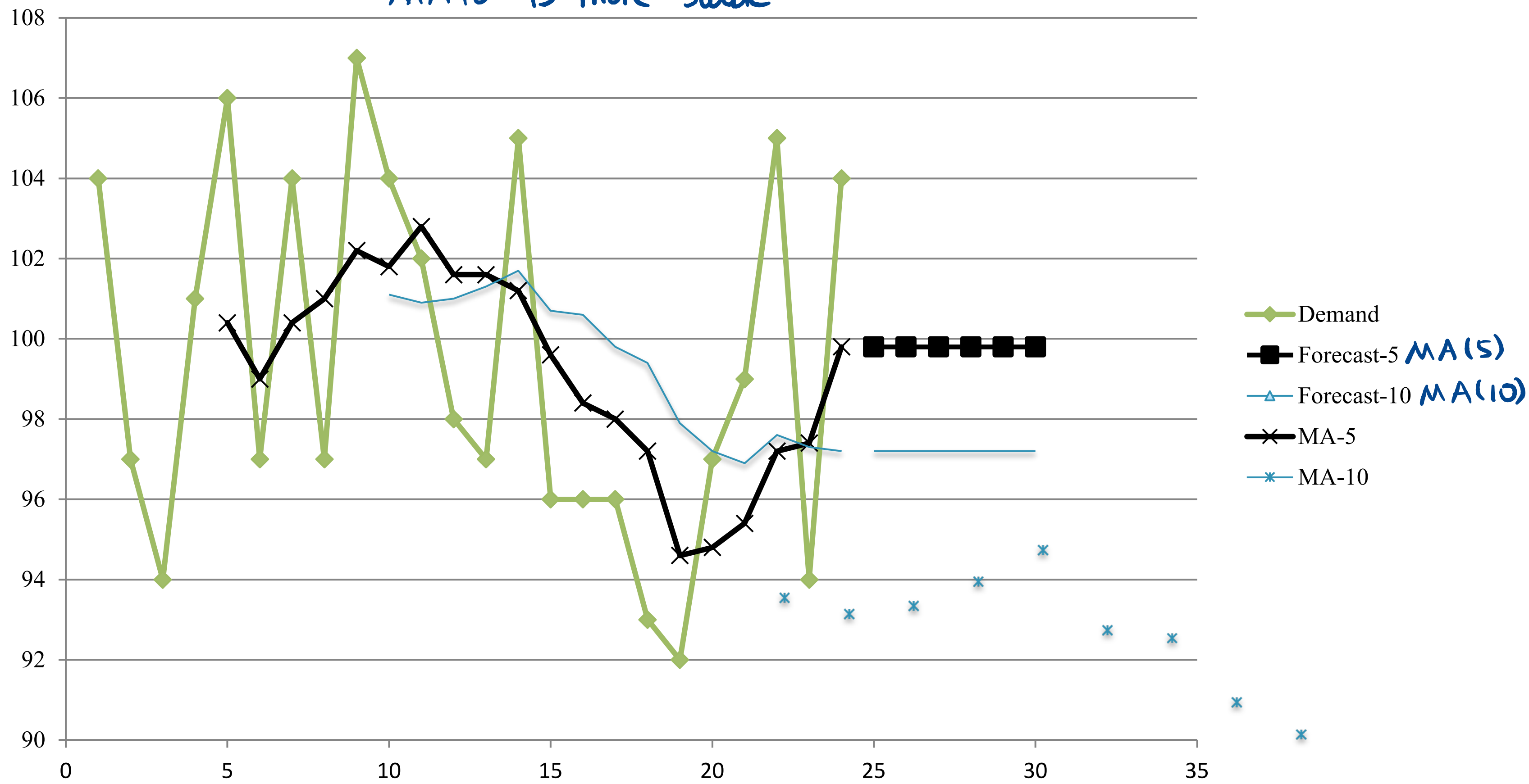
- Average only the most recent data points

$$y_t = \frac{1}{N} \sum_{i=t-N}^{t-1} D_i$$

- Smooth out noise
- Can respond to change in process

Moving Average (MA)

MA 5 is more sensitive
MA 10 is more stable



Weighted Moving Average

- A generalization of MA with weights

weight for data pt i

$$y_t = \frac{\sum_{i=t-N}^{t-1} w_i D_i}{\sum_{i=t-N}^{t-1} w_i}$$

SMA: simple moving average

$w_i \equiv 1 \ \forall i$

e.g.,

$$w_{t-1} = N, w_{t-2} = N-1, \ldots, w_{t-N} = 1$$

Exponential Smoothing

first point 不能 forecast
$y_1 = D_1$

- Adjust forecast based on the recent data point

previous demand data   previous forecast

$$y_t = \alpha D_{t-1} + (1-\alpha) y_{t-1} \quad \alpha \in (0,1)$$

$\alpha > \frac{1}{2}$: plays more emphasis on past demand

$\alpha < \frac{1}{2}$: plays more emphasis on past forecast

- It is a weighted average of all historical data points, with the weight decreasing exponentially with age

$$y_{t-1} = \alpha D_{t-2} + (1-\alpha) y_{t-2}$$

$$y_t = \alpha D_{t-1} + \alpha(1-\alpha) D_{t-2} + (1-\alpha)^2 y_{t-2}$$

$$y_t = \sum_{i=0}^{\infty} \alpha (1-\alpha)^i D_{t-i-1} = \sum_{i=0}^{\infty} \alpha_i D_{t-i-1}$$

对于之前demand的依赖在变小

$\alpha_i = \alpha(1-\alpha)^i$

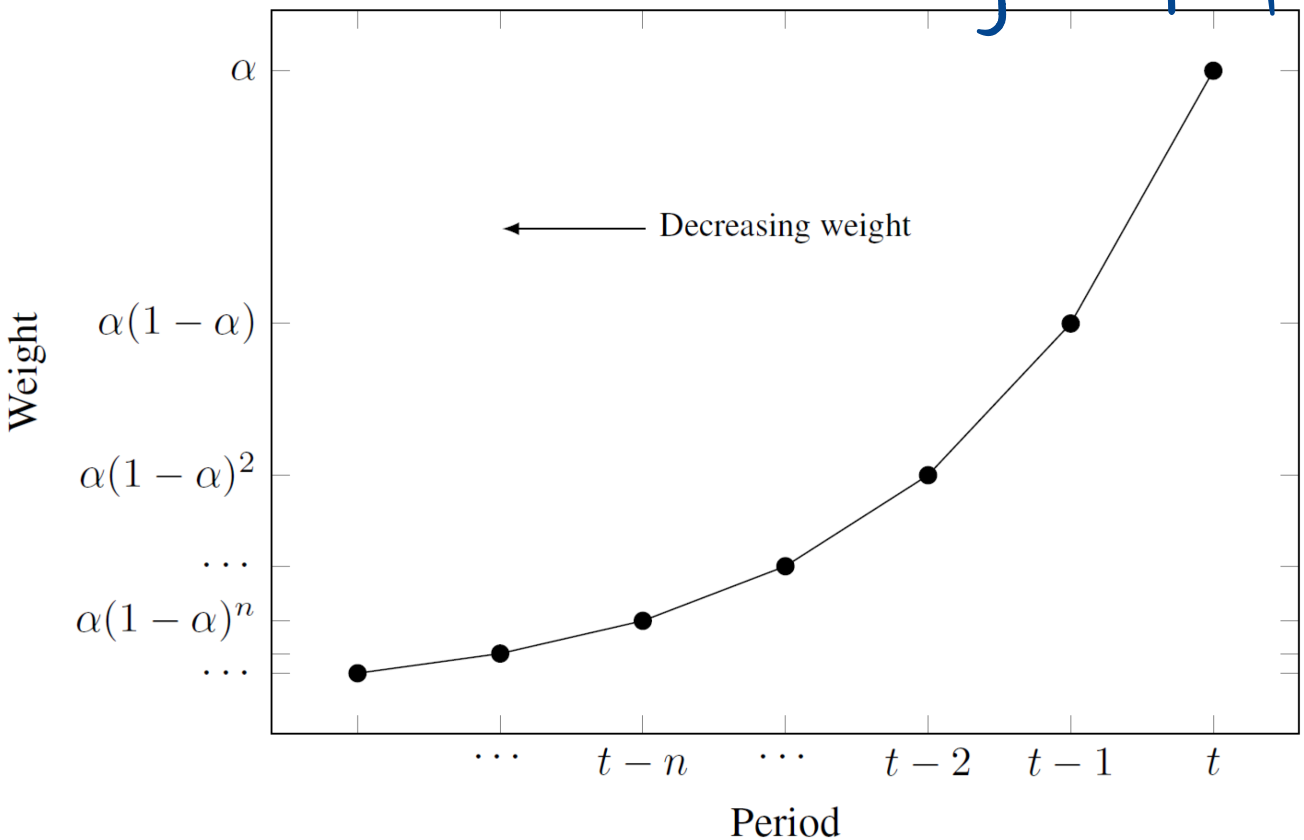
$\alpha \sim 0 \Rightarrow \alpha \underline{\underline{e^{-\alpha i}}}$

describe the weight

α_i is decreasing

Exponential Smoothing

weight drop expentially fast



Double Exponential Smoothing (Holt)

可以 predict trend

Double exponential smoothing can be used to forecast demands with a linear trend

initialize

$$D_t = I + \underbrace{tS}_{\text{capture the trend}} + \epsilon_t \qquad D_1, S_0 = 0$$

The predictor consists of base and slope:

forecast base slope

$$y_{t+1} = I_t + S_t$$

keep track of the base

$$I_t = \alpha D_t + (1-\alpha)\left(I_{t-1} + S_{t-1}\right)$$

$$S_t = \beta\left(I_t - I_{t-1}\right) + (1-\beta) S_{t-1}$$

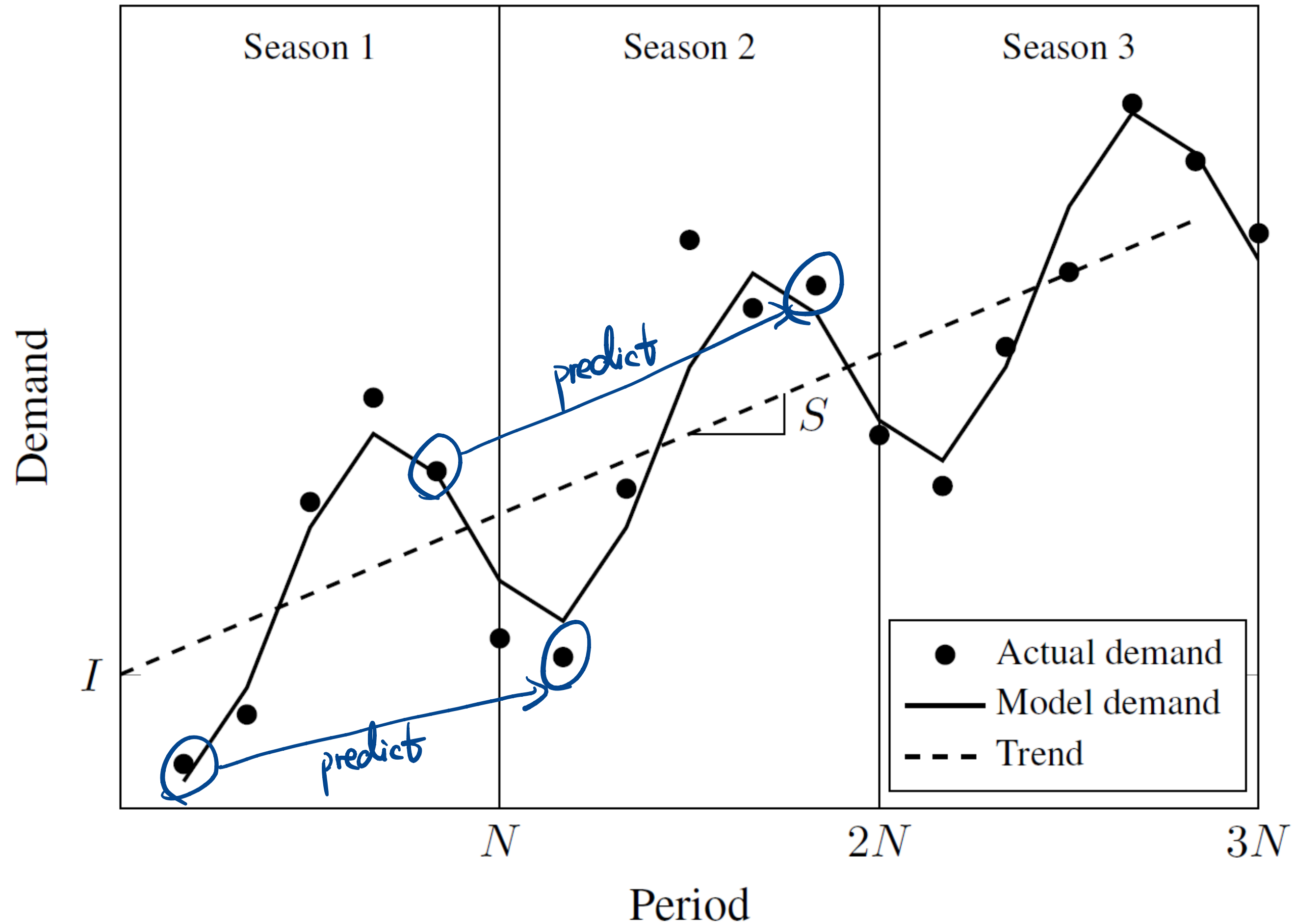
slope recent trend previous trend

Alpha is the smoothing constant and Beta is the trend constant

Triple Exponential Smoothing (Holt-Winters)

可以 capture seasonality

- Random demands with trend and seasonality



Triple Exponential Smoothing (Holt-Winters)

in the beginning, you could assume that

Demand model

slope   noise

$$D_t = (I + tS)c_t + \epsilon_t \qquad \sum c_t = N$$

base   adjustment for seasonality   length of one cycle

example: N=2

$C_1=1.5$ winter $C_2=0.5$ summer

The predictor

$$y_{t+1} = (I_t + S_t)\, c_{t+1-N}$$

= 如何选择 best N

Basic idea is to “de-trend” and “de-seasonalize”

base

$$I_t = \alpha \frac{D_t}{c_{t-N}} + (1-\alpha)(I_{t-1} + S_{t-1})$$

slope

$$S_t = \beta\,(I_t - I_{t-1}) + (1-\beta)S_{t-1}$$

$$c_t = \gamma \frac{D_t}{I_t} + (1-\gamma)c_{t-N}$$

adjustment   how much inflation or deflation you have

you must have the average $(D_1, \ldots, D_N)$ first season $c_i = \frac{D_i}{(Avg(D))}$

Python Time!

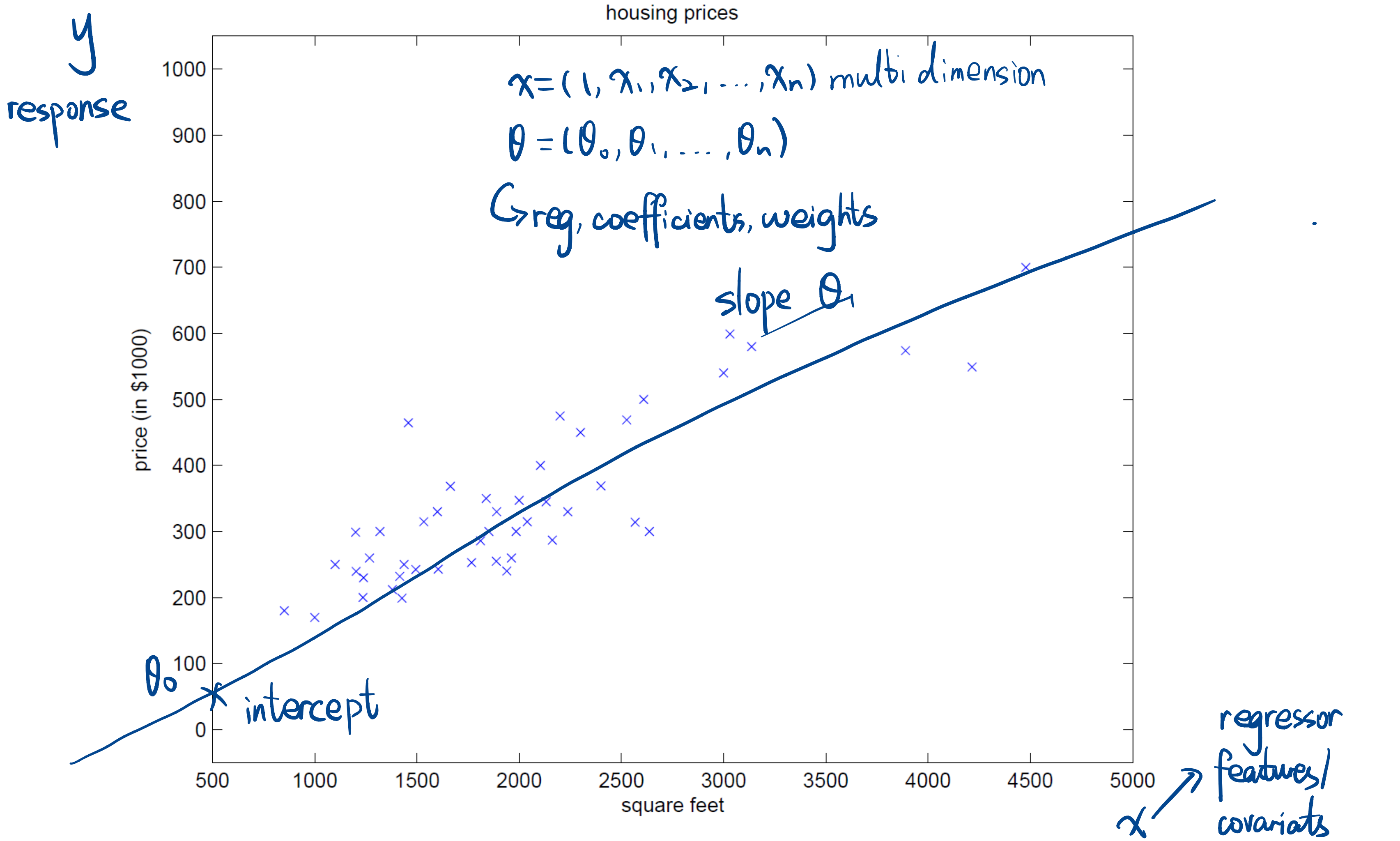
- statsmodels.tsa.seasonal 已经有 package 了

- statsmodels.tsa.holtwinters

Linear Regression

best line that minimize MSE
mean
square error

- "Best fitting line" $y = \theta_0 + \theta_1 x = h_\theta(x)$



Linear Regression

$\theta = (\theta_0, \ldots, \theta_n)$

$x = (x_0, x_1, \ldots, x_n)$

$\shortparallel$
1

- (Linear) Hypothesis Function:

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n = \theta \cdot x$$

MSE

- Minimizing the least-squares cost:

$$\min_{\theta} \left(J(\theta_{0\ldots n}) = \frac{1}{2m} \sum_{i=1}^{m} \left(\underbrace{h_\theta\left(x^{(i)}\right)}_{\text{predicted value}} - \overset{\text{real value}}{y^{(i)}} \right)^2 \right) \quad \left. \begin{matrix} x^{(1)} \\ \vdots \\ x^{(m)} \end{matrix} \right\} m \text{ data pts}$$

m data points

寻找best h_θ

Matrix Derivation

form of θ

- Turn everything into matrix notation

$$h_\theta(x) = \theta^T x \qquad \theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \dots \\ \theta_n \end{pmatrix} \in \mathbb{R}^{n+1} \qquad x = \begin{pmatrix} x_0 \triangleq 1 \\ x_1 \\ \dots \\ x_n \end{pmatrix} \in \mathbb{R}^{n+1}$$

求 θ

- Design matrix

$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(m-1)} \\ x^{(m)} \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(1)} & \dots & \dots & \dots & x_n^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{(m)} & \dots & \dots & \dots & x_n^{(m)} \end{bmatrix}$$

m data points

Normal Equation 警告.

caveat: (X^TX) should be invertible

$\Leftrightarrow$ nonsingular

$\Leftrightarrow$ full rank

- Cost function

$$J(\theta) = \underbrace{\frac{1}{2m}}_{\text{constant}}(X\theta - y)^T(X\theta - y)$$

- Throwing out the constant

$$J(\theta) = \theta^T X^T X\theta - 2(X\theta)^T y + y^T y$$

- The "famous" normal equation

$$\frac{\partial J}{\partial \theta} = 2X^T X\theta - 2X^T y = 0$$

$$X^T X\theta = X^T y$$

$$\theta = (X^T X)^{-1} X^T y$$

python: generalized inverse

review of matrix calculus

$$(A \cdot B)^T = B^T A^T \qquad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \qquad B = \begin{pmatrix} & \end{pmatrix}$$

$$\frac{\partial x^T B}{\partial x} = B \qquad \frac{\partial x^T B x}{\partial x} = 2Bx \text{ (B symmetric)}$$

example: $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ $\quad B = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$

$$\Rightarrow x^T B = (\underbrace{x_1 + 3x_2}_{f_1}, \underbrace{2x_1 + 4x_2}_{f_2})$$

$$\Rightarrow \frac{\partial (x^T B)}{\partial x} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} \\ \frac{\partial f_1}{\partial x_2} & \frac{\partial f_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = B$$

Residuals 一种 check residual 的方法,

- The residual should be normally distributed

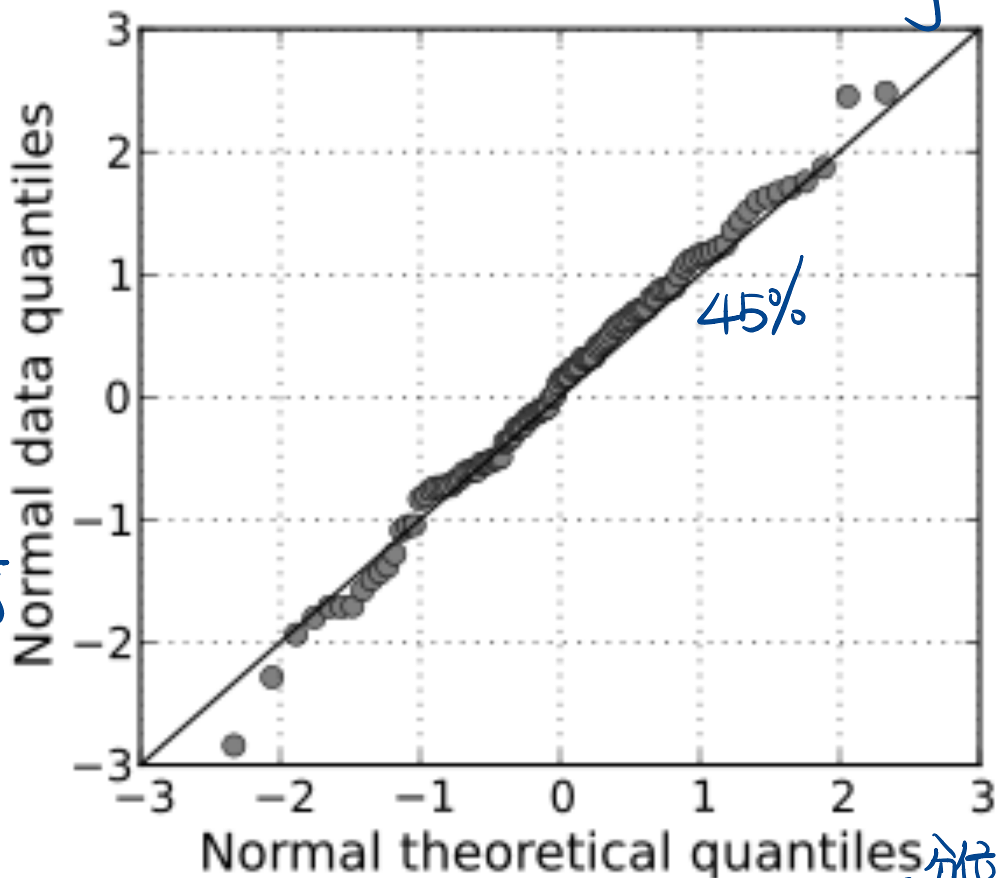
$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$$ N(0,6²)

non-bias - normally distributed

Q-Q plot

如果你的 Q-Q plot是 45°, 那么这个 model 就是 normal distributed quantile

empirical



分位数

linear regression

(1) what is R^2 (r-square) of this model

(2) If you get R^2, is that statistically significant

$R^2 \in [0,1]$: how much variation of y can be explained by x

① how do you compile R-square

$var(fit) = \frac{SS(fit)}{n}$ sum of squares $R^2 = \frac{var(mean) - var(fit)}{var(mean)}$

(2) Answer : hypothesis testing , we use F test

$F = \frac{[SS(mean) - SS(fit)]/(\#param - 1)}{SS(fit)/(n - \#param)} \rightarrow$ large $\Leftrightarrow R^2$ is more significant

find P-value, if P-value is small, reject null hypothesis

H_0: R^2 not significant

H_A: R^2 is significant

Probabilistic Interpretation

- Deriving the log-likilihood function

$$\epsilon^{(i)} = y^{(i)} - \theta^T x^{(i)} \sim \mathcal{N}\left(0, \sigma^2\right)$$

normal density

$$p\left(\epsilon^{(i)}\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(\epsilon^{(i)}\right)^2}{2\sigma^2}\right)$$

likelihood

$$\Downarrow$$

$$p\left(y^{(i)} | x^{(i)}; \theta\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right)$$

Probabilistic Interpretation

- Deriving the log-likilihood function

$$L(\theta) = L(\theta; X, \vec{y}) = p(\vec{y}|X; \theta)$$

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{m} p\left(y^{(i)}|x^{(i)}; \theta\right) \\ &= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right) \end{aligned}$$

Probabilistic Interpretation

- Maximum Likelihood Estimator (MLE)

$$
\begin{aligned}
\ell(\theta) &= \log L(\theta) \\
&= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right) \\
&= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right) \\
&= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^{m} \left(y^{(i)} - \theta^T x^{(i)}\right)^2
\end{aligned}
$$

Python Time!

- from sklearn import linear_model
- from statsmodels.api import OLS