Data Analysis - Lecture 6

Time series analysis

Dr. Jérémie Sublime

LISITE Laboratory - DaSSIP Team - ISEP
LIPN - CNRS UMR 7030

jeremie.sublime@isep.fr

Plan

1. Fundamentals of time series analysis
2. The ARIMA model
3. Introduction to Hidden Markov Models
4. Conclusion

Outline

1. Fundamentals of time series analysis

2. The ARIMA model

3. Introduction to Hidden Markov Models

4. Conclusion

Time data & Time series

Time data contain one or several attributes that describe when the observations took place: year, month, day, hour, elapsed time since the beginning of an experiment, timestamp, etc.

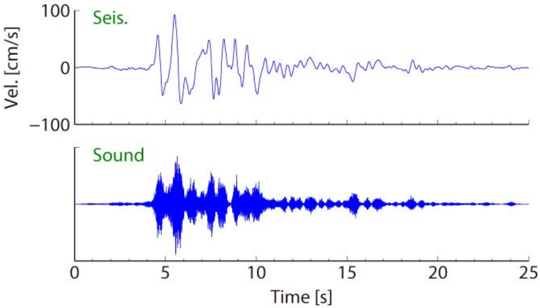
Times series: Definition

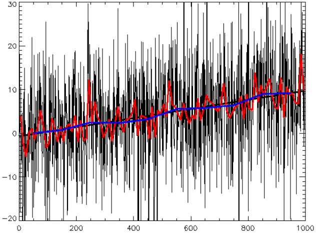
A time series is a series of data points indexed in time order.

- Most commonly, a time series is a sequence taken at successive equally spaced points in time.
- A time series usually describes the same observation evolving through time.

Time data & Time series

Time series can be used for a large number of applications: statistics, signal processing, weather forecast, earthquake prediction, finance analysis, budget predictions, tidal predictions, astrophysics, astronomy, tidal analysis, electroencephalography, control engineering, or any domain in science that involves time measurements.





Preliminary analysis of Time data

Processing time data first require to answer the following questions:

- Does my time series deals with one or several objects in time ?
- Are my observations equally spaced in time ? If not, can I fill in the gaps ?
  - If the observations are not equally spaced in time, most analysis are impossible to do.
- What am I looking for ?
  - Global trends ? Cyclic events ? Abnormalities ? Specific events ?
  - What is the time scale of interest for these data ?
- Should I differentiate some of the variables ?

Differentiating the variables ?

- When time is a parameter, the other variables can be differentiated with respect to time.

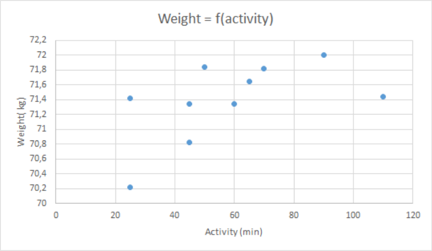
$$\Delta Y = Y_{t+1} - Y_t$$

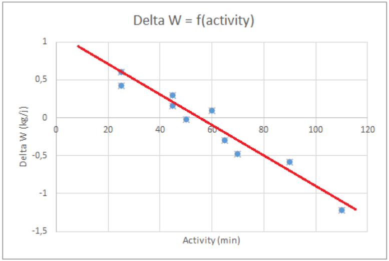
Why differentiating ?

- Sometimes the new variables can contain information that are more useful than the old ones.
- One must consider which is more interesting between the raw value of a variable at a point in time and its evolution between two points in time.

Differentiating the variables ?

Day	Sport (min)	Weight (kg)	Δ Weight
14/05	90	72.0	-0.58
15/05	25	71.42	0.42
16/05	50	71.84	-0.02
17/05	70	71.82	-0.48
18/05	45	71.34	0.3
19/05	65	71.64	-0.3
20/05	60	71.34	0.1
21/05	110	71.44	-1.22
22/05	25	70.22	0.6
23/05	45	70.82	0.16





Times series analysis

While creating new attributes using the time variable may lead to finding new correlations and could help analyzing the data, it is not time series analysis.

Times series analysis: Definition

Times series analysis is the study of how one or several variables, or even objects behave with respect to time. Time series analysis can have several goals.

- Finding and describing trends in a time series.
- Building predictive models from a time series.

Time series vs regular data

Variables indépendantes et identiquement distribuées

Cross-Sectional Data

$X_1, X_2, \cdots, X_N \quad \sim iid \quad \Rightarrow \quad LLN: \quad \frac{1}{N}\sum_i X_i = \mathbb{E}[X]$

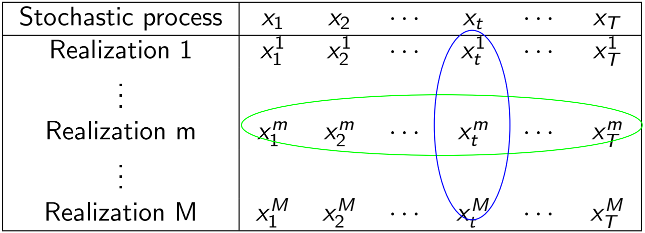
- Draws from a fixed distribution
- No ordering

Time series Data

$x_1, x_2, \cdots, x_t, \cdots, x_T$ with t the time index, and T the size of the sample

- each realization x_t is a draw from a random variable X_t
- Natural ordering $\Rightarrow$ Conditional models: $x_t | x_{t-1}, x_{t-2}$

Why do we need stationarity ?



Stochastic process	x_1	x_2	$\cdots$	x_t	$\cdots$	x_T
Realization 1	x_1^1	x_2^1	$\cdots$	x_t^1	$\cdots$	x_T^1
$\vdots$						
Realization m	x_1^m	x_2^m	$\cdots$	x_t^m	$\cdots$	x_T^m
$\vdots$						
Realization M	x_1^M	x_2^M	$\cdots$	x_t^M	$\cdots$	x_T^M

Without stationarity, these two measures are different:

- Ensemble Mean: $\widehat{\mathbb{E}(x_t)} = \frac{1}{M}\sum_{m=1}^{M} x_t^m$
- Time average of a realized sample path: $\bar{x_T} = \frac{1}{T}\sum_{i=1}^{T} x_i$

Stationarity: Definitions

Strict Stationarity

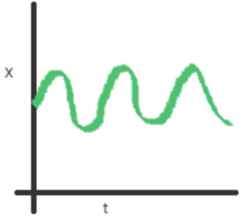
A time series is said to be strictly stationary if all its observations are drawn from the same distribution: the join probability does not change in time.

Weak Stationarity

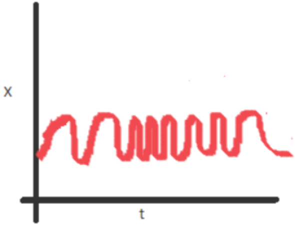
We do not require that each draw comes from the exact same distribution, only that the distributions have the same mean and variance (all of them not a function of time).

- Constant mean: $\mathbb{E}(x_t) = \mu$
- Constant variance: $Var(x_t) = \gamma_0$
- Constant co-variance: $Cov(x_t, x_{t-h}) = \gamma_h \quad \forall h \in [1..T]$

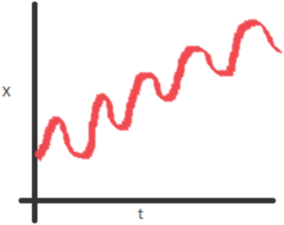
Stationarity: Examples



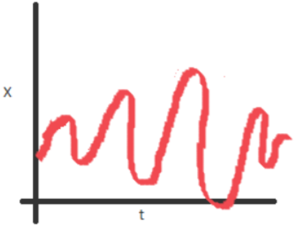
Stationary series



Non-Stationary series



Non-Stationary series



Non-Stationary series

Stationarity and weak dependence

Weak dependence hypothesis

x_t and x_{t+h} are approximately independent from each other when $h \rightarrow \infty$.

- Each observation contain new information about the distribution.

Under the weak dependence hypothesis and the weak stationarity hypothesis, we have: $\bar{x_T} \rightarrow \mathbb{E}(x_t)$
And with this, we can get information from the sample about the underlying distribution.

Remark

The assumptions of weak stationarity and weak dependence replaces the i.i.d hypothesis from cross-sectional data.

- Weak dependence $\Rightarrow$ independence
- Weak stationarity $\Rightarrow$ identically distributed

Outline

1 Fundamentals of time series analysis

2 The ARIMA model

3 Introduction to Hidden Markov Models

4 Conclusion

From regression to auto-regressive processes

While regression analysis can be used for time series analysis, it can only catch global trends and has the following weaknesses:

- It cannot be applied to several variables.
- It cannot detect cyclic or seasonal events.
- It cannot be used to predict "aftershock effects" after random shocks in a time line.
- It is ill adapted to segment a time series into several events.

From regression to auto-regressive processes

- In linear regressions, we try to find regressions coefficients to solve an equation of form $y = a \cdot x + b + \epsilon$.
- For time series predictions, the model is more like $x_t = \rho x_{t-1} + \epsilon_t$
  - ρ is a coefficient to be found
  - ϵ_t is an error term between the prediction and the real value.

This is what we call an **auto-regressive process** of order one: **AR(1)**

Auto-regressive processes

Auto-regressive process of order p

An auto-regressive process of order p is defined as follows:

$$AR(p): \qquad X_t = \sum_{i=1}^{p} \rho_i X_{t-i} + \epsilon_t$$

- the ρ_i are parameters
- ϵ_t is a white noise term (random noise i.i.d$(0, \sigma^2)$)

Auto-regressive process of order 1

$$AR(1): \qquad X_t = c + \rho X_{t-1} + \epsilon_t$$

Auto-regressive processes of order 1: Example (1/2)

Let us consider an oil price problem. The oil price change at a moment t ΔO_t is linked to the oil price change at the instant $t-1$:

$$\Delta O_t = 0.7 \times \Delta O_{t-1} + \epsilon_t$$

Auto-regressive processes of order 1: Example (1/2)

Let us consider an oil price problem. The oil price change at a moment t ΔO_t is linked to the oil price change at the instant $t-1$:

$$\Delta O_t = 0.7 \times \Delta O_{t-1} + \epsilon_t$$

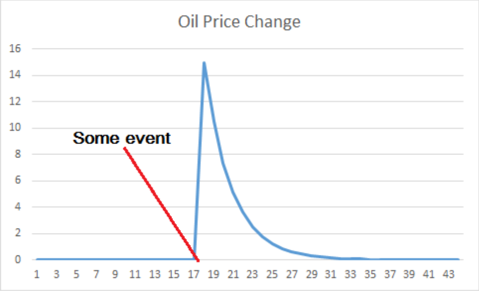
- If some event occurs, the oil price will suddenly change, resulting in a spike in the oil price change (ϵ spikes).

Auto-regressive processes of order 1: Example (1/2)

Let us consider an oil price problem. The oil price change at a moment t ΔO_t is linked to the oil price change at the instant $t-1$:

$$\Delta O_t = 0.7 \times \Delta O_{t-1} + \epsilon_t$$

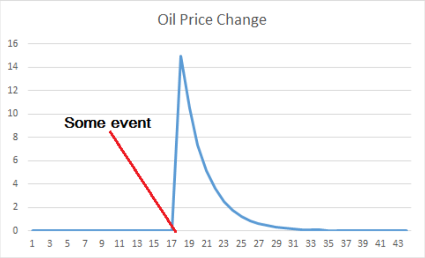
- If some event occurs, the oil price will suddenly change, resulting in a spike in the oil price change (ϵ spikes).
- Then following this model, the event has some persistent effect until the market stabilizes and the price stops changing.

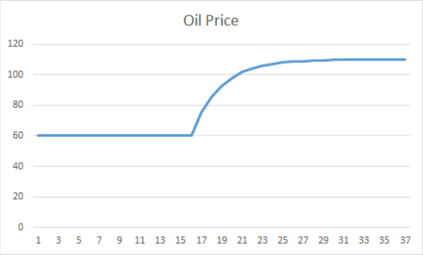


Auto-regressive processes of order 1: Example (2/2)

The oil price change at a moment t ΔO_t is linked to the oil price change at the instant $t-1$:

$$\Delta O_t = 0.7 \times \Delta O_{t-1} + \epsilon_t$$





Remark

This is an AR(1) model for the Oil Price Change, but since we use the first derivation of the Oil Price, we will see later that this model is an ARIMA(1,**1**,0) model of the Oil Price.

Stationarity conditions of an AR(1) Process 1/2

$X_t = \rho X_{t-1} + \epsilon_t$ with $\epsilon_t \sim iid(0, \sigma^2)$

Stationarity conditions of an AR(1) Process 1/2

$X_t = \rho X_{t-1} + \epsilon_t$ with $\epsilon_t \sim iid(0, \sigma^2)$
$X_t = \rho\left(\rho X_{t-2} + \epsilon_{t-1}\right) + \epsilon_t$

Stationarity conditions of an AR(1) Process 1/2

$X_t = \rho X_{t-1} + \epsilon_t$ with $\epsilon_t \sim iid(0, \sigma^2)$
$X_t = \rho \left(\rho X_{t-2} + \epsilon_{t-1} \right) + \epsilon_t$
$X_t = \rho^2 X_{t-2} + \rho \epsilon_{t-1} + \epsilon_t$

Stationarity conditions of an AR(1) Process 1/2

$X_t = \rho X_{t-1} + \epsilon_t$ with $\epsilon_t \sim iid(0, \sigma^2)$
$X_t = \rho\left(\rho X_{t-2} + \epsilon_{t-1}\right) + \epsilon_t$
$X_t = \rho^2 X_{t-2} + \rho\epsilon_{t-1} + \epsilon_t$

$$\Rightarrow X_t = \rho^t X_0 + \sum_{i=0}^{t-1} \rho^i \epsilon_{t-i}$$

Stationarity conditions of an AR(1) Process 1/2

$X_t = \rho X_{t-1} + \epsilon_t$ with $\epsilon_t \sim iid(0, \sigma^2)$
$X_t = \rho\left(\rho X_{t-2} + \epsilon_{t-1}\right) + \epsilon_t$
$X_t = \rho^2 X_{t-2} + \rho\epsilon_{t-1} + \epsilon_t$

$$\Rightarrow X_t = \rho^t X_0 + \sum_{i=0}^{t-1} \rho^i \epsilon_{t-i}$$

The first condition for stationarity is : $\mathbb{E}[X_t] = Cte$

Stationarity conditions of an AR(1) Process 1/2

$X_t = \rho X_{t-1} + \epsilon_t$ with $\epsilon_t \sim iid(0, \sigma^2)$
$X_t = \rho\left(\rho X_{t-2} + \epsilon_{t-1}\right) + \epsilon_t$
$X_t = \rho^2 X_{t-2} + \rho\epsilon_{t-1} + \epsilon_t$

$$\Rightarrow X_t = \rho^t X_0 + \sum_{i=0}^{t-1} \rho^i \epsilon_{t-i}$$

The first condition for stationarity is : $\mathbb{E}[X_t] = Cte$

$$\mathbb{E}[X_t] = \rho^t \mathbb{E}[X_0] + \sum_{i=0}^{t-1} \rho^i \mathbb{E}[\epsilon_{t-i}] = \rho^t \mathbb{E}[X_0]$$

Stationarity conditions of an AR(1) Process 1/2

$X_t = \rho X_{t-1} + \epsilon_t$ with $\epsilon_t \sim iid(0, \sigma^2)$
$X_t = \rho\left(\rho X_{t-2} + \epsilon_{t-1}\right) + \epsilon_t$
$X_t = \rho^2 X_{t-2} + \rho\epsilon_{t-1} + \epsilon_t$

$$\Rightarrow X_t = \rho^t X_0 + \sum_{i=0}^{t-1} \rho^i \epsilon_{t-i}$$

The first condition for stationarity is : $\mathbb{E}[X_t] = Cte$

$$\mathbb{E}[X_t] = \rho^t \mathbb{E}[X_0] + \sum_{i=0}^{t-1} \rho^i \mathbb{E}[\epsilon_{t-i}] = \rho^t \mathbb{E}[X_0]$$

Condition on the mean

For this term to be constant, we need $\mathbb{E}[X_0] = 0$ and therefore $\mathbb{E}[X_t] = 0$.

Stationarity conditions of an AR(1) Process 2/2

$X_t = \rho X_{t-1} + \epsilon_t$ with $\epsilon_t \sim iid(0, \sigma^2)$

Stationarity conditions of an AR(1) Process 2/2

$X_t = \rho X_{t-1} + \epsilon_t$ with $\epsilon_t \sim iid(0, \sigma^2)$

$Var(X_t) = \rho^2 Var(X_{t-1}) + Var(\epsilon_t)$

Stationarity conditions of an AR(1) Process 2/2

$X_t = \rho X_{t-1} + \epsilon_t$ with $\epsilon_t \sim iid(0, \sigma^2)$

$Var(X_t) = \rho^2 Var(X_{t-1}) + Var(\epsilon_t)$ NB: $var(aX) = a^2 var(X)$

Stationarity conditions of an AR(1) Process 2/2

$X_t = \rho X_{t-1} + \epsilon_t$ with $\epsilon_t \sim iid(0, \sigma^2)$

$Var(X_t) = \rho^2 Var(X_{t-1}) + Var(\epsilon_t)$ NB: $var(aX) = a^2 var(X)$

- We want the variance to be constant: $Var(X_t) = Var(X_{t-1})$

Stationarity conditions of an AR(1) Process 2/2

$X_t = \rho X_{t-1} + \epsilon_t$ with $\epsilon_t \sim iid(0, \sigma^2)$

$Var(X_t) = \rho^2 Var(X_{t-1}) + Var(\epsilon_t)$ NB: $var(aX) = a^2 var(X)$

- We want the variance to be constant: $Var(X_t) = Var(X_{t-1})$
- We substitute in the previous expression, and we get:

$Var(X_t) = \rho^2 Var(X_t) + \sigma^2$

Stationarity conditions of an AR(1) Process 2/2

$X_t = \rho X_{t-1} + \epsilon_t$ with $\epsilon_t \sim iid(0, \sigma^2)$

$Var(X_t) = \rho^2 Var(X_{t-1}) + Var(\epsilon_t)$ NB: $var(aX) = a^2 var(X)$

- We want the variance to be constant: $Var(X_t) = Var(X_{t-1})$
- We substitute in the previous expression, and we get:

$Var(X_t) = \rho^2 Var(X_t) + \sigma^2 \quad \Rightarrow \quad (1-\rho^2) Var(X_t) = \sigma^2$

Stationarity conditions of an AR(1) Process 2/2

$X_t = \rho X_{t-1} + \epsilon_t$ with $\epsilon_t \sim iid(0, \sigma^2)$

$Var(X_t) = \rho^2 Var(X_{t-1}) + Var(\epsilon_t)$ NB: $var(aX) = a^2 var(X)$

- We want the variance to be constant: $Var(X_t) = Var(X_{t-1})$
- We substitute in the previous expression, and we get:

$Var(X_t) = \rho^2 Var(X_t) + \sigma^2 \quad \Rightarrow \quad (1-\rho^2)Var(X_t) = \sigma^2$

Conditions on the variance and on ρ

$$Var(X_t) = \frac{\sigma^2}{1-\rho^2}$$

- Since we don't want a negative variance or a null denominator, we deduce that $|\rho| < 1$

Stationary covariance and weak dependence of an AR(1) process

From our previous calculi, we know that: $X_{t+h} = \rho^h X_t + \sum_{i=0}^{h-1} \rho^i \epsilon_{t+h-i}$

Stationary covariance and weak dependence of an AR(1) process

From our previous calculi, we know that: $X_{t+h} = \rho^h X_t + \sum_{i=0}^{h-1} \rho^i \epsilon_{t+h-i}$

$$Cov(X_t, X_{t+h}) = Cov(X_t, \rho^h X_t + \sum_{i=0}^{h-1} \rho^i \epsilon_{t+h-i})$$

Stationary covariance and weak dependence of an AR(1) process

From our previous calculi, we know that: $X_{t+h} = \rho^h X_t + \sum_{i=0}^{h-1} \rho^i \epsilon_{t+h-i}$

$$Cov(X_t, X_{t+h}) = Cov(X_t, \rho^h X_t + \sum_{i=0}^{h-1} \rho^i \epsilon_{t+h-i}) = Cov(X_t, \rho^h X_t)$$

Stationary covariance and weak dependence of an AR(1) process

From our previous calculi, we know that: $X_{t+h} = \rho^h X_t + \sum_{i=0}^{h-1} \rho^i \epsilon_{t+h-i}$

$$Cov(X_t, X_{t+h}) = Cov(X_t, \rho^h X_t + \sum_{i=0}^{h-1} \rho^i \epsilon_{t+h-i}) = Cov(X_t, \rho^h X_t)$$

$$= \rho^h Cov(X_t, X_t) = \rho^h Var(X_t)$$

Stationary covariance and weak dependence of an AR(1) process

From our previous calculi, we know that: $X_{t+h} = \rho^h X_t + \sum_{i=0}^{h-1} \rho^i \epsilon_{t+h-i}$

$$Cov(X_t, X_{t+h}) = Cov(X_t, \rho^h X_t + \sum_{i=0}^{h-1} \rho^i \epsilon_{t+h-i}) = Cov(X_t, \rho^h X_t)$$

$$= \rho^h Cov(X_t, X_t) = \rho^h Var(X_t) = \frac{\rho^h \sigma^2}{1-\rho^2}$$

Stationary covariance and weak dependence of an AR(1) process

From our previous calculi, we know that: $X_{t+h} = \rho^h X_t + \sum_{i=0}^{h-1} \rho^i \epsilon_{t+h-i}$

$$Cov(X_t, X_{t+h}) = Cov(X_t, \rho^h X_t + \sum_{i=0}^{h-1} \rho^i \epsilon_{t+h-i}) = Cov(X_t, \rho^h X_t)$$

$$= \rho^h Cov(X_t, X_t) = \rho^h Var(X_t) = \frac{\rho^h \sigma^2}{1-\rho^2}$$

$$Cor(X_t, X_{t+h}) = \frac{Cov(X_t, X_{t+h})}{Var(X_t)} = \rho^h$$

Stationary covariance and weak dependence of an AR(1) process

From our previous calculi, we know that: $X_{t+h} = \rho^h X_t + \sum_{i=0}^{h-1} \rho^i \epsilon_{t+h-i}$

$$Cov(X_t, X_{t+h}) = Cov(X_t, \rho^h X_t + \sum_{i=0}^{h-1} \rho^i \epsilon_{t+h-i}) = Cov(X_t, \rho^h X_t)$$

$$= \rho^h Cov(X_t, X_t) = \rho^h Var(X_t) = \frac{\rho^h \sigma^2}{1-\rho^2}$$

$$Cor(X_t, X_{t+h}) = \frac{Cov(X_t, X_{t+h})}{Var(X_t)} = \rho^h$$

Weak dependence condition

We want $lim_{h\to\infty} Cor(X_t, X_t + h) = 0$, so once we find once again that we need $|\rho| < 1$.

Stationary VS non Stationary AR(1)

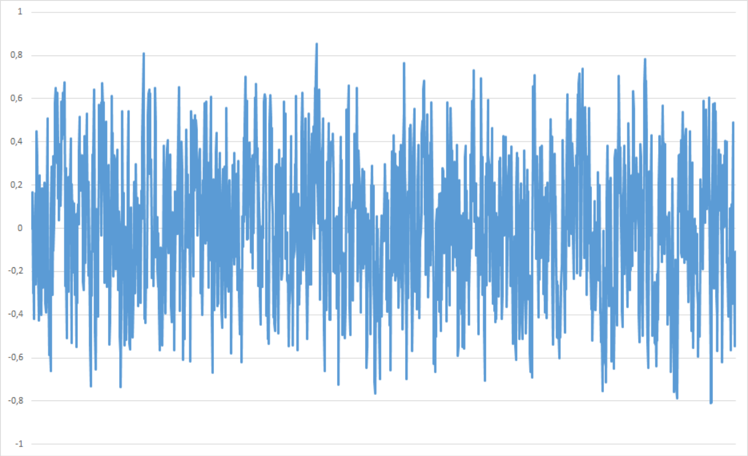


Figure: $\rho = 0.5$

Stationary VS non Stationary AR(1)

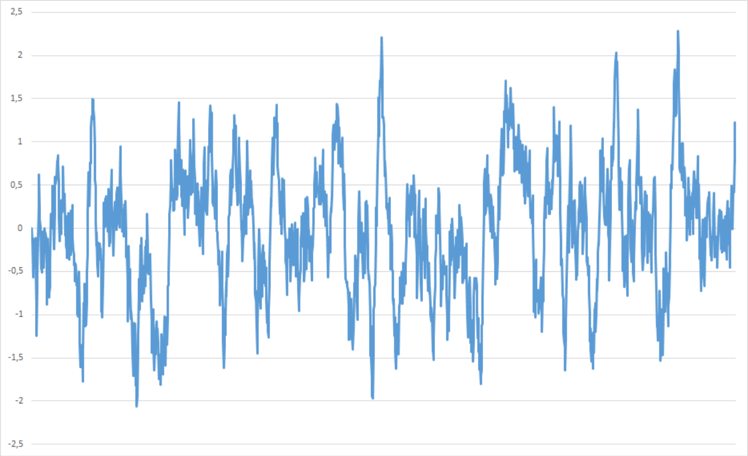


Figure: $\rho = 0.95$

Stationary VS non Stationary AR(1)

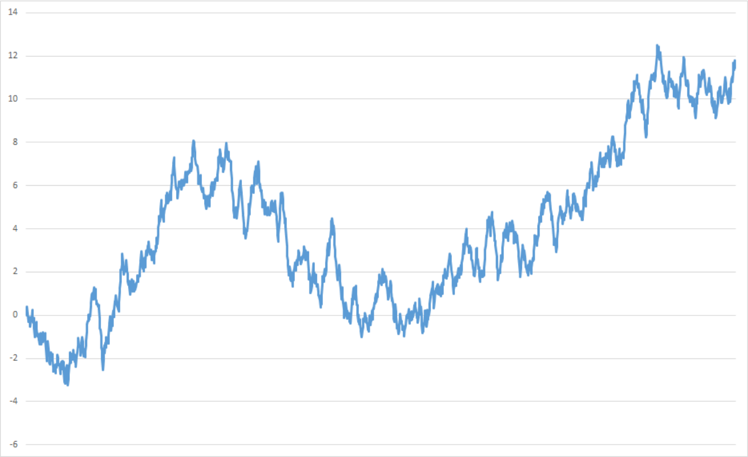


Figure: $\rho = 1$

Stationary VS non Stationary AR(1)

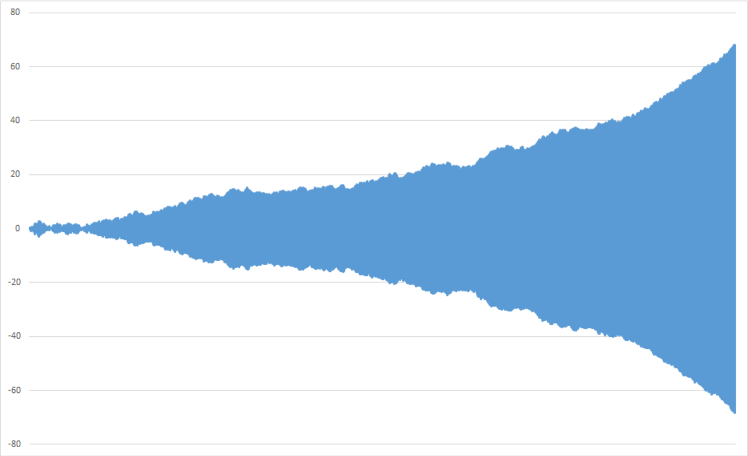


Figure: $\rho = -1.002$

Moving average processes

A moving average mode is a common approach for modeling univariate time series. In this model, the output variable depends linearly on the current and various past values of a stochastic (imperfectly predictable term).

Moving average model of order q: MA(q)

$$MA(q): \qquad X_t = \mu + \epsilon_t + \sum_{i=1}^{q} \theta_i \cdot \epsilon_{t-i}$$

- μ is the mean of the series (often assumed to be 0)
- The θ_i are the parameters
- $\epsilon_t, \cdots, \epsilon_{t-q}$ are white noise error terms.

Moving average model of order 1: MA(1)

$$MA(1): \qquad X_t = \mu + \epsilon_t + \theta \cdot \epsilon_{t-1}$$

Example of MA processes 1/2

Let us consider an example in which we want a model for the daily demand for soda bottles in a grocery store:

$$Demand = 25 + \epsilon_t - 0.5 \cdot \epsilon_{t-1}$$

- 25 is the average number of bottles usually sold in a day.
- In this example, ϵ_t could be the change in temperature: $\epsilon_t = \Delta_{temp}(t)$

Example of MA processes 1/2

Let us consider an example in which we want a model for the daily demand for soda bottles in a grocery store:

$$Demand = 25 + \epsilon_t - 0.5 \cdot \epsilon_{t-1}$$

- 25 is the average number of bottles usually sold in a day.
- In this example, ϵ_t could be the change in temperature: $\epsilon_t = \Delta_{temp}(t)$
- The explanation for the "$-0.5 \cdot \epsilon_{t-1}$" term is that, if the temperature already increased yesterday, the customers already bought soda and don't need to buy more today.

Example of MA processes 1/2

Let us consider an example in which we want a model for the daily demand for soda bottles in a grocery store:

$$Demand = 25 + \epsilon_t - 0.5 \cdot \epsilon_{t-1}$$

- 25 is the average number of bottles usually sold in a day.
- In this example, ϵ_t could be the change in temperature: $\epsilon_t = \Delta_{temp}(t)$
- The explanation for the "$-0.5 \cdot \epsilon_{t-1}$" term is that, if the temperature already increased yesterday, the customers already bought soda and don't need to buy more today.
- Further terms could be added:
  $Demand = 25 + \epsilon_t - 0.5 \cdot \epsilon_{t-1} - 0.25\epsilon_{t-2}$

Example of MA processes 2/2

We go back to an oil price example:

$$OilPrice = 45 + \epsilon_t + 0.5\epsilon_{t-1}$$

- 45$ is the usual average price for a barrel
- ϵ_t here could be used for modeling issues in oil delivery (hurricane at sea, blockade, strikes, armed conflicts, etc.)

Example of MA processes 2/2

We go back to an oil price example:

$$OilPrice = 45 + \epsilon_t + 0.5\epsilon_{t-1}$$

- 45$ is the usual average price for a barrel
- ϵ_t here could be used for modeling issues in oil delivery (hurricane at sea, blockade, strikes, armed conflicts, etc.)
- If there is a hurricane at sea at time t, the price will increase.
- If there was an issue at time $t - 1$, the supply may still be recovering so the prices are still higher than usual, hence the "$+0.5\epsilon_{t-1}$" term.

Example of MA processes 2/2

We go back to an oil price example:

$$OilPrice = 45 + \epsilon_t + 0.5\epsilon_{t-1}$$

- 45$ is the usual average price for a barrel
- ϵ_t here could be used for modeling issues in oil delivery (hurricane at sea, blockade, strikes, armed conflicts, etc.)
- If there is a hurricane at sea at time t, the price will increase.
- If there was an issue at time $t-1$, the supply may still be recovering so the prices are still higher than usual, hence the "$+0.5\epsilon_{t-1}$" term.

Remark

Unlike in the previous oil price example, we study the oil price, and not the oil price change ! It is not the same model.

MA(1) process: stationarity and weak dependency 1/2

$X_t = \mu + \epsilon_t + \theta \cdot \epsilon_{t-1}$ with $\epsilon_t \sim iid(0, \sigma^2)$

MA(1) process: stationarity and weak dependency 1/2

$X_t = \mu + \epsilon_t + \theta \cdot \epsilon_{t-1}$ with $\epsilon_t \sim iid(0, \sigma^2)$

$$\mathbb{E}[X_t] = \mathbb{E}[\mu + \epsilon_t + \theta \cdot \epsilon_{t-1}] = \mu + \mathbb{E}[\epsilon_t] + \theta \mathbb{E}[\epsilon_{t-1}] = \mu$$

MA(1) process: stationarity and weak dependency 1/2

$X_t = \mu + \epsilon_t + \theta \cdot \epsilon_{t-1}$ with $\epsilon_t \sim iid(0, \sigma^2)$

$$\mathbb{E}[X_t] = \mathbb{E}[\mu + \epsilon_t + \theta \cdot \epsilon_{t-1}] = \mu + \mathbb{E}[\epsilon_t] + \theta\mathbb{E}[\epsilon_{t-1}] = \mu$$

$$Var(X_t) = Var(\mu + \epsilon_t + \theta \cdot \epsilon_{t-1}) = Var(\epsilon_t) + \theta^2 Var(\epsilon_{t-1}) = \sigma^2(1 + \theta^2)$$

MA(1) process: stationarity and weak dependency 1/2

$X_t = \mu + \epsilon_t + \theta \cdot \epsilon_{t-1}$ with $\epsilon_t \sim iid(0, \sigma^2)$

$$\mathbb{E}[X_t] = \mathbb{E}[\mu + \epsilon_t + \theta \cdot \epsilon_{t-1}] = \mu + \mathbb{E}[\epsilon_t] + \theta\mathbb{E}[\epsilon_{t-1}] = \mu$$

$$Var(X_t) = Var(\mu + \epsilon_t + \theta \cdot \epsilon_{t-1}) = Var(\epsilon_t) + \theta^2 Var(\epsilon_{t-1}) = \sigma^2(1 + \theta^2)$$

Stationarity: conditions on the mean and variance

The condition for a weak stationarity are always respected since we have both the mean and the variance that are constant.

MA(1) process: stationarity and weak dependency 2/2

For the third stationarity hypothesis, we want $Cov(X_t, X_{t+h})$ to be a function of h independent from t.

MA(1) process: stationarity and weak dependency 2/2

For the third stationarity hypothesis, we want $Cov(X_t, X_{t+h})$ to be a function of h independent from t.

$$\begin{aligned} Cov(X_t, X_{t-1}) &= Cov(\epsilon_t + \theta \cdot \epsilon_{t-1}, \epsilon_{t-1} + \theta \cdot \epsilon_{t-2}) \\ &= \theta Cov(\epsilon_{t-1}, \epsilon_{t-1}) = \theta \sigma^2 \qquad \text{because the } \epsilon_t \text{ are iid} \end{aligned}$$

MA(1) process: stationarity and weak dependency 2/2

For the third stationarity hypothesis, we want $Cov(X_t, X_{t+h})$ to be a function of h independent from t.

$$\begin{aligned} Cov(X_t, X_{t-1}) &= Cov(\epsilon_t + \theta \cdot \epsilon_{t-1}, \epsilon_{t-1} + \theta \cdot \epsilon_{t-2}) \\ &= \theta Cov(\epsilon_{t-1}, \epsilon_{t-1}) = \theta\sigma^2 \qquad \text{because the } \epsilon_t \text{ are iid} \end{aligned}$$

From there we can infer that:

$$\forall \tau > 1 \quad Cov(X_t, X_{t-\tau}) = Cov(\epsilon_t + \theta \cdot \epsilon_{t-1}, \epsilon_{t-\tau} + \theta \cdot \epsilon_{t-1-\tau}) = 0$$

MA(1) process: stationarity and weak dependency 2/2

For the third stationarity hypothesis, we want $Cov(X_t, X_{t+h})$ to be a function of h independent from t.

$$\begin{aligned} Cov(X_t, X_{t-1}) &= Cov(\epsilon_t + \theta \cdot \epsilon_{t-1}, \epsilon_{t-1} + \theta \cdot \epsilon_{t-2}) \\ &= \theta Cov(\epsilon_{t-1}, \epsilon_{t-1}) = \theta\sigma^2 \qquad \text{because the } \epsilon_t \text{ are iid} \end{aligned}$$

From there we can infer that:

$$\forall \tau > 1 \quad Cov(X_t, X_{t-\tau}) = Cov(\epsilon_t + \theta \cdot \epsilon_{t-1}, \epsilon_{t-\tau} + \theta \cdot \epsilon_{t-1-\tau}) = 0$$

Stationarity and dependency

With the 3 conditions being respected, we can conclude that an MA(1) process is always stationary. Furthermore, with the third condition, the weak dependence hypothesis is easily verified.

MA(1) process: Example

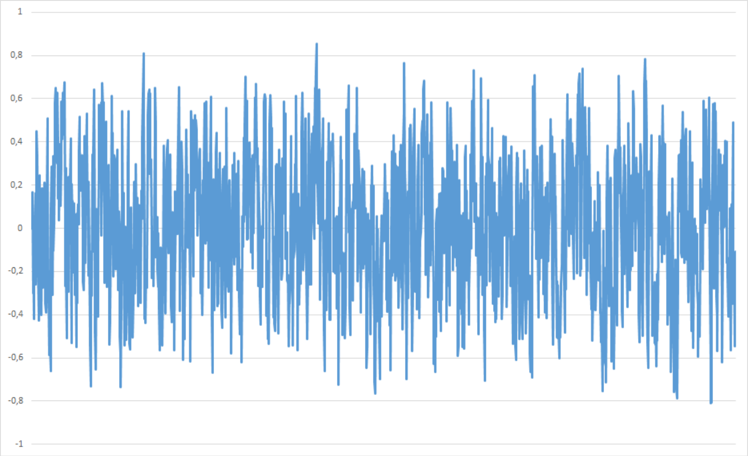


Figure: Example of a MA(1) process ($\theta = -0.3$)

Telling MA(1) from AR(1)

- Sometimes, just looking at the time series curb is not enough to tell whether it follows an AR(1) model, a MA(1) model, or something else.
- The best solution to guess MA(1) from AR(1) is to look at the correlation $Corr(X_t, X_{t+h})$

$$MA(1): \qquad Corr(X_t, X_{t+h}) = \begin{cases} \frac{\theta}{1-\theta^2} \text{ if } h = 1 \\ 0 \text{ if } h > 1 \end{cases}$$

$$AR(1): \qquad Corr(X_t, X_{t+h}) = \rho^h$$

ARMA model

ARMA(p,q)

An ARMA(p,q) process, is a model that contain an AR(p) process and and MA(q) process:

$$X_t = \mu + \epsilon_t + \sum_{i=1}^{p} \rho_i X_{t-i} + \sum_{i=1}^{q} \theta_i \cdot \epsilon_{t-i}$$

- The constant μ is the expectation of X_t (often assumed to be 0 and not taken into consideration in the regressive term).
- The ρ_i and θ_i are the parameters from the auto-regressive and moving average process respectively.
- The ϵ_i are white error terms.

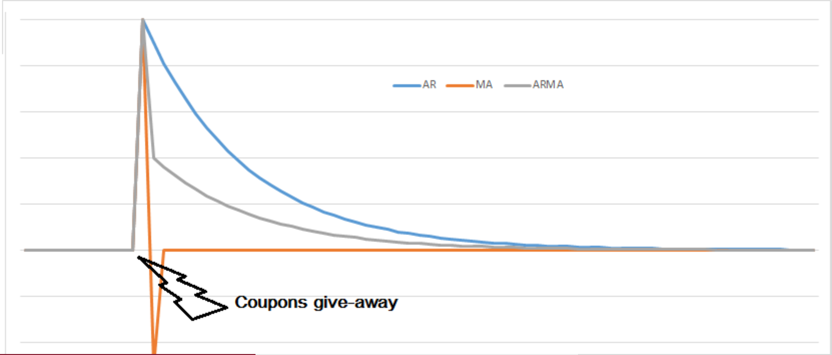
ARMA(1,1)

$$ARMA(1,1): \qquad X_t = \mu + \epsilon_t + \rho X_{t-1} + \theta \cdot \epsilon_{t-1}$$

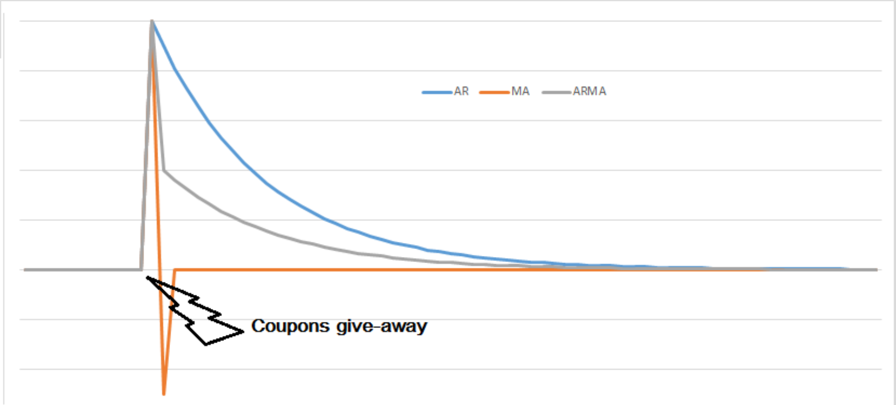
ARMA(1,1), Example 1

Let us consider a good sales problem:
$Sales_t = \epsilon_t + \rho Sales_{t-1} + \theta \cdot \epsilon_{t-1}$

- ρ is a loyalty effect from the customer to the good. Let's take $\rho = 0.9$
- ϵ_t represents free coupons given at a time t.
- θ is a negative effect due to the fact that people already bought the product at time $t-1$. Let's take $\theta = -0.3$



ARMA(1,1), Example 1

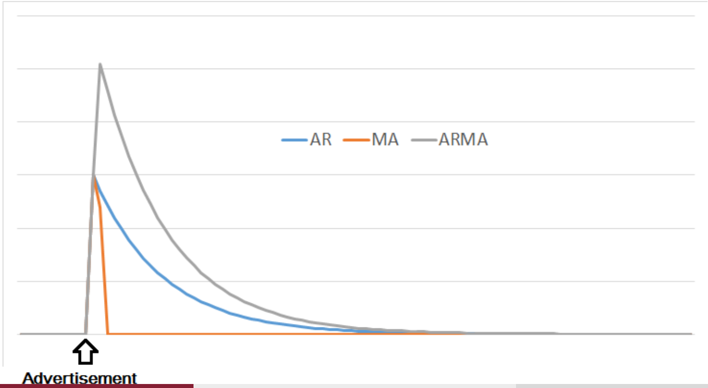


- We see that at the beginning the MA process causes a rapide decline in sales.
- Then, once the MA effect is over, the sales decrease following an AR process.

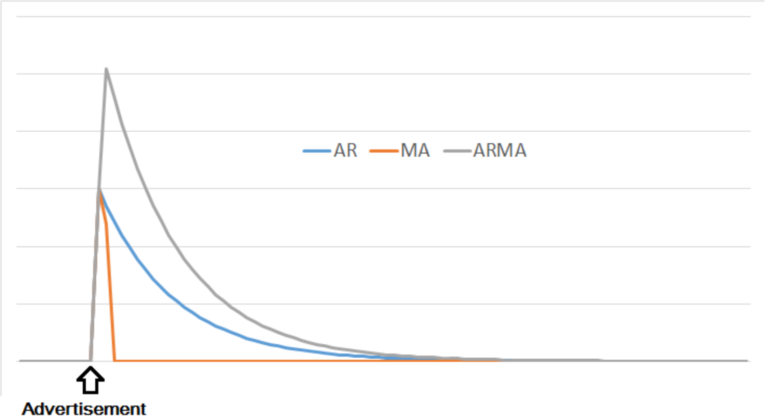
ARMA(1,1), Example 2

Let us consider the same good sales problem:
$Sales_t = \epsilon_t + \rho Sales_{t-1} + \theta \cdot \epsilon_{t-1}$

- ρ is a loyalty effect from the customer to the good. Let's take $\rho = 0.9$
- ϵ_t is this time advertisement for the product at a time t.
- θ becomes therefore a positive effect. Let's take $\theta = 0.8$.



ARMA(1,1), Example 1



- We see that with $\theta > 0$, there is a huge initial boost on the sale.
- It is again followed by an AR type decay process which takes more time due to the initial higher boost effect.

ARMA(1,1) behavior examples

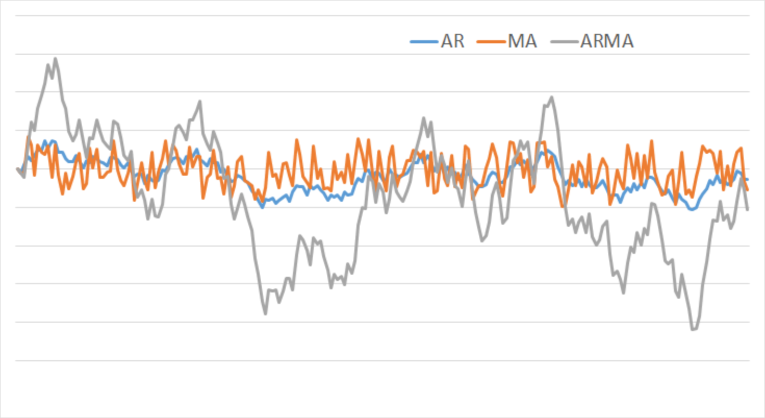


Figure: $\rho = 0.95$ and $\theta = 3$

ARMA(1,1) behavior examples

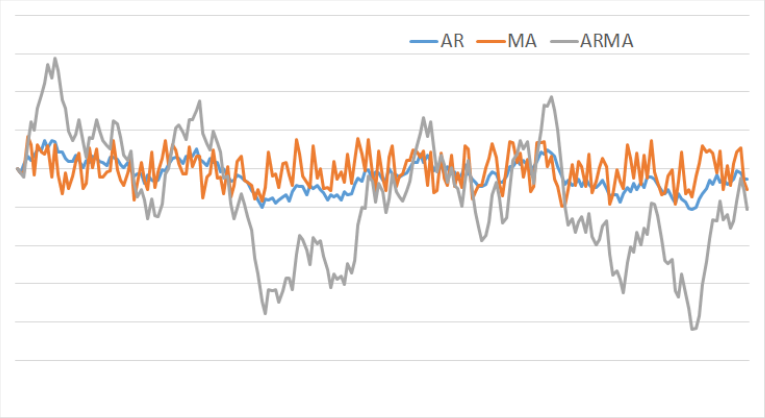


Figure: $\rho = 0.95$ and $\theta = 3$

ARMA(1,1) behavior examples

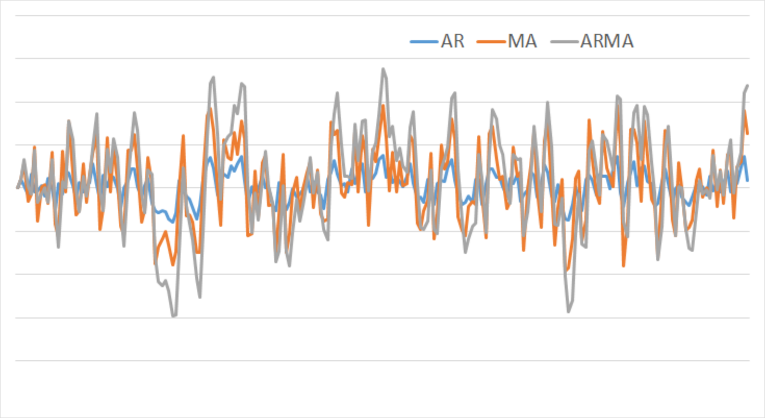


Figure: $\rho = 0.5$ and $\theta = 3$

ARMA(1,1) behavior examples

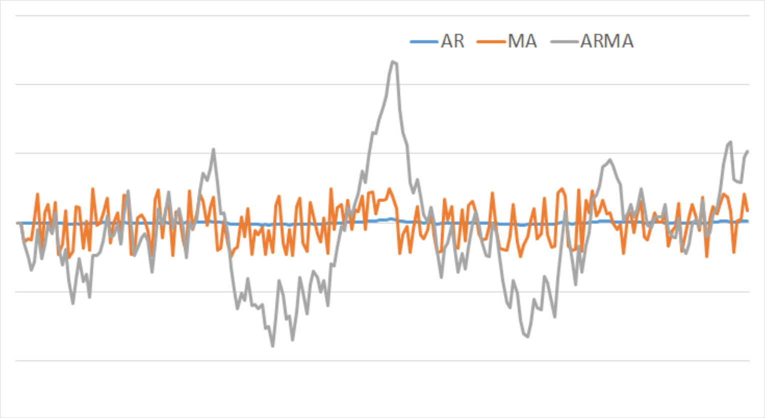


Figure: $\rho = 0.90$ and $\theta = 50$

ARMA(1,1) properties

ARMA processes have the same properties than AR processes when it comes to stationarity and the weak dependence hypothesis:

- $|\rho| < 1$ is required
- The mean value of the series should be 0.

Guessing the parameters of any ARMA(p,q) model (1/4)

The values p and q of any ARMA model can be guessed using two tools:

- The autocorrelogram for the q of the MA model.
- The partial autocorrelogram for p of the AR model.

Partial autocorrelation

In time series analysis, the **partial autocorrelation function (PACF)** gives the partial correlation of a time series with its own lagged values, controlling for the values of the time series at all shorter lags. It contrasts with the **autocorrelation function**, which does not control for other lags.

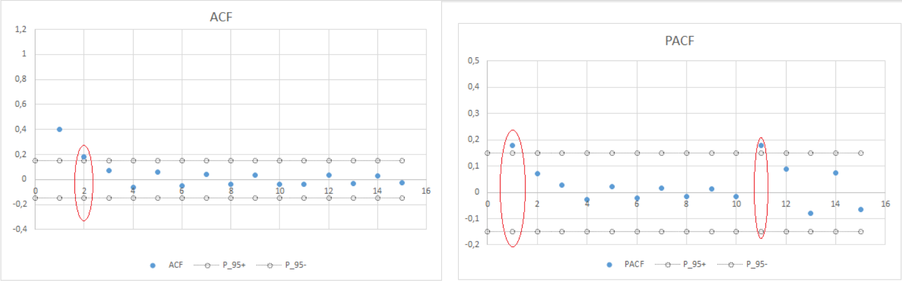
$$\alpha(h) = Cor(x_{t+h} - P_{t,h}(x_{t+h}), x_t - P_{t,h}(x_t))$$

with $P_{t,h}(x)$ the projection of x onto the space spanned by $x_{t+1}, \cdots, x_{t+h-1}$.

Guessing the parameters of any ARMA(p,q) model (2/4)

Autocorellation and partial autocorrelation confidence intervals.

Under the null hypothesis that we expect 0 autocorrelation, the confidence interval for 95% is: $P_{0.95} = 0 \pm \frac{2}{\sqrt{T}}$
Anything outside of these intervals denotes an autocorrelation and can be used to identify the type of ARMA(p,q) model.



(a) ACF showing MA(2)

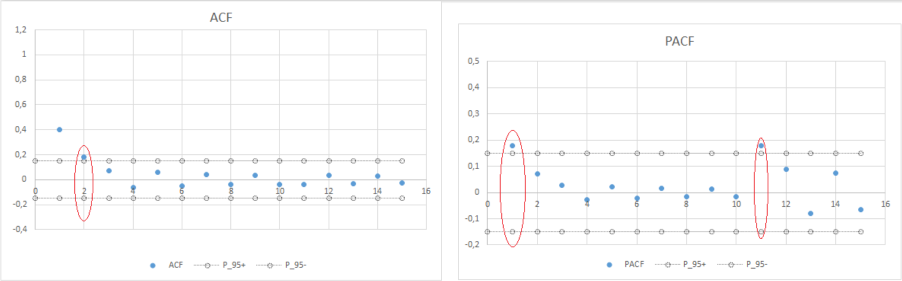
(b) PACF showing AR(1) or AR(11)

Figure: A possible ARMA(1,2) or ARMA(11,2)

Guessing the parameters of any ARMA(p,q) model (3/4)

ACF and PACF: how to read ?

- The ACF of a stationary AR(p) model goes to zero at an exponential rate, while the PACF becomes zero after lag p.
- For an MA(q) model, it is the opposite: the ACF cuts off brutally after lag q and the PACF goes to zero relatively quickly.



(a) ACF showing MA(2)

(b) PACF showing AR(1) or AR(11)

Figure: A possible ARMA(1,2) or ARMA(11,2)

Guessing the parameters of any ARMA(p,q) model (4/4)

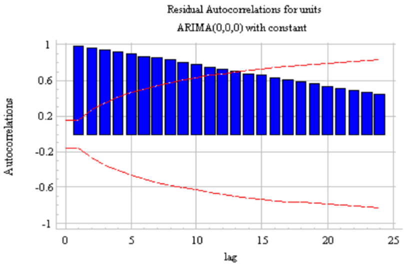
Tips to interpret PACF and ACF

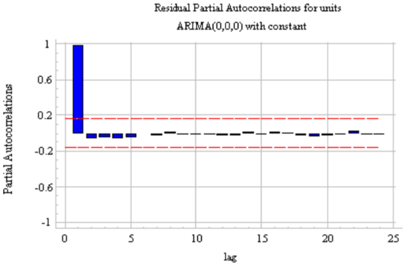
- High values for p and q are highly suspicious. Check if your result makes sense depending on the application field.
- AR model may have *echo phenomenons*: An AR(2) models may show echoes on lags 4 and 6. Once again, high values are suspicious.
- An isolated spike on a high value lag out of the confidence interval is probably just noise.

Important remarks

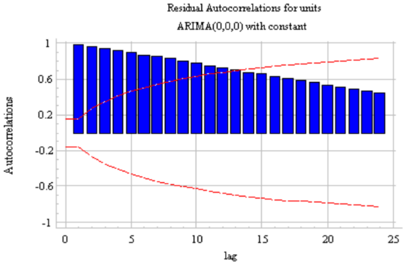
- Depending on your software, your ACF and PACF graphs may look very different and it can be confusing :
  - Lag 0 should have an auto-correlation of 1.
  - In R, the PACF starts at lag = the number of time the series was differentiated.
- ACF and PACF readings are not enough, and several models still need to be tested based on their likelihood and performances.

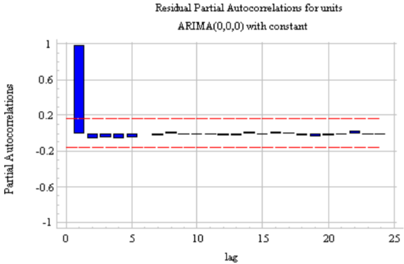
Guessing the parameters of any ARMA(p,q) model: Example





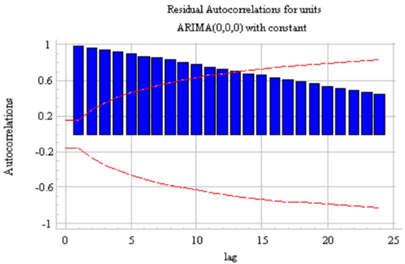
Guessing the parameters of any ARMA(p,q) model: Example

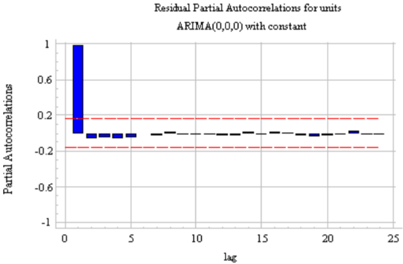




- The autocorrelations are significant for a large number of lags. But perhaps the autocorrelations beyond lag 1 are merely due to the propagation of autocorrelation at lag 1.
- This is confirmed by the PACF: This could be an ARMA(1,0)

Guessing the parameters of any ARMA(p,q) model: Example





- The autocorrelations are significant for a large number of lags. But perhaps the autocorrelations beyond lag 1 are merely due to the propagation of autocorrelation at lag 1.
- This is confirmed by the PACF: This could be an ARMA(1,0)
- Remark: The ACF does not go down exponentially which means that the series may need to be differenciated (see ARIMA)

From ARMA to ARIMA

We have seen that with time series we are not always working directly with the original series, but quite often with its differentiated version.

From ARMA to ARIMA

We have seen that with time series we are not always working directly with the original series, but quite often with its differentiated version.

ARIMA : definiton

The ARIMA(p,d,q) model is an ARMA(p,q) model applied to the d-th derivation of a time series.

From ARMA to ARIMA

We have seen that with time series we are not always working directly with the original series, but quite often with its differentiated version.

ARIMA : definiton

The ARIMA(p,d,q) model is an ARMA(p,q) model applied to the d-th derivation of a time series.

- The parameter d is usually found by trying several values incrementally until fitting model is found.
- Parameters p and q are found using the same methods (acf and pacf) than for classical ARMA models on a given d-th derivation.

From ARMA to ARIMA

We have seen that with time series we are not always working directly with the original series, but quite often with its differentiated version.

ARIMA : definiton

The ARIMA(p,d,q) model is an ARMA(p,q) model applied to the d-th derivation of a time series.

- The parameter d is usually found by trying several values incrementally until fitting model is found.
- Parameters p and q are found using the same methods (acf and pacf) than for classical ARMA models on a given d-th derivation.

Remark: Unless you are dealing with a very complicated phenomenon, d rarely goes beyond 2.

Rules to Guess the number *d* parameter in ARIMA (1/2)

- Rule 1 : If a series has a positive autocorrelation out to a high number of lags, then it probably needs a higher order of differencing.
- Rule 2 : If the lag-1 autocorrelation is 0 or negative, or the autocorrelations are all small and patternless, then the series does not need a higher order of differencing. If the lag-1 autocorrelation is -0.5 or more negative, the series may be overdifferenced. **BEWARE OF OVERDIFFERENCING !**
- Rule 3: The optimal order of differencing is often the order of differencing at which the standard deviation is lowest.

Rules to Guess the number *d* parameter in ARIMA (2/2)

- Rule 4:
  - A model without differencing assumes that the original series is stationary (mean-reverting).
  - A model with one order of differencing assumes that the original series has a constant average trend (e.g. a random walk or SES-type model, with or without growth).
  - A model with two orders of total differencing assumes that the original series has a time-varying trend (e.g. a random trend).
- Rule 5:
  - A model without differencing normally includes a constant term (which allows for a non-zero mean value). A model with two orders of total differencing normally does not include a constant term.
  - In a model with one order of total differencing, a constant term should be included if the series has a non-zero average trend.

Limits of the ARIMA model

ARIMA models have several limits:

- They can only handle univariate data
- The paramaters are quite complex to guess and the properties difficult to assess when the parameters p and q grow beyond 2.
- These models do not allow for an in depth analysis of a time series (series segmentation event detection, state detection, etc.)

Outline

1 Fundamentals of time series analysis

2 The ARIMA model

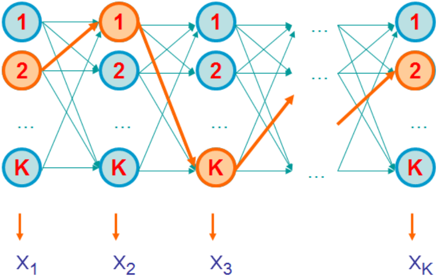
3 Introduction to Hidden Markov Models

4 Conclusion

Introduction to HMM

- Hidden Markov models (HMMs) are a ubiquitous tool for modeling time series data.
- Far from being limited to time series, they are widely used with all sorts of sequential data:
  - Speech recognition systems
  - Genome sequence analysis
  - Pattern recognition
  - Object tracking and computer vision
  - Etc.

Introduction to HMM



Definition

A hidden Markov model is a tool for representing probability distributions over sequences of observations. It is assumed that each observation X_t was generated by some process whose state $S_t \in [1..K]$ is unknown (hence the name hidden).

Introduction to HMM

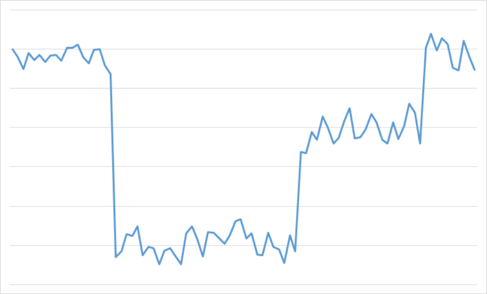


Figure: Example of a univariate sequence with 3 hidden states

HMM: Properties

- Each observation X_t depends only on the current state Q_t
- Markov property: Given the value of the state Q_{t-1}, the current state Q_t is independent from all states prior to $t-1$.

Definition of a HMM

A HMM is made of 5 key elements:

Definition of a HMM

A HMM is made of 5 key elements:

- **An alphabet** $\Sigma = \{o_1, \cdots, o_M\}$ which defines the form that the observations can take:
    - $\Sigma = \mathbb{R}^d, d \in \mathbb{N}^*$ for multi-dimensional observations
    - $\Sigma = \{1, 2, 3, 4, 5, 6\}$ for dice rolls
    - etc.

Definition of a HMM

A HMM is made of 5 key elements:

- **An alphabet** $\Sigma = \{o_1, \cdots, o_M\}$ which defines the form that the observations can take:
  - $\Sigma = \mathbb{R}^d, d \in \mathbb{N}^*$ for multi-dimensional observations
  - $\Sigma = \{1, 2, 3, 4, 5, 6\}$ for dice rolls
  - etc.
- **A set of states** $Q = \{1, \cdots, K\}$

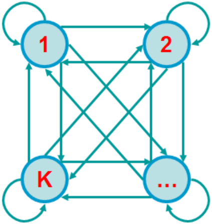


Figure: Example of a state transition diagram

Definition of a HMM

A HMM is made of 5 key elements:

- **An alphabet** $\Sigma = \{o_1, \cdots, o_M\}$ which defines the form that the observations can take:
  - $\Sigma = \mathbb{R}^d, d \in \mathbb{N}^*$ for multi-dimensional observations
  - $\Sigma = \{1, 2, 3, 4, 5, 6\}$ for dice rolls
  - etc.
- **A set of states** $Q = \{1, \cdots, K\}$

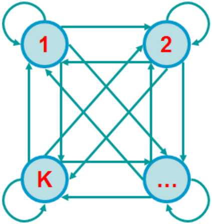


Figure: Example of a state transition diagram

- **A transition probability matrix** $A = (a_{ij})_{K \times K}, \quad \forall i \sum_j a_{ij} = 1$
  - a_{ij} is the probability of transition from state i to state j.

Definition of a HMM

A HMM is made of 5 key elements:

- **An alphabet** $\Sigma = \{o_1, \cdots, o_M\}$ which defines the form that the observations can take:
  - $\Sigma = \mathbb{R}^d, d \in \mathbb{N}^*$ for multi-dimensional observations
  - $\Sigma = \{1, 2, 3, 4, 5, 6\}$ for dice rolls
  - etc.
- **A set of states** $Q = \{1, \cdots, K\}$

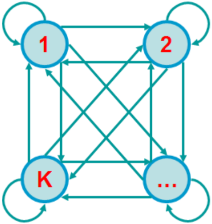


Figure: Example of a state transition diagram

- **A transition probability matrix** $A = (a_{ij})_{K \times K}, \quad \forall i \sum_j a_{ij} = 1$
  - a_{ij} is the probability of transition from state i to state j.
- **Emission probabilities** within each state:

$$e_i(x) = P(x|Q = i), \quad \forall i \in [1..K] \sum_{x \in \Sigma} e_i(x) = 1$$

Definition of a HMM

A HMM is made of 5 key elements:

- **An alphabet** $\Sigma = \{o_1, \cdots, o_M\}$ which defines the form that the observations can take:
  - $\Sigma = \mathbb{R}^d, d \in \mathbb{N}^*$ for multi-dimensional observations
  - $\Sigma = \{1,2,3,4,5,6\}$ for dice rolls
  - etc.
- **A set of states** $Q = \{1, \cdots, K\}$

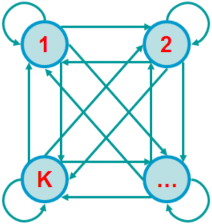


Figure: Example of a state transition diagram

- **A transition probability matrix** $A = (a_{ij})_{K \times K}, \quad \forall i \sum_j a_{ij} = 1$
  - a_{ij} is the probability of transition from state i to state j.
- **Emission probabilities** within each state:

$$e_i(x) = P(x|Q=i), \quad \forall i \in [1..K] \sum_{x \in \Sigma} e_i(x) = 1$$

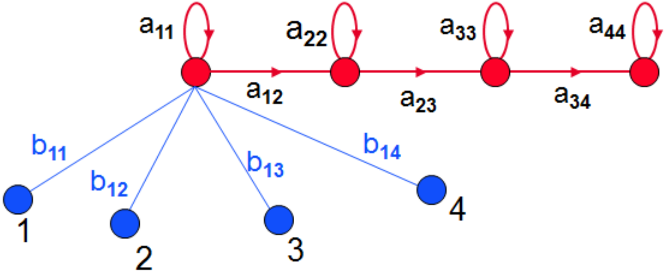
- **Starting probabilities**: $\pi_1, \cdots, \pi_K, \quad \sum_{i=1}^K \pi_i = 1$

Definition of a HMM

Notations

A HMM model is denoted $M = \{A, B, \pi\}$, where:

- A is the transition probability matrix
- B contains the emissions probability laws $e_i(x)$
- π the starting probabilities.



Question 1: Evaluating

The Evaluation Problem

- **Given**: a sequence of observations $X = \{x_1, \cdots, x_L\}$ and a model $M = \{A, B, \pi\}$
- **Question**: How do we efficiently compute $P(X|M)$, the probability of observing this sequence X knowing the model M.

- The probability $P(X|M)$ can be viewed as a measure of quality to evaluate the model M.
- It can be used to discriminate or select among alternative models $M_1, M_2, M_3, ...$

Question 2: Decoding

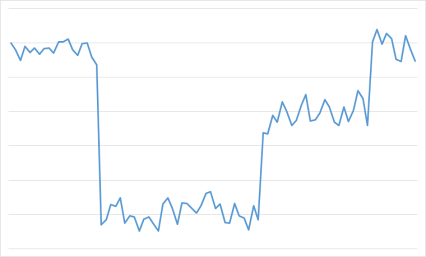
The Decoding Problem

- **Given**: a sequence of observations $X = \{x_1, \cdots, x_L\}$ and a model $M = \{A, B, \pi\}$
- **Question**: How do we compute the most probable sequence(s) of states $\boldsymbol{Q} = \{Q_1, \cdots, Q_L\}$?

Question 2: Decoding

The Decoding Problem

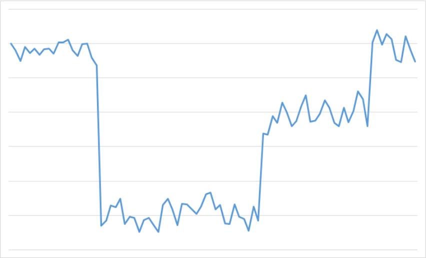
- **Given**: a sequence of observations $X = \{x_1, \cdots, x_L\}$ and a model $M = \{A, B, \pi\}$
- **Question**: How do we compute the most probable sequence(s) of states $\boldsymbol{Q} = \{Q_1, \cdots, Q_L\}$?

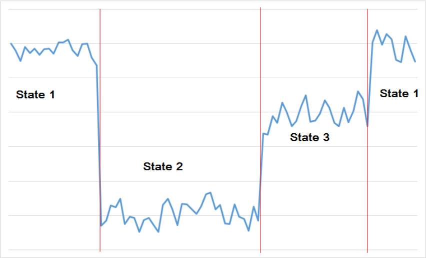


Question 2: Decoding

The Decoding Problem

- **Given**: a sequence of observations $X = \{x_1, \cdots, x_L\}$ and a model $M = \{A, B, \pi\}$
- **Question**: How do we compute the most probable sequence(s) of states $\boldsymbol{Q} = \{Q_1, \cdots, Q_L\}$?

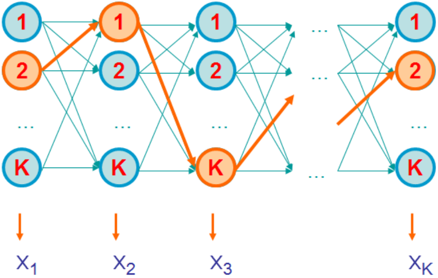




Question 3: Learning

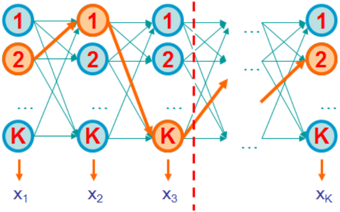
The Learning Problem

- **Given**: Just the sequence of observations $X = \{x_1, \cdots, x_L\}$
- **Question**: How do we learn $M = \{A, B, \pi\}$ and then figure out Q ?



The decoding problem

- Finding the sequence $\boldsymbol{Q} = \{Q_1, \cdots, Q_N\}$ of hidden states is the most common task with HMM.
- Given the observations $X = \{x_1, \cdots, x_N\}$, we want to find $\boldsymbol{Q}$ that maximizes $P(X, \boldsymbol{Q})$



Let us denote $V_k(t)$ the optimal sequence of state knowing that at time t we are in the state k:

$$V_k(t) = max_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_t, Q_1 \cdots Q_{t-1}, Q_t = k)$$

Decoding: Main Idea

Viterbi algorithm: Main idea

The idea of the Viterbi algorithm is to recursively build the $V_k(i)$

- Knowing that $V_k(t) = max_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_t, Q_1 \cdots Q_{t-1}, Q_t = k)$, how do we define $V_l(t+1)$?

Decoding: Main Idea

Viterbi algorithm: Main idea

The idea of the Viterbi algorithm is to recursively build the $V_k(i)$

- Knowing that $V_k(t) = max_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_t, Q_1 \cdots Q_{t-1}, Q_t = k)$, how do we define $V_l(t+1)$?

$$V_l(t+1) = max_{Q_1 \cdots Q_t} P(x_1 \cdots x_{t+1}, Q_1 \cdots Q_t, Q_{t+1} = l)$$

Decoding: Main Idea

Viterbi algorithm: Main idea

The idea of the Viterbi algorithm is to recursively build the $V_k(i)$

- Knowing that $V_k(t) = max_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_t, Q_1 \cdots Q_{t-1}, Q_t = k)$, how do we define $V_l(t+1)$?

$$\begin{aligned} V_l(t+1) &= max_{Q_1 \cdots Q_t} P(x_1 \cdots x_{t+1}, Q_1 \cdots Q_t, Q_{t+1} = l) \\ &= max_{Q_1 \cdots Q_t} P(x_{t+1}, Q_{t+1} = l | x_1 \cdots x_t, Q_1 \cdots Q_t) P(x_1 \cdots x_t, Q_1 \cdots Q_t) \end{aligned}$$

Decoding: Main Idea

Viterbi algorithm: Main idea

The idea of the Viterbi algorithm is to recursively build the $V_k(i)$

- Knowing that $V_k(t) = max_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_t, Q_1 \cdots Q_{t-1}, Q_t = k)$, how do we define $V_l(t+1)$?

$$\begin{aligned}
V_l(t+1) &= max_{Q_1 \cdots Q_t} P(x_1 \cdots x_{t+1}, Q_1 \cdots Q_t, Q_{t+1} = l) \\
&= max_{Q_1 \cdots Q_t} P(x_{t+1}, Q_{t+1} = l | x_1 \cdots x_t, Q_1 \cdots Q_t) P(x_1 \cdots x_t, Q_1 \cdots Q_t) \\
&= max_{Q_1 \cdots Q_t} P(x_{t+1}, Q_{t+1} = l | Q_t) P(x_1 \cdots x_t, Q_1 \cdots Q_t)
\end{aligned}$$

Decoding: Main Idea

Viterbi algorithm: Main idea

The idea of the Viterbi algorithm is to recursively build the $V_k(i)$

- Knowing that $V_k(t) = max_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_t, Q_1 \cdots Q_{t-1}, Q_t = k)$, how do we define $V_l(t+1)$?

$$\begin{aligned}
V_l(t+1) &= max_{Q_1 \cdots Q_t} P(x_1 \cdots x_{t+1}, Q_1 \cdots Q_t, Q_{t+1} = l) \\
&= max_{Q_1 \cdots Q_t} P(x_{t+1}, Q_{t+1} = l | x_1 \cdots x_t, Q_1 \cdots Q_t) P(x_1 \cdots x_t, Q_1 \cdots Q_t) \\
&= max_{Q_1 \cdots Q_t} P(x_{t+1}, Q_{t+1} = l | Q_t) P(x_1 \cdots x_t, Q_1 \cdots Q_t) \\
&= max_k \left[P(x_{t+1}, Q_{t+1} = l | Q_t = k) max_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_t, Q_1 \cdots Q_{t-1}, Q_t = k) \right]
\end{aligned}$$

Decoding: Main Idea

Viterbi algorithm: Main idea

The idea of the Viterbi algorithm is to recursively build the $V_k(i)$

- Knowing that $V_k(t) = max_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_t, Q_1 \cdots Q_{t-1}, Q_t = k)$, how do we define $V_l(t+1)$?

$$\begin{aligned}
V_l(t+1) &= max_{Q_1 \cdots Q_t} P(x_1 \cdots x_{t+1}, Q_1 \cdots Q_t, Q_{t+1} = l) \\
&= max_{Q_1 \cdots Q_t} P(x_{t+1}, Q_{t+1} = l | x_1 \cdots x_t, Q_1 \cdots Q_t) P(x_1 \cdots x_t, Q_1 \cdots Q_t) \\
&= max_{Q_1 \cdots Q_t} P(x_{t+1}, Q_{t+1} = l | Q_t) P(x_1 \cdots x_t, Q_1 \cdots Q_t) \\
&= max_k \left[P(x_{t+1}, Q_{t+1} = l | Q_t = k) max_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_t, Q_1 \cdots Q_{t-1}, Q_t = k) \right] \\
&= max_k \left[P(x_{t+1} | Q_{t+1} = l) P(Q_{t+1} = l | Q_t = k) V_k(t) \right]
\end{aligned}$$

Decoding: Main Idea

Viterbi algorithm: Main idea

The idea of the Viterbi algorithm is to recursively build the $V_k(i)$

- Knowing that $V_k(t) = max_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_t, Q_1 \cdots Q_{t-1}, Q_t = k)$, how do we define $V_l(t+1)$?

$$\begin{aligned}
V_l(t+1) &= max_{Q_1 \cdots Q_t} P(x_1 \cdots x_{t+1}, Q_1 \cdots Q_t, Q_{t+1} = l) \\
&= max_{Q_1 \cdots Q_t} P(x_{t+1}, Q_{t+1} = l | x_1 \cdots x_t, Q_1 \cdots Q_t) P(x_1 \cdots x_t, Q_1 \cdots Q_t) \\
&= max_{Q_1 \cdots Q_t} P(x_{t+1}, Q_{t+1} = l | Q_t) P(x_1 \cdots x_t, Q_1 \cdots Q_t) \\
&= max_k \left[P(x_{t+1}, Q_{t+1} = l | Q_t = k) max_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_t, Q_1 \cdots Q_{t-1}, Q_t = k) \right] \\
&= max_k \left[P(x_{t+1} | Q_{t+1} = l) P(Q_{t+1} = l | Q_t = k) V_k(t) \right] \\
&= e_l(x_{t+1}) max_k \left[a_{kl} V_k(t) \right]
\end{aligned}$$

The Viterbi Algorithm

Initialization:

- $V_0(0) = 1$
- $\forall k > 0 \quad V_k(0) = 0$

Iteration:

- $V_k(t) = e_j(x_t) max_q \left[a_{qk} V_q(t-1)\right]$
- $Ptr_k(t) = argmax_q \left[a_{qk} V_q(t-1)\right]$

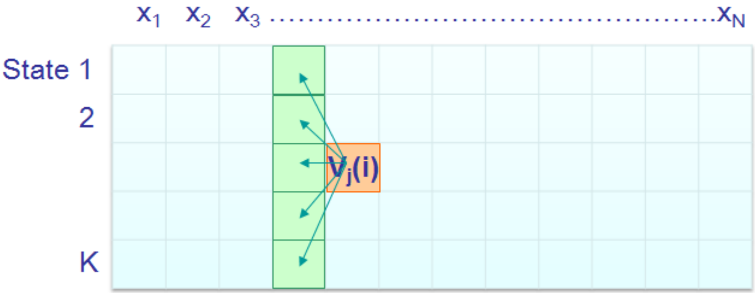
Termination:

$$P(X, \boldsymbol{Q}^*) = max_k V_k(N)$$

Traceback:

$Q_N^* = argmax_k V_k(N)$
$Q_{t-1}^* = Ptr_{Q_t}(t)$

The Viterbi Algorithm



complexity of the Viterbi algorithm

- Time complexity $O(K^2N)$
- Space complexity $O(KN)$

The Viterbi Algorithm

An important remark on a practical detail

Underflows are a major problem with the Viterbi algorithm.

$$P(x_1, \cdots, x_t, Q_1, \cdots, Q_t) = \pi_{Q_1} \cdot a_{Q_1,Q_2} \cdot a_{Q_2,Q_3} \cdots a_{Q_{t-1},Q_t} \cdot e_{Q_1}(x_1) \cdots e_{Q_t}(x_t)$$

These numbes are extremely small, thus leading to a quick underflow.

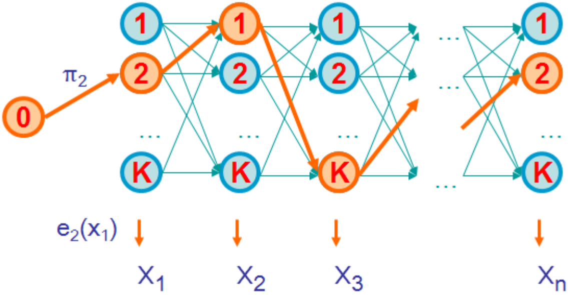
A good solution is to use the logarithm of all values, e.g:

$$\log V_k(t) = \log e_j(x_t) + \log\left(max_q\left[a_{qk} \times V_q(t-1)\right]\right)$$

Generating a sequence by the model

Given a HMM, we can generate a sequence of length n as follows:

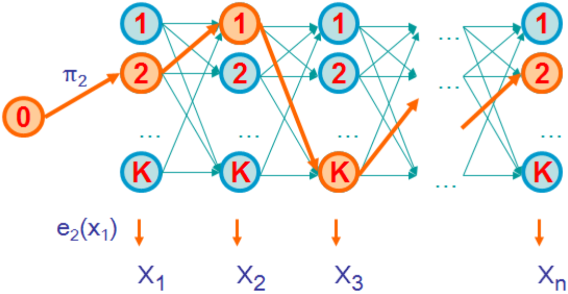
1. Start at state Q_1 according to π_k
2. Emit observation x_1 according to $e_{Q_1}(x_1)$
3. Go to state Q_2 according to a_{Q_1,Q_2}
4. ... until emitting x_n



Evaluation

We want an algorithm that can compute:

- $P(X)$ the probability of X given the model
- $P(x_i \cdots x_j)$ the probability of a substring of X given the model
- $P(Q_i = k|X)$ the posterior probability that the i^{th} state is k, given X



The Forward Algorithm

We want to calculate $P(X)$, the probability of the whole sequence given the HMM.
We can sum all possible ways of generating X:

$$P(X) = \sum_{\boldsymbol{Q}} P(X, \boldsymbol{Q}) = \sum_{\boldsymbol{Q}} P(X|\boldsymbol{Q})P(\boldsymbol{Q})$$

To avoid summing over an exponential number of paths $\boldsymbol{Q}$, we define the **forward probability**:

$$f_k(i) = P(x_1 \cdots x_t, Q_t = k)$$

It is the probability of observing the sequence $x_1 \cdots x_t$ and having the t^{th} state being k.

The Forward Algorithm: Derivation

$$f_k(t) = P(x_1 \cdots x_t, Q_t = k)$$

The Forward Algorithm: Derivation

$$
\begin{aligned}
f_k(t) &= P(x_1 \cdots x_t, Q_t = k) \\
&= \sum_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_{t-1}, Q_1 \cdots Q_{t-1}, Q_t = k) P(x_t | Q_t = k)
\end{aligned}
$$

The Forward Algorithm: Derivation

$$\begin{aligned} f_k(t) &= P(x_1 \cdots x_t, Q_t = k) \\ &= \sum_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_{t-1}, Q_1 \cdots Q_{t-1}, Q_t = k) P(x_t | Q_t = k) \\ &= \sum_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_{t-1}, Q_1 \cdots Q_{t-1}, Q_t = k) e_k(x_t) \end{aligned}$$

The Forward Algorithm: Derivation

$$\begin{aligned}
f_k(t) &= P(x_1 \cdots x_t, Q_t = k) \\
&= \sum_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_{t-1}, Q_1 \cdots Q_{t-1}, Q_t = k) P(x_t | Q_t = k) \\
&= \sum_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_{t-1}, Q_1 \cdots Q_{t-1}, Q_t = k) e_k(x_t) \\
&= \sum_{l} \sum_{Q_1 \cdots Q_{t-2}} P(x_1 \cdots x_{t-1}, Q_1 \cdots Q_{t-2}, Q_{t-1} = l) a_{lk} e_k(x_t)
\end{aligned}$$

The Forward Algorithm: Derivation

$$
\begin{aligned}
f_k(t) &= P(x_1 \cdots x_t, Q_t = k) \\
&= \sum_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_{t-1}, Q_1 \cdots Q_{t-1}, Q_t = k) P(x_t | Q_t = k) \\
&= \sum_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_{t-1}, Q_1 \cdots Q_{t-1}, Q_t = k) e_k(x_t) \\
&= \sum_{l} \sum_{Q_1 \cdots Q_{t-2}} P(x_1 \cdots x_{t-1}, Q_1 \cdots Q_{t-2}, Q_{t-1} = l) a_{lk} e_k(x_t) \\
&= \sum_{l} P(x_1 \cdots x_{t-1}, Q_{t-1} = l) a_{lk} e_k(x_t)
\end{aligned}
$$

The Forward Algorithm: Derivation

$$
\begin{aligned}
f_k(t) &= P(x_1 \cdots x_t, Q_t = k) \\
&= \sum_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_{t-1}, Q_1 \cdots Q_{t-1}, Q_t = k) P(x_t | Q_t = k) \\
&= \sum_{Q_1 \cdots Q_{t-1}} P(x_1 \cdots x_{t-1}, Q_1 \cdots Q_{t-1}, Q_t = k) e_k(x_t) \\
&= \sum_l \sum_{Q_1 \cdots Q_{t-2}} P(x_1 \cdots x_{t-1}, Q_1 \cdots Q_{t-2}, Q_{t-1} = l) a_{lk} e_k(x_t) \\
&= \sum_l P(x_1 \cdots x_{t-1}, Q_{t-1} = l) a_{lk} e_k(x_t) \\
&= e_k(x_t) \sum_l f_l(t-1) a_{lk}
\end{aligned}
$$

The Forward Algorithm

We can compute all the $f_k(t)$ using dynamic programming:

Initialization:

- $f_0(0) = 1$
- $\forall k > 0 \quad f_k(0) = 0$

Iteration:

$$f_k(i) = e_k(x_t) \sum_l f_l(t-1) a_{lk}$$

Termination:

$$P(X) = \sum_k f_k(N)$$

The Backward Algorithm

Motivation

To unlock the sequence of hidden states, want to compute $P(Q_t = k|X)$ the probability distribution on the t^{th} position given X.

Since we have $P(Q_t = k|X) = \frac{P(Q_t=k,X)}{P(X)}$, we start by computing $P(Q_t = k, X)$:

The Backward Algorithm

Motivation

To unlock the sequence of hidden states, want to compute $P(Q_t = k|X)$ the probability distribution on the t^{th} position given X.

Since we have $P(Q_t = k|X) = \frac{P(Q_t=k,X)}{P(X)}$, we start by computing $P(Q_t = k, X)$:

$$P(Q_t = k, X) = P(x_1 \cdots x_t, Q_t = k, x_{t+1} \cdots x_N)$$

The Backward Algorithm

Motivation

To unlock the sequence of hidden states, want to compute $P(Q_t = k|X)$ the probability distribution on the t^{th} position given X.

Since we have $P(Q_t = k|X) = \frac{P(Q_t=k,X)}{P(X)}$, we start by computing $P(Q_t = k, X)$:

$$\begin{aligned} P(Q_t = k, X) &= P(x_1 \cdots x_t, Q_t = k, x_{t+1} \cdots x_N) \\ &= P(x_1 \cdots x_t, Q_t = k) P(x_{t+1} \cdots x_N | Q_t = k, x_1 \cdots x_t) \end{aligned}$$

The Backward Algorithm

Motivation

To unlock the sequence of hidden states, want to compute $P(Q_t = k|X)$ the probability distribution on the t^{th} position given X.

Since we have $P(Q_t = k|X) = \frac{P(Q_t=k,X)}{P(X)}$, we start by computing $P(Q_t = k, X)$:

$$\begin{aligned} P(Q_t = k, X) &= P(x_1 \cdots x_t, Q_t = k, x_{t+1} \cdots x_N) \\ &= P(x_1 \cdots x_t, Q_t = k) P(x_{t+1} \cdots x_N | Q_t = k, x_1 \cdots x_t) \\ &= P(x_1 \cdots x_t, Q_t = k) P(x_{t+1} \cdots x_N | Q_t = k) \end{aligned}$$

The Backward Algorithm

Motivation

To unlock the sequence of hidden states, want to compute $P(Q_t = k|X)$ the probability distribution on the t^{th} position given X.

Since we have $P(Q_t = k|X) = \frac{P(Q_t=k,X)}{P(X)}$, we start by computing $P(Q_t = k, X)$:

$$\begin{aligned} P(Q_t = k, X) &= P(x_1 \cdots x_t, Q_t = k, x_{t+1} \cdots x_N) \\ &= P(x_1 \cdots x_t, Q_t = k) P(x_{t+1} \cdots x_N | Q_t = k, x_1 \cdots x_t) \\ &= P(x_1 \cdots x_t, Q_t = k) P(x_{t+1} \cdots x_N | Q_t = k) \\ &= f_k(t) b_k(t) \end{aligned}$$

Let us note $b_k(t)$ the backward probability.

The Backward Algorithm: Derivation

$$b_k(t) = P(x_{t+1} \cdots x_N | Q_t = k)$$

The Backward Algorithm: Derivation

$$\begin{aligned} b_k(t) &= P(x_{t+1} \cdots x_N | Q_t = k) \\ &= \sum_{Q_{t+1} \cdots Q_N} P(x_{t+1} \cdots x_N, Q_{t+1} \cdots Q_N | Q_t = k) \end{aligned}$$

The Backward Algorithm: Derivation

$$
\begin{aligned}
b_k(t) &= P(x_{t+1} \cdots x_N | Q_t = k) \\
&= \sum_{Q_{t+1} \cdots Q_N} P(x_{t+1} \cdots x_N, Q_{t+1} \cdots Q_N | Q_t = k) \\
&= \sum_{l} \sum_{Q_{t+1} \cdots Q_N} P(x_{t+1} \cdots x_N, Q_{t+1} = l, Q_{t+2} \cdots Q_N | Q_t = k)
\end{aligned}
$$

The Backward Algorithm: Derivation

$$
\begin{aligned}
b_k(t) &= P(x_{t+1} \cdots x_N | Q_t = k) \\
&= \sum_{Q_{t+1} \cdots Q_N} P(x_{t+1} \cdots x_N, Q_{t+1} \cdots Q_N | Q_t = k) \\
&= \sum_{l} \sum_{Q_{t+1} \cdots Q_N} P(x_{t+1} \cdots x_N, Q_{t+1} = l, Q_{t+2} \cdots Q_N | Q_t = k) \\
&= \sum_{l} P(x_{t+1} | Q_{t+1} = l) a_{kl} \sum_{Q_{t+2} \cdots Q_N} P(x_{t+2} \cdots x_N, Q_{t+2} \cdots Q_N | Q_{t+1} = l)
\end{aligned}
$$

The Backward Algorithm: Derivation

$$
\begin{aligned}
b_k(t) &= P(x_{t+1} \cdots x_N | Q_t = k) \\
&= \sum_{Q_{t+1} \cdots Q_N} P(x_{t+1} \cdots x_N, Q_{t+1} \cdots Q_N | Q_t = k) \\
&= \sum_l \sum_{Q_{t+1} \cdots Q_N} P(x_{t+1} \cdots x_N, Q_{t+1} = l, Q_{t+2} \cdots Q_N | Q_t = k) \\
&= \sum_l P(x_{t+1} | Q_{t+1} = l) a_{kl} \sum_{Q_{t+2} \cdots Q_N} P(x_{t+2} \cdots x_N, Q_{t+2} \cdots Q_N | Q_{t+1} = l) \\
&= \sum_l e_l(x_{t+1}) a_{kl} \sum_{Q_{t+2} \cdots Q_N} P(x_{t+2} \cdots x_N, Q_{t+2} \cdots Q_N | Q_{t+1} = l)
\end{aligned}
$$

The Backward Algorithm: Derivation

$$
\begin{aligned}
b_k(t) &= P(x_{t+1}\cdots x_N | Q_t = k) \\
&= \sum_{Q_{t+1}\cdots Q_N} P(x_{t+1}\cdots x_N, Q_{t+1}\cdots Q_N | Q_t = k) \\
&= \sum_l \sum_{Q_{t+1}\cdots Q_N} P(x_{t+1}\cdots x_N, Q_{t+1} = l, Q_{t+2}\cdots Q_N | Q_t = k) \\
&= \sum_l P(x_{t+1} | Q_{t+1} = l) a_{kl} \sum_{Q_{t+2}\cdots Q_N} P(x_{t+2}\cdots x_N, Q_{t+2}\cdots Q_N | Q_{t+1} = l) \\
&= \sum_l e_l(x_{t+1}) a_{kl} \sum_{Q_{t+2}\cdots Q_N} P(x_{t+2}\cdots x_N, Q_{t+2}\cdots Q_N | Q_{t+1} = l) \\
&= \sum_l e_l(x_{t+1}) a_{kl} b_l(t+1)
\end{aligned}
$$

The Backward Algorithm

We can compute all the $b_k(t)$ using dynamic programming:

Initialization:

- $\forall k \quad b_k(N) = 0$

Iteration:

$$b_k(i) = \sum_l e_l(x_{t+1}) a_{kl} b_l(t+1)$$

Termination:

$$P(X) = \sum_l \pi_l e_l(x_1) b_l(1)$$

Computational complexity

Complexity analysis for the Forward and Backward algorithms

- Time complexity $O(K^2N)$
- Space complexity $O(KN)$

Like for the Viterbi algorithm, underflows can be a problem:

- Method 1: Use sums of log
- Method 2: Rescale every few positions by multiplying by a constant

Posterior decoding

Using both Forward and backward algorithms, we can now calculate:

$$P(Q_t = k|X) = \frac{f_k(t)b_k(t)}{P(X)}$$

Posterior decoding

Using both Forward and backward algorithms, we can now calculate:

$$P(Q_t = k|X) = \frac{f_k(t)b_k(t)}{P(X)}$$

And from there we can deduce the most likely hidden state at a position i of a sequence X:

$$\widehat{Q_t} = argmax_k P(Q_t = k|X)$$

Posterior decoding

Using both Forward and backward algorithms, we can now calculate:

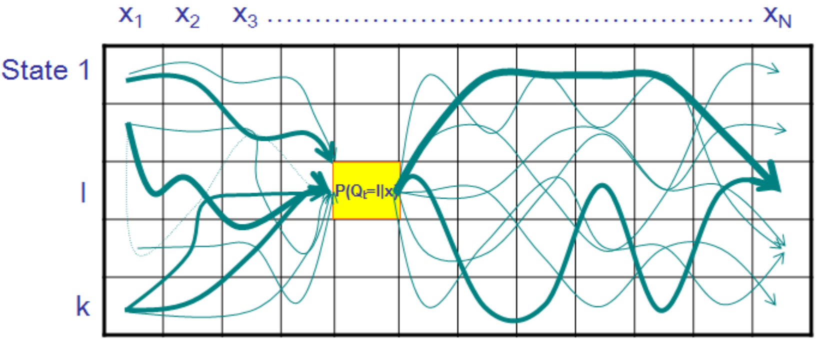
$$P(Q_t = k|X) = \frac{f_k(t)b_k(t)}{P(X)}$$

And from there we can deduce the most likely hidden state at a position i of a sequence X:

$$\widehat{Q_t} = argmax_k P(Q_t = k|X)$$

- From there, the posterior decoding gives us a likelihood curve of states for each position.
- This is sometimes more informative than Viterbi path $\boldsymbol{Q}^*$

Posterior decoding



The learning problem

- If the hidden states are known, it is easy to guess the parameters:
  - Count the transitions to guess A
  - Count or measure the distribution of the observations in each states to guess the emission parameters$B$.
- If the parameters A, B and π are known, the sequence of hidden states can be guessed using Viterbi or the Forward-Backward algorithm.

The learning problem

- If the hidden states are known, it is easy to guess the parameters:
  - Count the transitions to guess A
  - Count or measure the distribution of the observations in each states to guess the emission parameters$B$.
- If the parameters A, B and π are known, the sequence of hidden states can be guessed using Viterbi or the Forward-Backward algorithm.
- We have a "Chicken and egg" problem.

The learning problem: Use the EM algorithm

This kind of problem can be solved using the EM algorithm (Cf. Lecture 4) to alternatively optimize both the model and the most likely path.

- **Initialization**: Guess the initial HMM parameters $M = \{A, B, \pi\}$
- Repeat until convergence:
    - **E-Step**: Compute the distribution overs paths.
    - **M-Step**: Compute the maximum likelihood parameters.

The Baum-Welch algorithm

The Forward-Backward algorithm can be adapted into an EM like procedure to learn the model. This is known as the Baum-Welch algorithm.

Repeat until convergence:

- Compute the probability of each state at each position using forward and backward probabilities
    - This gives the expected distribution of the observations for each state using Bayes Theorem.
- Compute the probability of each pair of states at each pair of consecutive positions t and $t+1$ using forward(t) and backward(t+1)
    - This gives the expected transition counts.

$$Count(k \to l) = \frac{\sum_i f_k(t) a_{kl} b_l(t+1)}{P(X)}$$

Outline

1 Fundamentals of time series analysis

2 The ARIMA model

3 Introduction to Hidden Markov Models

4 Conclusion

ARMA or HMM, when to use which ?

Type of data

- If your observations/data seem (or are known) to depend upon previous observations, use ARMA.
- If your observations/data are independent in time, use HMM.
- For HMM series, each state is assumed to have stationary properties. With ARMA series, it depends on the parameters.
- With cyclic data, ARMA models (and their evolved ARIMA version) are usually better.
- For stationary data with a single observable event, HMM are usualy a bit too much and ARMA models are enough.
- For multi-dimensional data, HMM should be preferred.

ARMA or HMM, when to use which ?

Type of tasks

- ARMA models are dedicated to modeling time dependent events and to predict the evolution of a phenomenon in time.
- HMM usually perform poorly in long term predictions.
- For time data segmentation and modeling, HMM should be used.
- Both ARMA and HMM can be used to fill in missing data.
- For data sequences analysis other than time data, HMM are usually preferred.