MAP 565
Time series analysis : Lecture V

François Roueff
http://perso.telecom-paristech.fr/~roueff/

Telecom ParisTech – École Polytechnique

January 13, 2016

Outline of the course

- ▷ Stochastic modeling
  - I Random processes.
  - II Spectral representation.
- ▷ Linear models
  - III Linear filtering, innovation process.
  - IV ARMA processes.
  - V Linear forecasting. $\leftarrow$
- ▷ Statistical inference
  - VI Overview of goals and methods.
  - VII Asymptotic statistics in a dependent context.
- ▷ Non-linear models
  - VIII Standard models for financial time series.
  - IX Complements.

$\leftarrow$: we are here.

Outline of lectures V

1. Linear prediction
   - Prediction VS linear prediction
   - Linear prediction for weakly stationary processes

2. An illustrative example with R

1. Linear prediction
   - Prediction VS linear prediction
   - Linear prediction for weakly stationary processes

2. An illustrative example with R

1. Linear prediction
   - Prediction VS linear prediction
   - Linear prediction for weakly stationary processes

2. An illustrative example with R

Conditional expectation

Definition : conditional expectation

Let X be a real valued random variable defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Let $\mathcal{G}$ be a sub-σ-field of $\mathcal{F}$.

(a) Suppose $X \in L^2(\Omega, \mathcal{F}, \mathbb{P})$. The conditional expectation of X given $\mathcal{G}$ is defined by

$$\mathbb{E}\left[X \middle| \mathcal{G}\right] = \operatorname{proj}\left(X \middle| L^2(\Omega, \mathcal{G}, \mathbb{P})\right) \ .$$

(b) It is equivalently characterized (in the a.s. sense) by

(i) $\mathbb{E}\left[X \middle| \mathcal{G}\right] \in L^1(\Omega, \mathcal{G}, \mathbb{P})$.

(ii) For all $A \in \mathcal{G}$, we have $\mathbb{E}\left[X \mathbb{1}_A\right] = \mathbb{E}\left[\mathbb{E}\left[X \middle| \mathcal{G}\right] \mathbb{1}_A\right]$.

If $X \in L^1(\Omega, \mathcal{F}, \mathbb{P})$, definition (b) remains valid.

Conditional expectation with respect to random variables

If $\mathcal{G} = \sigma(Z_t, t \in T)$, we denote

$$\mathbb{E}\left[X \middle| Z_t, t \in T\right] = \mathbb{E}\left[X \middle| \mathcal{G}\right] \ .$$

Basic properties

Conditional density

If (X, Y) admits a density f with respect to $\xi \otimes \xi'$, then, for all real valued g, $\mathbb{E}\left[g(X) \mid Y\right] = \widehat{g}(Y)$ with $\widehat{g}(y) = \int g(x)\, f(x|y)\, \xi(\mathrm{d}x)$ and $f(x|y) = f(x, y) / \displaystyle\int f(x', y)\, \xi(\mathrm{d}x')$.

Some standard properties

Let $X \in L^1(\Omega, \mathcal{F}, \mathbb{P})$.

(P-i) If X is $\mathcal{G}$-measurable, $\mathbb{E}\left[X \mid \mathcal{G}\right] = X$.

(P-ii) If X is independent of $\mathcal{G}$, $\mathbb{E}\left[X \mid \mathcal{G}\right] = \mathbb{E}\left[X\right]$

(P-iii) If Y is $\sigma(\mathcal{G})$-meas. and $\mathbb{E}[|XY|] < \infty$, $\mathbb{E}\left[XY \mid \mathcal{G}\right] = Y\mathbb{E}\left[X \mid \mathcal{G}\right]$.

(P-iv) If $\mathcal{G} \subset \mathcal{H}$, $\mathbb{E}\left[\mathbb{E}\left[X \mid \mathcal{H}\right] \mid \mathcal{G}\right] = \mathbb{E}\left[X \mid \mathcal{G}\right]$ (tower property).

(P-v) If $X = F(Y, Z)$ with Y $\mathcal{G}$-measurable and Z independent of $\mathcal{G}$, then $\mathbb{E}\left[X \mid \mathcal{G}\right] = \widehat{F}(Y)$, where, for all y, $\widehat{F}(y) = \mathbb{E}\left[F(y, Z)\right]$.

Prediction VS linear prediction

If $X \in L^2(\Omega, \mathcal{F}, \mathbb{P})$, then $\mathbb{E}\left[X \middle| \mathcal{G}\right] = \mathrm{proj}\left(X \middle| L^2(\Omega, \mathcal{G}, \mathbb{P})\right)$, hence

$$\mathbb{E}\left[\left(X - \mathbb{E}\left[X \middle| \mathcal{G}\right]\right)^2\right] = \inf\left\{\mathbb{E}\left[(X - Y)^2\right] \ : \ Y \in L^2(\Omega, \mathcal{G}, \mathbb{P})\right\} .$$

We say that $\mathbb{E}\left[X \middle| \mathcal{G}\right]$ is the best predictor of X given $\mathcal{G}$.

In particular, if $\mathcal{G} = \sigma(Z_t, t \in T)$, then $\mathbb{E}\left[X \middle| \mathcal{G}\right]$ can be any measurable function of $(Z_t)_{t \in T}$. Thus, in the case where $(Z_t)_{t \in T}$ is an L^2 process,

$$\mathbb{E}\left[\left(X - \mathbb{E}\left[X \middle| Z_t, t \in T\right]\right)^2\right] \leq \mathbb{E}\left[\left(X - \mathrm{proj}\left(X \middle| \overline{\mathrm{Span}}\left(Z_t, t \in T\right)\right)\right)^2\right] ,$$

and the equality only occurs when

$$\mathbb{E}\left[X \middle| Z_t, t \in T\right] = \mathrm{proj}\left(X \middle| \overline{\mathrm{Span}}\left(Z_t, t \in T\right)\right) \quad \text{a.s.}$$

In general, the best linear predictor does not achieve the same prediction error as the best predictor but is much easier to determine.

The Gaussian assumption

Theorem

Let $(X_t)_{t \in T}$ be a Gaussian process. Then, for any $t \in T$ and any countable set $I \subset T$, the conditional expectation of X_t given $(X_s)_{s \in I}$ is in $\overline{\mathrm{Span}}\left(1, (X_s)_{s \in I}\right)$, that is,

$$\mathbb{E}\left[X_t \middle| X_s, s \in I\right] = \mathrm{proj}\left(X_t \middle| \overline{\mathrm{Span}}\left(1, X_s, s \in I\right)\right) \quad \text{a.s.} \tag{1}$$

In other words,

Under the Gaussian assumption,
best predictor = best linear (or affine) predictor.

Remark

Since Gaussian processes are L^2 processes, we can rely on projections in the Hilbert space L^2 and (1) is equivalent to

$$\mathrm{proj}\left(X_t \middle| L^2(\Omega, \sigma(X_s, s \in I), \mathbb{P})\right) = \mathrm{proj}\left(X_t \middle| \overline{\mathrm{Span}}\left(1, X_s, s \in I\right)\right) .$$

From a countable to a finite I.

The following lemma for general Hilbert spaces implies that it suffices to consider the case where I is finite.

Lemma

Let $\mathcal{H}$ be a Hilbert space and $(E_p)_{p\geq 1}$ be a non-decreasing sequence of closed linear subspaces of $\mathcal{H}$. Then, for all $x \in \mathcal{H}$,

$$\lim_{p\to\infty} \operatorname{proj}\left(x \middle| E_p\right) = \operatorname{proj}\left(x \middle| E\right) \quad \text{with} \quad E = \overline{\bigcup_{p\geq 1} E_p}\,.$$

Let $I = \{t_1, t_2, t_2, \dots\}$. Then we use that

$$\overline{\operatorname{Span}}\left(1, X_s, s \in I\right) = \overline{\bigcup_{p\geq 1} \operatorname{Span}\left(1, X_{t_k}, k = 1, \dots, p\right)}$$

and

$$L^2(\Omega, \sigma(X_s, s \in I), \mathbb{P}) = \overline{\bigcup_{p\geq 1} L^2(\Omega, \sigma(X_{t_k}, k = 1, \dots, p), \mathbb{P})}\,.$$

Proof for a finite I.

Let $\begin{bmatrix} X & Z^T \end{bmatrix}^T$ be a Gaussian vector.

Start with the best linear predictor. Denote

$$\widehat{X} = \operatorname{proj}\left(X \middle| \operatorname{Span}(1, Z) \right) .$$

We may thus write $X = Y + \widehat{X}$ with $Y = X - \widehat{X}$, and notice that

$$\mathbb{E}[Y] = \langle Y, 1 \rangle = 0 \quad \text{and} \quad \operatorname{Cov}(Y, Z) = \langle Y, Z \rangle = 0 .$$

On the other hand, since $\widehat{X} \in \operatorname{Span}(1, Z)$, we have that $\begin{bmatrix} Y & Z^T \end{bmatrix}^T$ is an affine function of $\begin{bmatrix} X & Z^T \end{bmatrix}^T$ and thus a Gaussian vector. We conclude that Y and Z are independent and, by (P-ii), we get

$$\mathbb{E}\left[X \middle| Z \right] = \mathbb{E}[Y] + \mathbb{E}\left[\widehat{X} \middle| Z \right] = 0 + \widehat{X} = \operatorname{proj}\left(X \middle| \operatorname{Span}(1, Z) \right) .$$

1. Linear prediction
   - Prediction VS linear prediction
   - Linear prediction for weakly stationary processes

2. An illustrative example with R

Linear prediction : general idea

Basic assumption

Let $X = (X_t)_{t\in\mathbb{Z}}$ be a weakly stationary time series.

Recall the Wold decomposition

$$X_t = \text{mean} + \underbrace{\sum_{k\geq 0} \psi_k \epsilon_{t-k}}_{\text{purely non-det. process}} + \text{deterministic process} ,$$

where $(\epsilon_t)_{t\in\mathbb{Z}}$ is the innovation (white noise) defined by

$$\begin{aligned}\epsilon_t &= X_t - \operatorname{proj}\left(X_t \middle| \mathcal{H}_{t-1}^X \right) \\ &= X_t - \lim_{p\to\infty} \operatorname{proj}\left(X_t \middle| \mathcal{H}_{t-1,p}^X \right) .\end{aligned}$$

From now on, we only consider the centered purely non-deterministic part, so we assume that X is centered and purely non-deterministic.

Linear prediction : general idea (cont.)

There are two different ways to consider the problem of linear prediction :

- Model-based linear prediction : a parametric model has been determined (either because the system producing the data is well known and understood or because a model has been statistically inferred from historical data). In this case, use the best linear predictor of the corresponding model. Examples of models :
  - ARMA(p, q) models.
  - Dynamic linear models.
  - Extension to non-linear models (and non-linear prediction).
- Direct calculation of linear prediction coefficients using the autocovariance function γ.

AR(p) model

Consider an AR(p) model

$$X_t = \sum_{k=1}^{p} \phi_k X_{t-k} + \epsilon_t \; .$$

Many successful applications:

- Statistical inference : estimate the parameters of the model and use them for time series analysis.
- Forecasting : use the AR(p) best linear predictor.
- Coding : use the AR(p) representation to code, transmit and reconstruct a signal (as in speech coding in the standards of GSM).

Yule Walker equations

Definition : linear prediction coefficients of order p

Let $p \geq 1$. The forward linear prediction coefficients of order p, denoted by $\boldsymbol{\phi}_p^+ = \begin{bmatrix} \phi_{1,p}^+ & \dots & \phi_{p,p}^+ \end{bmatrix}$, are defined by

$$\operatorname{proj}\left(X_t \middle| \mathcal{H}_{t-1,p}^X \right) = \sum_{k=1}^{p} \phi_{k,p}^+ X_{t-k} \;,$$

which is equivalent to

$$\Gamma_p^+ \boldsymbol{\phi}_p^+ = \gamma_p^+ \;, \tag{2}$$

where $\quad \gamma_p^+ = \operatorname{Cov}\left(X_t, \begin{bmatrix} X_{t-1} \\ \vdots \\ X_{t-p} \end{bmatrix} \right)^T$ and $\quad \Gamma_p^+ = \operatorname{Cov}\left(\begin{bmatrix} X_{t-1} \\ \vdots \\ X_{t-p} \end{bmatrix} \right)^T$

Linear prediction coefficients (cont)

Note that we have

$$\gamma_p^+ = \begin{bmatrix} \gamma(1) \\ \vdots \\ \gamma(p) \end{bmatrix} \text{ and } \Gamma_p^+ = \begin{bmatrix} \gamma(0) & \gamma(-1) & \cdots & & \gamma(-p+1) \\ \gamma(1) & \gamma(0) & \gamma(-1) & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \\ \vdots & & & & \gamma(-1) \\ \gamma(p-1) & \gamma(p-2) & \cdots & \gamma(1) & \gamma(0) \end{bmatrix} .$$

Moreover the variance of the error is given by

$$\sigma^2(p) := \mathrm{Var}\left(X_t - \mathrm{proj}\left(X_t | \mathcal{H}_{t-1,p}^X\right)\right) = \gamma(0) - (\phi_p^+)^T \overline{\gamma_p^+} . \qquad (3)$$

Eq. (2) and (3) are called the Yule-Walker equations.

Innovation algorithm

Let $\epsilon^+_{1,0} = X_1$ and, for all $t \geq 2$,

$$\epsilon^+_{t,t-1} = X_t - \mathrm{proj}\left(X_t \middle| \mathrm{Span}\left(X_s,\ s = 1, \dots, t-1 \right) \right) \ .$$

Then $(\epsilon^+_{t,t-1})_{t \geq 1}$ is an orthogonal sequence such that

$$\|\epsilon^+_{t,t-1}\|^2 = \sigma^2_{t-1}$$

(which decreases with t) and

$$\mathrm{Span}\left(X_s,\ s = 1, \dots, t \right) = \mathrm{Span}\left(\epsilon^+_{s,s-1},\ s = 1, \dots, t \right) \ .$$

Innovation algorithm (cont.)

Denote the prediction coefficients in the innovation basis $\left(\epsilon^+_{k,k-1}\right)_{k=1,\dots,p}$ by $\boldsymbol{\theta}_p = (\theta_{k,p})_{k=1,\dots,p}$, that is,

$$\operatorname{proj}\left(X_{p+1} \middle| \operatorname{Span}\left(X_s,\ s=1,\dots,p\right)\right) = \sum_{k=1}^{p} \theta_{k,p}\epsilon^+_{k,k-1} \;.$$

Then one gets back to the prediction coefficients in the observation basis recursively by identifying

$$\sum_{k=1}^{p} \phi^+_{k,p} X_{p+1-k} = \sum_{k=1}^{p} \theta_{k,p} \left(X_k - \sum_{j=1}^{k-1} \phi^+_{j,k-1} X_{k-j} \right)$$
$$= \theta_{p,p} X_p + \sum_{k=2}^{p} \left(\theta_{p+1-k,p} - \sum_{j=1}^{k-1} \theta_{p+1-j,p} \phi^+_{k-j,p-j} \right) X_{p+1-k}$$

Algorithm 1: Innovation algorithm

Data: $\gamma(k,j)$, $1 \leq j \leq k \leq K+1$, $X_1, \ldots, X_{K+1}$

Result: $\epsilon^+_{1,0}, \ldots, \epsilon^+_{K+1,K}$, $\boldsymbol{\theta}_p$ and σ^2_p for $p = 1, \ldots, K$.

Initialization: set $\sigma^2_0 = \gamma(1,1)$ and $\epsilon^+_{1,0} = X_1$.

for $p = 1, \ldots, K$ **do**

  for $m = 1, \ldots, p$ **do**

    Set $\theta_{m,p} = \sigma^{-2}_{m-1} \left(\gamma(p+1,m) - \sum_{j=1}^{m-1} \overline{\theta_{j,m-1}}\, \theta_{j,p}\, \sigma^2_j \right)$

  end

  Set

$$\sigma^2_p = \gamma(p+1,p+1) - \sum_{m=1}^{p} |\theta_{m,p}|^2\, \sigma^2_{m-1}$$

$$\epsilon^+_{p+1,p} = X_{p+1} - \sum_{m=1}^{p} \theta_{m,p} \epsilon^+_{m,m-1} \, .$$

end

Innovation algorithm : numerical complexity

- $O(p^3)$ operations are needed to compute $\boldsymbol{\theta}_p$ and $\boldsymbol{\phi}_p^+$.
- If X is known to be an MA(q) process, then for all $p \geq q$, we have

$$\boldsymbol{\theta}_p = \begin{bmatrix} 0 & \dots & 0 & \theta_{p-q+1,p} & \dots & \theta_{p,p} \end{bmatrix}^T$$

  Hence, in this special case, the innovation algorithm can be performed in $O(p)$ operations.
- The innovation algorithm is also valid for a non-stationary L^2 sequence $(X_t)_{t\geq 1}$.
- Exploiting the stationarity to compute $\boldsymbol{\phi}_p^+$, Levinson's Algorithm can be performed in $O(p^2)$ operations.

Partial auto-correlation function

Recall that the sequence $\kappa := (\phi^+_{p,p})_{p\geq 1}$ is called the partial autocorrelation function of X.

Then we have $\kappa(1) = \dfrac{\gamma(1)}{\gamma(0)} = \rho(1) = \dfrac{\langle X_t, X_{t-1}\rangle}{\|X_t\|\,\|X_{t-1}\|}$.

If $p \geq 2$, this formula can be extended as follows.

Denote the forward and backward linear prediction errors by

$$\epsilon^+_{t,p} = X_t - \mathrm{proj}\left(X_t \middle| \mathcal{H}^X_{t-1,p}\right) \quad \text{and} \quad \epsilon^-_{t,p} = X_t - \mathrm{proj}\left(X_t \middle| \mathcal{H}^X_{t+p,p}\right)$$

Then we have

$$\kappa(p) = \frac{\left\langle \epsilon^+_{t,p-1}, \epsilon^-_{t-p,p-1}\right\rangle}{\|\epsilon^+_{t,p-1}\|\,\|\epsilon^-_{t-p,p-1}\|} = \frac{\mathrm{Cov}\left(\epsilon^+_{t,p-1}, \epsilon^-_{t-p,p-1}\right)}{\sigma^2_{p-1}} .$$

$$X_{t-p} \quad \underbrace{X_{t-p+1} \quad \ldots \quad X_{t-1}}_{\mathcal{H}^X_{t-1,p-1}} \quad X_t$$

Algorithm 2: Levinson-Durbin algorithm.

Data: $\gamma(k)$, $k = 0, \ldots, K$

Result: $\{\phi^+_{m,p}\}_{1 \leq m \leq p, 1 \leq p \leq K}$, $\kappa(1), \ldots, \kappa(K)$

Initialization: set $\kappa(1) = \phi^+_{1,1} = \gamma(1)/\gamma(0)$ and $\sigma_1^2 = \gamma(0)(1 - \kappa(1)^2)$.

for $p = 1, 2, \ldots, K-1$ **do**

Set

$$\kappa(p+1) = \sigma_p^{-2} \left(\gamma(p+1) - \sum_{k=1}^{p} \phi^+_{k,p} \gamma(p+1-k) \right)$$

$$\sigma^2_{p+1} = \sigma^2_p (1 - \kappa(p+1)^2)$$

$$\phi^+_{p+1,p+1} = \kappa(p+1)$$

for $m \in \{1, \cdots, p\}$ **do**

Set

$$\phi^+_{m,p+1} = \phi^+_{m,p} - \kappa(p+1) \, \overline{\phi^+_{p+1-m,p}} \; .$$

end

end

1 Linear prediction

2 An illustrative example with R

```
################################################
#        Linear Predictive Coding              #
################################################
quantize <- function(x,corder=3){
# code input x with quantiz. levels given by corder
  if (length(corder)>1)  # corder are quant. levels
    {
      ql <- corder
    } else # set quant. levels from normal quantiles
    {
      ql <- qnorm(seq(from=0,to=1,by=1/corder),
                  mean = mean(x), sd = sqrt(3*var(x)))
    }
  xc <- NULL
  for (t in 1:(length(x))){
    xc= c(xc,ql[which.min(abs(ql-x[t]))])}
  return(list(xc=xc,ql=ql))
}
lpcoding <- function(x,corder=3,rorder=10){
# estimation of ar parameters
  ac <- acf(x,type=c('covariance'),plot= FALSE)
  arc <- acf2AR(ac$acf[1:(rorder+1)])
  arc <- arc[nrow(arc),]
  # residuals computation
  res <- NULL
  xinitz <- c(rep(0,rorder),x)
  for (t in ((rorder+1):length(xinitz))){
    res <- c(res,xinitz[t]-
             t(as.vector(xinitz[(t-1):(t-rorder)]))
             %*% as.vector(arc))
  }
  # coded residuals
  resc <- quantize(res, corder=corder)
  # reconstructed time series from coded residuals
  return(list(xc=filter(resc$xc,arc,method='recursive'),
              ar=arc))
```

```
}
lpcodingblocks <- function(x,bt=0.02,freq,
                           corder=3,rorder=10){
  bl <- floor(bt*freq) # block length
  rorder <- min(c(rorder,floor(bl/5)))
  xc <- NULL
  for (k in 1:floor(length(x)/bl)){
    xc <- c(xc,lpcoding(x[((k-1)*bl+1):(k*bl)],
                        corder=corder,rorder=rorder)$xc)
  }
  return(xc)
}
# get the original speech audio sample
require(audio)
X <- load.wave('/home/roueff/data/dataset/audio/speech/3meninaboat.wav')
fr <- X$rate
subsamp <- 2**3
extract <- ts(X[seq(from=1,to=length(X),by=subsamp)]/max(abs(X)),
              frequency=fr/subsamp)
ts.plot(extract)

# AR Coding and direct coding
extractc <- ts(lpcodingblocks(extract,
                              freq=frequency(extract),
                              corder=5,rorder=20),
               start=start(extract),
               frequency=frequency(extract))

extractbadc <- ts(quantize(extract,corder=5)$xc,
                  start=start(extract),
                  frequency=frequency(extract))

# plot coded signals
lines(extractc,col=2)
lines(extractbadc,col=3)
# plot within a time window
```

```
for (term in c('','c','badc')){
  eval(parse(text=paste('we',term,
               ' <- window(extract',
               term,',start=3.5,end=4)',
               sep='')))
}
ts.plot(we)
lines(wec,col=2)
lines(webadc,col=3)
# save wav files
require(audio)
save.wave(audioSample(extract,
                      rate=frequency(extract)),
          '/tmp/3.wav')
save.wave(audioSample(extractc,
                      rate=frequency(extract)),
          '/tmp/3c.wav')
save.wave(audioSample(extractbadc,
                      rate=frequency(extract)),
          '/tmp/3badc.wav')
```