10/03 - The Maximum Likelihood approach

Identification of ARMA / ARMAX models

For an ARMA / ARMAX model we consider a model class defined as:

$$M_\theta : \quad y(t) = \frac{B(z)}{A(z)} u(t-d) + \frac{C(z)}{A(z)} e(t) \qquad e(t) \sim WN(0, \lambda^2)$$

where (written in canonical form):

- $A(z) = 1 - a_1 z^{-1} - \dots - a_m z^{-m}$
- $B(z) = b_0 + b_1 z^{-1} + \dots + b_p z^{-p+1}$
- $C(z) = 1 + c_1 z^{-1} + \dots + c_n z^{-n}$

and the set of parameter is $\theta = [a_1 \dots a_m \; b_0 \dots b_{p-1} \; c_1 \dots c_n]^T$ and the overall size of θ is $n_\theta = m + p + n$

So

$$J_N(\theta) = \frac{1}{N} \sum (y(t) - \hat{y}(t|t-1, \theta))^2$$

The predictor, in ARMAX case is $\hat{y}(t+k|t, \theta) = \frac{B(z)}{C(z)} E(z) u(t+k-d) + \frac{F(z)}{C(z)} y(t+k)$. If we want the 1-step ahead, using the long division, we find that (for $k=1$) we perform only 1 step of long division, and

$$E(z) = 1 \qquad F(z) z^{-k} = C(z) - A(z)$$

$C(z)$	$A(z)$
$-A(z)$	$1 \quad E(z)$
$F(z)z^{-1} \quad C(z) - A(z)$	

The final predictor for the ARMA / ARMAX model will be

$$\hat{y}(t|t-1) = \frac{C(z) - A(z)}{C(z)} y(t) + \frac{B(z)}{C(z)} u(t-d)$$

prediction error: $\varepsilon(t|t-1, \theta) = y(t) - \hat{y}(t|t-1, \theta) = y(t) - \frac{C(z)-A(z)}{C(z)} y(t) - \frac{B(z)}{C(z)} u(t-d)$

$$= \left(1 - \frac{C(z)-A(z)}{C(z)}\right) y(t) - \frac{B(z)}{C(z)} u(t-d)$$

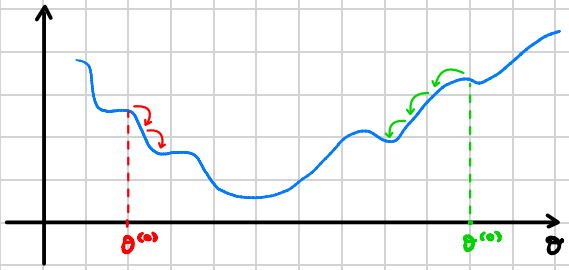
$$= \frac{A(z)}{C(z)} y(t) - \frac{B(z)}{C(z)} u(t-d)$$

θ appears in the denominator of $\varepsilon(t|t-1, \theta)$

Now $\hat{y}(t|t-1, \theta) \neq \varphi^T(t)\theta$ non-linear optimization problem

We need an iterative numerical approach

- we initialize the algorithm with an initial guess $\theta^{(0)}$
- we define an update rule: $\theta^{(i+1)} = f(\theta^{(i)})$. The sequence of the estimates should converge to $\hat{\theta}_N = \arg\min_\theta J_N(\theta)$



We have two problems:

1) Update rule
2) multiple minima $\Rightarrow$ many local optimum

Problem of local optimal

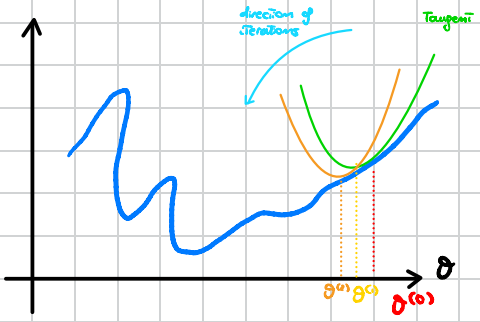
Iterative algorithms are guaranteed to converge to a local minimum, not to the global one. We will follow an heuristic approach

iterations

$$\theta^{(0,1)} \to \theta^{(1,1)} \to \theta^{(N,1)} \Rightarrow J_N(\theta^{(N,1)})$$
$$\theta^{(0,2)} \to \theta^{(1,2)} \to \theta^{(N,2)} \Rightarrow J_N(\theta^{(N,2)})$$
$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$
$$\theta^{(0,M)} \to \theta^{(1,M)} \to \theta^{(N,M)} \Rightarrow J_N(\theta^{(N,M)})$$

We will take the one leading the lowest cost

How to update $\vartheta^{(i)}$? We can use the the Newton parabola method. Mathematically we define the Tangent paraboloid as (2^o order Taylor expansion)



$$V^{(i)}(\vartheta) = J_N(\vartheta^{(i)}) + \frac{\partial}{\partial \vartheta^{(i)}} J_N(\vartheta)\Big|_{\vartheta=\vartheta^{(i)}} (\vartheta-\vartheta^{(i)}) + \frac{1}{2}(\vartheta-\vartheta^{(i)})^T \frac{\partial^2}{\partial \vartheta^{(i)2}} J_N(\vartheta)\Big|_{\vartheta=\vartheta^{(i)}} (\vartheta-\vartheta^{(i)})$$

The idea is to take $\vartheta^{(i+1)}$ as the minimizer of $V^{(i)}(\vartheta)$. So we have to find that

$$\frac{\partial}{\partial \vartheta} V^{(i)}(\vartheta^{(i+1)})\Big|_{\vartheta=\vartheta^{(i)}} = 0$$

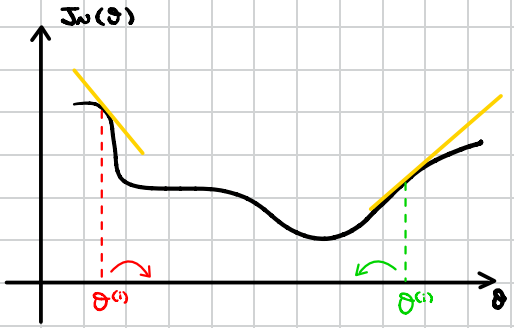
with:

$$\frac{\partial}{\partial \vartheta} V^{(i)}(\vartheta^{(i+1)})\Big|_{\vartheta=\vartheta^{(i)}} = \frac{\partial}{\partial \vartheta} J_N(\vartheta)\Big|_{\vartheta=\vartheta^{(i)}} + \frac{\partial^2}{\partial \vartheta^2} J_N(\vartheta)\Big|_{\vartheta=\vartheta^{(i)}} (\vartheta^{(i+1)}-\vartheta^{(i)})$$

and we obtain the condition for $\vartheta^{(i+1)}$: $\frac{\partial}{\partial \vartheta} J_N(\vartheta)\Big|_{\vartheta=\vartheta^{(i)}} + \frac{\partial^2}{\partial \vartheta^2} J_N(\vartheta)\Big|_{\vartheta=\vartheta^{(i)}} (\vartheta^{(i+1)}-\vartheta^{(i)}) = 0$

So the

$$\vartheta^{(i+1)} = \vartheta^{(i)} - \underbrace{\left[\frac{\partial^2}{\partial \vartheta^2} J_N(\vartheta)\Big|_{\vartheta=\vartheta^{(i)}}\right]^{-1}}_{\text{inverse of the Hessian matrix in } \vartheta^{(i)}} \underbrace{\frac{\partial}{\partial \vartheta} J_N(\vartheta)\Big|_{\vartheta=\vartheta^{(i)}}}_{\text{gradient of } \vartheta^{(i)}}$$



Looking at the gradient, we have that:
- if it's positive $\Rightarrow$ $\vartheta^{(i+1)} < \vartheta^{(i)}$
- if it's negative $\Rightarrow$ $\vartheta^{(i+1)} \geq \vartheta^{(i)}$

We need an explicit form related to our signal

Computation of the gradient and the Hessian Matrix

Remembering that $J_N(\vartheta) = \frac{1}{N}\sum \varepsilon(t|t-1,\vartheta)^2$ we derive it

gradient: $$\frac{\partial}{\partial \vartheta} J_N(\vartheta) = \frac{d}{d\vartheta}\left[\frac{1}{N}\sum \varepsilon(t|t-1,\vartheta)^2\right] = \frac{1}{N}\sum \frac{d}{d\vartheta}\varepsilon(t|t-1,\vartheta)^2 = \frac{2}{N}\sum \varepsilon(t|t-1,\vartheta)\frac{\partial}{\partial \vartheta}\varepsilon(t|t-1,\vartheta)$$

(ε: scalar; $\frac{\partial}{\partial \vartheta}\varepsilon$: vector)

Hessian: $$\frac{\partial^2}{\partial \vartheta^2} J_N(\vartheta) = \frac{d}{d\vartheta}\left[\frac{\partial}{\partial \vartheta} J_N(\vartheta)\right] = \frac{d}{d\vartheta}\left[\frac{2}{N}\sum \varepsilon(t|t-1,\vartheta)\frac{\partial}{\partial \vartheta}\varepsilon(t|t-1,\vartheta)\right]$$

$$= \frac{2}{N}\sum \frac{\partial}{\partial \vartheta}\varepsilon(t|t-1,\vartheta)\left(\frac{\partial}{\partial \vartheta}\varepsilon(t|t-1,\vartheta)\right)^T + \frac{2}{N}\sum \varepsilon(t|t-1,\vartheta)\frac{\partial^2}{\partial \vartheta^2}\varepsilon(t|t-1,\vartheta)$$

($\left(\frac{\partial}{\partial \vartheta}\varepsilon\right)^T$: square ≥ 0 (always); remove this because could be negative)

We remove the second term to be sure to have a paraboloid with a minimum (and not a maximum).

Newton's law (update rule for ϑ)

$$\vartheta^{(i+1)} = \vartheta^{(i)} - \left[\sum \frac{\partial}{\partial \vartheta}\varepsilon(t|t-1,\vartheta)\Big|_{\vartheta^{(i)}} \frac{\partial}{\partial \vartheta}\varepsilon(t|t-1,\vartheta)\Big|^T_{\vartheta^{(i)}}\right]^{-1} \sum \varepsilon(t|t-1,\vartheta^{(i)})\frac{\partial}{\partial \vartheta}\varepsilon(t|t-1,\vartheta)\Big|_{\vartheta^{(i)}}$$

we need:
- $\varepsilon(t|t-1,\vartheta^{(i)}) = y(t) - \hat{y}(t|t-1,\vartheta^{(i)})$ — no need to further job, everything is in the data
- $\frac{\partial}{\partial \vartheta}\varepsilon(t|t-1,\vartheta)\Big|_{\vartheta^{(i)}} = ?$

Computation of $\frac{\partial}{\partial\vartheta}\varepsilon(t|t-1,\vartheta)$

Being $\vartheta = [a_1 \dots a_m\ b_0 \dots b_{p-1}\ c_1 \dots c_n]^T$ the set of model parameters, we compute the prediction error writing it in the canonical form

$$\varepsilon(t|t-1,\vartheta) = \frac{A(z)}{C(z)}y(t) - \frac{B(z)}{C(z)}u(t-d) = \frac{1-a_1z^{-1}-\dots-a_mz^{-m}}{1+c_1z^{-1}+\dots+c_nz^{-n}}y(t) - \frac{b_0+b_1z^{-1}+\dots+b_{p-1}z^{-p+1}}{1+c_1z^{-1}+\dots+c_nz^{-n}}u(t-d)$$

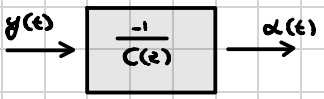
So, when we derive $\varepsilon(t|t-1,\vartheta)$ by ϑ we obtain the vector

$$\frac{\partial}{\partial\vartheta}\varepsilon(t|t-1,\vartheta) = \left[\frac{\partial}{\partial a_1}\varepsilon(t|t-1,\vartheta) \dots \frac{\partial}{\partial b_0}\varepsilon(t|t-1,\vartheta) \dots \frac{\partial}{\partial c_n}\varepsilon(t|t-1,\vartheta) \dots\right]^T$$

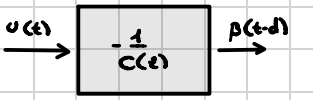
That being $\vartheta = [a_1 \dots a_m\ b_0 \dots b_{p-1}\ c_1 \dots c_n]^T$ we derive for each element for ϑ: for example

- $\frac{\partial}{\partial a_1}\varepsilon(t|t-1,\vartheta) = \frac{\partial}{\partial a_1}\left[\frac{1-a_1z^{-1}-\dots-a_mz^{-m}}{C(z)}y(t) - \frac{B(z)}{C(z)}u(t-d)\right]$

  $= \frac{\partial}{\partial a_1}\left[\frac{-a_1z^{-1}}{C(z)}y(t)\right] = -\frac{z^{-1}}{C(z)}y(t) \triangleq \alpha(t-1)$

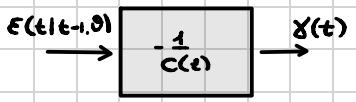


- $\frac{\partial}{\partial a_2}\varepsilon(t|t-1,\vartheta) = \frac{\partial}{\partial a_2}\left[\frac{-a_2z^{-2}}{C(z)}y(t)\right] = -\frac{z^{-2}}{C(z)}y(t) \triangleq \alpha(t-2)$



- $\frac{\partial}{\partial b_0}\varepsilon(t|t-1,\vartheta) = \frac{\partial}{\partial b_0}\left[\frac{A(z)}{C(z)}y(t) - \frac{b_0+b_1z^{-1}+\dots+b_{p-1}z^{-p+1}}{C(z)}u(t-d)\right]$

  $= \frac{\partial}{\partial b_0}\left[-\frac{b_0}{C(z)}u(t-d)\right] = -\frac{1}{C(z)}u(t-d) \triangleq \beta(t-d)$



- $\frac{\partial}{\partial c_i}\varepsilon(t|t-1,\vartheta) = \frac{\partial}{\partial c_i}\left[\frac{A(z)}{1+c_1z^{-1}+\dots+c_nz^{-n}}y(t) - \frac{B(z)}{1+c_1z^{-1}+\dots+c_nz^{-n}}u(t-d)\right]$?

Computing $\partial/\partial c_i\ \varepsilon(t|t-1,\vartheta)$ is not straightforward since $C(z)$ is at the denominator. So we have to rewrite $\varepsilon(t|t-1,\vartheta)$ as

$$C(z)\varepsilon(t|t-1,\vartheta) = A(z)y(t) - B(z)u(t-d)$$

and now derive it by c_i:

$$\frac{\partial}{\partial c_i}[C(z)\varepsilon(t|t-1,\vartheta)] = \frac{\partial}{\partial c_i}[A(z)y(t) - B(z)u(t-d)] = 0$$

$$= \frac{\partial}{\partial c_i}C(z)\varepsilon(t|t-1,\vartheta) + C(z)\frac{\partial}{\partial c_i}\varepsilon(t|t-1,\vartheta)$$

So the final derivation would be

$$\frac{\partial}{\partial c_i}\varepsilon(t|t-1,\vartheta) = -\frac{1}{C(z)}\frac{\partial}{\partial c_i}C(z)\cdot\varepsilon(t|t-1,\vartheta) = -\frac{1}{C(z)}z^{-i}\varepsilon(t|t-1,\vartheta) \triangleq \gamma(t-1)$$

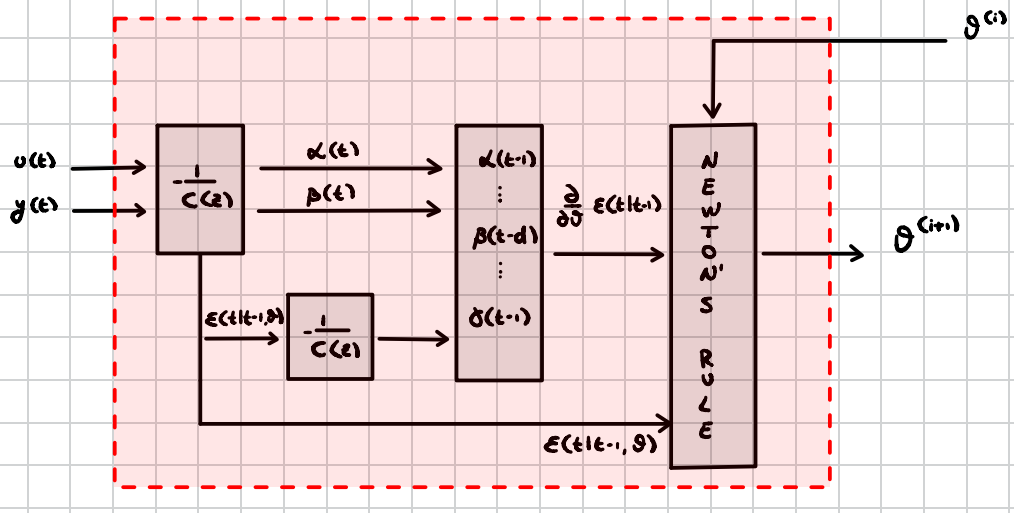
where $\frac{\partial}{\partial c_i}[1+c_1z^{-1}+\dots+c_nz^{-n}]$

Final expression

$$\frac{\partial}{\partial\vartheta}\varepsilon(t|t-1,\vartheta) = [\alpha(t-1) \dots \alpha(t-m)\ \beta(t-d) \dots \beta(t-d-p+1)\ \gamma(t-1) \dots \gamma(t-n)]$$

where

$$\alpha = -1/C(z)\cdot y(t) \qquad \beta = -1/C(z)\cdot u(t-d) \qquad \gamma = -1/C(z)\cdot z^{-1}\varepsilon(t|t-1,\vartheta)$$



Processing scheme for the Newton's update law

Remark: the Newton's rule is one of the possible way to solve this nonlinear optimization problem that takes advantage of the computation of the gradient and the second derivative of the cost.

- Newton's Rule $\theta^{(i+1)} = \theta^{(i)} - [\text{Hessian}]^{-1} \cdot \text{gradient}$
- gradient descent: $\theta^{(i+1)} = \theta^{(i)} - \eta \cdot \text{gradient}$ η scalar and fixed (step size)
- quasi-Newton's rule: $\theta^{(i+1)} = \theta^{(i)} - [\text{approximate Hessian}]^{-1} \text{gradient}$

↳ like when we have removed the last piece from the Hessian; this is what we do! computationally lighter than Newton's, still more accurate but slower than GD