



Análisis de Series de Tiempo

Carrera de Especialización en Inteligencia Artificial

Temas

- Estimación del orden del modelo
- Estimación de parámetros
- Ejemplos de función de autocorrelación y autocorrelación parcial
- Objeto de resultados ARIMA
- Diagnósticos
- Revisión de la primera entrega

Estimación del orden

¿Cómo estimar el orden del modelo?

Usando la función de autocorrelación y la función de autocorrelación parcial.

Si $k > p$, la **autocorrelación parcial** es nula

Si $k > q$ la autocorrelación es nula

Función de autocorrelación parcial: define el efecto que tiene $Y(t-k)$ sobre $Y(t)$

$$\alpha(1) = \text{corr}(Y_{t+1}, Y_t), \text{ for } k = 1$$

$$\alpha(k) = \text{corr}((Y_{t+k} - P_{t,k}(Y_{t+k}), Y_t - P_{t,k}(Y_t)), \text{ para } k \geq 2$$

donde $P_{t,k}(x)$ es el operador de proyección ortogonal de x sobre el espacio generado por $Y_{t+1}, \dots, Y_{t+k-1}$

¿Cómo estimar el orden del modelo?

Función de autocorrelación parcial: define el efecto que tiene $Y(t-k)$ sobre $Y(t)$

$$\alpha(1) = \text{corr}(Y_{t+1}, Y_t), \text{ for } k = 1$$

$$\alpha(k) = \text{corr}((Y_{t+k} - P_{t,k}(Y_{t+k}), Y_t - P_{t,k}(Y_t)), \text{ para } k \geq 2$$

donde $P_{t,k}(x)$ es el operador de proyección ortogonal de x sobre el espacio generado por $Y_{t+1}, \dots, Y_{t+k-1}$

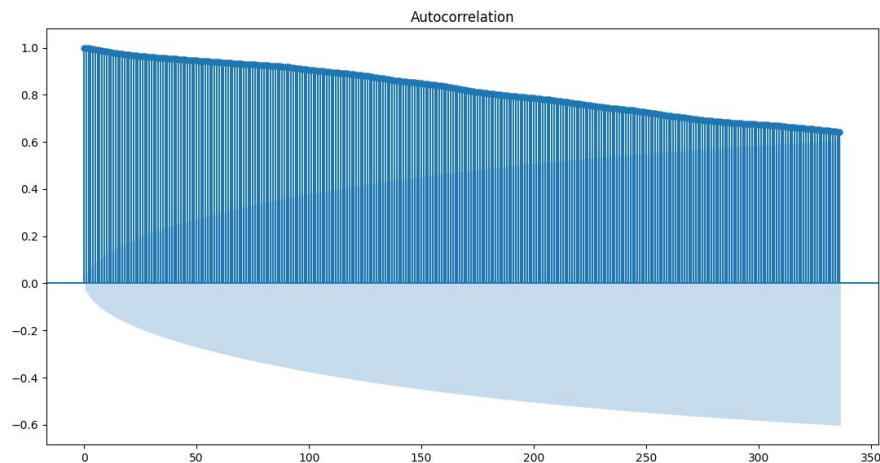
Podemos estimar el orden del modelo usando la función de autocorrelación y la función de autocorrelación parcial.

- Si $k > p$, la autocorrelación parcial es nula
- Si $k > q$ la autocorrelación es nula

¿Cómo estimar el orden del modelo?

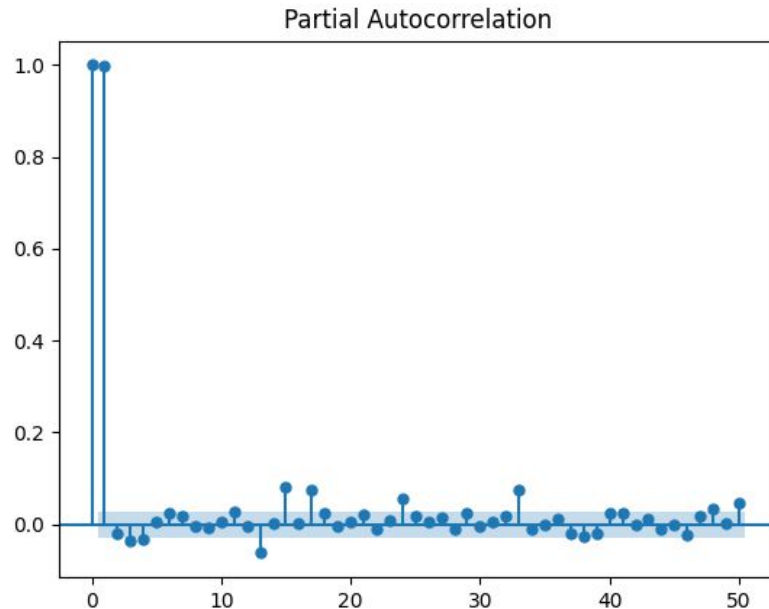
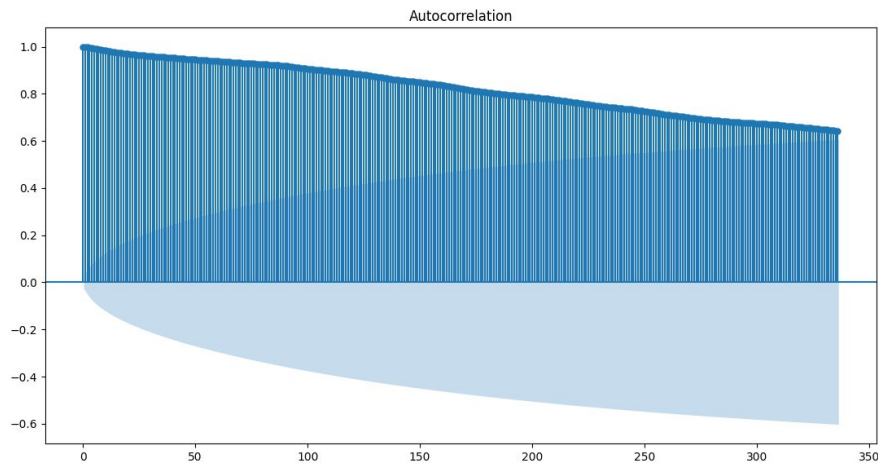
Model	PACF
White noise	The partial autocorrelation is 0 for all lags.
Autoregressive model	The partial autocorrelation for an $AR(p)$ model is nonzero for lags less than or equal to p and 0 for lags greater than p .
Moving-average model	If $\phi_{1,1} > 0$, the partial autocorrelation <i>oscillates</i> to 0.
	If $\phi_{1,1} < 0$, the partial autocorrelation <i>geometrically</i> decays to 0.
Autoregressive–moving-average model	An $ARMA(p, q)$ model's partial autocorrelation geometrically decays to 0 but only after lags greater than p .

Función de autocorrelación



→ Indicio de que la serie es NO estacionaria

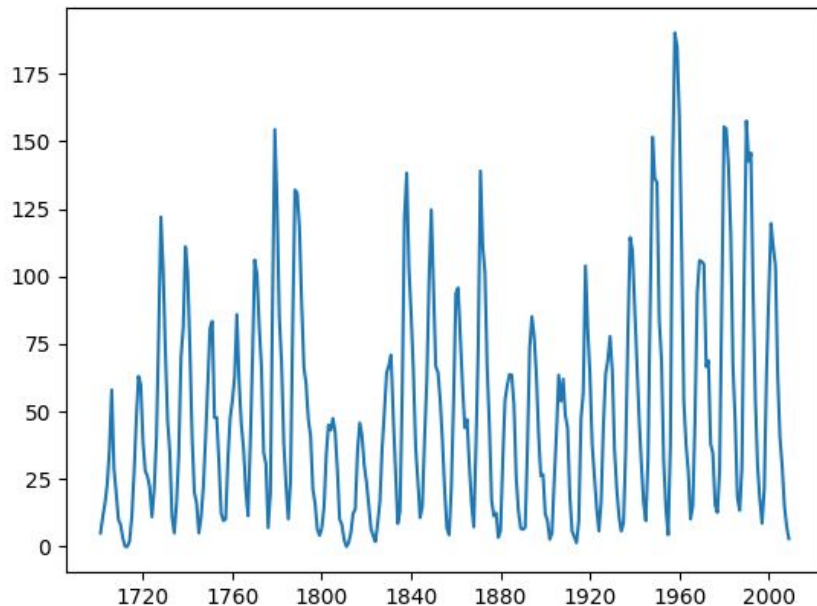
Función de autocorrelación parcial



Función de autocorrelación

Ejemplo con dataset público

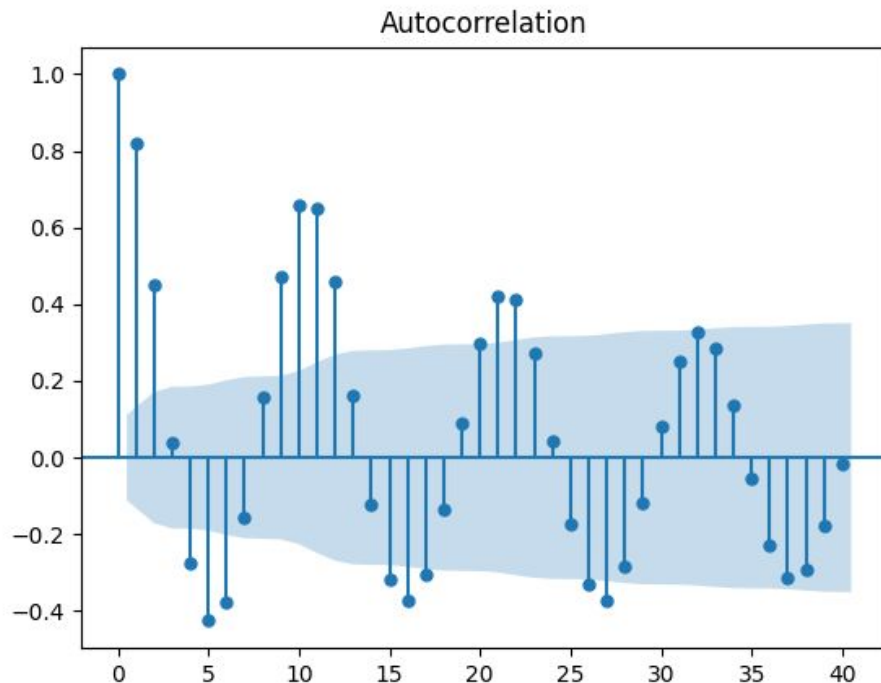
```
import statsmodels.api  
as sm  
  
dta =  
sm.datasets.sunspots.load_pandas().data
```



Función de autocorrelación

Ejemplo con dataset público

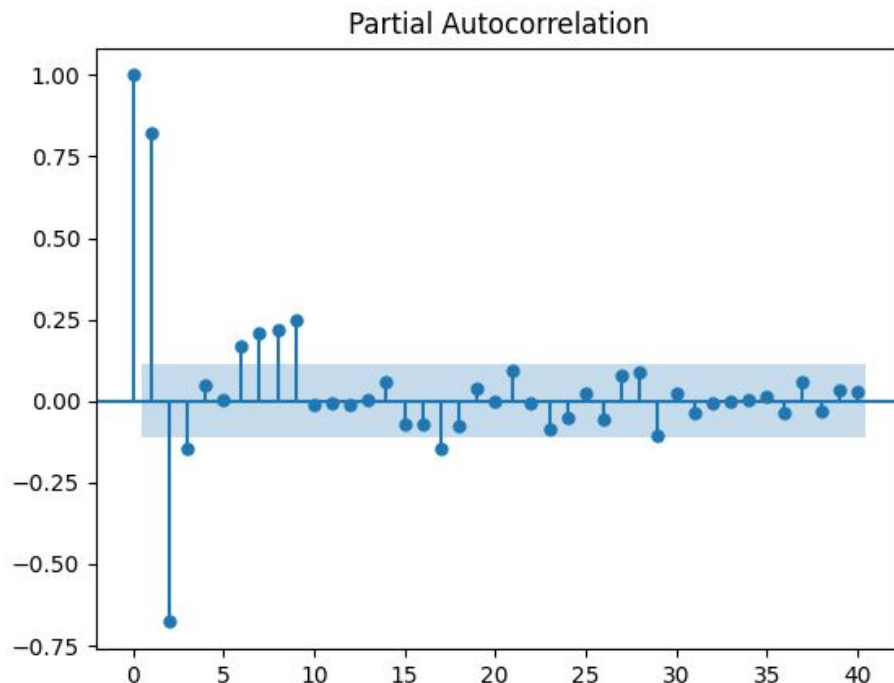
```
sm.graphics.tsa.plot_acf  
(dta.values.squeeze(),  
lags=40)
```



Función de autocorrelación parcial

Ejemplo con dataset público

```
sm.graphics.tsa.plot_pacf  
(dta.values.squeeze(),  
lags=40, method="ywm")
```



Sobre la estimación del orden de integración

Al estimar el orden d en el modelo ARIMA(p, d, q) se debe tener cuidado de no diferenciar de más.

Si bien la diferencia de cualquier serie estacionaria sigue siendo estacionaria, si diferenciamos más de lo necesario estamos introduciendo correlaciones innecesarias en los datos, complicando el modelado.

Además sobrediferenciar lleva a modelos no invertibles, los cuales traen problemas a la hora de estimar los parámetros del modelo

Ejemplo en Modelos estacionales

Modelo SARIMA multiplicativo

Una herramienta importante es el análisis de procesos estacionales **no estacionarios** es la diferenciación estacional de período s para la serie $\{Y_t\}$, denotada

$$\nabla_s Y_t = Y_t - Y_{t-s}$$

Se dice que una serie estacional no estacionaria sigue un modelo SARIMA(p,d,q)x(P,D,Q) $_s$ de período s si la serie diferenciada

$$W_t = \nabla^d \nabla_s^D Y_t$$

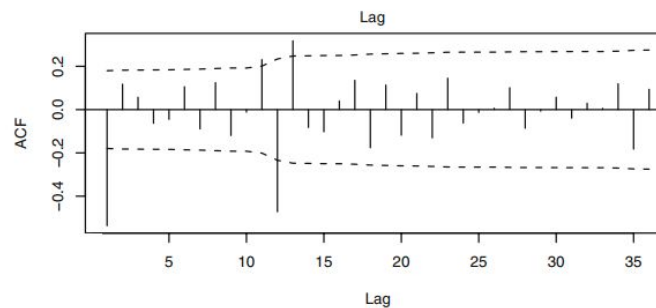
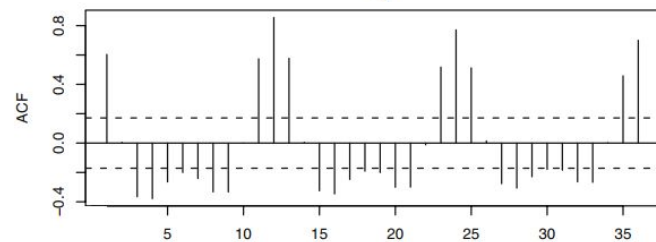
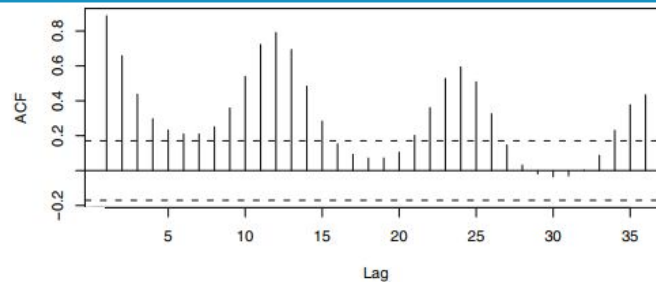
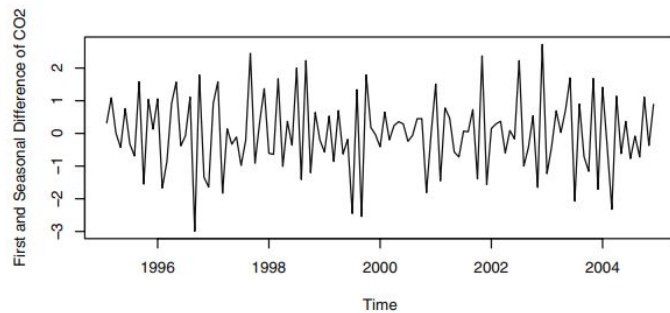
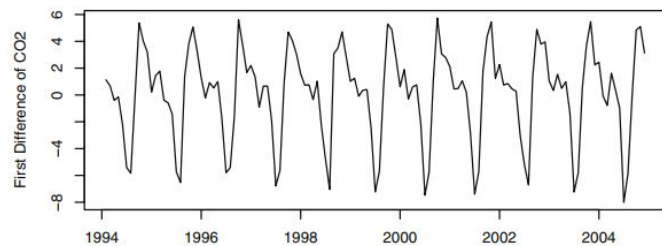
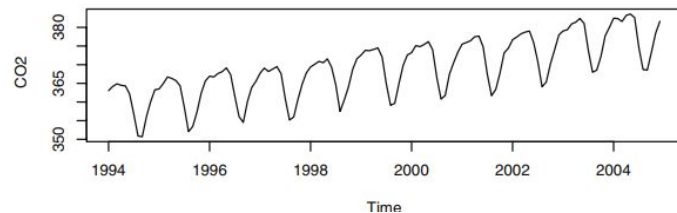
sigue un proceso SARMA(p,q)x(P,Q) $_s$.

Especificación del modelo

Se usan las mismas ideas introducidas para modelos ARMA. y ARIMA.

- 1) Inspeccionar la serie de tiempo y su función de autocorrelación muestral.
¿Observo alguna tendencia y/o algún comportamiento estacional?
- 2) Proponer transformaciones (ej. diferenciar) y volver a analizar la serie resultante

Especificación del modelo - ejemplo



Estimación de parámetros

¿Cómo estimar los parámetros de un modelo?

En general no conocemos los valores de a_1, \dots, a_p y b_1, \dots, b_q y debemos estimarlos a partir de las observaciones de la serie de tiempo.

Existen distintos enfoques:

1. Usando las ecuaciones de Yule-Walker - Sólo sirve para modelos **AR**
2. Cuadrados mínimos (no lo vemos)
3. Método de máxima verosimilitud basado en el modelo de estados

Estimación del modelo AR mediante las ecs. de Y-W

Usando la expresión de las ecs. de Y-W, podemos reemplazar los valores de R_i por \hat{R}_i y resolver el sistema de ecuaciones para a_1, \dots, a_p :

$$\begin{cases} \hat{R}_1 = a_1 + a_2 \hat{R}_1 + \dots + a_p \hat{R}_{p-1} \\ \vdots \\ \hat{R}_p = a_1 \hat{R}_{p-1} + a_2 \hat{R}_{p-2} + \dots + a_p \hat{R}_p \end{cases}$$

Se puede demostrar que los estimadores de Y-W minimizan el ECM.

Una forma eficiente de hacer este cálculo es mediante el algoritmo de Levinson:

1. Set $\hat{\sigma}_0^2 = \hat{C}_0$ and $\text{AIC}_0 = N(\log 2\pi \hat{\sigma}_0^2 + 1) + 2$
2. For $m = 1, \dots, M$, repeat the following steps

$$(a) \hat{a}_m^m = \left(\hat{C}_m - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} \hat{C}_{m-j} \right) (\hat{\sigma}_{m-1}^2)^{-1},$$

$$(b) \hat{a}_i^m = \hat{a}_i^{m-1} - \hat{a}_m^m \hat{a}_{m-i}^{m-1} \text{ for } i = 1, \dots, m-1,$$

$$(c) \hat{\sigma}_m^2 = \hat{\sigma}_{m-1}^2 \{1 - (\hat{a}_m^m)^2\},$$

$$(d) \text{AIC}_m = N(\log 2\pi \hat{\sigma}_m^2 + 1) + 2(m+1). \quad 19$$

Estimación del modelo AR por MV

Bajo la suposición de que el ruido $e_i \sim \mathcal{N}(0, \sigma_e^2)$, la serie de tiempos

$$\{Y_1, \dots, Y_n\} \sim \mathcal{N}(0, \Sigma), \quad \Sigma = \begin{bmatrix} C_0 & C_1 & \dots & C_{n-1} \\ C_1 & C_0 & \dots & C_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n-1} & C_{n-2} & \dots & C_0 \end{bmatrix}$$

Hallar el EMV puede ser costoso computacionalmente pues hay que calcular Σ^{-1}

Una alternativa es descomponer $L(\theta) = f_{Y_1, \dots, Y_n}(y_1, \dots, y_n; \theta) = \prod_{i=1}^n f(y_i | y_1, \dots, y_{i-1}; \theta)$,

pero ¿cómo calculamos de forma sencilla cada uno de los términos?

Modelo de estados

Un modelo de estados general está dado por

vector de estados $\rightarrow \boxed{x_t} = F_t x_{t-1} + G_t \boxed{e_t} \leftarrow$ ruido blanco del sistema

vector de observaciones $\rightarrow \boxed{y_t} = H_t x_t + \boxed{w_t} \leftarrow$ ruido blanco de medición

$$e_t \sim \mathcal{N}(0, Q_t)$$

$$w_t \sim \mathcal{N}(0, R_t)$$

Modelo AR como modelo de estados

Si definimos $x_t = (y_t, y_{t-1}, \dots, y_{t-p+1})^T$, nos queda que

$$F = \begin{bmatrix} a_1 & a_2 & \dots & a_p \\ 1 & 0 & \dots & 0 \\ & \ddots & & \\ & & 1 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Además, considerando $H = [1 \ 0 \ \dots \ 0]$, recuperamos $y_t = Hx_t$. Fijando $Q = \sigma^2$ y $R = 0$ obtenemos el modelo de estados del modelo AR.

Modelo ARMA como modelo de estados

Para el modelo ARMA vamos a definir $x_t = [y_t, y_{t-1}, \dots, y_{t-p+1}]^T$ al igual que en el modelo AR. Las matrices F y H también van a ser las mismas:

$$F = \begin{bmatrix} a_1 & a_2 & \dots & a_p \\ 1 & 0 & \dots & 0 \\ & \ddots & & \\ & & 1 & 0 \end{bmatrix} \quad H = [1 \ 0 \ \dots \ 0]$$

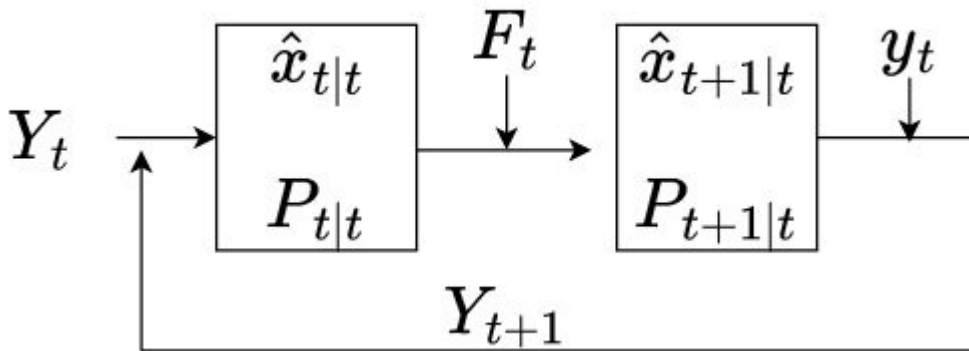
Lo que va a cambiar en el modelo ARMA es la matriz G, que debe incluir la dependencia con ruidos anteriores:

$$G = \begin{bmatrix} 1 & b_1 & \dots & b_q \\ 0 & 0 & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

Estimación de parámetros con un modelo de estados

Vamos a buscar predecir el vector de estados a partir de las observaciones anteriores, ya que nos va a facilitar para descomponer la verosimilitud.

$$\hat{x}_{t+1|t} = \mathbb{E}[x_{t+1} | \underbrace{y_t, y_{t-1}, \dots, y_1}_{Y_t}] = \mathbb{E}[x_{t+1} | Y_t]$$



Ecuaciones del filtro de Kalman

$$\hat{x}_{t|t-1} = F_{t-1} \hat{x}_{t-1|t-1}$$

$$P_{t|t-1} = F_{t-1} P_{t-1|t-1} A_{t-1}^T + B_{t-1} Q_{t-1} B_{t-1}^T$$

$$K_t = P_{t|t-1} H_t^T (R_t + H_t P_{t|t-1} H_t^T)^{-1}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t (y_t - H_t \hat{x}_{t|t-1})$$

$$P_{t|t} = (I - K_t H_t) P_{t|t-1}$$

Donde $\hat{x}_{t|t-1} = \mathbb{E}[x_t | Y_{t-1}]$, $\hat{x}_{t|t} = \mathbb{E}[x_t | Y_t]$, $P_{t|t-1} = \mathbb{E}[(x_t - \hat{x}_{t|t-1})(x_t - \hat{x}_{t|t-1})^T]$ y

$$P_{t|t} = \mathbb{E}[(x_t - \hat{x}_{t|t})(x_t - \hat{x}_{t|t})^T]$$

¿Qué tiene que ver todo esto con MV?

Recordemos que el objetivo es descomponer la verosimilitud como:

$$\begin{aligned} L(\theta) &= f(y_t | y_{t-1}, \dots, y_1; \theta) f(y_{t-1} | y_{t-2}, \dots, y_1; \theta) \dots f(y_2 | y_1; \theta) f(y_1; \theta) \\ &= f(y_t | Y_{t-1}; \theta) f(y_{t-1} | Y_{t-2}; \theta) \dots f(y_2 | y_1; \theta) f(y_1; \theta) \end{aligned}$$

O lo que es lo mismo $\log(L(\theta)) = \sum_{i=1}^n \log(f(y_i | Y_{i-1}; \theta))$

El filtro de Kalman me devuelve justamente estas condicionales

Modelos de estados y MV

Además, dado que el ruido $e_t \sim \mathcal{N}(0, Q_t)$ las distribuciones condicionales son de la forma

$$y_t | Y_{t-1} \sim \mathcal{N}(y_{t|t-1}, d_{t|t-1})$$

donde

$$y_{t|t-1} = H_t \hat{x}_{t|t-1} \text{ y } d_{t|t-1} = H_t P_{t|t-1} H_t^T + Q_t$$

Luego

$$\log(L(\theta)) \propto \sum_{i=1}^n d_{t|t-1} + \sum_{i=1}^n (y_t - y_{t|t-1})^T d_{t|t-1}^{-1} (y_t - y_{t|t-1})$$

Cómo usamos el KF + MV para estimar los parámetros

Básicamente los pasos a seguir son:

1. Asignar un valor para θ
2. Dado θ , calcular las matrices del filtro de Kalman
3. Asignar los valores iniciales para $x_{0|0} = 0$, y $P_{0|0}$
4. Para $t=1, \dots, n$ evolucionar el FK y obtener $x_{t|t-1}$, y $P_{t|t-1}$
5. Con estos valores hallar la expresión de MV
6. Actualizar θ y volver a 1.

Bibliografía y material complementario

- [State-Space modelling](#)
- [Estimating State-Space models through Maximum Likelihood](#)
- “[Kalman Filtering](#)”, Charles K. Chui and Guanrong Chen, Springer
- “Optimal Filtering”, Brian D.O. Anderson and John B. Moore, Dover

Redes Neuronales para Series de Tiempo

Redes Neuronales

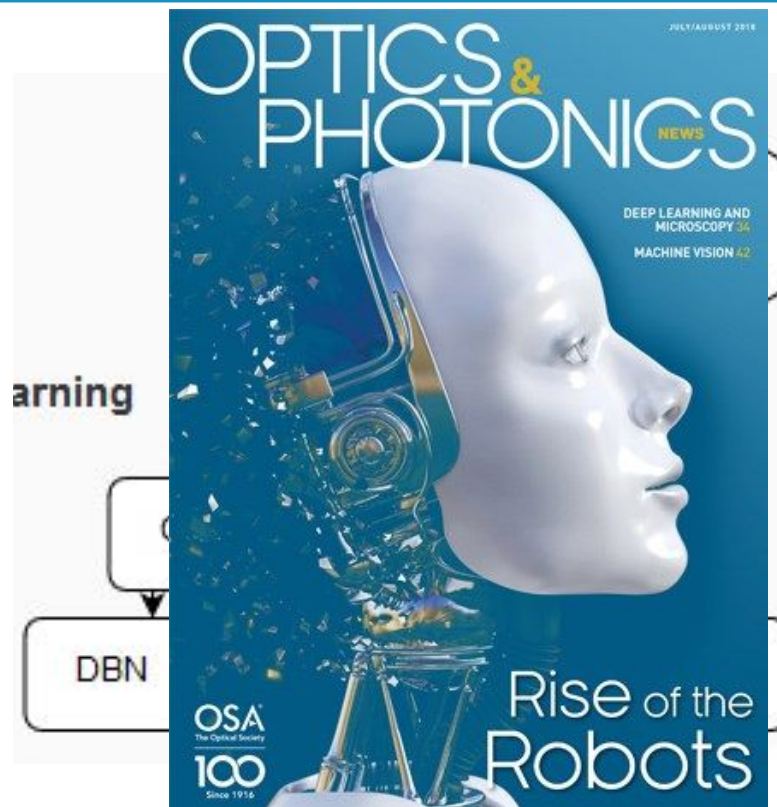
Bibliografía

- **Deep Learning**, Ian Goodfellow and Yoshua Bengio and Aaron Courville
- **A Survey on Deep Learning: Algorithms, Techniques, and Applications**

SAMIRA PUYANFAR, SAAD SADIQ, YILIN YAN, HAIMAN TIAN, YUDONG TAO, MARIA PRESA REYES, MEI-LING SHYU, SHU-CHING CHEN, S.S.IYENGAR, ACM Computing Surveys, Vol. 51, No. 5, Article 92. **September 2018**

Frameworks

- Caffe, DL4j, Torch, Neon, Theano, MXNet, TensorFlow, CNTK



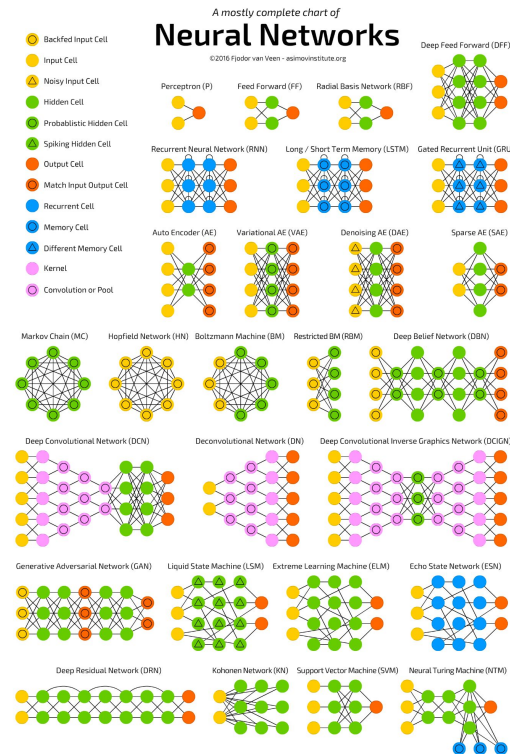
Redes Neuronales Recurrentes y Recursivas

10

Sequence Modeling: Recurrent and Recursive Nets

Recurrent neural networks, or RNNs (Rumelhart et al., 1986a), are a family of neural networks for processing sequential data. Much as a convolutional network is a neural network that is specialized for processing a grid of values \mathbf{X} such as an image, a recurrent neural network is a neural network that is specialized for processing a sequence of values $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(\tau)}$. Just as convolutional networks can readily scale to images with large width and height, and some convolutional networks can process images of variable size, recurrent networks can scale to much longer sequences than would be practical for networks without sequence-based specialization. Most recurrent networks can also process sequences of variable length.

Ref.: Yoshua Bengio, Deep Learning Adaptive Computation and Machine Learning

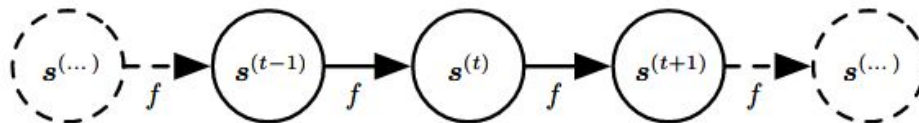


Redes Neuronales Recurrentes

$$s^{(t)} = f(s^{(t-1)}; \theta)$$

- $s^{(t)}$ es el **estado** del sistema
- la definición recurrente se refiere a t-1
- la función $f(s, \theta)$ mapea estados

$$\begin{aligned} s^{(3)} &= f(s^{(2)}; \theta) \\ &= f(f(s^{(1)}; \theta); \theta) \end{aligned}$$



$$s^{(t)} = f(s^{(t-1)}, x^{(t)}; \theta)$$

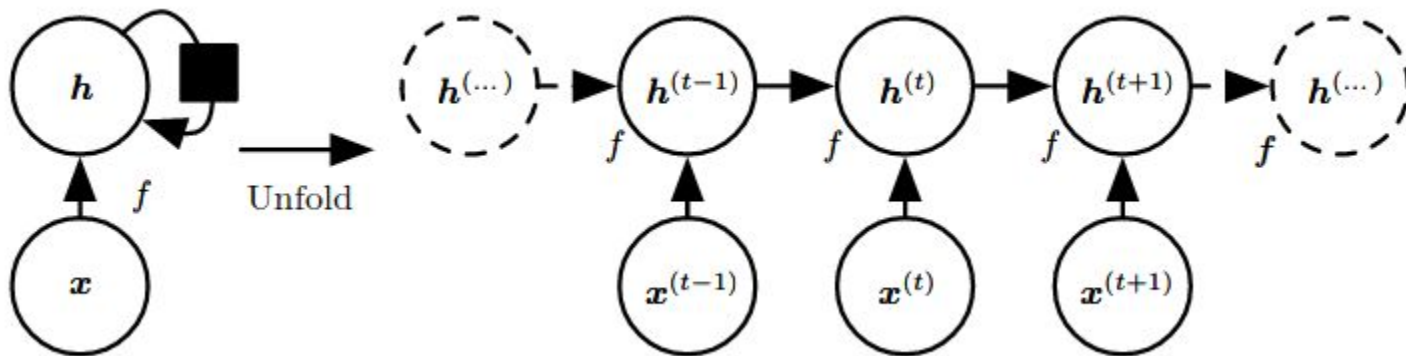
- $x^{(t)}$ es una señal externa
- la definición recurrente se refiere a t-1

Redes Neuronales Recurrentes

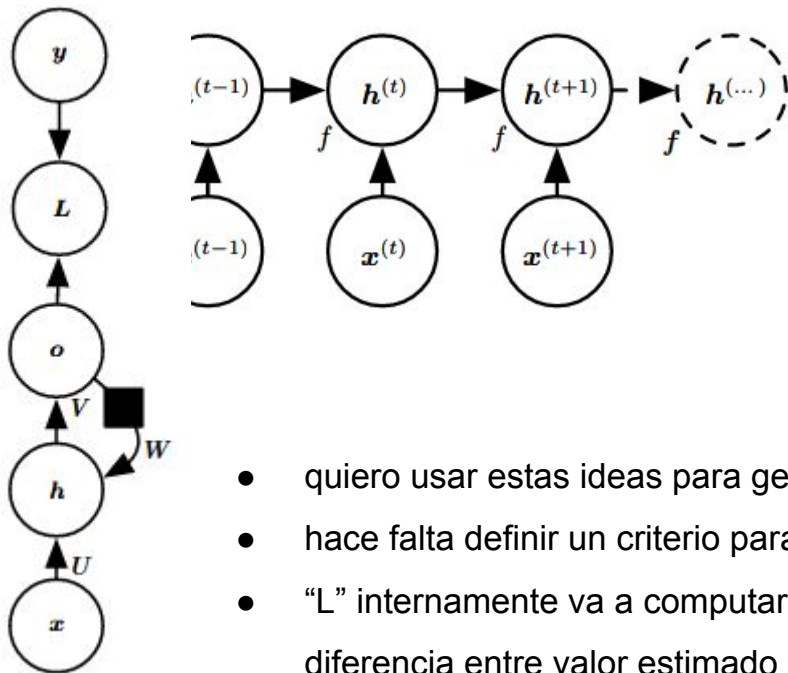
$$h^{(t)} = f(h^{(t-1)}, x^{(t)}; \theta)$$

$$\begin{aligned} h^{(t)} &= g^{(t)}(x^{(t)}, x^{(t-1)}, x^{(t-2)}, \dots, x^{(2)}, x^{(1)}) \\ &= f(h^{(t-1)}, x^{(t)}; \theta). \end{aligned}$$

- Usamos $h^{(t)}$ para representar un **estado oculto** (hidden) del sistema
- Una notación gráfica compacta simplifica la vista desplegada de recurrencia
- $g^{(t)}$ usa toda la serie para el estado $s(t)$

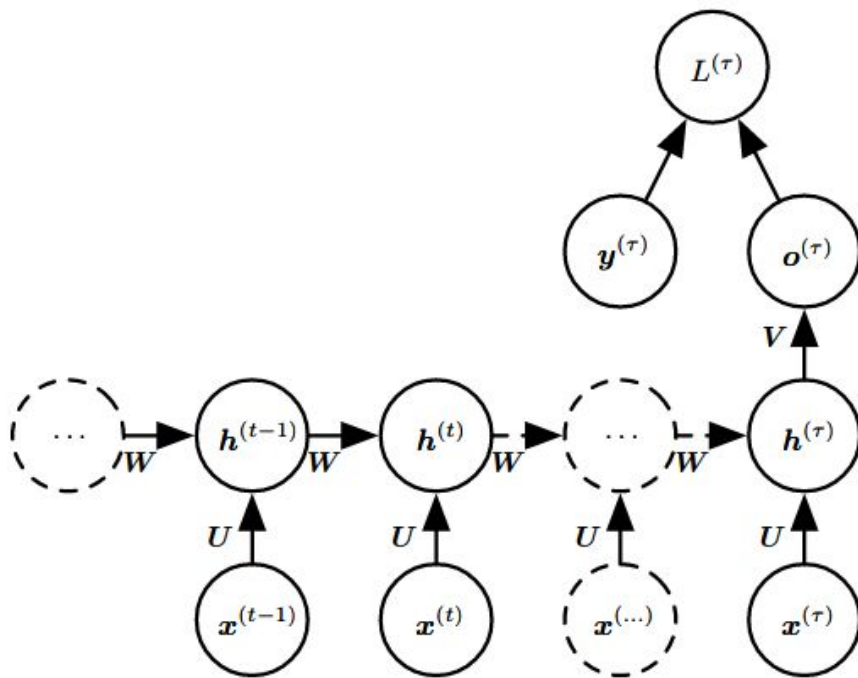


Redes Neuronales Recurrentes



- hay una ventaja en esta notación que refiere al largo de la red: no es el largo de la serie, sino de transiciones de estados.
- es posible usar la misma función de transición f con los mismos parámetros en cada paso. Se posibilita **generalización**.
- quiero usar estas ideas para generar una salida $o^{(t)}$
- hace falta definir un criterio para optimizar esa salida usando una función de costo L (loss)
- “ L ” internamente va a computar un estimador $\hat{y}^{(t)}$ para $y^{(t)}$ y va a buscar minimizar la diferencia entre valor estimado y valor obtenido.

Redes Neuronales Recurrentes



- Distintas **topologías** producen distintos resultados
- RNN que produce una salida en cada paso a partir de conexiones entre unidades ocultas
- RNN que produce una salida en cada paso a partir de sólo el estado anterior
- RNN con conexiones ocultas que leen toda la secuencia y producen una sola salida
- RNN con múltiples salidas...(próx. clase)

Terminología

- Celdas de entrada, ocultas y de salida
- Perceptrones o unidades funcionales
- Circuitos, topologías, arquitectura ([archs](#))
- Función de costo
- Funciones de activación
- Entrenamiento / Test
- Sobreajuste / sub-ajuste (overfitting / underfitting)

Ejemplo con Torch

- Identificar las etapas del algoritmo
 - dataset
 - formato
 - definir arquitectura (input, hidden, output)
 - inicializar
 - entrenar, iterar (epochs)
 - función de costo
 - predicción como etapa de test

Diagnóstico de modelos

Diagnóstico de los modelos

Se basa en determinar la bondad de la estimación del modelo, y en caso de tener un mal ajuste proponer modificaciones apropiadas.

Vamos a ver dos enfoques complementarios:

- Análisis residual
- Análisis de modelos sobre-parametrizados

1. Análisis de residuos

Al igual que en los problemas de regresión, vamos a llamar residuo a la diferencia entre el valor verdadero y el estimado por el modelo:

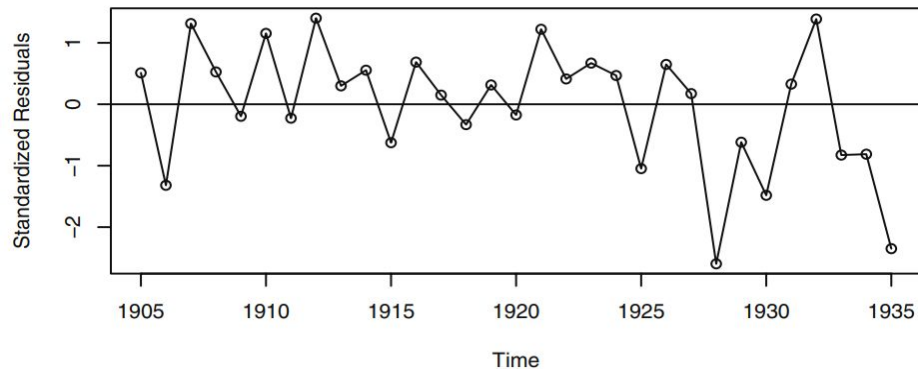
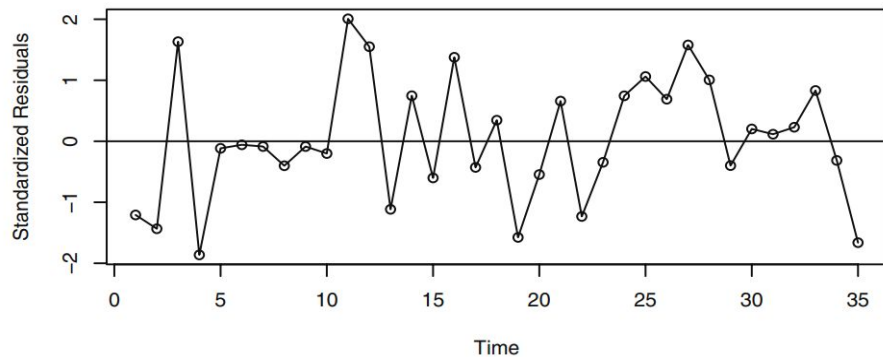
$$\hat{e}_t = Y_t - \hat{Y}_t$$

Si el modelo se encuentra bien estimado, los residuos deberían tener aproximadamente las propiedades del ruido blanco.

Las desviaciones de esta propiedad pueden ayudarnos a corregir el modelo.

Gráfico de residuos

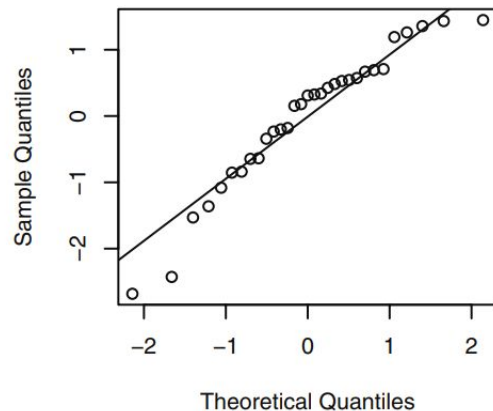
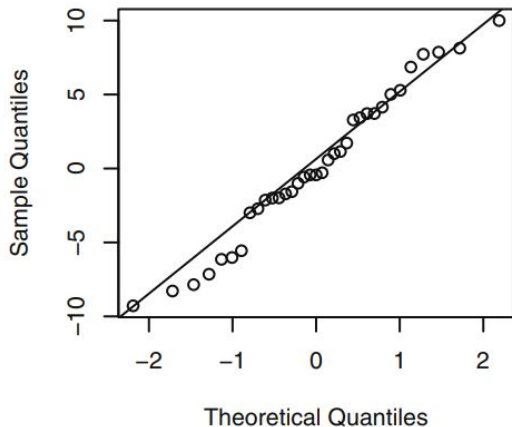
El primer testeo de diagnóstico consiste en graficar los residuos obtenidos a lo largo del tiempo.



Normalidad de los residuos

Una buena forma de analizar la normalidad de los residuos es mediante un QQplot.

También pueden analizarse tests estadísticos como el Shapiro-Wilk (recordar que si la cantidad de muestras es muy grande este camino puede no ser recomendable).

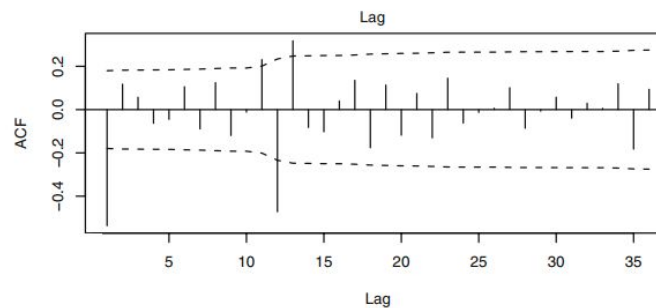
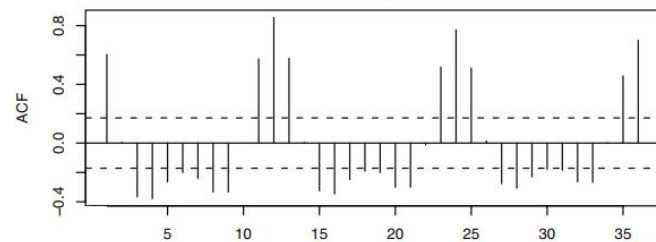
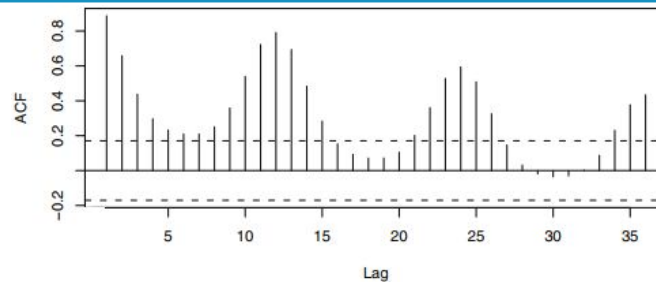
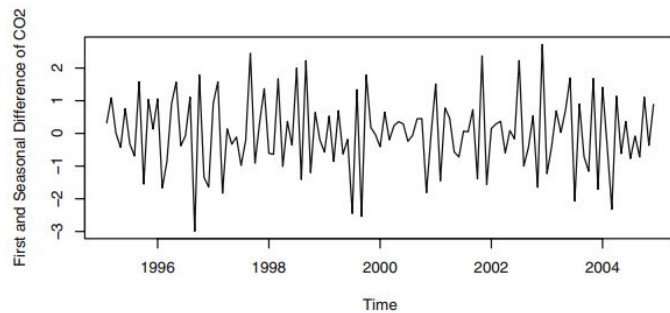
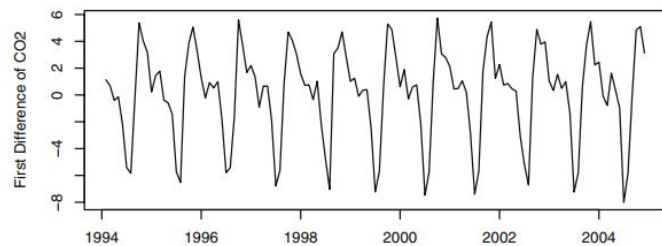
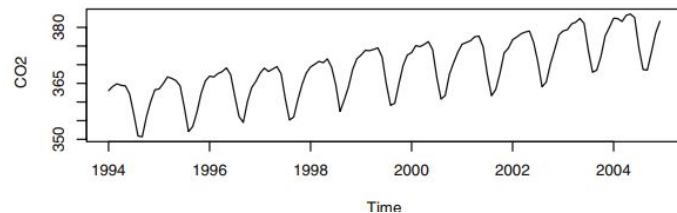


Autocorrelación de los residuos

Para analizar la independencia de los residuos podemos estimar su función de autocorrelación.

Idealmente, para tamaños grandes de muestra, las estimaciones de las correlaciones seguir una distribución normal de media 0 y varianza $1/n$. Sin embargo esto no resulta del todo cierto para lags pequeños j, k , donde la varianza puede resultar significativamente menor y estar altamente correlacionados

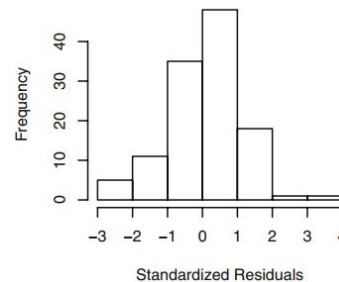
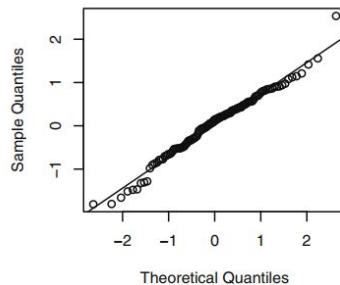
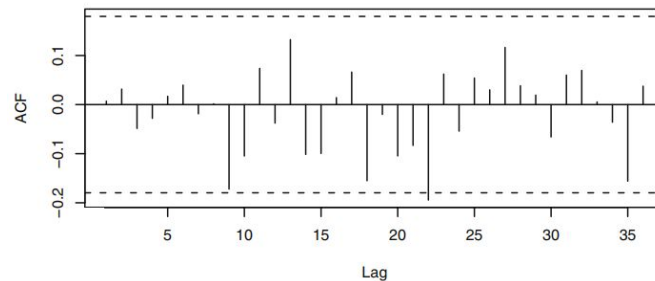
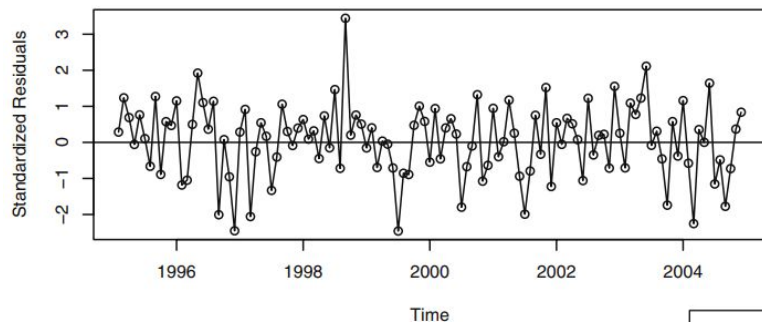
Especificación del modelo - ejemplo



Ajuste del modelo

Una vez especificado el modelo (en el ejemplo anterior un candidato sería $\text{SARIMA}(0,1,1) \times (0,1,1)_{12}$) debemos ajustar los parámetros.

Nuevamente vamos a analizar los residuos de la estimación



Test de Ljung-Box

Contempla las magnitudes de estas **autocorrelaciones en conjunto**.

Box y Pierce propusieron el estadístico $Q = n(\hat{r}_1^2 + \hat{r}_2^2 + \dots + \hat{r}_k^2)$. Mostraron que si los órdenes p y q del ARMA están bien estimados, y n es grande, $Q \approx \chi_{k-p-q}^2$. El problema es que la dist. asintótica está basada en un teorema límite.

Ljung y Box demostraron que esta dist. no se cumple para tamaños comunes de muestras. Propusieron $Q_* = n(n+2)(\frac{\hat{r}_1^2}{n-1} + \frac{\hat{r}_2^2}{n-1} + \dots + \frac{\hat{r}_k^2}{n-1})$ que se asemeja mucho más a la distribución χ^2 .

En ambos casos el test asociado es **H0: “Los residuos están descorrelacionados”**. Busco **no rechazar** la hipótesis nula

2. Overfitting y redundancia de parámetros

Cuando tenemos un modelo ARMA, se presenta el problema de redundancia de parámetros o falta de identificabilidad. Si $a(B)Y_t = b(B)e_t$ es el modelo correcto, luego también es correcto el modelo $(1 - cB)a(B)Y_t = (1 - cB)b(B)e_t$ para cualquier constante c . Sin embargo, si el modelo original se correspondía con un ARMA(p,q), el segundo es un ARMA(p+1,q+1). Decimos en este caso que hay redundancia de parámetros.

1. Especificamos el modelo más sencillo que se vea factible (antes de probar alguno más complejo)
2. Al hacer overffiting, agrandamos la parte MA y la AR por separado
3. Expandir el modelo en la dirección sugerida por el análisis de residuos.

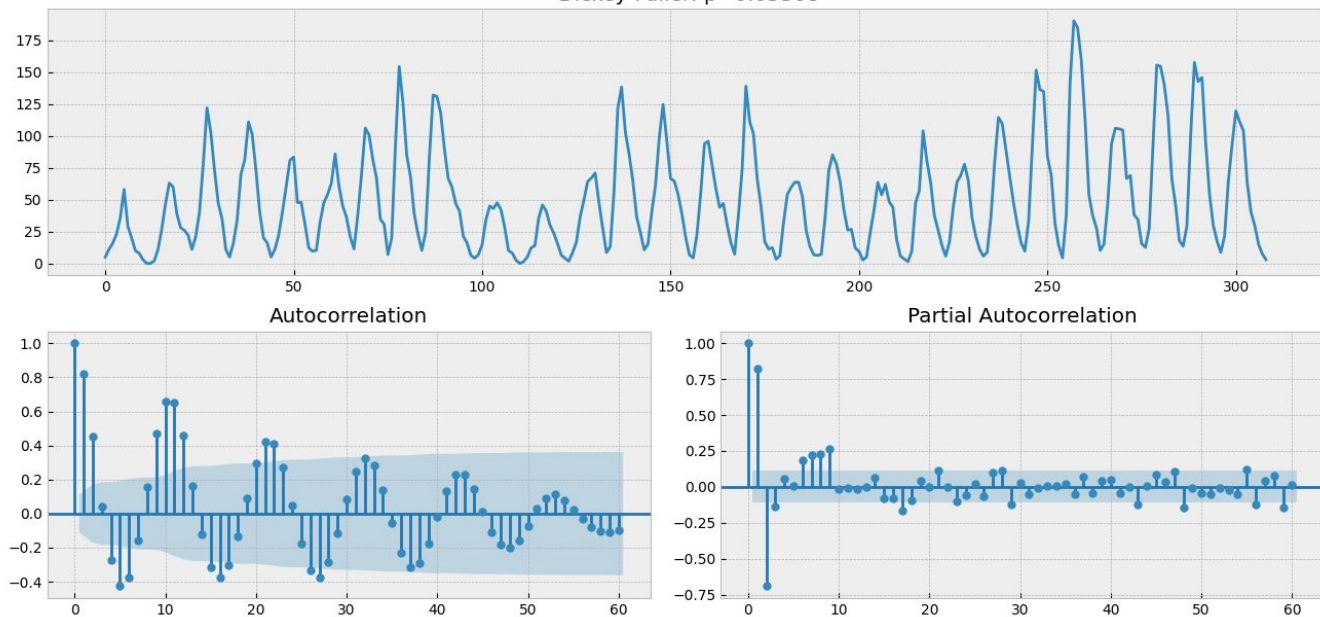
Resultados (S) ARIMA

```
1 SARIMAX Results
2 =====
3 Dep. Variable: y No. Observations: 4840
4 Model: ARIMA(5, 1, 0) Log Likelihood: -10901.009
5 Date: Thu, 11 Nov 2021 AIC: 21814.019
6 Time: 17:51:37 BIC: 21852.926
7 Sample: 0 HQIC: 21827.678
8 - 4840
9 Covariance Type: opg
10 =====
11      coef      std err      z      P>|z|      [0.025      0.975]
12 -----
13 ar.L1      0.0367      0.004      9.613      0.000      0.029      0.044
14 ar.L2      0.0218      0.005      4.612      0.000      0.013      0.031
15 ar.L3      0.0451      0.005      9.124      0.000      0.035      0.055
16 ar.L4     -0.0205      0.005     -3.941      0.000     -0.031     -0.010
17 ar.L5      0.0056      0.005      1.041      0.298     -0.005      0.016
18 sigma2      5.2995      0.027     199.035      0.000      5.247      5.352
19 =====
20 Ljung-Box (L1) (Q): 0.00 Jarque-Bera (JB): 227927.17
21 Prob(Q): 1.00 Prob(JB): 0.00
22 Heteroskedasticity (H): 953.07 Skew: -0.30
23 Prob(H) (two-sided): 0.00 Kurtosis: 36.62
24 =====
25
```

Volvamos al ejemplo de sun activity

Serie de actividad del sol (SUNACTIVITY)

Time Series Analysis Plots
Dickey-Fuller: $p=0.05308$

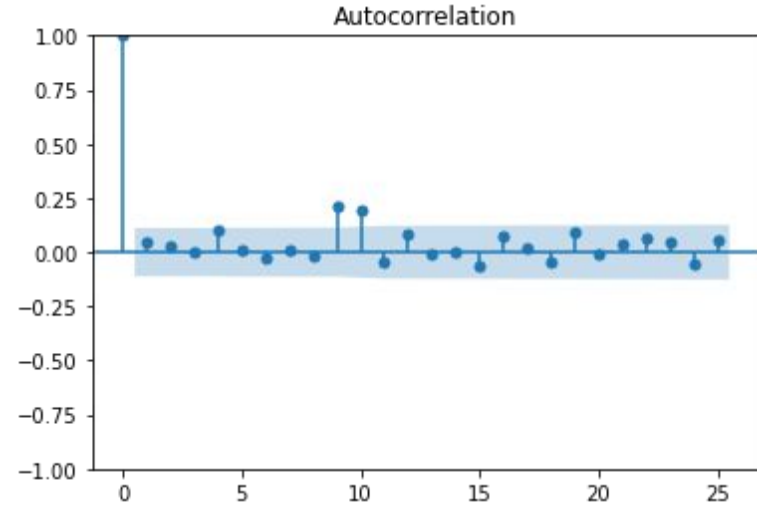
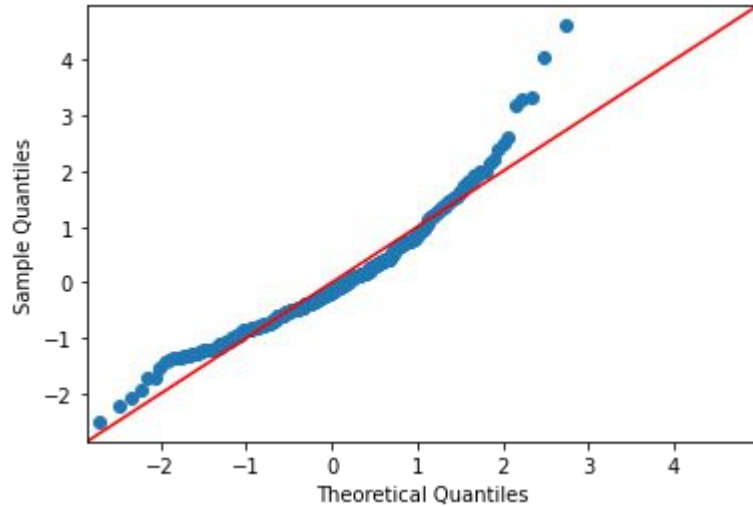


Ajuste de modelo SARIMAX(2,0,3)x(1,1,1)10

No. Observations:	309
Log Likelihood	-1257.865
AIC	2531.730
BIC	2561.307
HQIC	2543.569

	coef	std err	z	P> z	[0.025	0.975]
ar.L1	1.2853	0.183	7.037	0.000	0.927	1.643
ar.L2	-0.6071	0.108	-5.632	0.000	-0.818	-0.396
ma.L1	-0.0998	0.187	-0.533	0.594	-0.467	0.267
ma.L2	0.0945	0.138	0.685	0.493	-0.176	0.365
ma.L3	0.0224	0.122	0.184	0.854	-0.217	0.261
ar.S.L11	0.2089	0.074	2.826	0.005	0.064	0.354
ma.S.L11	-0.9780	0.144	-6.787	0.000	-1.260	-0.696
sigma2	247.3620	30.231	8.182	0.000	188.110	306.614
Ljung-Box (L1) (Q):	0.08	Jarque-Bera (JB): 155.97				
Prob(Q):	0.78	Prob(JB): 0.00				
Heteroskedasticity (H):	1.43	Skew: 1.12				
Prob(H) (two-sided):	0.08	Kurtosis: 5.74				

¿Qué pasa con los residuos?



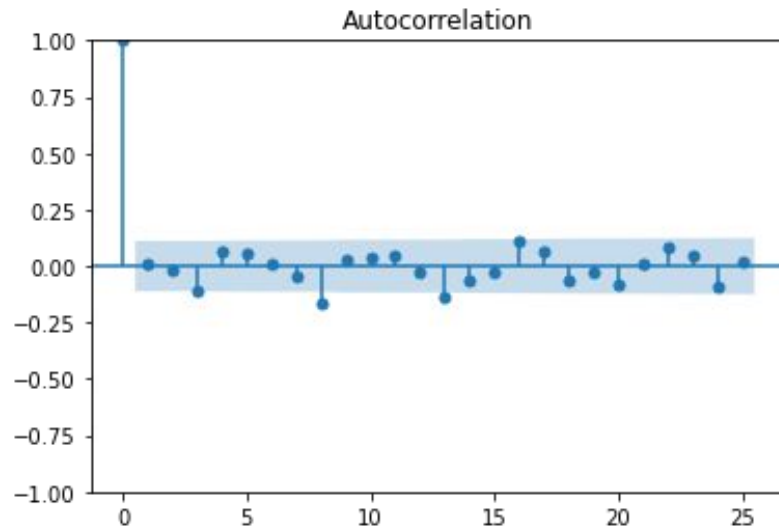
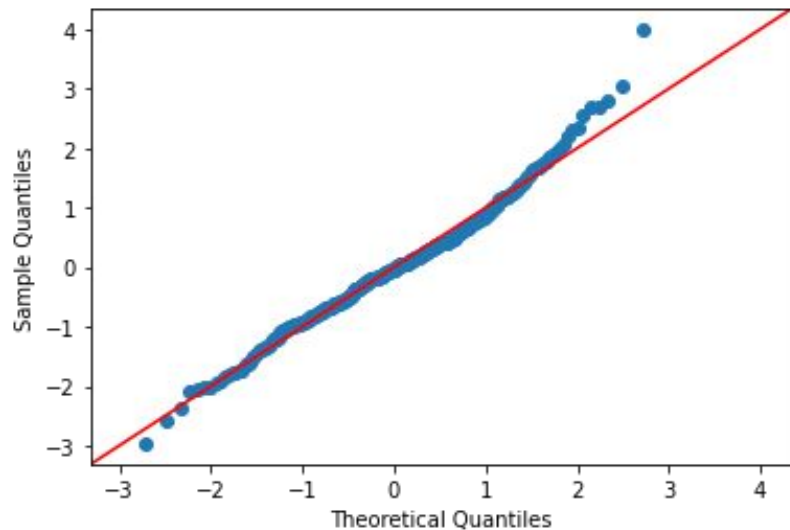
El modelo
todavía se
puede mejorar

Probemos con otro: SARIMA(2,0,3)x(1,1,1)10

No. Observations:	309
Log Likelihood	-1251.934
AIC	2517.868
BIC	2543.701
HQIC	2528.211

	coef	std err	z	P> z	[0.025	0.975]
ar.L1	1.6084	0.029	55.070	0.000	1.551	1.666
ar.L2	-0.9328	0.024	-38.205	0.000	-0.981	-0.885
ma.L1	-1.4432	0.061	-23.733	0.000	-1.562	-1.324
ma.L2	0.4691	0.101	4.651	0.000	0.271	0.667
ma.L3	0.1220	0.064	1.902	0.057	-0.004	0.248
ma.S.L12	-0.9987	2.759	-0.362	0.717	-6.406	4.409
sigma2	241.3748	662.225	0.364	0.715	-1056.563	1539.313
Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB): 24.02				
Prob(Q):	0.97	Prob(JB): 0.00				
Heteroskedasticity (H):	1.41	Skew: 0.42				
Prob(H) (two-sided):	0.09	Kurtosis: 4.11				

Probemos con otro: SARIMA(2,0,3)x(1,1,1)₁₀



¡Este se ve mejor!

Conclusiones

- Para este modelo resultaron algunas observaciones particulares.
- Las componentes estacionales requieren datos interanuales
- Pronósticos de corto plazo: LSTM
- Pronósticos a largo plazo: ARIMA
- Compromiso entre cantidad de datos y método utilizado
- Posibles mejoras extendiendo el tamaño de la red LSTM
- Trabajo futuro: monitoreo, descomposición y combinación de métodos