Análisis de Series de Tiempo

Carrera de Especialización en Inteligencia Artificial

Clase

Ing. Magdalena Bouza, Ing. Carlos German Carreño Romano

Agenda

1. Estacionariedad
2. Estacionalidad
3. Modelos ARIMA y SARIMA
4. Criterios de bondad de modelos

ARMA(p,q)

Modelo ARMA

El modelo arma es una combinación de un proceso AR con un MA. Diremos que $\{Y_t\}$ sigue un modelo ARMA(p,q) si

$$Y_t = a_1 Y_{t-1} + \ldots + a_p Y_{t-p} + e_t - b_1 e_{t-1} - \ldots - b_q e_{t-q}$$

Si se satisfacen las condiciones de estacionariedad, el modelo ARMA(p,q) puede reescribirse como un proceso lineal general con coeficientes $\psi_1, \psi_2, \ldots$ dados por:

$$\begin{cases} \psi_0 = 1 \\ \psi_1 = -b_1 + a_1 \\ \psi_2 = -b_2 + a_2 + a_1\psi_1 \\ \vdots \\ \psi_j = -b_j + a_p\psi_{j-p} + a_{p-1}\psi_{j-p+1} + \ldots + a_1\psi_{j-1} \end{cases} \qquad \psi_j = 0, si\ j < 0\ y\ b_j = 0\ si\ j > q$$

ARMA(p,q)

Se puede ver que la función de autocorrelación está dada por:

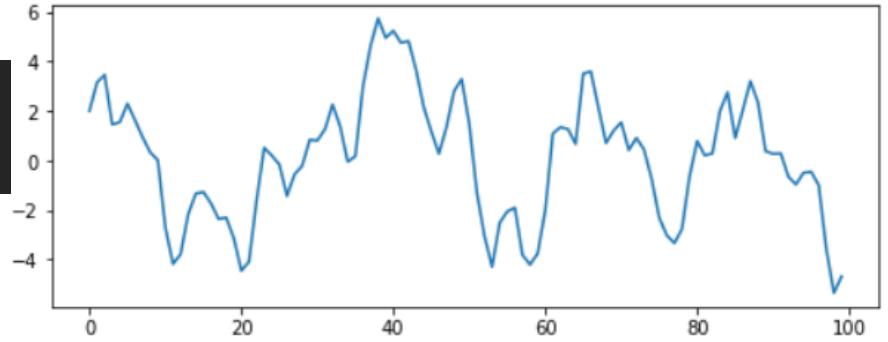
$$\begin{cases} C_0 = a_1 C_1 + a_2 C_2 + \ldots - \sigma_e^2 (b_0 + b_1 \psi_1 + \ldots + b_q \psi_q) \\ C_1 = a_1 C_0 + a_2 C_1 + \ldots + a_p C_{p-1} - \sigma e^2 (b_1 + b_2 \psi_1 + \ldots + b_1 \psi_{q-1}) \\ \quad \vdots \\ C_p = a_1 C_{p-1} + a_2 C_{p-2} + \ldots + a_p C_0 - \sigma_e^2 (b_p + b_{p+1} \psi_1 + \ldots + b_q \psi_{q-p}) \end{cases}$$

Si k>q entonces la expresión puede simplificarse como:

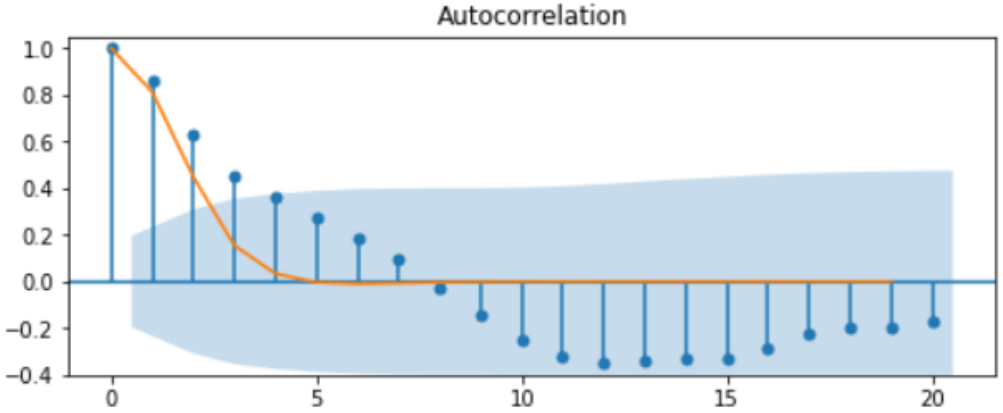
$$C_k = a_1 C_{k-1} + a_2 C_{k-2} + \ldots + a_p C_{k-p}$$

Ejemplo!

```
arma_ts = arma_generate_sample(
    ar=np.r_[1,-ar_coef], ma=np.r_[1,ma_coef],
    nsample =100)
```



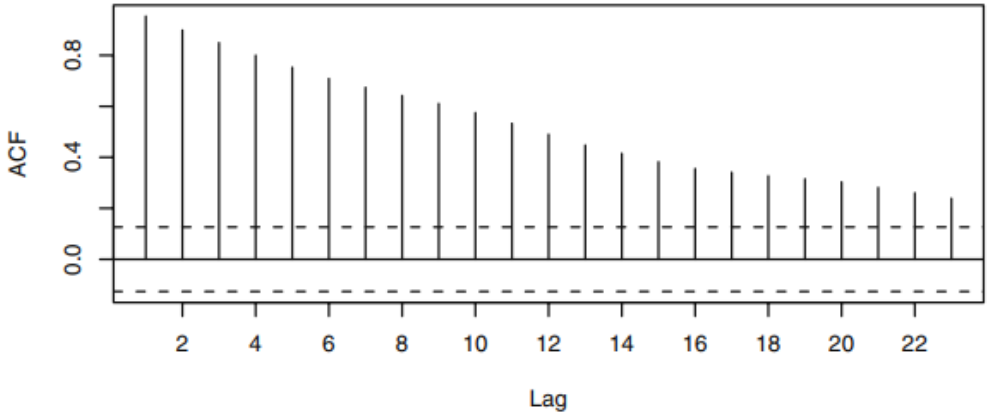

```
plot_acf(arma_ts, ax=ax)
plt.plot(arma_acf(ar=np.r_[1, -ar_coef],
          ma=np.r_[1,ma_coef], lags=20))
```



Estacionariedad

Autocorrelación

Un método relativamente fácil, aunque bastante a ojo, para verificar que una serie **no** es **estacionaria** es a través de su función de autocorrelación muestral.



Si la gráfica no alcanza valores nulos para lags grandes es porque posiblemente no sea estacionaria

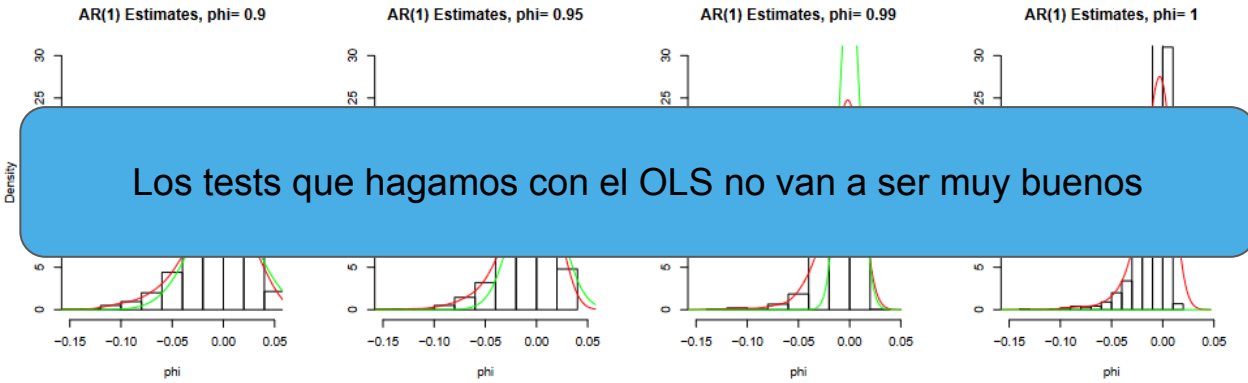
Tests para determinar estcionariedad

Tengo un modelo AR(1) de la forma $Y_t = a_1 Y_{t-1} + e_t$ y quiero saber si $a_1 = 1$.

Si desconocemos la estacionariedad del modelo, sólo podemos estimar el parámetro por OLS:

$$\hat{a}_1 = \frac{\sum_{t=1}^{n} Y_t Y_{t-1}}{\sum_{t=1}^{n} Y_{t-1}^2}$$

Se puede demostrar que $\sqrt{n}(\hat{a}_1 - a_1) \sim \mathcal{N}(0, 1 - a_1^2)$ si $|a_1| < 1$.
Cuando $a_1 \approx 1$, esta aproximación deja de ser válida.



Test de Dickey-Fuller

Propone el modelo $Y_t = aY_{t-1} + W_t$ ← Proceso estacionario

Si **a=1** → El proceso es **no estacionario** (hay caminante aleatorio)

Si miramos la serie diferenciada una vez, tenemos que

$$Y_t - Y_{t-1} = aY_{t-1} + W_t - Y_{t-1} = (a-1)Y_{t-1} + W_t$$

Dickey y Fuller proponen entonces el test

$$H_0 : (a-1) = 0 \quad vs. \quad H_1 : (a-1) \neq 0$$

Busco rechazar el test

Su gran aporte fue hallar la distribución asintótica de $n(\widehat{a-1})$ bajo H0.

Es un test para determinar si la serie (**a una diferenciación**), posee una componente de RW

Test de dickey-Fuller Aumentado

Incorpora al modelo un término de ruido dependiente (pero estacionario)

$$Y_t = aY_{t-1} + X_t \qquad X_t = \sum_{j=1}^{p} \rho_j X_{t-j} + w_t$$

w_t ← Ruido Blanco

Si tomamos la primera diferencia, y observamos que bajo H0

$X_t = Y_t - Y_{t-1}$ tenemos que

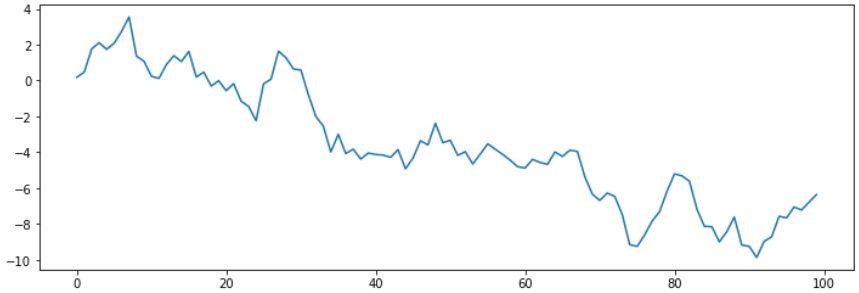
$$Y_t - Y_{t-1} = (a-1)Y_{t-1} + \sum_{j=1}^{p} \rho_j (X_{t-j} - X_{t-j-1}) + W_t$$

Nuevamente se definen las hipótesis:

$$H_0 : (a-1) = 0 \quad vs. \quad H_1 : (a-1) \neq 0$$

p habría que estimarlo, pero podemos dejar que se encargue el software.

¿Cómo implementar en statsmodels?




```
from statsmodels.tsa.stattools import adfuller
adfuller(y)
```

✓ 0.9s

```
(-1.394757894883641,
 0.584786653359185,
 4,
 95,
 {'1%': -3.5011373281819504,
  '5%': -2.8924800524857854,
  '10%': -2.5832749307479226},
```

Estadístico

p-valor

lags usados

observaciones usadas

umbrales

regression : {"c","ct","ctt","n"}

Constant and trend order to include in regression.

- "c" : constant only (default).
- "ct" : constant and trend.
- "ctt" : constant, and linear and quadratic trend.
- "n" : no constant, no trend.

Scripts/Dickey-Fuller_demo.ipynb

Ejemplo con datos reales

```
inputfile = "../Datasets/TECO2.2000.2021.csv"


ts = pd.read_csv(inputfile, header=0, index_col=0, squeeze=True)
ts.fechaHora = pd.to_datetime(ts.fechaHora)
ts.fechaHora=pd.to_datetime(ts.fechaHora).dt.date
ts.fechaHora=pd.DatetimeIndex(ts.fechaHora)
ts=ts.sort_index(ascending=False)
```

Scripts/Dickey-Fuller_dataset_example.ipynb

```
adfuller(ts.ultimoPrecio)
(0.57890231841091,
 0.9870838971395222,
 32,
 4807,
 {'1%': -3.431711097447145,
  '5%': -2.862141452575749,
  '10%': -2.5670901548483767},
```

Ejemplo con datos reales

```
log_ultimoPrecio = np.log(ts.ultimoPrecio)
```

```
adfuller(log_ultimoPrecio, regression='ct')
```

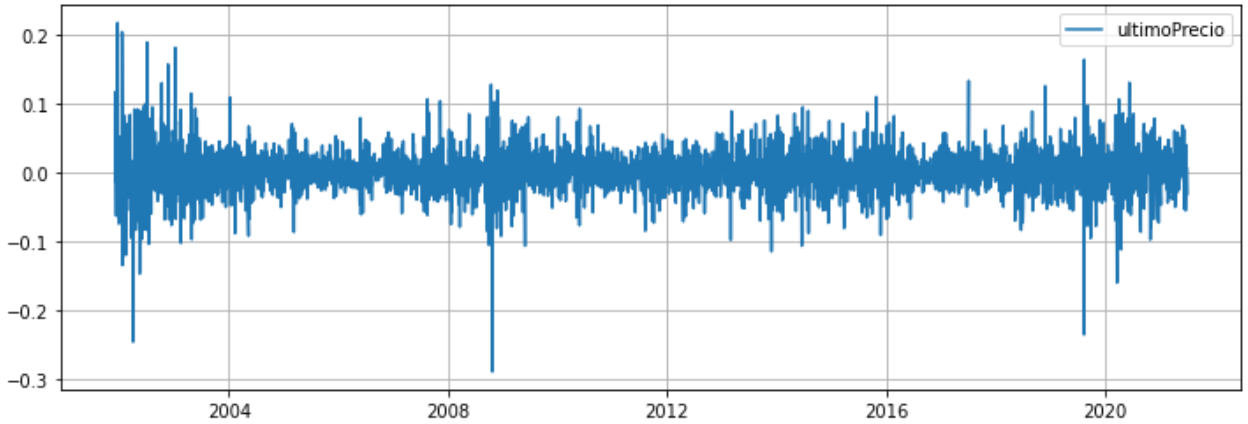
```
(-3.037699968528972,
 0.1218440464079108,
 2,
 4837,
 {'1%': -3.9606428515465306,
  '5%': -3.4113980566548285,
  '10%': -3.127584714163724},
```

Ejemplo con datos reales

```
diff_log_ultimoPrecio = log_ultimoPrecio - log_ultimoPrecio.shift(1)
```

```
adfuller(diff_log_ultimoPrecio.dropna())
```

```
(-46.783753322231384,
 0.0,
 1,
 4837,
 {'1%': -3.4317026511738518,
  '5%': -2.862137721117907,
  '10%': -2.567088168437432},
 -20357.65641243886)
```



Test de Kwiatkowski–Phillips–Schmidt–Shin (KPSS)

Es un test para determinar si la serie es **estacionaria alrededor de una tendencia determinística**.

Plantea que la serie se puede descomponer como

$$Y_t = W_t + X_t + \mu_t$$

Caminante aleatorio

Tendencia determinística

Para luego plantear las hipótesis:

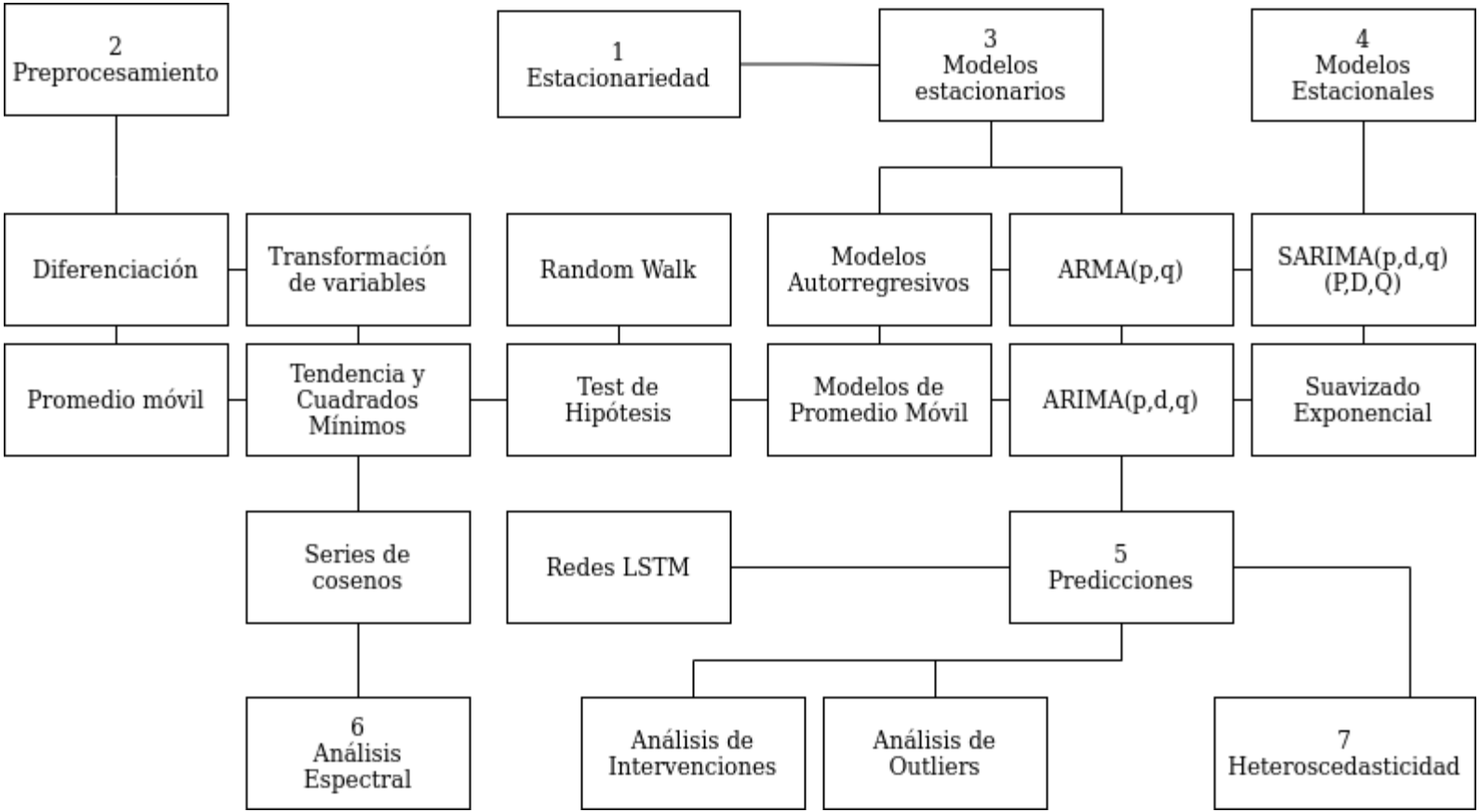
$$H_0 : var(W_t) = 0 \quad H_1 : var(W_t) > 0$$

Busco **no** rechazar el test

Bibliografía extra para D-F y KPSS

- Testing for unit roots
- Augmented Dickey-Fuller root tests
- Statsmodels: Stationarity and detrending

ARIMA(p,d,q)


2
Preprocesamiento
1
Estacionariedad
3
Modelos
estacionarios
4
Modelos
Estacionales
Diferenciación
Transformación
de variables
Random Walk
Modelos
Autorregresivos
ARMA(p,q)
SARIMA(p,d,q)
(P,D,Q)
Promedio móvil
Tendencia y
Cuadrados
Mínimos
Test de
Hipótesis
Modelos de
Promedio Móvil
ARIMA(p,d,q)
Suavizado
Exponencial
Series de
cosenos
Redes LSTM
5
Predicciones
6
Análisis
Espectral
Análisis de
Intervenciones
Análisis de
Outliers
7
Heteroscedasticidad

Operadores

- Operador de Backshift (B), Operador Nabla
- Polinomio característico
- AR, MA compactos
- SARIMA
- Ejemplo

AR, MA + Operador de Backshift (B)

MA(q):

$$\begin{aligned} Y_t &= e_t - b_1 e_{t-1} - \ldots - b_q e_{t-q} \\ &= e_t - b_1 B e_t - \ldots b_q B^q e_t \\ &= (1 - b_1 B - \ldots - b_q B^q) e_t \end{aligned}$$

AR(p)

$$\begin{aligned} Y_t &= e_t + a_1 Y_{t-1} + \ldots + a_p Y_{t-p} \\ Y_t - a_1 Y_{t-1} - \ldots - a_p Y_{t-p} &= e_t \\ Y_t - a_1 B Y_t - \ldots - a_p B^p Y_t &= e_t \\ (1 - a_1 B - \ldots - a_p B^p) Y_t &= e_t \end{aligned}$$

- **Proceso Lineal General**: es útil para encontrar propiedades generales, por ejemplo para condicionar los coeficientes del modelo
- **Polinomio característico**: es útil para expresar relaciones analíticas entre las raíces del polinomio y las expresiones de autocorrelación y autocovarianza del modelo.

AR, MA usando Operadores

AR(p)

$$e_t = Y_t - \sum_{k=1}^{p} a_k Y_{t-k}$$

$$e_t = (1 - \sum_{k=1}^{p} a_k B^k) Y_t$$

$$e_t = \phi(B) Y_t$$

MA(q)

$$Y_t = e_t - \sum_{k=1}^{p} b_k e_{t-k}$$

$$Y_t = (1 - \sum_{k=1}^{p} b_k B^k) e_t$$

$$Y_t = \theta(B) e_t$$

Modelo ARIMA

Diremos que *{Yt}* sigue un modelo ARIMA si $W_t = \nabla^d Y_t$ es un proceso ARMA(p,q) estacionario. Diremos en este caso que *Yt* sigue un proceso ARIMA(p,d,q).

Operador de *backshift*: el operador de backshift (B) opera sobre el índice temporal de la serie de tiempo y la desplaza en una unidad de tiempo:

$$BY_t = Y_{t-1}$$

El operador B es lineal: $B(aY_t + bX_t + c) = aY_{t-1} + bX_{t-1} + c$

En general escribimos que $B^d Y_t = Y_{t-d}$

AR, MA + operador Backshift

MA(q):

$$\begin{aligned} Y_t &= e_t - b_1 e_{t-1} - \ldots - b_q e_{t-q} \\ &= e_t - b_1 B e_t - \ldots b_q B^q e_t \quad \Rightarrow Y_t = b(B) e_t \\ &= (1 - b_1 B - \ldots - b_q B^q) e_t \end{aligned}$$

ARMA(p,q)

$$a(B) Y_t = b(B) e_t$$

AR(p)

$$\begin{aligned} Y_t &= e_t + a_1 Y_{t-1} + \ldots + a_p Y_{t-p} \\ Y_t - a_1 Y_{t-1} - \ldots - a_p Y_{t-p} &= e_t \\ Y_t - a_1 B Y_t - \ldots - a_p B^p Y_t &= e_t \quad \Rightarrow a(B) Y_t = e_t \\ (1 - a_1 B - \ldots - a_p B^p) Y_t &= e_t \end{aligned}$$

ARIMA(p,d,q)

$$a(B)(1 - B)^d Y_t = b(B) e_t$$

ARMA, ARIMA + operador Backshift

ARMA(p,q)

$$Y_t = \theta(B)e_t$$

$$e_t = \phi(B)Y_t$$

$$\phi(B)Y_t = \theta(B)e_t$$

$$a(B)Y_t = b(B)e_t$$

ARIMA(p,d,q)

$$\nabla^d Y_t = W_t$$

$$W_t : ARMA(p,q)$$

$$a(B)(1-B)^d Y_t = b(B)e_t$$

Ejemplos

https://github.com/charlieromano/TimeSeries/blob/main/Scripts/**predicciones_AR_MA.py**

Statsmodels

https://www.statsmodels.org/devel/generated/statsmodels.tsa.arima.model.ARIMA.html

Estacionalidad

Modelo SARMA multiplicativo

Cuando el proceso es estacionario podemos definir un modelo SARMA multiplicativo **ARMA(p,q)x(P,Q)s**, con período estacional **s** como un modelo AR con polinomio característico $a(x)\alpha(x)$ más un MA con polinomio característico $b(x)\beta(x)$. Con

$$
\begin{aligned}
a(x) &= 1 - a_1 x - a_2 x^2 - \ldots - a_p x^p \\
\alpha(x) &= 1 - \alpha_1 x^s - \alpha_2 x^{2s} - \ldots - \alpha_P x^{Ps}, \\
b(x) &= 1 - b_1 x - b_2 x^2 - \ldots b_q x^q \\
\beta(x) &= 1 - \beta_1 x^s - \beta_2 x^{2s} - \ldots - \beta_P x^{Ps},
\end{aligned}
$$

Modelo SARIMA multiplicativo

In general, then, we define a **multiplicative seasonal ARMA$(p,q)\times(P,Q)_s$ model with seasonal period s** as a model with AR characteristic polynomial $\phi(x)\Phi(x)$ and MA characteristic polynomial $\theta(x)\Theta(x)$, where

$$\left.\begin{aligned}\phi(x) &= 1-\phi_1 x-\phi_2 x^2-\cdots-\phi_p x^p\\ \Phi(x) &= 1-\Phi_1 x^s-\Phi_2 x^{2s}-\cdots-\Phi_P x^{Ps}\end{aligned}\right\} \tag{10.2.6}$$

and

$$\left.\begin{aligned}\theta(x) &= 1-\theta_1 x-\theta_2 x^2-\cdots-\theta_q x^q\\ \Theta(x) &= 1-\Theta_1 x^s-\Theta_2 x^{2s}-\cdots-\Theta_Q x^{Qs}\end{aligned}\right\} \tag{10.2.7}$$

The model may also contain a constant term θ_0. Note once more that we have just a special ARMA model with AR order $p+Ps$ and MA order $q+Qs$, but the coefficients are not completely general, being determined by only $p+P+q+Q$ coefficients. If $s=12$, $p+P+q+Q$ will be considerably smaller than $p+Ps+q+Qs$ and will allow a much more parsimonious model.

Modelo SARIMA multiplicativo

Una herramienta importante es el análisis de procesos estacionales **no estacionarios** es la diferenciación estacional de período s para la serie {Yt}, 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.

$$\nabla_s^d Y_t = (1 - B_s)^d Y_t$$

Criterios de Bondad de modelos

Criterios de bondad de modelo

Uno de los objetivos cuando analizamos series de tiempo es poder modelarlas. Esto incluye tanto elegir la familia de modelos correcta como hallar todos los parámetros de la misma.

Es necesario entonces poder contar con algún criterio que nos permita saber cuán bueno es nuestro modelo, es decir cuán cerca se encuentra la distribución especificada por el modelo a la distribución verdadera de los datos.

Divergencia de Kullback-Leibler (DK-L)

- Mis datos siguen un **proceso desconocido** $g(y)$
- Proponemos un modelo $f(y)$ para aproximarnos a $g(y)$
- "Métrica" para ver si los modelos son parecidos: Divergencia de Kullback-Leibler

$$D_{\mathrm{KL}}(g \parallel f) = \int_{-\infty}^{\infty} g(y) \log\left(\frac{g(y)}{f(y)}\right) dy = \mathbb{E}\left[\log\left(\frac{g(Y)}{f(Y)}\right)\right]$$

→ Da una medida de la pérdida de información que tenemos al aproximar $f(y)$ por $g(y)$.

Cuanto más pequeño sea el valor de la divergencia K-L, más cerca estarán $f(y)$ y $g(y)$.

El mejor modelo será el que minimice la DK-L.

Estimación de la divergencia K-L

Problema: en general no se conoce el valor verdadero de la distribución g(y),
Solución: Estimamos $D_{\mathrm{KL}}(g \parallel f)$ a partir de las muestras $y_1, \ldots, y_n$.
Supuesto: las muestras son observadas de forma **independiente** de g(y).

En primer lugar, vemos que podemos reescribir

$$D_{\mathrm{KL}}(g \parallel f) = \mathbb{E}\left[\log\left(\frac{g(Y)}{f(Y)}\right)\right] = \mathbb{E}[\log(g(Y))] - \mathbb{E}[\log(f(Y))]$$

no puede ser calculado (pero no nos interesa para comparar)

Estimación de la divergencia K-L

Problema: $\mathbb{E}[\log(f(Y))] = \int \log(f(y))g(y)dy$ tampoco se puede calcular.

Solución: la ley de los grandes números garantiza que

$$\underbrace{\frac{1}{n}\sum_{i=1}^{n}\log(f(y_i))}_{\text{log-verosimilitud}} \to \mathbb{E}[\log(f(Y))]$$

$$\ell = \sum_{i=1}^{n}\log(f(y_i)) = \log(L)$$

Obs: Minimizar $D_{\mathrm{KL}}(g \parallel f)$equivale a maximizar la log-verosimiliud. Por lo tanto una forma natural de hallar los mejores parámetros para un modelo dado es utilizando el estimador de máxima verosimilitud.

Criterio de Información de Akaike (AIC)

El modelo de información de Akaike sirve para comparar la bondad de dos modelos propuestos, y se basa en los resultados analizados previamente.

$$\mathrm{AIC} = 2k - 2\ell(\hat{\theta})$$

cantidad de parámetros parámetros estimados por MV

Se elegirá entonces el modelo que alcance el menor AIC.

Observaciones:

- El resultado de AIC es asintótico pues se basa en la LGN
- AIC no nos dice cuán bueno es cada modelo sino cuál es el mejor de todos (podría ser que sean todos malos)

Criterio de información Bayesiano

El criterio de información Bayesiano también se basa en la log-verosimilitud, pero penaliza más una mayor cantidad de parámetros.

$$\mathrm{BIC} = k\log(n) - 2\ell(\hat{\theta})$$

Nuevamente, se preferirán los modelos con BIC más bajo.

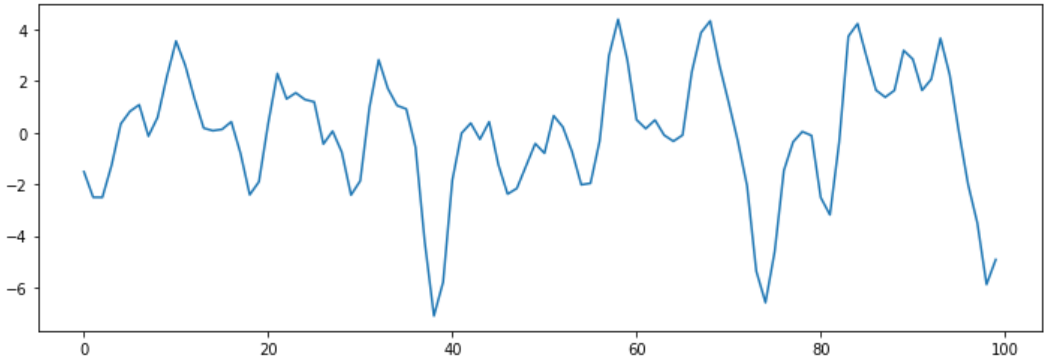
El BIC aparece como consecuencia de un enfoque Bayesiano, donde se busca el modelo f que maximice .

$$\mathbb{P}(f|y_1, \ldots, y_n)$$

Para una explicación más detallada ver “Elements of Statistical Learning”, Trevor Hastie, Robert Tibshirani, Jerome Friedman.

Ejemplo

```
ar_coef  = np.array([0.8,-0.2])
ma_coef = np.array([0.9,0.5,-.3])
arma_ts = arma_generate_sample(
ar=np.r_[1,-ar_coef], ma=np.r_[1,ma_coef], nsample =100)
plt.figure(figsize=(8,3))
plt.plot(arma_ts)
```



compare_model.ipynb

Ejemplo

```
model_bad = ARIMA(arma_ts, order=(5, 0, 5))
model_bad_res = model_bad.fit()
print(model_bad_res.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  100
Model:                 ARIMA(5, 0, 5)   Log Likelihood                -139.350
Date:                Sat, 14 May 2022   AIC                            302.699
Time:                        12:55:57   BIC                            333.961
Sample:                             0   HQIC                           315.351
                                  - 100
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.8097      0.432     -1.873      0.061      -1.657       0.037
ar.L1          2.0463      0.526      3.888      0.000       1.015       3.078
ar.L2         -2.0668      0.904     -2.285      0.022      -3.840      -0.294
ar.L3          0.8230      0.966      0.852      0.394      -1.071       2.716
ar.L4          0.1352      0.489      0.277      0.782      -0.822       1.093
ar.L5         -0.1899      0.158     -1.204      0.229      -0.499       0.119
ma.L1         -0.4931      0.554     -0.891      0.373      -1.578       0.592
ma.L2          0.3442      0.172      1.998      0.046       0.006       0.682
ma.L3         -0.0143      0.288     -0.050      0.960      -0.578       0.549
ma.L4          0.6987      0.164      4.250      0.000       0.376       1.021
ma.L5         -0.5023      0.485     -1.035      0.301      -1.453       0.449
sigma2         0.9075      0.139      6.523      0.000       0.635       1.180
==============================================================================
```

```
model_good = ARIMA(arma_ts, order=(2, 0, 3))
model_good_res = model_good.fit()
print(model_good_res.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  100
Model:                 ARIMA(2, 0, 3)   Log Likelihood                -142.279
Date:                Sat, 14 May 2022   AIC                            298.559
Time:                        12:55:48   BIC                            316.795
Sample:                             0   HQIC                           305.939
                                  - 100
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.8281      0.434     -1.908      0.056      -1.679       0.023
ar.L1          1.1745      0.620      1.894      0.058      -0.041       2.390
ar.L2         -0.3519      0.225     -1.565      0.118      -0.793       0.089
ma.L1          0.4292      0.647      0.663      0.507      -0.839       1.697
ma.L2         -0.0531      0.792     -0.067      0.946      -1.605       1.499
ma.L3         -0.6309      0.574     -1.100      0.271      -1.755       0.494
sigma2         0.9665      0.139      6.963      0.000       0.694       1.239
==============================================================================
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):                 0.32
Prob(Q):                              0.88   Prob(JB):                         0.85
Heteroskedasticity (H):               1.09   Skew:                            -0.14
Prob(H) (two-sided):                  0.80   Kurtosis:                         2.96
==============================================================================
```

¿preguntas?