



Análisis de Series de Tiempo

Carrera de Especialización para Inteligencia Artificial

Predicciones

Predicciones

Uno de los principales motivos de entrenar un modelo es para poder hacer predicciones acerca de los valores que va a tomar la serie de tiempo.

Es sumamente importante también conocer la precisión de estas estimaciones.



Para la mayor parte de lo que veamos esta clase vamos a suponer que conocemos perfectamente el modelo verdadero. Si bien es una suposición que en la práctica no se cumple, el uso de parámetros estimados cuando se cuenta con una cantidad grande de muestras no modifica significativamente los resultados presentados.

Objetivo

Dadas la historia disponible en un instante t , Y_1, \dots, Y_t queremos predecir el valor de Y_{t+k} que va a ocurrir dentro de k instantes de tiempo.

De la materia Probabilidad y Estadística, sabemos que el mejor predictor de Y_{t+k} basado en las muestras Y_1, \dots, Y_t es la esperanza condicional

$$\hat{Y}_t(\ell) = \mathbb{E}[Y_{t+\ell} | Y_1, \dots, Y_t]$$

Predicción con tendencias determinísticas

Consideremos el caso visto en la clase 3 donde Y_t podía modelarse como

$$Y_t = X_t + \mu_t$$

donde X_t es una serie de tiempo de media nula y μ_t es la tendencia determinística.

Si además X_t tiene las propiedades de ruido blanco de varianza C_0^2 , ocurre que X_t, X_l son independientes $\forall t \neq l$ y

$$\hat{Y}_{t+l} = \mathbb{E}[Y_{t+l} | Y_1, \dots, Y_t] = \mathbb{E}[\mu_{t+l} + X_{t+l} | Y_1, \dots, Y_t] = \mu_{t+l}$$

El error de predicción en este caso resulta

$$e_t(l) = Y_{t+l} - \hat{Y}_t(l) = X_{t+l}$$

con

$$\mathbb{E}[e_t(l)] = 0 \text{ y } \text{var}(e_t(l)) = C_0$$

Predicción de ARIMA - ejemplo AR(1)

Consideremos un AR(1) con media distinta de cero:

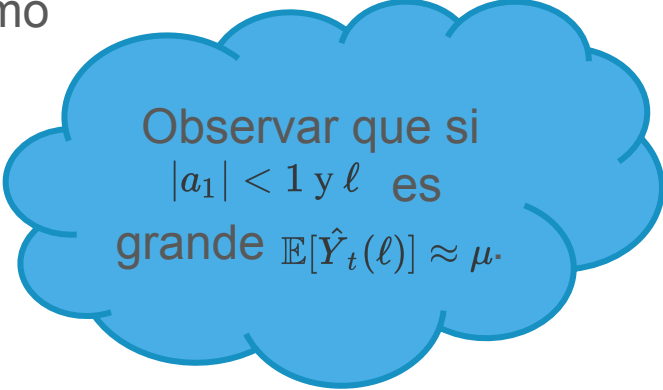
$$Y_t - \mu = a_1(Y_{t-1} - \mu) + e_t$$

Luego, la predicción a un paso resulta

$$\hat{Y}_t(1) - \mu = \mathbb{E}[Y_{t+1}|Y_1, \dots, Y_t] = \mathbb{E}[a_1(Y_t - \mu) + e_t|Y_1, \dots, Y_t] = a_1(Y_t - \mu)$$

Podemos generalizar la predicción a ℓ pasos como

$$\hat{Y}_t(\ell) = \mu + a_1^\ell(Y_t - \mu)$$



Observar que si $|a_1| < 1$ y ℓ es grande $\mathbb{E}[\hat{Y}_t(\ell)] \approx \mu$.

Predicción de ARIMA - ejemplo AR(1)

Siguiendo la misma lógica de antes,

$$\begin{aligned} e_t(1) &= Y_{t+1} - \hat{Y}_t(1) = Y_{t+1} - (a_1(Y_t - \mu) + \mu) \\ &= a_1(Y_t - \mu) + \mu + e_{t+1} - (a_1(Y_t - \mu) + \mu) = e_{t+1} \end{aligned}$$

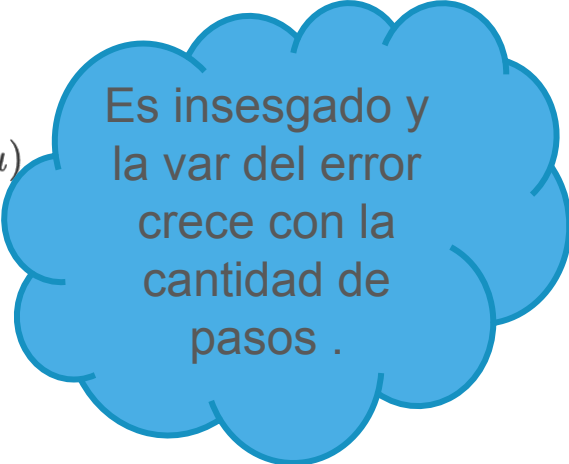
De donde obtenemos que

$$\mathbb{E}[e_t(1)] = 0 \text{ y } \text{var}(e_t(1)) = C_0$$

Para el caso general: $e_t(\ell) = Y_{t+\ell} - \hat{Y}_t(\ell) = Y_{t+\ell} - (a_1^\ell(Y_t - \mu) + \mu)$

$$\begin{aligned} &= e_{t+\ell} + a_1 e_{t+\ell-1} + \dots + a_1^{\ell-1} e_t \\ &= e_{t+\ell} + \psi_1 e_{t+\ell-1} + \dots + \psi_{\ell-1} e_{t+1} \end{aligned}$$

$$\begin{aligned} \mathbb{E}[e_t(\ell)] &= 0 \text{ y } \text{var}(e_t(\ell)) = (1 + \psi_1 + \dots + \psi_{\ell-1})\sigma_e^2 \\ &= \sigma_e^2 \left(\frac{1 - a_1^{2\ell}}{1 - a_1^2} \right) \end{aligned}$$



Es insesgado y la var del error crece con la cantidad de pasos .

Predicción de ARIMA - ejemplo MA(1)

Consideremos ahora un MA con media no nula: $Y_t = \mu + e_t - b_1 e_{t-1}$

En este caso resulta que la predicción a un paso se puede aproximar como

$$\hat{Y}_t(1) = \mathbb{E}[\mu + e_{t+1} - b_1 e_t | Y_1, \dots, Y_t] = \mu - b_1 \mathbb{E}[e_t | Y_1, \dots, Y_t] \approx \mu - b_1 e_t$$

Nuevamente, podemos obtener una expresión más general

$$\hat{Y}_t(\ell) = \mathbb{E}[\mu + e_{t+\ell} - b_1 e_{t+\ell-1} | Y_1, \dots, Y_t] = \mu \quad \ell > 1$$

Para el caso de $\ell = 1$, tenemos que el error resulta

$$e_t(1) = e_{t+1}$$

Predicción de ARIMA - ejemplo Random Walk

Consideremos el caso del caminante aleatorio con deriva (*drift*)

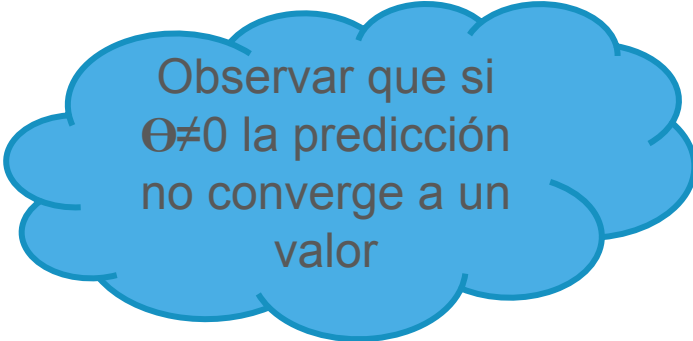
$$Y_t = Y_{t-1} + \theta - e_t$$

donde la predicción a un paso resulta

$$\hat{Y}_t(1) = \mathbb{E}[Y_t + \theta - e_{t+1} | Y_1, \dots, Y_t] = Y_t + \theta$$

Similar al caso AR(1) la predicción a ℓ pasos se puede obtener a partir de una expresión recursiva como

$$\hat{Y}_t(\ell) = Y_t + \theta\ell, \quad \ell \geq 1$$



Observar que si $\theta \neq 0$ la predicción no converge a un valor

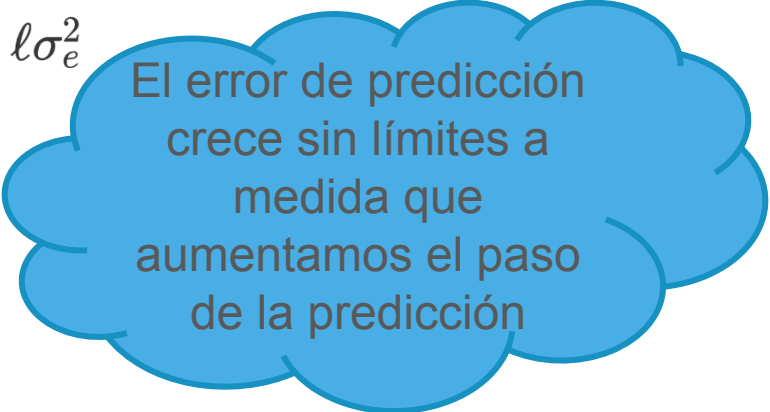
Predicción de ARIMA - ejemplo Random Walk

El error para este ejemplo del caminante aleatorio resulta

$$\begin{aligned}e_t(\ell) &= Y_{t+1} - \hat{Y}_t(\ell) = (Y_t + e_{t+1} + \dots + e_{t+\ell} + \ell\theta) - (Y_t + \theta\ell) \\ &= e_{t+1} + \dots + e_{t+\ell}\end{aligned}$$

Se obtiene entonces que

$$\mathbb{E}[e_t(\ell)] = 0 \text{ y } \text{var}(e_t(\ell)) = \ell\sigma_e^2$$



El error de predicción
crece sin límites a
medida que
aumentamos el paso
de la predicción

Predicción de ARIMA - ejemplo ARMA(p,q)

Consideremos ahora un modelo ARMA(p,q)

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

En este caso, se puede ver que

$$\hat{Y}_t(\ell) = \mathbb{E}[Y_{t+\ell} | Y_1, \dots, Y_t] = a_1 \hat{Y}_t(\ell-1) + \dots + a_p \hat{Y}_t(p) + \theta_0 - b_1 \mathbb{E}[e_{t-1} | Y_1, \dots, Y_t] + \dots + b_q \mathbb{E}[e_{t-q} | Y_1, \dots, Y_t]$$

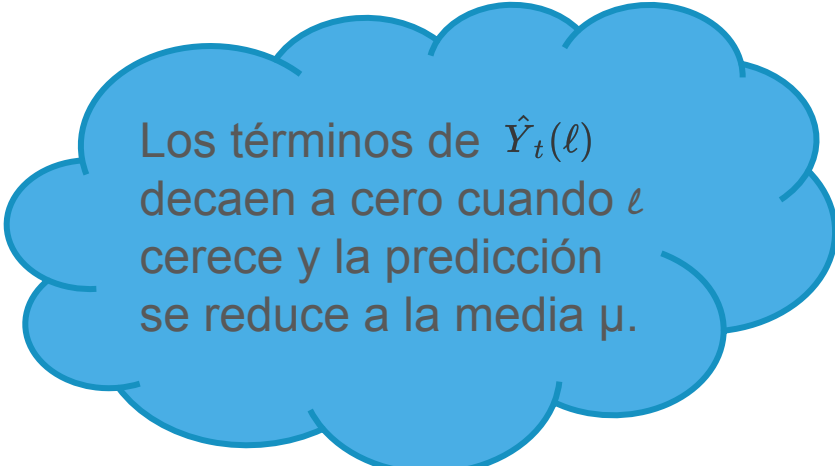
donde $\theta_0 = \mu(1 - a_1 - \dots - a_p)$.

Se observa que si $\ell > q$ los términos asociados al MA desaparecen y el comportamiento queda regido por la componente AR del modelo. Con lo cual tenemos que

$$\hat{Y}_t(\ell) = a_1 (\hat{Y}_t(\ell-1) - \mu) + \dots + a_p (\hat{Y}_t(p) - \mu) + \mu \quad \ell > q$$

Predicción de ARIMA - ejemplo ARMA(p,q)

Observar que la expresión hallada de $\hat{Y}_t(\ell)$ sigue la misma recursión de Yule-Walker que la función de autocorrelación del proceso, y por lo tanto las raíces de la ec. característica determinarán el comportamiento para ℓ grande. En particular, se puede escribir a $\hat{Y}_t(\ell)$ como sumas de términos que decaen exponencialmente con ℓ , y senoidales amortiguadas.



Los términos de $\hat{Y}_t(\ell)$ decaen a cero cuando ℓ crece y la predicción se reduce a la media μ .

Modelo ARMA como proceso lineal truncado

Se puede demostrar (ver. apéndice G de Time Series Analysis) que todo modelo ARMA(p,q) se puede representar usando un proceso lineal truncado:

$$Y_{t+\ell} = C_t(\ell) + I_t(\ell), \quad \ell > 1$$

donde $C_t(\ell)$ es una función de Y_t, Y_{t-1}, \dots y

$$I_t(\ell) = e_{t+\ell} + \psi_1 e_{t+\ell-1} + \dots + \psi_{\ell-1} e_{t+1}$$

Además, si ℓ es lo suficientemente grande y el sistema es invertible $C_t(\ell)$ depende sólo de Y_t, Y_{t-1}, \dots, Y_1 , y

$$\hat{Y}_t(\ell) = C_t(\ell) \text{ y } e_t(\ell) = Y_{t+\ell} - \hat{Y}_t(\ell) = I_t(\ell)$$

$$\mathbb{E}[e_t(\ell)] = 0$$

$$\text{var}(e_t(\ell)) = (1 + \psi_1 + \dots + \psi_{\ell-1})\sigma_e^2$$

$$\text{var}(e_t(\ell)) \approx C_0 \text{ para } \ell \text{ grande}$$

Predicción de ARIMA - ARIMA(p,d,q)

Se puede ver que un modelo ARIMA(p,d,q) se puede reescribir como un modelo ARMA(p+d,q) no estacionario:

$$Y_t = \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \dots + \varphi_{t-p-d} Y_{t-p-d} + e_t - b_1(e_t - 1) - \dots - b_q e_{t-q}$$

y las predicciones pueden calcularse igual que para el caso ARMA(p+d,q) y

$$e_t(\ell) = e_{t+\ell} + \psi_1 e_{t-1} + \dots + \psi_{\ell-1} e_{t+\ell-1}$$

Nuevamente es válido que

$$\mathbb{E}[e_t(\ell)] = 0 \text{ y } \text{var}(e_t(\ell)) = \sigma_e^2 \sum_{i=1}^{\ell-1} \psi_i^2$$

La diferencia es que al ser un proceso no estacionario los coeficientes no convergen a cero a medida que aumenta i , y la varianza aumenta sin cota a medida que ℓ aumenta.

Límites de la predicción

Como siempre, queremos saber la bondad de nuestra estimación. Hasta ahora analizamos la media y la varianza del error para cada instante predicho.

El objetivo es poder brindar un intervalo de confianza alrededor del valor predicho para cada valor de ℓ .

Si las innovaciones (e_t) siguen una distribución gaussiana, luego el error $e_t(\ell)$ también va a seguir una distribución gaussiana, de media y varianza ya calculadas. Luego,

$$\mathbb{P} \left(z_{\alpha/2} \leq \frac{Y_{y+\ell} - \hat{Y}_t(\ell)}{\sqrt{\text{var}(e_t(\ell))}} \leq z_{1-\alpha/2} \right) = 1 - \alpha$$

resultando que

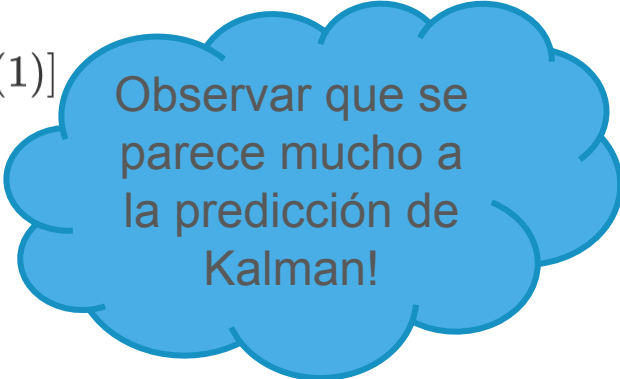
$$Y_{t+\ell} \in [\hat{Y}_t(\ell) - z_{1-\alpha/2} \sqrt{\text{var}(e_t(\ell))}, \hat{Y}_t(\ell) + z_{1-\alpha/2} \sqrt{\text{var}(e_t(\ell))}]$$

Actualizando las predicciones ARIMA

¿Cómo actualizamos las predicciones realizadas una vez que nos llega una nueva muestra de la serie?

Supongamos que tengo la predicción $\hat{Y}_t(\ell + 1)$, que es la predicción del instante $t + \ell + 1$, basados en las observaciones hasta tiempo t . Supongamos que de pronto me llega información acerca del instante $t + 1$. Luego, quiero actualizar mi estimación de $Y_{t+\ell+1}$: $\hat{Y}_{t+1}(\ell)$

$$\hat{Y}_{t+1}(\ell) = \hat{Y}_t(\ell + 1) + \psi_\ell [Y_{t+1} - \hat{Y}_t(1)]$$



Observar que se parece mucho a la predicción de Kalman!

Predicción de series transformadas

Pongamos el ejemplo de la transformación logarítmica. En este caso tenemos Y_t la serie original y $W_t = \log(Y_t)$. Se puede demostrar que

$$\mathbb{E}[Y_{t+\ell} | Y_1, \dots, Y_t] \geq e^{\mathbb{E}[W_{t+\ell} | Y_1, \dots, Y_t]}$$

con lo cual el estimador naive $\hat{Y}_t(\ell) = e^{\hat{W}_t(\ell)}$ no es el de ECM. Si W_t tiene distribución normal, vale que predictor de ECM es

$$\hat{Y}_t(\ell) = e^{\hat{W}_t(\ell) + 0.5 \text{var}(e_t(\ell))}$$

Próximas clases

Clase 5	Predicciones
Clase 6	Aplicaciones. Análisis espectral
Clase 7	Heteroscedasticidad. Markov.
Clase 8	Evaluación final

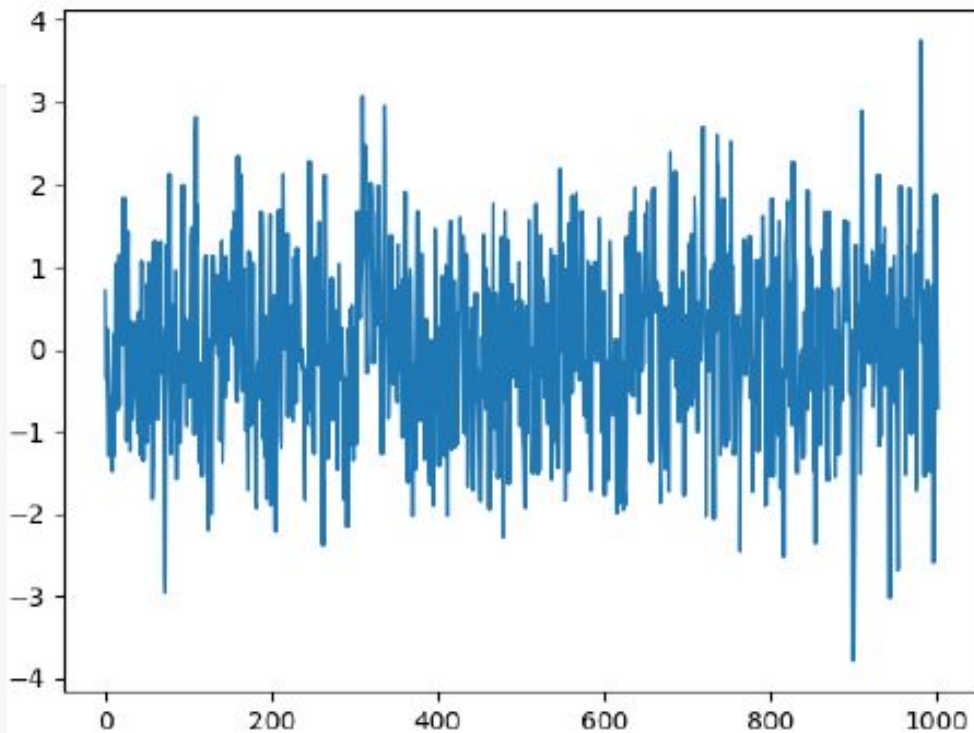
Ejemplos

Modelo de Promedio Móvil MA(1)

$$y_t = b_1 \epsilon_{t-1} + \epsilon_t$$

```
# MA(1)
N = 1000
b1 = 0.25
e_t0 = np.random.normal(0,1)
e_t1 = np.random.normal(0,1)
y = np.append(e_t0, b1*e_t0+e_t1) #

for i in range(N):
    e_t0 = e_t1
    e_t1 = np.random.normal(0,1)
    y = np.append(y, b1*e_t0+e_t1)
```



Modelo de Promedio Móvil MA(1)

$$y_t = b_1 \epsilon_{t-1} + \epsilon_t$$

Predict

```
ma1 = ARIMA(y, order=(0, 0, 1))
```

```
ma1_res = ma1.fit()
```

```
print(ma1_res.summary())
```

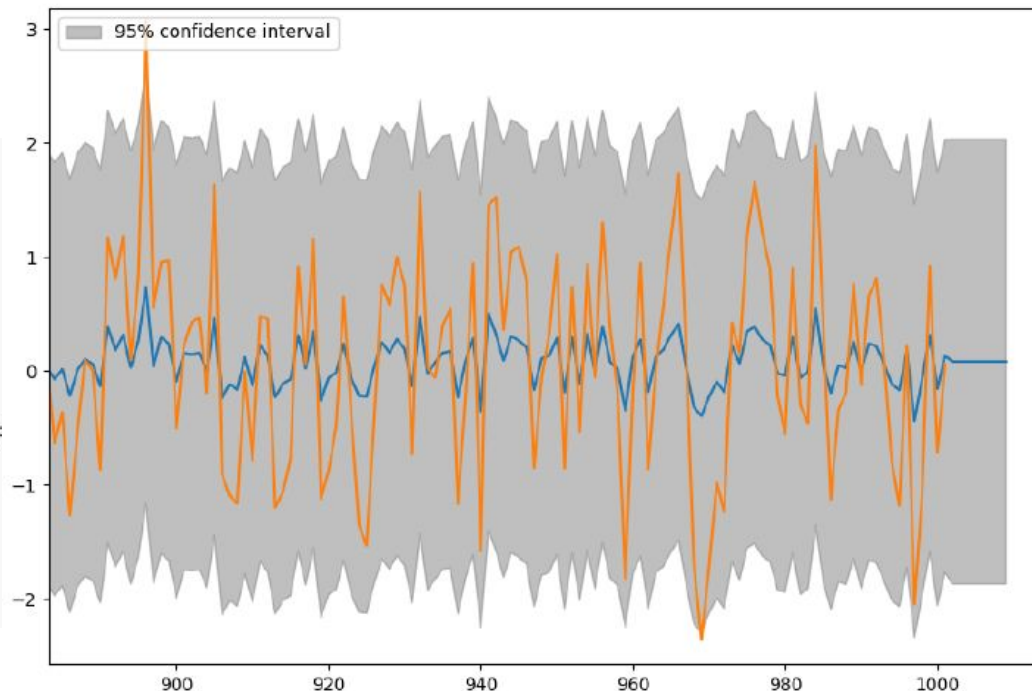
```
fig, ax = plt.subplots(figsize=(10, 8))
```

```
fig = plot_predict(ma1_res, start=1, end=
```

```
plt.plot(y)
```

```
legend = ax.legend(loc="upper left")
```

```
plt.show()
```



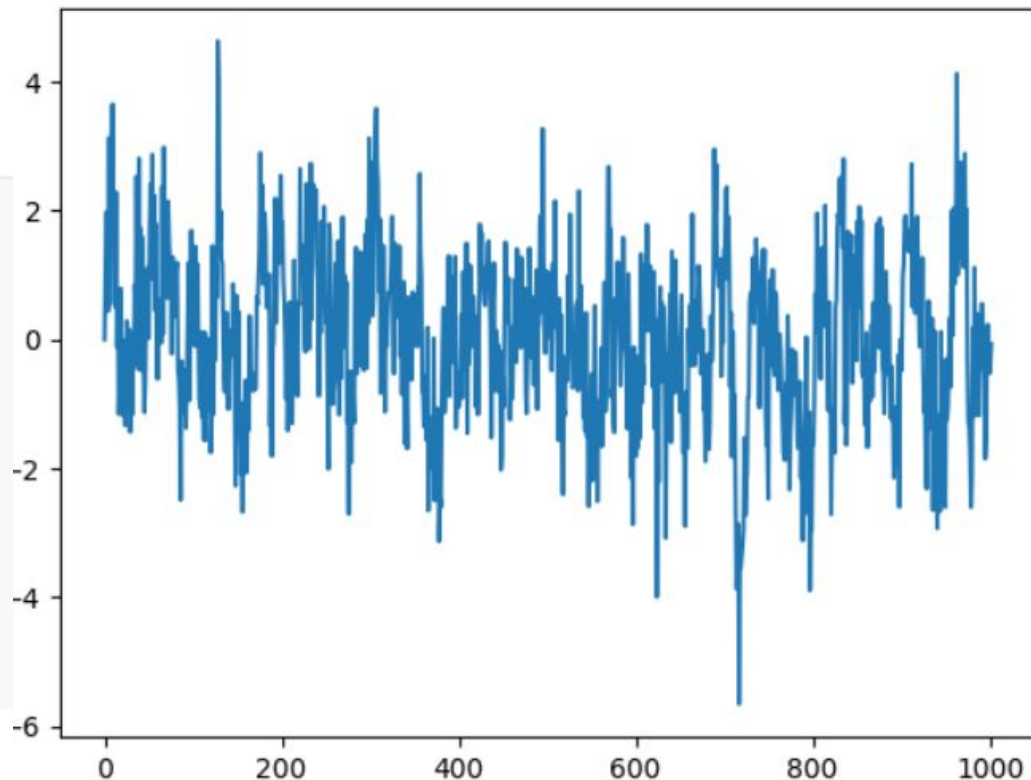
$$\gamma_0 = (1 + \theta_1^2 + \theta_2^2 + \dots + \theta_q^2) \sigma_e^2$$

Modelo Autoregresivo AR(2)

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \epsilon_t$$

```
# AR(2)
N=1000
a1=0.4
a2=0.35
x=np.arange(2)
e_t=np.random.normal(0,1)
y=np.append(x,a1*x[1]+a2*x[0]+e_t)

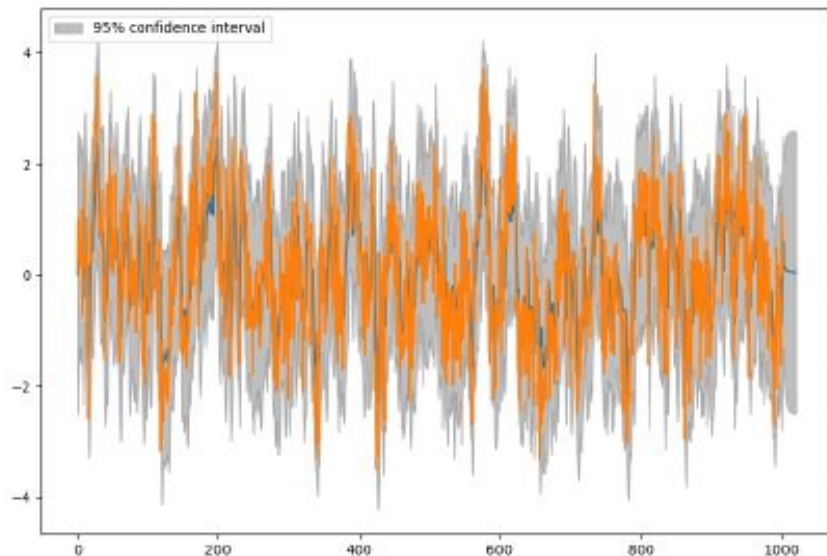
for i in range(N):
    e_t=np.random.normal(0,1)
    y=np.append(y,a1*y[-1]+a2*y[-2]+e_t)
```



Modelo Autoregresivo AR(2)

```
# Predict
l=20
ar2 = ARIMA(y, order=(2, 0, 0))
ar2_res = ar2.fit()
print(ar2_res.summary())

fig, ax = plt.subplots(figsize=(10, 8))
fig = plot_predict(ar2_res, start=0, end=N+l, ax=ax)
plt.plot(y)
legend = ax.legend(loc="upper left")
plt.show()
```



→ ver resultados del modelo

Modelo ARMA(2,1)

```
# Predict
```

```
y_arma = ARIMA(y, order=(2, 0, 1))
```

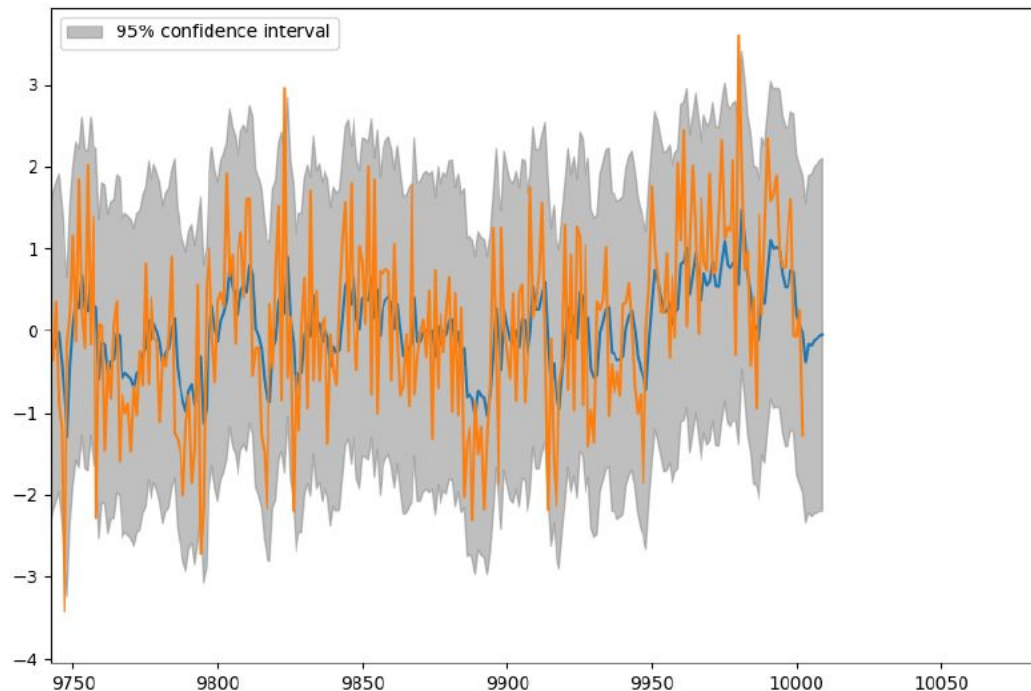
```
y_arma_res = y_arma.fit()
```

```
print(y_arma_res.summary())
```

```
fig, ax = plt.subplots(figsize=(10, 8))
```

```
fig = plot_predict(y_arma_res, start=1, end=N
```

```
plt.plot(y)
```



Modelo ARIMA(2,1,1)

```
y=np.append(y,a1*y[-1]+a2*y[-2]+ b1*e_0 + e_t)
```

```
t = np.arange(int(N*0.01), step=0.01)  
y=y[2:N+2]
```

```
y = y+t
```

```
# Predict
```

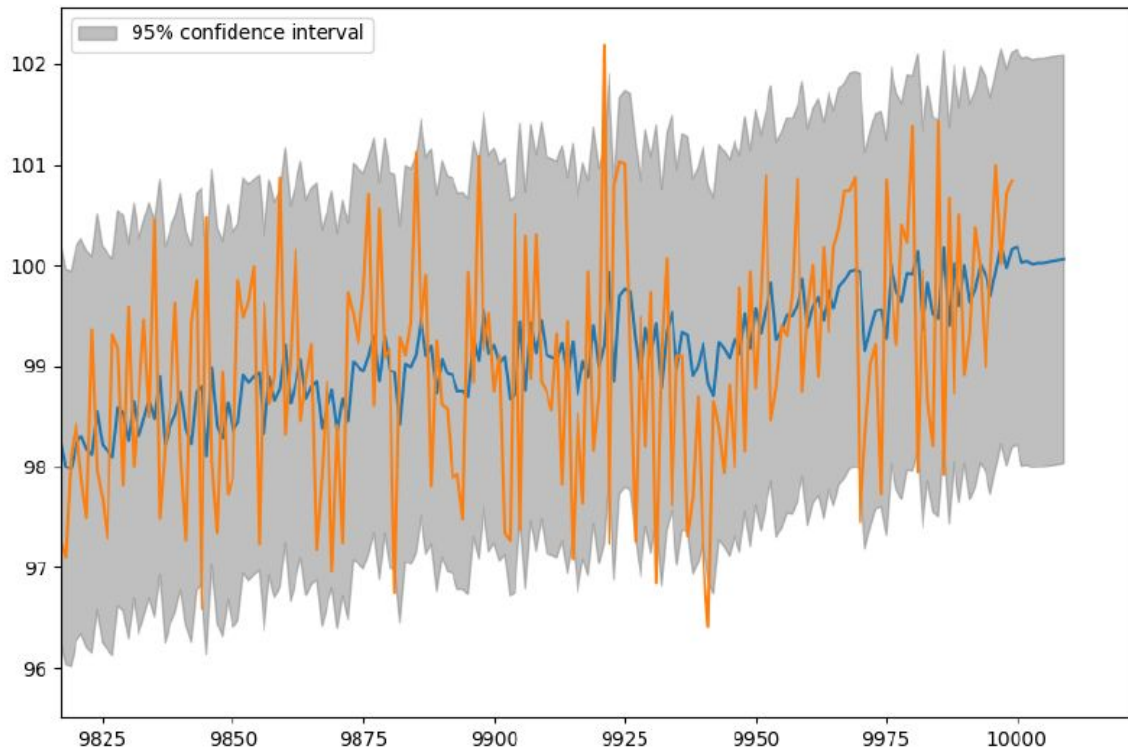
```
y_arma = ARIMA(y, order=(2, 1, 1), trend='t')  
y_arma_res = y_arma.fit()  
print(y_arma_res.summary())
```

```
fig, ax = plt.subplots(figsize=(10, 8))  
fig = plot_predict(y_arma_res, start=1,end=N+10, ax=ax)
```

Modelo ARIMA(2,1,1)

```
1 SARIMAX Results
2 =====
3 Dep. Variable: y No. Observations: 10000
4 Model: ARIMA(2, 1, 1) Log Likelihood: -14245.820
5 Date: jue, 18 nov 2021 AIC: 28501.641
6 Time: 19:40:41 BIC: 28537.692
7 Sample: 0 HQIC: 28513.844
8 - 10000
9 Covariance Type: opg
10 =====
11 coef std err z P>|z| [0.025 0.975]
12 -----
13 x1 0.0100 9.94e-06 1006.340 0.000 0.010 0.010
14 ar.L1 0.0335 0.010 3.470 0.001 0.015 0.052
15 ar.L2 0.2474 0.010 25.610 0.000 0.228 0.266
16 ma.L1 -0.9996 0.001 -1875.605 0.000 -1.001 -0.999
17 sigma2 1.0100 0.014 70.300 0.000 0.982 1.038
18 =====
19 Ljung-Box (L1) (Q): 4.66 Jarque-Bera (JB): 0.24
20 Prob(Q): 0.03 Prob(JB): 0.89
21 Heteroskedasticity (H): 0.98 Skew: 0.00
22 Prob(H) (two-sided): 0.61 Kurtosis: 2.98
23 =====
```

Modelo ARIMA(2,1,1)



#ARIMA

N=10000

a1=0.35

$$a_2 = 0.25$$

`b1 = -0.35`

```
t = np.arange(int(N*0.01), step=0.01)
```

```

10 =====
11 |         |         |         |         | coef
12 -----
13 x1              0.0100
14 ar.L1           0.0335
15 ar.L2           0.2474
16 ma.L1          -0.9996
17 sigma2          1.0100
18 =====

```

Modelos con tendencia determinística

- Procesos estocásticos con tendencia determinística
- Si puedo extraer la tendencia con un modelo analítico, entonces voy a poder usar modelos de forma adecuada

$$Y_t = X_t + \mu_t$$

$$\mu_t := \begin{cases} \text{Constante} & \mu_t = \mu \\ \text{Lineal} & \mu_t = \beta_0 + \beta_1 t \\ \text{Cuadrática} & \mu_t = \beta_0 + \beta_1 t + \beta_2 t^2 \\ \text{Cíclica} & \mu_t = \mu_{t-T} \\ \text{Senoidal} & \mu_t = \beta \cos(\omega t + \phi) \end{cases}$$

→ ver extensión al modelo SARIMA

Conclusiones

- Los modelos **AR**, **MA**, **ARMA** se basan en procesos estacionarios y están bien estudiados.
- Si un proceso no estacionario pueda ser **diferenciado** 'i' veces y volverse estacionario, entonces podemos usar la extensión del modelo **ARIMA**
- Para procesos con **tendencia constante o lineal**, el framework funciona normalmente bien con **ARIMA.fit()**
- Con la misma idea, procesos a los que se les puede extraer una tendencia determinística conviene tratarlos con modelos conocidos.
- Vimos que **Cuadrados Mínimos** puede ser útil para ajustar tendencias determinísticas y obtener los coeficientes del modelo
- La extensión **SARIMA** usa estos métodos con la misma idea de extraer componentes que se pueden modelar por descomposición
- Los **coeficientes** son **todo** a la hora de modelar para predecir. Una buena predicción se da cuando el modelo es el adecuado y los coeficientes están bien ajustados.

Algunas referencias de otros métodos de
predicción

Bibliografía

1. [Godfrey2017] Neural Decomposition of Time-Series Data
2. [Chrobok2005] Theory and Application of Advanced Traffic Forecast Methods
3. [Hochreiter1997] LONG SHORT-TERM MEMORY
4. [Gers2000] Learning to Forget: Continual Prediction with LSTM

Bibliografía

