Predicción de Energías Renovables como Series Temporales

Predicción

Contents

Regresión Lineal

Repaso: regresión lineal (I)

- También conocido como **mínimos cuadrados ordinarios** (OLS – Ordinary Least-Squares).
- Método más común para hacer aprendizaje estadístico.
- Variables predictivas (X), salida (y):

$$\hat{y} = \beta_0 + \beta_1 X$$

- Entrenamiento: buscar β_0 y β_1.
  - Función de pérdida: mínimos cuadrados.

$$L(X) = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2.$$

  - Minimizamos la función de pérdida.

Repaso: regresión lineal (II)

- **Ventajas:**
  - Poco costoso computacionalmente.
  - Fácil de interpretar.

- **Inconvenientes:**
  - Es muy rígido.
  - Probablemente no encuentre la relación real entre X e y.

Modelos clásicos de predicción de ST

Modelos clásicos de predicción de ST

- Modelos estadísticos sencillos.
- Se basan principalmente en la búsqueda de relaciones lineales.
- Bien aplicados pueden ser la mejor solución para predecir una ST.
- Modelos:
  - **Familia ARIMA**: AR, MA, ARMA, ARIMA, SARIMA
  - **Exponential Smoothing**: simple, damped, holt-winters

Horizontes de predicción

Horizonte de predicción

El **horizonte de predicción** es el número de intervalos de tiempo que llevamos sin conocer información real.

Train_1	Train_2	$\cdots$	Train_N	Test_1	Test_2	$\cdots$	Test_M
				$h{=}0$	$h{=}1$	$\cdots$	$h{=}M$

- Es importante decidir el horizonte de predicción para el cual nos interesa predecir.

Horizontes de predicción

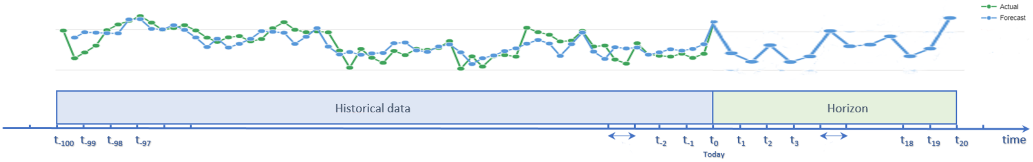
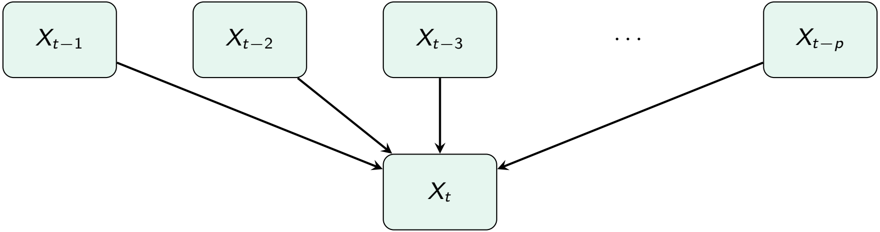


Imagen obtenida de https://www.sapanalytics.cloud/time-series-forecasting-smart-predict/

Modelos Autorregresivos: AR(p) (I)

- Modeliza el valor siguiente de la serie como una función lineal de las **observaciones** anteriores.
- Son modelos estacionarios.
- Parámetro: orden **p**
  - p es el número de retrasos que cogemos hacia atrás.



Modelos Autorregresivos: AR(p) (II)

- Regresión lineal sobre los retrasos.
- Modelo:

$$\boxed{X_t = C + \sum_{j=1}^{p} \phi_j X_{t-j} + \epsilon_t \ .}$$

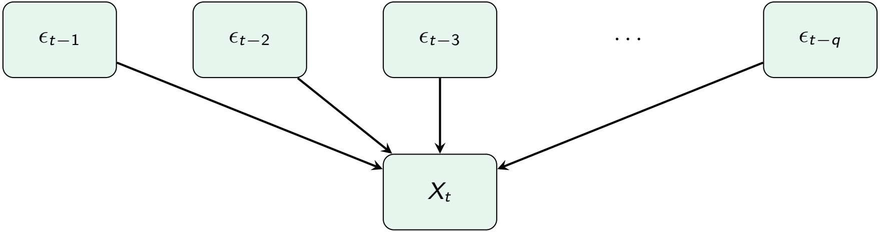
  donde
  - C es una constante.
  - ϕ_i coeficientes de la regresión lineal
  - ϵ_i ruido blanco

Modelos Autorregresivos: AR(p) (III)

- Elección del orden p:
  - ACF.
  - PACF (sobre esta es más sencillo ver patrones).
  - Búsqueda del mejor parámetro en un conjunto de validación.
- Búsqueda de coeficientes ϕ_i:
  - Búsqueda por mínimos cuadrados (los que minimizan el error cuadrático).

Modelos de Medias Móviles: MA(q) (I)

- Modelo de regresión sobre los **errores** de la serie temporal.
- Son siempre modelos estacionarios.
- Parámetro: orden **q**
  - q es el número de retrasos que cogemos de ruido hacia atrás.



Modelos de Medias Móviles: MA(q) (II)

- Regresión lineal sobre los errores previos.
- Modelo:

$$\boxed{X_t = \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \ .}$$

  donde

  - ϵ_k son variables aleatorias i.i.d. que siguen una normal (ruido blanco).
  - θ_i son los coeficientes de la regresión.

Modelos de Medias Móviles: MA(q) (III)

- Elección del orden q:
  - La gráfica ACF nos puede dar pistas.
  - Búsqueda del mejor parámetro en un conjunto de validación.
- Búsqueda de coeficientes θ_i:
  - Es más complicado que en AR, porque no vemos los términos de error previos.
  - Se suelen usar procedimientos no lineales iterativos.

Modelos Autorregresivos de Medias Móviles: ARMA(p, q) (I)

- Modeliza el siguiente valor de la serie como función lineal de las observaciones y los errores residuales.
- Combina **AR** y **MA**.
- Parámetros:
  - **p**: orden de AR.
  - **q**: orden de MA.
- Funciona para series sin tendencias ni componentes estacionales (ST estacionarias).

Modelos Autorregresivos de Medias Móviles: ARMA(p, q) (II)

- Regresión lineal sobre los retrasos de la serie y los errores previos.
- Modelo:

$$\boxed{X_t = C + \sum_{j=1}^{p} \phi_j X_{t-j} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \ .}$$

  donde
  - ϵ_k, son variables aleatorias i.i.d, obtenidas a partir de una gaussiana.
  - X_{t-k} son las observaciones anteriores de la serie.
  - ϕ_k y θ_k son los coeficientes de autorregresión y de medias móviles respectivamente.
  - C es una constante.

Modelos Autorregresivos de Medias Móviles: ARMA(p, q) (III)

- Estimar el orden:
  - ACF, PACF (subjetivo, vemos a continuación cómo estimarlo)
  - Proceso de validación.
- Búsqueda de coeficientes:
  - Estimación por mínimos cuadrados.

Función en Python

statsmodel

```
statsmodels.tsa.statespace.sarimax.SARIMAX(endog, exog=None, order=(1, 0, 0),
    seasonal_order=(0, 0, 0, 0), trend=None, measurement_error=False,
    time_varying_regression=False, mle_regression=True, simple_differencing=False,
    enforce_stationarity=True, enforce_invertibility=True, hamilton_representation=False,
    concentrate_scale=False, trend_offset=1, use_exact_diffuse=False, dates=None,
    freq=None, missing='none', validate_specification=True, **kwargs)
```

endog La serie temporal.

order El orden del modelo, que en nuestro caso será **(p,0,q)**.

Validación de modelos de ST

Método de Box-Jenkins

Asumimos que si el proceso es estacionario se puede aproximar por ARMA.
Proceso iterativo, según se reciben nuevos datos:

1. **Identificación**. Elegir un submodelo (definir p,q) que resuma los datos.
   - AR: si en ACF hay una bajada gradual tras un tiempo p, que en PACF es una caída brusca.
   - MA: si en ACF tras un tiempo q hay una caída brusca.
   - AR+MA: si presenta los dos comportamientos
2. **Estimación**. Búsqueda de coeficientes.
   - Método numérico para minimizar la función de pérdida.
   - Lo gestiona el método elegido en cuestión.
   - Ejemplos: ajuste por mínimos cuadrados, por máxima verosimilitud, BFGS,...
3. **Comprobación**. Ver si el modelo se adecúa a los datos.
   - Overfitting
   - Los errores residuales deben seguir siendo gaussianos e independientes:
     - Usar histogramas, diagramas de densidad,...
     - ACF y PACF sobre los residuos.

Método de Box-Jenkins: fase de Identificación

	MA(q)	**AR(p)**	**ARMA(p,q)**
ACF	Primeros q coefs significativos. Caída brusca a 0.	Decrecimiento exponencial rápido u ondas sinusoidales.	Comportamiento irregular, con q picos. Decaimiento posterior.
PACF	Decrecimiento exponencial rápido u ondas sinusoidales.	Primeros p coefs significativos. Caída brusca a 0.	Decrece paulatinamente.

Validación de modelos de ST

- En ST, como en cualquier modelo de Aprendizaje Automático, nos interesa validar la definición del modelo sobre datos no vistos en el entrenamiento $\Rightarrow$ **conjunto de validación**.
- k-CV no se puede utilizar directamente en ST.
  - Perdemos la estructura temporal!
- **Backtesting** es el nombre que se utiliza en ST para el proceso de validación.

Tipos de backtesting (I)

División del train en train-validation

- Útil cuando tenemos muchos datos $\Rightarrow$ train-validation representativo.
- Fijamos un porcentaje para la división: 50-50, 60-40,...

División del train en múltiples conjuntos de train-validation

- Más costoso computacionalmente.
- Resultados más robustos.
- El conjunto de validación en todas las divisiones debe tener el mismo tamaño.

Tipos de backtesting (II)

Validación walk-forward

- Para validar modelos que se van a actualizar con frecuencia.
- Vemos la partición de train como una ventana.
- Establecemos:
  - Tamaño de la ventana inicial.
  - Si la ventana es corrida o se va ampliando.
- Es el método más robusto, pero evalúa muchos modelos (equivalente a LOOCV).

Notebook

Predicción utilizando Modelos Clásicos

Modelo Autorregresivo Integrado de Medias Móviles

Modelo Autorregresivo Integrado de Medias Móviles: ARIMA(p,d,q) (I)

- Generalización de ARMA.
- Combina **AR** y **MA**, pero calculados sobre la **serie de diferencias** de los datos indicados.
- Parámetros:
  - p: orden de AR.
  - d: grado de diferenciación.
  - q: orden de MA.
- Estimamos el orden por backtesting.

Modelo Autorregresivo Integrado de Medias Móviles: ARIMA(p,d,q) (II)

- Modelo:

$$\boxed{D_t = C + \sum_{j=1}^{p} \phi_j D_{t-j} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \ .}$$

  donde

  - ϵ_k, son variables aleatorias i.i.d, obtenidas a partir de una gaussiana.
  - D_{t-k} son los pasos anteriores de la serie de diferencias.
  - ϕ_k y θ_k son los coeficientes de autorregresión y de medias móviles respectivamente.
  - C es una constante.

Función en Python

statsmodel

```
statsmodels.tsa.statespace.sarimax.SARIMAX(endog, exog=None, order=(1, 0, 0),
    seasonal_order=(0, 0, 0, 0), trend=None, measurement_error=False,
    time_varying_regression=False, mle_regression=True, simple_differencing=False,
    enforce_stationarity=True, enforce_invertibility=True, hamilton_representation=False,
    concentrate_scale=False, trend_offset=1, use_exact_diffuse=False, dates=None,
    freq=None, missing='none', validate_specification=True, **kwargs)
```

endog La serie temporal.

order El orden del modelo, que en nuestro caso será **(p,d,q)**.

Notebook

ARIMA

Modelo Autorregresivo Integrado de Medias Móviles Estacional

Modelo Autorregresivo Integrado de Medias Móviles Estacional: SARIMA(p,d,q)(P,D,Q,m) (I)

- Extensión de ARIMA que considera además periodos estacionales.
- Parámetros:
  - **Tendencias**:
    - p: orden de AR.
    - d: grado de diferenciación.
    - q: orden de MA.
  - **Estacionalidad**:
    - P: orden de AR para la parte estacional.
    - D: grado de diferenciación estacional.
    - Q: orden de MA para la parte estacional.
    - m: número de pasos para un periodo estacional.

Modelo Autorregresivo Integrado de Medias Móviles Estacional: SARIMA(p,d,q)(P,D,Q,m) (II)

- **Problema**: son 7 parámetros!!
- Hay que hacer una búsqueda en rejilla, para encontrar la mejor configuración del modelo.
  - Mejor no hacerlo en un notebook.
  - Intentar agilizar cálculos utilizando varias CPUs en la medida de lo posible.

Función en Python

statsmodel

```
statsmodels.tsa.statespace.sarimax.SARIMAX(endog, exog=None, order=(1, 0, 0),
    seasonal_order=(0, 0, 0, 0), trend=None, measurement_error=False,
    time_varying_regression=False, mle_regression=True, simple_differencing=False,
    enforce_stationarity=True, enforce_invertibility=True, hamilton_representation=False,
    concentrate_scale=False, trend_offset=1, use_exact_diffuse=False, dates=None,
    freq=None, missing='none', validate_specification=True, **kwargs)
```

endog La serie temporal.

order El orden del modelo, que en nuestro caso será **(p,d,q)**.

seasonal_order El orden de la parte estacional,es decir, **(P,D,Q,m)**.

Notebook

SARIMA

Variables Exógenas

Modelos Autorregresivos con Variables Exógenas

- A veces la ST no es suficiente para predecir valores futuros y necesitamos información extra.
- Este es el caso de la predicción de **energías renovables** como la eólica y la solar, donde conocer la meteorología (que es muy cambiante) ayuda a la predicción.
- Todos los modelos anteriores admiten variables exógenas:
  - **ARX**, **MAX**, **ARMAX**, **ARIMAX**, **SARIMAX**.

Función en Python

statsmodel

```
statsmodels.tsa.statespace.sarimax.SARIMAX(endog, exog=None, order=(1, 0, 0),
    seasonal_order=(0, 0, 0, 0), trend=None, measurement_error=False,
    time_varying_regression=False, mle_regression=True, simple_differencing=False,
    enforce_stationarity=True, enforce_invertibility=True, hamilton_representation=False,
    concentrate_scale=False, trend_offset=1, use_exact_diffuse=False, dates=None,
    freq=None, missing='none', validate_specification=True, **kwargs)
```

endog La serie temporal.

exog Las variables exógenas.

order El orden del modelo, que en nuestro caso será **(p,d,q)**.

seasonal_order El orden de la parte estacional,es decir, **(P,D,Q,m)**.

Notebook

SARIMAX

Modelos de suavizado exponencial

Modelos de Suavizado Exponencial (ETS)

- **Semejanzas** con modelos de la familia ARIMA:
  - Predicción es una suma ponderada de las observaciones anteriores.
- **Diferencias** con modelos de la familia ARIMA:
  - Se utiliza un decaimiento exponencial de los pesos.

- Diferentes modelos:
  - Suavizado exponencial simple.
  - Suavizado exponencial doble.
  - Suavizado exponencial triple.

Suavizado Exponencial Simple (STS)

- Modelos aptos para ST sin tendencia ni estacionalidad.
- Parámetros:
  - Coeficiente de suavizado: $\boxed{\alpha} \in [0, 1]$.
    - $\alpha = 1$, aprendizaje rápido,
    - $\alpha = 0$, aprendizaje lento.
- Definición del modelo:

$$
\begin{aligned}
s_0 &= x_0 \\
s_t &= \alpha x_t + (1 - \alpha) s_{t-1}.
\end{aligned}
$$

Suavizado Exponencial Doble

- Modelos que consideran explícitamente tendencias en la ST.
- Parámetros:
  - Coeficiente de suavizado: $\boxed{\alpha} \in [0, 1]$.
  - Coeficiente de suavizado de la tendencia: $\boxed{\beta} \in [0, 1]$.
- Definición del modelo (siguiendo un modelo aditivo):

$$\begin{aligned} F_{t+m} &= s_t + mb_t \\ s_1 &= x_1 \; ; b_1 = x_1 - x_0 \\ s_t &= \alpha x_t + (1-\alpha)(s_{t-1} + b_{t-1}) \\ b_t &= \beta(s_t - s_{t-1}) + (1-\beta) b_{t-1} \; . \end{aligned}$$

Suavizado Exponencial Triple (Holt-Winters ES)

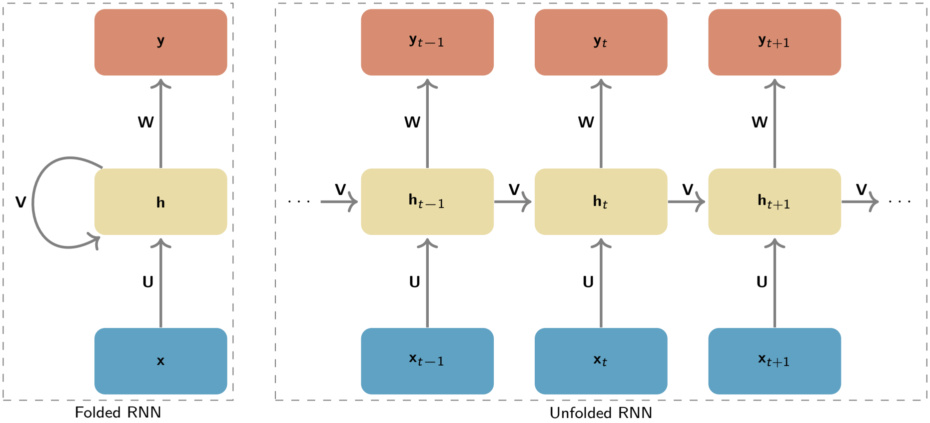
- Modelos que consideran además estacionalidad en la ST.
- Son modelos adaptativos que modelan variaciones en el tiempo de la media, la tendencia y la estacionalidad.
- Parámetros:
  - Coeficiente de suavizado: $\boxed{\alpha} \in [0,1]$.
  - Coeficiente de suavizado de la tendencia: $\boxed{\beta} \in [0,1]$.
  - Coeficiente de suavizado de la estacionalidad: $\boxed{\gamma} \in [0,1]$.

Long Short Term Memory (LSTM)

Redes Neuronales Recurrentes (I)

- La mayoría de modelos de ML asumen independencia entre los ejemplos.
- En la predicción de series temporales esta hipótesis no es cierta, y los patrones son muy dependientes del contexto.
- Las **Redes Neuronales Recurrentes (RNNs)** permiten conexiones hacia atrás entre neuronas, paliando este problema.
  - Valores producidos previamente por la red se introducen como entradas de nuevo en la red.
- Estas redes presentan dependencias entre los pesos por lo que requieren un entrenamiento especial (Back-propagation Through Time).

Redes Neuronales Recurrentes (II)



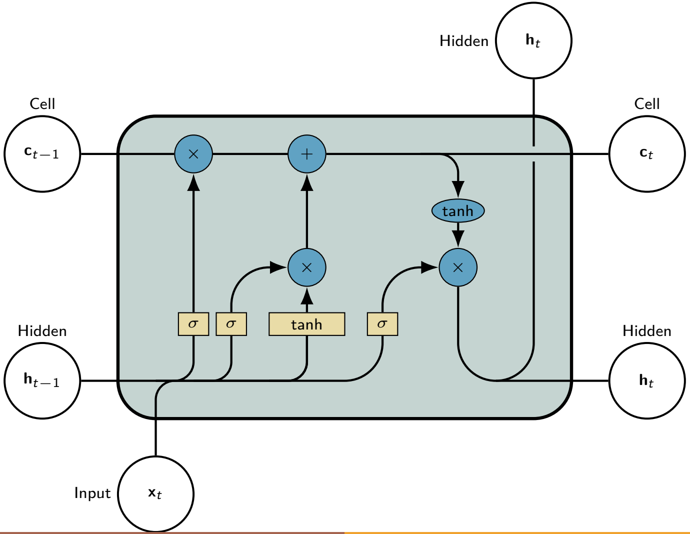
Long Short Term Memory (I)

- Una **unidad de memoria a largo-corto plazo (LSTM)** es una componente especializada, diseñada para retener información durante periodos de tiempo mas largos que las RNNs.
- Las LSTMs tienen varias puertas de control de la red.
- Estas unidades suelen estar formadas por:

Cell	Es la memoria de la unidad LSTM.
Input Gate	Controla la influencia de la nueva entrada en la celda.
Forget Gate	Controla cuánto se retiene un valor en la celda.
Output Gate	Controla cuánto se usa un valor en la celda para calcular la unidad de salida.

- Existen conexiones hacia dentro y hacia fuera de las puertas LSTM (algunos son recurrentes).
- Ventajas:
  - Los pesos de las puertas se aprenden durante el entrenamiento.
  - La red aprende qué patrón se debe retener.

Long Short Term Memory (II)



Notebook

LSTM

Resumen

Resumen (I)

- **Análisis** de la ST:
  - Visualización.
  - Análisis estadístico de los datos.
  - Interpolación de huecos en la serie.
  - Análisis de autocorrelaciones y autocorrelaciones parciales.
- **Predicción** de la ST:
  - Dividir la serie en entrenamiento y test.
  - Para buscar los hiperparámetros: búsqueda en rejilla sobre backtesting.
  - Decidir el horizonte de predicción que nos interesa.

Resumen (II)

- Modelos de predicción vistos:
  - Familia de modelos ARIMA:
    - **AR**: ST estacionarias
    - **MA**: ST estacionarias
    - **ARMA**: ST estacionarias
    - **ARIMA**: ST con tendencia
    - **SARIMA**: ST con tendencia y estacionalidad
    - **SARIMAX**: modelos anteriores con variables exógenas
  - Familia de modelos de suavizado exponencial:
    - **Suavizado exponencial simple**: ST estacionarias
    - **Suavizado exponencial doble**: ST con tendencia
    - **Holt-Winters suavizado exponencial**: ST con tendencia y estacionalidad
  - **LSTM**

Gracias por vuestra atención.