Suavizamiento Exponencial Triple de Holt–Winters con Selección de Parámetros por Validación sin Fuga de Información

(Borrador técnico)

1. Planteamiento del problema y notación

Sea

$$\{P_t\}_{t=1}^N$$

una serie de precios diarios ajustados de un activo financiero (acción, índice, tipo de cambio, cripto, etc.). Definimos la serie de *log–precios*

$$Z_t = \log P_t, \qquad t = 1, \dots, N.$$

Por qué usar log–precios

Trabajar con $Z_t = \log P_t$ en lugar de P_t tiene varias ventajas:

- Los rendimientos simples se aproximan por diferencias de log:

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}} \approx \log P_t - \log P_{t-1} = Z_t - Z_{t-1}.$$

- Transformaciones multiplicativas en precios se vuelven aditivas en log–precios, lo cual es coherente con el modelo aditivo de Holt–Winters.
- Frecuentemente los errores en Z_t son más cercanos a normales y de varianza aproximadamente constante que los errores en P_t.

Partición en entrenamiento, validación y prueba

Sea N el número total de observaciones. Fijamos proporciones

$$0 < f_{\text{train}}, f_{\text{val}} < 1, \qquad f_{\text{train}} + f_{\text{val}} < 1,$$

y definimos los tamaños enteros

$$N_{\text{train}} = \lfloor f_{\text{train}} N \rfloor, \qquad N_{\text{val}} = \lfloor f_{\text{val}} N \rfloor, \qquad N_{\text{test}} = N - N_{\text{train}} - N_{\text{val}}.$$

Partimos así la serie

$$Z_1, \dots, Z_N$$

en tres segmentos consecutivos:

$$\begin{aligned} \text{Entrenamiento: } & Z_1, \dots, Z_{N_{\text{train}}}, \\ \text{Validación: } & Z_{N_{\text{train}}+1}, \dots, Z_{N_{\text{train}}+N_{\text{val}}}, \\ \text{Prueba: } & Z_{N_{\text{train}}+N_{\text{val}}+1}, \dots, Z_N. \end{aligned}$$

Definición 1.1 (Sin fuga de información)**.** *Diremos que un procedimiento de ajuste y selección de hiperparámetros no tiene* fuga de información *si:*

1. *Los parámetros del modelo (en este caso los estados de Holt–Winters: nivel, tendencia y estacionalidad) se actualizan exclusivamente con el segmento de entrenamiento.*
2. *El segmento de validación se usa únicamente para evaluar pronósticos verdaderamente fuera de muestra.*
3. *El segmento de prueba se mantiene completamente fuera de la selección de hiperparámetros y sólo se utiliza para una evaluación final.*

Nuestro objetivo es ajustar un modelo de suavizamiento exponencial triple (Holt–Winters aditivo) sobre $\{Z_t\}$, elegir:

$$s = \alpha \in (0,1), \quad m \in \{m_1, \ldots, m_K\}$$

mediante el error cuadrático medio (MSE) en validación, y producir pronósticos para la parte de validación y prueba, todo sin fuga de información.

2. Modelo Holt–Winters aditivo (triple suavizamiento exponencial)

2.1. Descomposición aditiva

El modelo Holt–Winters aditivo postula que el log–precio Z_t puede escribirse como:

$$Z_t = \ell_t + b_t + s_{p_t} + \varepsilon_t, \qquad t = 1, \ldots, N,$$

donde:

- ℓ_t es el **nivel** (tendencia lenta) en el tiempo t.
- b_t es el **componente de crecimiento** o *trend* local (pendiente).
- s_{p_t} es el **componente estacional aditivo** en la posición p_t dentro del ciclo estacional.
- ε_t es un término de error (ruido) con media cero.
- p_t es el índice estacional:

  $$p_t = t \bmod m \in \{0, 1, \ldots, m-1\},$$

  donde m es la longitud del periodo estacional (por ejemplo, $m = 5$ para patrón semanal en días hábiles).

Interpretación: en cada instante t, la serie es la suma de un nivel global, una tendencia lineal local y una corrección estacional que depende únicamente de la posición dentro del ciclo de longitud m.

2.2. Parámetros de suavizamiento

En una versión simplificada usamos un sólo hiperparámetro de suavizamiento $\alpha \in (0,1)$ y lo reutilizamos para:

- Actualización del nivel,
- Actualización del componente de tendencia,
- Actualización del componente estacional.

Esto implica fijar

$$\beta = \alpha, \qquad \gamma = \alpha,$$

aunque en versiones más generales se podría tratar a β y γ como parámetros independientes.

3. Inicialización de nivel, tendencia y estacionalidad

Sea m un candidato a longitud estacional. Suponemos que el conjunto de entrenamiento tiene al menos $2m$ observaciones:

$$N_{\text{train}} \geq 2m,$$

de modo que existe al menos dos ciclos completos dentro del entrenamiento.

Definición 3.1 (Inicialización con promedios estacionales)**.** *Sea*

$$Z_1, \ldots, Z_{N_{train}}$$

la serie de entrenamiento y m la longitud estacional. Definimos:

- $n_{seas} = \lfloor N_{train}/m \rfloor$ *número de estaciones completas;*
- *elegimos un número n_{use} de estaciones a usar para inicialización, típicamente*

  $$n_{use} = \text{máx}\{2, \text{mín}(n_{seas}, 8)\};$$

- *Para cada estación $k = 0, \ldots, n_{use} - 1$ tomamos el bloque*

  $$\{Z_{km+1}, \ldots, Z_{km+m}\}$$

  y definimos el promedio de la k–ésima estación

  $$\bar{Z}^{(k)} = \frac{1}{m} \sum_{j=1}^{m} Z_{km+j}.$$

A partir de esto definimos:

- **Nivel inicial**:

  $$\ell_0 = \frac{1}{n_{\text{use}}} \sum_{k=0}^{n_{\text{use}}-1} \bar{Z}^{(k)}.$$

  Es el promedio de las estaciones iniciales.

- **Tendencia inicial** (crecimiento promedio por unidad de tiempo):

  $$b_0 = \frac{\bar{Z}^{(1)} - \bar{Z}^{(0)}}{m},$$

  es decir, la diferencia entre las medias de las primeras dos estaciones, dividida por m (número de pasos entre ellas). Si sólo hay una estación útil, podemos tomar $b_0 = 0$ como aproximación.

- **Índices estacionales iniciales** aditivos: para cada posición $j = 0, \ldots, m-1$ dentro del ciclo, tomamos
$$s_j = \frac{1}{n_{\text{use}}} \sum_{k=0}^{n_{\text{use}}-1} \left(Z_{km+(j+1)} - \bar{Z}^{(k)} \right).$$
Es decir, la desviación promedio de la observación en la posición j con respecto a la media de su estación.

Así obtenemos el vector inicial estacional
$$\boldsymbol{s}^{(0)} = (s_0, \ldots, s_{m-1}).$$

4. Actualización recursiva en el conjunto de entrenamiento

Trabajamos ahora únicamente con las observaciones de entrenamiento
$$Z_1, \ldots, Z_{N_{\text{train}}},$$
y mantenemos la *restricción clave*: las observaciones de validación y prueba no se usan para actualizar los estados.

4.1. Índice estacional

Para cada tiempo $t \geq 1$ definimos
$$p_t = (t-1) \bmod m \in \{0, \ldots, m-1\},$$
de forma que la observación Z_t corresponde a la posición p_t dentro del ciclo estacional.

4.2. Ecuaciones de Holt–Winters aditivo

Con parámetros $\alpha \in (0,1)$, $\beta = \alpha$, $\gamma = \alpha$, el algoritmo de actualización es:

$$\text{Nivel:} \quad \ell_t = \alpha\,(Z_t - s_{p_t}) + (1-\alpha)(\ell_{t-1} + b_{t-1}), \tag{1}$$
$$\text{Tendencia:} \quad b_t = \beta\,(\ell_t - \ell_{t-1}) + (1-\beta)\,b_{t-1}, \tag{2}$$
$$\text{Estacionalidad:} \quad s_{p_t} = \gamma\,(Z_t - \ell_t) + (1-\gamma)\,s_{p_t}. \tag{3}$$

Interpretación paso a paso:

- En (1), la nueva estimación del nivel ℓ_t es una combinación convexa de:
  - la observación *deseasonalizada* $Z_t - s_{p_t}$;
  - el nivel previo proyectado un paso adelante $\ell_{t-1} + b_{t-1}$.
- En (2), la nueva tendencia b_t se actualiza como combinación convexa de:
  - el *incremento* observado del nivel $\ell_t - \ell_{t-1}$;
  - la tendencia previa b_{t-1}.
- En (3), el índice estacional en la posición p_t se actualiza con:
  - la desviación $Z_t - \ell_t$ entre la observación y el nivel,
  - el valor estacional previo en esa misma posición.

4.3. Valores ajustados en entrenamiento

Definimos el *pronóstico un paso adelante* (ajuste in–sample) en el tiempo t como

$$\hat{Z}_t = \ell_{t-1} + b_{t-1} + s_{p_t}, \qquad t \geq 2.$$

Para $t = 1$ no hay historia anterior, así que se puede tomar

$$\hat{Z}_1 = Z_1$$

por conveniencia. De este modo se obtiene una serie ajustada

$$\hat{Z}_1, \ldots, \hat{Z}_{N_{\text{train}}}$$

que suaviza la serie original de entrenamiento.

5. Pronóstico fuera de muestra: validación y prueba

Sea $T_{\text{train}} = N_{\text{train}}$ el último índice de entrenamiento. Después de procesar $Z_1, \ldots, Z_{N_{\text{train}}}$ mediante (1)–(3), tenemos:

$$\ell_{T_{\text{train}}}, \quad b_{T_{\text{train}}}, \quad s_0, \ldots, s_{m-1}, \quad p_{T_{\text{train}}} = (T_{\text{train}} - 1) \bmod m.$$

Para un horizonte de pronóstico $h = 1, 2, \ldots,$ el modelo Holt–Winters aditivo propone el pronóstico de $Z_{T_{\text{train}}+h}$ como

$$\hat{Z}_{T_{\text{train}}+h|\text{train}} = \ell_{T_{\text{train}}} + h\, b_{T_{\text{train}}} + s_{p_{T_{\text{train}}+h}}, \qquad p_{T_{\text{train}}+h} := (p_{T_{\text{train}}} + h) \bmod m. \tag{4}$$

Definición 5.1 (Pronósticos de validación y prueba)**.** *Para cada $k = 1, \ldots, N_{val}$ (validación) y $h = k$, se define*

$$\hat{Z}_{N_{train}+k|train} = \ell_{N_{train}} + k\, b_{N_{train}} + s_{p_{N_{train}+k}},$$

y para el segmento de prueba, con $k' = 1, \ldots, N_{test}$ (horizontes continuando desde el final de validación),

$$\hat{Z}_{N_{train}+N_{val}+k'|train} = \ell_{N_{train}} + (N_{val} + k')\, b_{N_{train}} + s_{p_{N_{train}+N_{val}+k'}}.$$

En todos los casos, ningún dato de validación ni de prueba modifica los estados del modelo.

6. Función objetivo en validación y búsqueda en rejilla

6.1. Función de pérdida en validación para (α, m)

Fijemos un candidato de suavización $s = \alpha \in (0, 1)$ y un candidato de periodo estacional $m \in \mathbb{N}$. Procedemos así:

1. Inicializar $(\ell_0, b_0, \boldsymbol{s}^{(0)})$ usando únicamente $Z_1, \ldots, Z_{N_{\text{train}}}$ y m.
2. Aplicar las actualizaciones (1)–(3) de $t = 1$ a $t = N_{\text{train}}$.
3. Obtener $(\ell_{N_{\text{train}}}, b_{N_{\text{train}}}, \boldsymbol{s})$.
4. Generar los pronósticos de validación $\hat{Z}_{N_{\text{train}}+k|\text{train}}$ para $k = 1, \ldots, N_{\text{val}}$ usando (4).

Definimos los errores de validación para (α, m) como

$$e_k(\alpha, m) := Z_{N_{\text{train}}+k} - \hat{Z}_{N_{\text{train}}+k|\text{train}}, \qquad k = 1, \ldots, N_{\text{val}}.$$

Definición 6.1 (Error cuadrático medio de validación). *La función objetivo que queremos minimizar es el MSE de validación:*

$$J(\alpha, m) := \frac{1}{N_{val}} \sum_{k=1}^{N_{val}} \big(e_k(\alpha, m)\big)^2 = \frac{1}{N_{val}} \sum_{k=1}^{N_{val}} \left(Z_{N_{train}+k} - \hat{Z}_{N_{train}+k|train} \right)^2.$$

Observación clave: **los pronósticos de validación se generan con el modelo entrenado únicamente en el segmento de entrenamiento**, por lo que $J(\alpha, m)$ es un verdadero error fuera de muestra.

6.2. Rejilla de búsqueda en α y m

Elegimos:

- Una rejilla finita de suavizaciones:

  $$\alpha_1, \ldots, \alpha_{K_\alpha} \in (0, 1),$$

  típicamente un conjunto de puntos equiespaciados en $(0{,}01, 0{,}99)$.

- Un conjunto discreto de periodos estacionales candidatos:

  $$\mathcal{M} = \{m_1, \ldots, m_{K_m}\} \subset \mathbb{N},$$

  por ejemplo $\{5, 10, 20\}$ para tratar de capturar patrones semanales, quincenales o mensuales.

Definimos la rejilla total

$$\mathcal{G} = \{(\alpha_k, m_i) : k = 1, \ldots, K_\alpha,\ i = 1, \ldots, K_m\}.$$

Para cada par $(\alpha_k, m_i) \in \mathcal{G}$:

1. Verificamos que $N_{\text{train}} \geq 2m_i$; si no se cumple, no podemos inicializar correctamente, y marcamos $J(\alpha_k, m_i) = \text{NaN}$.
2. En caso contrario, calculamos $J(\alpha_k, m_i)$ siguiendo el procedimiento anterior (actualizar sólo con entrenamiento y pronosticar validación).

Finalmente, elegimos

$$(\alpha^*, m^*) \in \arg \min_{(\alpha_k, m_i) \in \mathcal{G}} J(\alpha_k, m_i).$$

Es decir, el par de parámetros que produce el menor MSE de validación dentro de la rejilla considerada.

Comentario 6.1 (Interpretación). *El periodo estacional m^* se selecciona* por evidencia empírica de validación. *Si la serie realmente tiene una estacionalidad fuerte en niveles (por ejemplo semanal), el m correspondiente tenderá a generar un $J(\alpha, m)$ menor que otros valores. Si la serie es esencialmente no estacional en niveles, la componente estacional ajustada tenderá a ser pequeña y distintos m producirán resultados similares.*

7. Ajuste final y pronóstico sobre validación + prueba

Una vez seleccionado (α^*, m^*):

1. Reajustamos el modelo Holt–Winters aditivo en el conjunto de entrenamiento usando exactamente las mismas ecuaciones (1)–(3), pero fijando $\alpha = \alpha^*$ y $m = m^*$.
2. Obtenemos:
$$\ell^*_{N_{\text{train}}}, \quad b^*_{N_{\text{train}}}, \quad \boldsymbol{s}^*, \quad p^*_{N_{\text{train}}} = (N_{\text{train}} - 1) \bmod m^*.$$
3. Definimos los pronósticos multi–paso, ahora tanto para validación como para prueba, como
$$\hat{Z}_{N_{\text{train}}+h|\alpha^*,m^*} = \ell^*_{N_{\text{train}}} + h\, b^*_{N_{\text{train}}} + s^*_{(p^*_{N_{\text{train}}}+h) \bmod m^*}, \qquad h = 1, \dots, N_{\text{val}} + N_{\text{test}}.$$

Los errores en el conjunto de prueba se calculan como
$$e_k^{\text{test}} = Z_{N_{\text{train}}+N_{\text{val}}+k} - \hat{Z}_{N_{\text{train}}+N_{\text{val}}+k|\alpha^*,m^*}, \qquad k = 1, \dots, N_{\text{test}},$$
y el MSE de prueba como
$$\text{MSE}_{\text{test}} = \frac{1}{N_{\text{test}}} \sum_{k=1}^{N_{\text{test}}} \left(e_k^{\text{test}}\right)^2.$$

8. Resumen simbólico

Resumiendo de forma compacta:

- Dado $Z_t = \log P_t$, se propone el modelo aditivo
$$Z_t = \ell_t + b_t + s_{p_t} + \varepsilon_t, \qquad p_t = (t-1) \bmod m.$$
- Se inicializa $(\ell_0, b_0, \boldsymbol{s})$ a partir de promedios de estaciones completas en el conjunto de entrenamiento.
- Sobre $\{Z_t\}_{t=1}^{N_{\text{train}}}$ se actualizan los estados mediante:
$$\boxed{\begin{aligned} \ell_t &= \alpha\,(Z_t - s_{p_t}) + (1-\alpha)(\ell_{t-1} + b_{t-1}), \\ b_t &= \alpha\,(\ell_t - \ell_{t-1}) + (1-\alpha)\, b_{t-1}, \\ s_{p_t} &= \alpha\,(Z_t - \ell_t) + (1-\alpha)\, s_{p_t}. \end{aligned}}$$
- El pronóstico multi–paso desde el final de entrenamiento es:
$$\boxed{\hat{Z}_{N_{\text{train}}+h|\text{train}} = \ell_{N_{\text{train}}} + h\, b_{N_{\text{train}}} + s_{(p_{N_{\text{train}}}+h) \bmod m}, \quad h \geq 1.}$$
- La función objetivo de validación para un par (α, m) es
$$\boxed{J(\alpha, m) = \frac{1}{N_{\text{val}}} \sum_{k=1}^{N_{\text{val}}} \left(Z_{N_{\text{train}}+k} - \hat{Z}_{N_{\text{train}}+k|\text{train}}\right)^2.}$$
- Se selecciona
$$\boxed{(\alpha^*, m^*) \in \arg \min_{(\alpha,m)\in\mathcal{G}} J(\alpha, m).}$$
- Finalmente, se rehace el ajuste sobre entrenamiento con (α^*, m^*) y se usa el modelo resultante para pronosticar tanto la parte de validación como la de prueba, sin emplear nunca esos datos para reentrenar o actualizar estados (sin fuga de información).