



Universidade de Vigo

Trabajo Fin de Máster

Estudio de la calidad de la información estadística difundida por el IGE empleando técnicas de Big Data y Machine Learning

Eugenio Fernández Vidal

Máster en Técnicas Estadísticas

Curso 2018-2019

Propuesta de Trabajo Fin de Máster

Título en galego: Estudo da calidade da información estadística difundida polo IGE empregando técnicas de Big Data e Machine Learning
Título en español: Estudio de la calidad de la información estadística difundida por el IGE empleando técnicas de Big Data y Machine Learning
English title: Checking the quality of the statistical information published by the IGE using Big Data and Machine Learning techniques
Modalidad: Modalidad B
Autor/a: Eugenio Fernández Vidal, Universidad de Santiago de Compostela
Director/a: María José Lombardía Cortiña, Universidad de la Coruña
Tutor/a: Esther López Vizcaíno, Instituto Gallego de Estadística
Breve resumen del trabajo: <p>Realizar un control de calidad de los datos es una tarea muy importante para asegurar la calidad de la información difundida. En este trabajo se resuelve el problema de la coherencia con el último dato a partir de la creación de un método de predicción y de generación de intervalos de predicción para series temporales utilizando técnicas de <i>machine learning</i>. Para verificar que los datos que van a ser introducidos son coherentes con los demás, se comprobará que dichos datos están dentro de los intervalos de predicción calculados para las predicciones, asumiendo que estas últimas están en consonancia con los demás datos.</p>

Doña María José Lombardía Cortiña, profesora de la Universidad de la Coruña y doña Esther López Vizcaíno, responsable del Servicio de Difusión e Información del Instituto Gallego de Estadística, informan que el Trabajo Fin de Máster titulado

Estudio de la calidad de la información estadística difundida por el IGE empleando técnicas de Big Data y Machine Learning

fue realizado bajo su dirección por don Eugenio Fernández Vidal para el Máster en Técnicas Estadísticas. Estimando que el trabajo está terminado, dan su conformidad para su presentación y defensa ante un tribunal.

En Santiago de Compostela, a 3 de julio de 2019.

La directora:

Doña María José Lombardía Cortiña

La tutora:

Doña Esther López Vizcaíno

El autor:

Don Eugenio Fernández Vidal

Agradecimientos

Tengo que manifestar mi más sincero agradecimiento a todas las personas que durante todo este tiempo han contribuido al proceso y conclusión del presente trabajo.

En primer lugar deseo expresar mi agradecimiento a la tutora Esther López Vizcaíno por darme todas las facilidades para el desarrollo del trabajo, por su tiempo, por confiar en mí y animarme a superarme constantemente. Sin sus correcciones, experiencia y consejos no hubiera sido posible la elaboración de este trabajo.

A mi directora María José Lombardía Cortiña, por su gran dedicación, sus buenas ideas y su gran paciencia. Gracias por el respeto a mis sugerencias e ideas y por el rigor y la coherencia que han brindado desde el primer momento a este trabajo.

Gracias a Pablo Montero Manso por la rápida resolución de las dudas surgidas con respecto a su método.

Agradecer también a mi familia y amigos, sobretodo a mis padres, por ser los principales promotores de mis sueños, gracias a ellos por confiar cada día en mí y en mis expectativas, gracias por sus consejos y por cada una de sus palabras que me guiaron durante toda mi vida.

A todos, muchas gracias.

Índice general

Resumen	XI
1. Introducción	1
2. Metodología	5
2.1. Conceptos básicos	5
2.2. Métodos	8
2.2.1. Paseo aleatorio	8
2.2.2. Suavización exponencial	8
2.2.3. Naïve	9
2.2.4. Naïve estacionario	9
2.2.5. Bagging	10
2.2.6. BATS	10
2.2.7. TBATS	11
2.2.8. STL	12
2.2.9. Redes neuronales	12
2.2.10. Theta	13
2.2.11. X13-ARIMA-SEATS	14
2.2.12. TRAMO-SEATS	15
3. Desarrollo de los métodos	19
3.1. Trabajo preparatorio	20
3.2. Errores	20
3.3. Best Forecasting Methods (BFM)	21
3.3.1. Metodología BFM	21
3.4. FFORMA	22
3.4.1. Metodología	24
3.4.2. Intervalos de predicción FFORMA	26
3.4.3. Modified FFORMA (MFFORMA)	28
3.4.4. Intervalos de predicción MFFORMA	29
4. Aplicación a datos reales	31
4.1. Análisis exploratorio	31
4.2. BFM	36

4.3. Comparación del FFORMA con el MFFORMA	43
5. Conclusiones	55
A. Tablas	57
A.1. Errores OWA	57
A.2. Mejor modelo, tiempo y OWA	67
A.3. Comparación de métodos	77
B. Código de R	81
B.1. Trabajo preparatorio	81
B.2. Método BFM	83
B.3. Método MFFORMA	86
Bibliografía	93

Resumen

Resumen en español

El objetivo de este trabajo es el de hacer un control de calidad de los datos publicados por el Instituto Gallego de Estadística. En concreto, centramos nuestra atención en resolver el problema de la coherencia del último dato, es decir, comprobar que los datos que van a ser añadidos a las series temporales ya existentes son coherentes con los anteriores.

Nos hemos planteado resolver este problema mediante la predicción y creación de intervalos de predicción, comprobando que los datos que van a ser añadidos a los demás están dentro de estos intervalos de predicción calculados.

Para ello, hemos creado un método de predicción basado en el método FFORMA de Montero-Manso et al. (2018) que realiza una combinación lineal de diferentes métodos de predicción para generar las predicciones finales utilizando técnicas de *machine learning*. El método produce predicciones muy precisas y calcula intervalos de predicción eficientes, resolviendo así el problema objetivo del trabajo. Este método se ilustra con datos reales ofrecidos por el Instituto Gallego de Estadística.

English abstract

The aim of this project is to carry out a quality control of the data published by the Galician Institute of Statistics. Specifically, we focus our attention on solving the problem of the coherence of the last data, that's to say, checking that the data that will be added to the existing time series are coherent with the previous ones.

We have decided to solve this problem by calculating predictions and creating prediction intervals, to check that the data that will be added to the others are within these calculated prediction intervals.

For this we have created a prediction method based on the FFORMA method of Montero-Manso et al. (2018) that performs a linear combination of different prediction methods to generate the final predictions using *machine learning* techniques. The method generates very accurate predictions and calculates efficient prediction intervals, solving the objective problem of the project. This method is tested with real data provided by the Galician Institute of Statistics.

Capítulo 1

Introducción

El Instituto Gallego de Estadística (IGE) es una de las principales fuentes de información de datos de carácter socioeconómico de la Comunidad Autónoma de Galicia y de sus principales divisiones territoriales. Esta circunstancia ha motivado un incremento notable de la cantidad de datos disponibles en el Instituto, teniendo así, no sólo datos de producción propia, sino también de los de diversos organismos oficiales que ofrecen información estadística de nuestra Comunidad Autónoma.

En el IGE, y en general en los institutos de estadística, se está continuamente añadiendo períodos de tiempo a su información. Por tanto, el control que tendremos que hacer es verificar que el dato que se va a añadir es coherente con los datos disponibles para períodos anteriores, para así asegurarnos de que no vamos a cometer errores a la hora de introducir nuevos datos a las plataformas de difusión, que en el caso del IGE, se encuentra en el siguiente enlace: <http://www.ige.eu>.

Este problema de *coherencia del último dato* es una técnica utilizada para controlar la calidad de los datos difundidos. Actualmente en la literatura no existe ninguna documentación sobre como afrontar este problema, por lo que el objetivo de nuestro trabajo será el de buscar una solución a dicho problema. Como objetivo adicional, nos proponemos hacer este control de una forma rápida y para un gran número de series simultáneamente. Como generalmente, los métodos de predicción existentes nos generan datos en consonancia con los demás, podemos considerar que las predicciones son buenos estimadores de los datos futuros. Por tanto, para afrontarlo, utilizaremos diferentes métodos de predicción para intentar obtener el siguiente valor que sucede a los demás de la forma más precisa posible, y a su vez calcular su intervalo de predicción con una probabilidad determinada. Una vez hecho esto quedaría comprobar que el dato que queremos introducir está dentro de este intervalo. Durante todo este proceso haremos uso de la teoría de las series temporales y de métodos de predicción de series temporales automáticos, es decir, algoritmos programados en distintos software como **R**, **Python** o **SAS** y que calculan los parámetros de los métodos utilizados de forma automática. Nosotros, en concreto, trabajaremos con el software libre **R**.

Que los métodos de predicción de series temporales automáticos sean precisos y robustos es realmente importante hoy en día en multitud de ámbitos como pueden ser: el comercio, para la predicción de ventas de sus productos y tener una estimación de la cantidad de producto que abastecerse; las finanzas, para saber como fluctuará el precio de los activos y actuar en función de esas predicciones; o el medio ambiente, para calcular la predicción de los niveles de oxido de azufre y tomar las medidas necesarias en caso de que estos niveles sean muy elevados.

La creación de los métodos de predicción se remonta a mediados del siglo XX, con los modelos de medias móviles y los autoregresivos. Unos años más tarde, con la necesidad de crear métodos de predicción que fuesen más precisos se empiezan a crear otros más complejos como el modelo de suavización exponencial simple (Brown, 1956), el Holt-Winters (Winters, 1960) o el famoso ARIMA (Box y Jenkins, 1970) entre otros.

Los resultados del artículo publicado por Spyros Makridakis y Michèle Hibon en 1979 (Makridakis et al., 1979) causaron gran polémica, tal y como se muestra en Hyndman (2019) y en Montero-Manso (2019). Esta polémica se debe a que en este artículo se muestra que modelos muy simples, como los de medias móviles, en la práctica funcionaban mejor que otros más complejos como los famosos modelos ARIMA (Box y Jenkins, 1970), dejándolos así en evidencia.

En la actualidad, los métodos de predicción automáticos que tienen implementados los modelos ARIMA (Box y Jenkins, 1970), el método X-13ARIMA-SEATS (Research Staff U.S. Census Bureau, 2017) desarrollado por la Oficina del Censo de los Estados Unidos, o el método TRAMO-SEATS (Gómez y Maravall, 1996) desarrollado por el Banco de España, funcionan realmente bien en la práctica, tal y como veremos en las conclusiones obtenidas por el método de la tercera sección del Capítulo 3 (*Best Forecasting Methods, BFM*).

Gracias a los avances en tecnología y computación hechos hasta ahora, podemos realizar predicciones de series temporales de forma alternativa con técnicas de *Machine Learning* y de *redes neuronales*, que son una rama de la inteligencia artificial, en las que se desarrollan técnicas para que las computadoras aprendan a base de la experiencia sobre un tema concreto. Este tipo de métodos son buenos a la hora de trabajar con *Big Data* (datos de gran volumen, gran variabilidad y crecimiento a gran velocidad) ya que una vez ha aprendido el algoritmo, nos ofrece soluciones buenas y rápidas. Algoritmos donde se hace uso de este tipo de procedimientos son los utilizados este último año en la competición M4 (M4 Competitor's Guide, 2018), llamada así por su principal organizador, Spyros Makridakis y por ser la cuarta edición. En esta competición investigadores desarrollan modelos de predicción de series temporales cada vez más precisos. Los métodos que destacaremos por hacer uso de estas técnicas son los que han obtenido la primera y segunda posición de la competición M4. Estos métodos son el ES-RNN de Slawek Smyl (2018) con un modelo de *redes neuronales* y el FFORMA del gallego Pablo Montero-Manso et al. (2018) con técnicas de *Machine Learning*, respectivamente.

Con respecto a los datos de trabajo, el IGE nos proporcionará varias bases de datos para ilustrar esta metodología. El análisis y los resultados se muestran en el Capítulo 4. Los conjuntos de datos reales proporcionados son:

- Conjunto 1: viajeros, noches y estancia media en establecimientos hoteleros y de turismo rural en España, Galicia y sus provincias (Datos mensuales).
- Conjunto 2: población de 16 y más años por sexo, grupos de edad y relación con la actividad económica en Galicia (Datos trimestrales).
- Conjunto 3: contratos registrados según la modalidad (Datos mensuales).
- Conjunto 4: índice de producción industrial general y por destino económico de los bienes (Datos mensuales).
- Conjunto 5: transacciones inmobiliarias por régimen y tipo de vivienda. Número, valor total y valor medio (Datos trimestrales).

La elección de estos conjuntos de series proviene de la diferente naturaleza entre ellos, ya que al ser todos estos conjuntos tan heterogéneos, conseguimos trabajar y probar nuestros métodos con series de gran variabilidad y características muy distintas. Aunque la suma total de datos que componen estos conjuntos de series de tiempo está cerca de 70,000 no podemos considerarlos como *Big Data* ya que no crecen a gran velocidad (como mucho, un dato al mes por serie temporal).

Una primera aproximación que hacemos para intentar resolver el problema de la coherencia del último dato es tratar de averiguar el método de predicción automática que produce mejores resultados en la práctica de forma general, es decir, tratar de encontrar el método que predice mejor para un mayor número de series. Una vez contrastados los resultados de esta aproximación nos decantamos por trabajar con el método FFORMA (Montero-Manso et al., 2018) para realizar la predicción y el cálculo de intervalos de predicción, haciéndole unas pequeñas modificaciones con el fin de obtener mejores predicciones y de una forma más rápida. Tanto la primera aproximación como el método FFORMA y sus modificaciones las explicaremos de forma rigurosa en el Capítulo 3.

El resto del trabajo se organizará de la siguiente forma. En el Capítulo 2 se exponen los conceptos básicos para poder comprender todo lo que viene a continuación, así como los métodos que utilizaremos en los algoritmos de su posterior capítulo. Los métodos desarrollados en esta memoria se explican con detalle en el Capítulo 3, y en el Capítulo 4 se muestra una aplicación a los datos proporcionados por el IGE para el estudio del control de calidad de los mismos. Finalmente, en el capítulo de conclusiones se exponen las conclusiones obtenidas a lo largo del trabajo, finalizando el documento con los anexos del código de R utilizado y las tablas con sus resultados.

Capítulo 2

Metodología

En este capítulo nos centraremos en introducir los conceptos necesarios que nos servirán como soporte para poder comprender lo que se explicará tanto en este como en capítulos posteriores. Después explicaremos los principales métodos que utilizaremos para la predicción de valores y el cálculo de intervalos de predicción de series temporales en los métodos del Capítulo 3, indicando las funciones de R que hacen referencia a dichos métodos y en qué paquetes se encuentran cada una.

Es decir, asentamos todos los conceptos necesarios para abordar el problema de la *coherencia del último dato* mediante la predicción y creación de intervalos de predicción de series temporales, comprobando que el nuevo valor que va a ser introducido es coherente con respecto a los anteriores, viendo que está dentro de los intervalos de predicción calculados.

2.1. Conceptos básicos

Comenzaremos definiendo los conceptos que se repetirán a lo largo de todo el documento en numerosas ocasiones y que necesitamos conocer antes de pasar a abordar el problema objetivo del trabajo. Estos conceptos se pueden encontrar en cualquier libro que trate sobre series temporales, como por ejemplo: Aznar y Trivez (1993) o Woodward et al. (2017). Estos conceptos son: la definición de *serie de tiempo* o *serie temporal*, características de las series temporales como la *tendencia*, o la *estacionalidad*, lo que significa hacer una *predicción de una serie de tiempo* y qué son el *error de predicción*, un *intervalo de predicción* y los *modelos ARIMA*.

Una serie de tiempo es una secuencia de datos numéricos que son tomados a lo largo de un período de tiempo. Ejemplos de serie de tiempo pueden ser los índices del precio del petróleo, el número de habitantes en cierto país por año, la medición diaria del nivel de un río o la temperatura a cada hora del día en una ciudad determinada.

Dos características muy importantes que presentan muy a menudo las series de tiempo, y que darán lugar a descomposiciones de series en torno a ellas, son la tendencia y la estacionalidad. La tendencia es el comportamiento a largo plazo, que se produce en relación al nivel medio de la serie, y la estacionalidad es el comportamiento periódico de la serie.

Se habla de predicciones de series de tiempo cuando se calcula (de manera más o menos exacta) los valores que siguen al último dato de la serie de tiempo a partir de los demás. Por ejemplo, intentar pronosticar el precio que tendrá el petróleo en los próximos días, o el número de habitantes de un país en

los años venideros. Al número de valores que queremos predecir le llamamos horizonte de predicción. Para resolver el problema que tenemos entre manos sería suficiente con tomar como horizonte de predicción 1, ya que se va actualizando la información dato a dato. Sin embargo, en el Capítulo 3, para tratar de averiguar cuál es el método que mejor predice vamos a tomar como horizonte de predicción el número de observaciones que tenemos por año (frecuencia) en cada serie con la que estamos tratando. Por ejemplo, si estamos trabajando con series mensuales, la frecuencia es 12, y si trabajamos con series trimestrales, la frecuencia es 4. Esto lo haremos así ya que para una única predicción no podríamos afirmar con seguridad que un modelo genera buenas predicciones, puesto que no tendríamos pruebas suficientes para garantizar la veracidad de este resultado.

Una forma de medir cómo de buena es una predicción, es utilizando el error de predicción, siempre y cuando sepamos el valor de la observación real que estamos intentando predecir. Existen multitud de formas para medir este error de predicción, como por ejemplo el Error Relativo, el Error Cuadrático Medio (ECM), el Error cuadrado Integrado (ECI) o el Error OWA (Overall Weighting Average), entre otros. Las formas de medir el error de predicción que utilizaremos nosotros en esta memoria serán el ECM y el error OWA que explicaremos en la Sección 3.2.

Un intervalo de predicción es un rango de valores entre los cuales se estima que estará la predicción con un grado de probabilidad determinada. El intervalo de predicción depende del tamaño de los errores de predicción. A continuación se muestra la fórmula de los intervalos de predicción más habituales en la literatura:

$$\text{predicción} \pm K * \text{error estandar de predicción},$$

donde K es el multiplicador del error estándar considerando la distribución normal. En esta memoria se trabajará con los intervalos de predicción del 95 %, por tanto, tomaremos el percentil 95 de la distribución normal, teniendo así $K = 1,96$.

Los modelos de series temporales se usan para describir el comportamiento de las series de tiempo e intentar predecir a partir de ellos. El modelo ARIMA (Box y Jenkins, 1970), también llamado Box-Jenkins por sus creadores, es fundamental en la teoría y aplicaciones de las series de tiempo, siendo uno de los modelos más importantes e utilizados hasta el momento. Este modelo aparece en más de uno de los métodos de predicción que utilizaremos en la siguiente sección para hacer el control de calidad de los datos.

El enfoque del modelo de Box-Jenkins es eliminar de la serie de tiempo la tendencia y la parte estacional (mediante transformaciones o filtros) para que quede únicamente la parte probabilística, con el fin de aplicarle a esta última parte un ajuste de modelos paramétricos.

Sea $\{X_t\}_t$ una serie de tiempo, un modelo ARIMA(p,d,q) es aquel que admite una representación del tipo

$$\phi(B)(1 - B)^d X_t = c + \theta(B)a_t, \quad (2.1)$$

donde X_t es el valor que toma la serie en el instante t , B denota el operador retardo definido por $BX_t = X_{t-1}$, $\phi(B) = (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)$, $\theta(B) = (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q)$; c , $\phi_1, \phi_2, \dots, \phi_p$, $\theta_1, \theta_2, \dots, \theta_q$ son constantes, d es el número de diferencias necesarias para eliminar la tendencia a la serie, a_t es un proceso de ruido blanco (colección de datos incorrelados, con media cero ($\mu = 0$) y varianza finita ($\sigma_a^2 = \text{cte}$), o lo que es lo mismo, datos independientes e idénticamente distribuidos (i.i.d.), y p y q son los órdenes de los procesos autorregresivos (AR) y de medias móviles

(MA) respectivamente.

Los modelos ARIMA también se pueden considerar como una mezcla de los primeros modelos de series temporales creados, los procesos AR y los MA. Los procesos AR(p) están definidos como $X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_q X_{t-q} + a_t$ y los procesos MA(q) como

$$X_t = c + a_t + \theta_1 a_{t-1} + \theta_2 a_{t-2} + \dots + \theta_q a_{t-q}.$$

Cuando el proceso que genera la serie de tiempo presenta un comportamiento periódico a lo largo del tiempo, esto es, existen relaciones entre las observaciones que distan entre sí s periodos o múltiplos de s , siendo s el período estacional; se debe utilizar un modelo ARIMA estacional multiplicativo (ARIMA(p,d,q)×(P,D,Q)) que admite la siguiente representación

$$\phi(B)\Phi(B^s)(1-B)^d(1-B^s)^D X_t = c + \theta(B)\Theta(B^s)a_t, \quad (2.2)$$

donde D es el número de diferenciaciones estacionales aplicadas para eliminarle la componente estacional y Φ_i y Θ_j son las constantes respectivas a la parte estacional para $i = 1, \dots, P$ y $j = 1, \dots, Q$, con P igual al orden del proceso AR estacional y Q al orden del proceso MA estacional.

Existe una función de R del paquete `forecast` que calcula los parámetros adecuados del mejor modelo ARIMA para una serie de tiempo basándose en los criterios AIC AICc o BIC (que explicaremos a continuación). Esta función se llama `auto.arima` y genera muy buenos resultados en la práctica. El inconveniente que presenta es que en muchas ocasiones puede tardar considerablemente en generar estas predicciones.

El Criterio de Información de Akaike (AIC), El Criterio de Información de Akaike corregido (AICc) y el Criterio de Información de Bayes (BIC) son medidas globales que tienen en cuenta tanto el ajuste como el número de parámetros. Con estos criterios se pretende buscar un equilibrio entre estas dos características. La idea es encontrar el modelo que incorpore parámetros realmente útiles para incrementar la verosimilitud (que depende de la distribución que siga el modelo utilizado). Estos criterios se definen de la siguiente forma:

- $$AIC = -2\log(\text{verosimilitud}) + 2p \quad (2.3)$$

- $$AICc = AIC + \frac{2p^2 + 2p}{n - p - 1} \quad (2.4)$$

- $$BIC = -2\log(\text{verosimilitud}) + 2p \log(n) \quad (2.5)$$

donde p es el número de parámetros y n el número de observaciones. Para más información sobre estos criterios podemos consultar Burnham y Anderson (2004).

Ahora que conocemos todos estos conceptos podemos pasar a explicar los métodos de predicción que utilizaremos en el siguiente capítulo. Estos métodos realizan una modelización previa de la serie temporal con el objetivo de conocer el modelo probabilístico que haya sido susceptible de generarla y crear predicciones e intervalos de predicción a partir de él.

2.2. Métodos

A continuación explicaremos los principales métodos de predicción y de generación de intervalos de predicción que utilizaremos para resolver el problema de *coherencia con el último dato*.

Para cada uno de los métodos, explicaremos su funcionamiento, mencionaremos algunas ventajas o desventajas de los mismos, diremos donde podemos encontrarlos y las funciones del software libre R que hacen referencia a dichos métodos, junto con los paquetes a los que pertenecen.

2.2.1. Paseo aleatorio

Un paseo aleatorio (Random Walk) es aquel proceso aleatorio generado por el siguiente modelo:

$$X_t = c + X_{t-1} + a_t,$$

donde $\{a_t\}_t$ es ruido blanco y c una constante. Las predicciones están dadas por

$$\hat{X}_{n+h} = ch + X_n,$$

siendo n el número máximo de observaciones y h el horizonte de predicción.

Este modelo es equivalente a un modelo $ARIMA(0, 1, 0)$. Es conveniente utilizar este modelo cuando sabemos que la serie de tiempo depende únicamente del valor que lo precede, ahorrándonos así una búsqueda entre los parámetros del mejor modelo ARIMA, generando así resultados en un tiempo mucho menor.

El ajuste de este método se realiza bajo la función `rwf` (Random Walk Forecast) del paquete `forecast` (Hyndman, 2019), que calcula las predicciones y los intervalos de predicción tras ajustar a las series su modelo ARIMA homólogo ($ARIMA(0, 1, 0)$).

2.2.2. Suavización exponencial

Los métodos de suavización exponencial tratan de captar el comportamiento a largo plazo, que se produce en relación al nivel medio de la serie (tendencia) y patrones no estacionales ya que se asume una estacionalidad local y una media que varía. Esto lo hacen realizando una descomposición muy común de las series de tiempo, que es la de separar la serie en torno a su componente de tendencia (comportamiento a largo plazo de la serie con respecto a su nivel medio) y a su componente estacional (comportamiento periódico de la serie). Según sea la componente de la tendencia y la componente estacional, podemos clasificar los métodos de suavizado exponencial en 15 diferentes tipos, recogidos de forma taxonómica en Taylor (2003).

Los primeros métodos de suavización exponencial propuestos son el método de suavización exponencial simple creado por Brown (1956) y el de Holt (1957); le siguen otros métodos como el de Winters (1960), el de Gardner (1985), el de Ord et al. (1997) o el de Hyndman et al. (2002). Estos 15 métodos únicamente nos permiten el cálculo de predicciones y no de sus intervalos de predicción. Por lo que, para solventar este problema, Hyndman et al. (2008b) añaden a la descomposición de series anterior la componente del error, creando así para cada uno de los 15 métodos de suavización exponencial, dos posibles modelos, uno con errores aditivos y otro con errores multiplicativos, de forma que estos nuevos modelos nos proporcionan las mismas predicciones (tomando los mismos parámetros para los

dos modelos) pero diferentes intervalos de predicción. Estos últimos modelos son denominados modelos de espacio de estado de las innovaciones (errores). Teniendo así 30 métodos diferentes con esta clasificación (Métodos de suavización exponencial de espacio de estado).

Los métodos de suavización exponencial lineales son casos especiales de los modelos ARIMA. Las equivalencias entre ambos modelos se discuten en Hyndman et al. (2008), sin embargo, también se pone de manifiesto que los métodos de suavización exponencial no lineales no tienen su equivalente ARIMA y lo mismo sucede al revés, muchos modelos ARIMA, no tienen su método homólogo de suavización exponencial, por lo que cada uno tiene sus puntos fuertes y sus debilidades.

Cabe destacar que para series estacionarias, los modelos ARIMA son mejores ya que los métodos exponenciales son todos no estacionarios.

La función de **R** del paquete `forecast` que recorre todos estos modelos (si no se le especifica ningún modelo concreto) hasta quedarse con el modelo de suavización exponencial más adecuado basándose en los criterios BIC o AIC es la `ets` (Error, Trend and Seasonality), ya que descompone la serie en torno a esas tres componentes. Este procedimiento funcionó de manera realmente buena en la competición M3 (ver Hyndman et al. (2002), donde podemos comprobar además la metodología de este método).

2.2.3. Naïve

El método naïve es equivalente al paseo aleatorio o al modelo ARIMA(0.1.0) fijando la constante $c = 0$. El modelo de este método es

$$X_t = X_{t-1} + a_t,$$

siendo $\{a_t\}_t$ ruido blanco, es decir, errores normales i.i.d. Este modelo utiliza la última observación como futura predicción, por lo que no resultaría ser bueno para series temporales con mucha variabilidad ni para series con estacionalidad. En este caso, es conveniente usar mejor el método de suavización exponencial simple de Brown (1956) o el de medias móviles ya que las predicciones cambiarán con cada nueva observación.

Este modelo se encuentra implementado también en el paquete `forecast` bajo la función `naive`.

2.2.4. Naïve estacionario

El método naïve estacionario (seasonal naïve) se crea con la misma idea subyacente que el método naïve para series con estacionalidad, es decir, tomar como predicciones las observaciones respectivas al último año o periodo estacional. Su ecuación del modelo es de la siguiente forma

$$X_t = X_{t-s} + a_t,$$

donde $\{a_t\}_t$ es ruido blanco y s es el periodo estacional.

Este modelo es el equivalente al modelo $ARIMA(0, 0, 0)(0, 1, 0)_s$ donde s es el periodo estacional. Las predicciones generadas por este modelo son iguales al valor de la serie un periodo estacional antes, por ejemplo, si la serie es mensual, y la serie sigue cierto patrón cada 12 meses, la predicción del siguiente dato será el mismo valor que 12 instantes de tiempo antes.

Este modelo tampoco resulta adecuado para series que presentan tendencia y mucha variabilidad. La función del paquete `forecast` con la que podemos ajustar este modelo es la `snaive` y lo que hace

es calcular las predicciones y los intervalos de predicción a partir de su modelo ARIMA equivalente ($ARIMA(0, 0, 0)(0, 1, 0)_s$).

2.2.5. Bagging

El *bagging* o *bootstrap aggregating* es una forma de hacer *Machine Learning*. En términos generales, el *bagging* lo que hace es dividir conjunto de entrenamiento (muestra original) en distintos sub-conjuntos de datos, obteniendo como resultado diferentes muestras aleatorias con las siguientes características:

- Muestra uniforme, es decir, la misma cantidad de individuos en cada conjunto de datos.
- Muestras con reemplazo, esto es, los individuos pueden repetirse en el mismo conjunto de datos.
- El tamaño de la muestra es igual al tamaño del conjunto de entrenamiento, pero no contiene a todos los individuos ya que algunos se repiten.
- Si se usan muestras sin reemplazo, suele elegirse el 50% de los datos como tamaño de muestra.

Luego se crea un modelo predictivo con cada conjunto, obteniendo modelos diferentes y finalmente se construye o ensambla un único modelo predictivo, que es el promedio de todos los modelos para predecir a partir de él.

Como nuestro objetivos a la hora de resolver el problema de coherencia con el último dato es el de modelizar muchas series de tiempo simultáneamente de una forma rápida y efectiva, y predecir a partir de estos modelos, descartamos este método que acabamos de describir ya que tarda mucho en generar predicciones y es muy costoso computacionalmente.

Si se desea, se puede estimar el modelo *bagging* para las series con la función `baggedmodel` del paquete `forecast`.

2.2.6. BATS

El modelo BATS (Exponential smoothing state space model with Box-Cox transformation, ARMA errors, Trend and Seasonal components) lo proponen De Livera y Alysha (2010). Ellos plantean modificaciones al método de suavización exponencial expuesto en Hyndman et al. (2008) para que pueda contemplar una mayor variedad de patrones estacionales y también solucione el problema de errores correlados. A estos modelos se los denomina BATS como acrónimo de las características claves del modelo: transformación Box-Cox, errores ARMA, componentes de tendencia y estacionalidad.

La transformación Box-Cox se utiliza para convertir un conjunto de observaciones en lineales y homocedásticas (variabilidad constante de las serie) en caso de no serlo, y dicha transformación es de la siguiente forma:

$$x_t^{(\lambda)} = \begin{cases} \frac{x_t^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \log(x_t), & \lambda = 0, \end{cases}$$

donde $x_t^{(\lambda)}$ se utiliza para representar la observación transformada por el método Box-Cox con parámetro λ .

Las demás ecuaciones que completan el modelo son

$$x_t^{(\lambda)} = l_{t-1} + \omega b_{t-1} + \sum_{i=1}^T s_{t-m_i}^{(i)} + d_t, \quad (2.6)$$

$$l_t = l_{t-1} + \omega b_{t-1} + \alpha d_t,$$

$$b_t = (1 - \omega)b + \omega b_{t-1} + \beta d_t,$$

$$s_t^{(i)} = s_{t-m_i}^{(i)} + \gamma_i d_t, \quad (2.7)$$

$$d_t = \sum_{i=1}^p \phi_i d_{t-i} + \sum_{i=1}^q \theta_i a_{t-i} + a_t$$

donde x_t es la observación en el momento t , m_1, \dots, m_T son los períodos estacionales, l_t es el nivel local en el período t , b es la tendencia a largo plazo, b_t es la tendencia a corto plazo en el período t , $s_t^{(i)}$ representa la i -ésima componente estacional en el momento t , $\{d_t\}$ puede ser un modelo ARMA(p,q) y el proceso $\{a_t\}$ es un proceso de ruido blanco gaussiano con media cero y varianza constante σ^2 . Los parámetros de suavización están dados por α , β y γ_i para $i = 1, \dots, T$ y ω es la constante de amortiguación de la tendencia. Este cambio asegura que el valor de la tendencia a corto plazo b_t converje al valor b (tendencia a largo plazo), en lugar de tender a cero.

Este modelo tiene las ventajas de que la transformación Box-Cox puede corregir la no linealidad y la heterocedasticidad (cambios de variabilidad) de la serie, el modelo ARMA para los residuos puede resolver la autocorrelación (relación o dependencia de una variable consigo misma en instantes temporales diferentes) y funciona mejor que los métodos de suavización exponencial de estado de espacio.

La función `bats` del paquete `forecast` ajusta el modelo BATS a las series de la forma que se describe en De Livera et al. (2011).

2.2.7. TBATS

El método TBATS es más flexible que el método BATS ya que De Livera et al. (2011) le introducen a este último modelo una representación trigonométrica (de donde procede la primera T del modelo TBATS) de las componentes estacionales basada en series de Fourier de la siguiente manera:

$$s_t^{(i)} = \sum_{j=1}^{T_i} s_{j,t}^{(i)}, \quad (2.8)$$

$$s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \cos \Lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \Lambda_j^{(i)} + \gamma_1^{(i)} d_t,$$

$$s_{j,t}^{*(i)} = s_{j,t-1}^{(i)} \sin \Lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \Lambda_j^{(i)} + \gamma_2^{(i)} d_t,$$

donde t es el periodo de tiempo de estudio, $\gamma_1^{(i)}$ y $\gamma_2^{(i)}$ son parámetros de suavización, $\Lambda_j^{(i)} = \frac{2\pi j}{m_i}$ describe el nivel estocástico de la i -ésima componente estacional y T es el número total de estacionalidades.

La diferencia entre el modelo BATS y el TBATS es el intercambio de la ecuación de la componente

estacional del modelo BATS 2.7 por la ecuación trigonométrica 2.8 y la ecuación del modelo 2.6 por

$$x_t^{(\lambda)} = l_{t-1} + \omega b_{t-1} + \sum_{i=1}^T s_{t-1}^{(i)} + d_t.$$

Las ventajas que ofrece este modelo con respecto al BATS es que puede trabajar con datos de alta frecuencia y con series con multi-estacionalidad sin incrementar demasiado el número de parámetros. La función de R que hace referencia a este método es `tbats`, del paquete `forecast`, que ajusta el modelo TBATS a las series de la forma que descrita en De Livera et al. (2011), artículo que también podemos consultar para conocer ventajas y desventajas o conceptos más teóricos.

2.2.8. STL

El método STL es aquel que realiza una descomposición de la serie temporal en su componente de tendencia y de estacionalidad utilizando la descomposición de Loess (método utilizado para estimar relaciones no lineales). Este método fue desarrollado por Cleveland et al. (1990) y recibe este nombre debido a sus siglas en inglés, Seasonal and Trend decomposition using Loess. Las ventajas que presenta este método es que puede trabajar con cualquier tipo de estacionalidad; la componente estacional puede cambiar con el tiempo y puede ser controlado por el usuario; la fluidez del ciclo de tendencia también puede ser controlado por el usuario; y puede ser robusto ante valores outliers, con lo que observaciones inusuales no afectan a las estimaciones de las componentes del ciclo de tendencia y de la componente estacional.

La función `stlm` del paquete `forecast` le aplica a la serie dada una descomposición STL y modeliza los datos ajustados estacionalmente por el modelo especificado en el argumento `modelfunction`.

2.2.9. Redes neuronales

Una forma rápida de definir las *redes neuronales* podría ser: modelo para encontrar una combinación de parámetros y aplicarlos al mismo tiempo con un objetivo concreto. Su estructura se inspira en el funcionamiento del sistema nervioso. En términos generales, una *red neuronal* consiste en un gran número de unidades simples de proceso, denominadas neuronas, que actúan en paralelo, están agrupadas en capas y están conectadas mediante vínculos ponderados (pesos).

Para encontrar esta combinación de pesos que mejor se ajusta a los datos es necesario “entrenar” la red neuronal. Este aprendizaje se basa en minimizar el error de la red neuronal. Una vez la red está ya entrenada, la podemos utilizar para hacer predicciones o clasificaciones, es decir, “aplicar” los pesos.

La estructura principal de las *redes neuronales* está formada por: una capa de entrada, que en nuestro caso sería una serie de tamaño n que utilizaría para el entrenamiento y sobre la que calculará las predicciones; una capa oculta de m neuronas que aprenderá a partir de los datos de entrada generando un conjunto de pesos, W_k , para $k = 1, \dots, n$, que nos indican la importancia de cada observación para el modelo; y una capa de salidas que nos devolvería las predicciones de la serie a partir de los pesos W_j , para $j = 1, \dots, m$. Esta estructura la podemos ver en el siguiente diagrama.

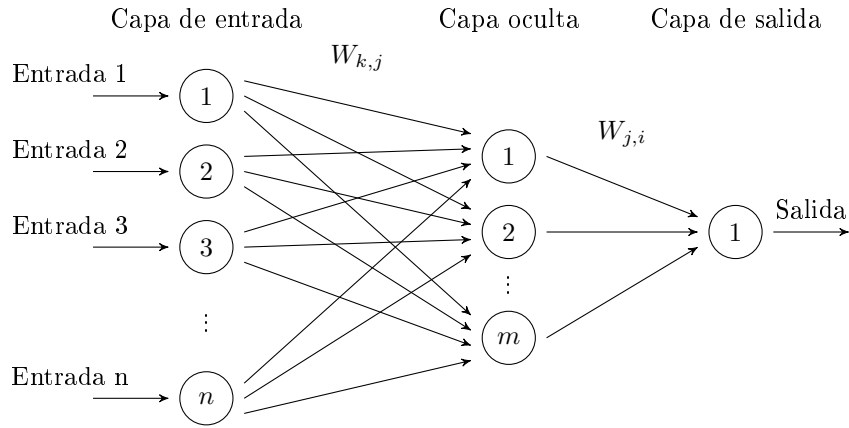


Figura 2.1: Esquema general de una red neuronal.

El aprendizaje puede ser supervisado, si conocemos de antemano la salida deseada a la hora de entrenar las neuronas, o no supervisado, si no lo conocemos. Para el supervisado, la red trata de minimizar el error entre la salida que calcula y las observaciones reales, de modo que la salida calculada termine siendo la deseada.

La función `nnetar` del paquete `forecast` ajusta la serie de interés por un modelo de *redes neuronales* pre-alimentadas, con aprendizaje supervisado. Para ello necesitamos especificarle el número de nodos adecuados (neuronas) de la capa oculta. La red se entrena para que realice una única predicción, si se desean obtener más predicciones, estas se calculan de forma recursiva.

Los principales inconvenientes de este método son que tiende a sobre-ajustar las predicciones y que la función no genera intervalos de predicción para el valor predicho.

2.2.10. Theta

El método Theta para predicciones fue introducido por Assimakopoulos y Nikolopoulos (2000) (A&N). Su compleja formulación llevó a otros autores a examinar el modelo utilizado y deducir que puede ser expresado de una manera mucho más simple, ver Hyndman y Billah (2001). Además en ese artículo podemos ver un estudio que muestra que las predicciones obtenidas con ese método son equivalentes al las del método de suavización exponencial simple.

Sea $\{X_1, \dots, X_n\}$ una serie de tiempo univariante, el modelo propuesto por (A&N) construye una nueva serie Y de la forma

$$\Delta^2 Y_t(\gamma) = \gamma \Delta^2 X_t, \quad t = 3, \dots, n \quad (2.9)$$

donde Δ es el operador diferencia (es decir, $\Delta X_t = X_t - X_{t-1}$).

Los valores iniciales Y_1 e Y_2 se calculan minimizando $\sum_{t=1}^n (X_t - Y_t(\gamma))^2$. Sin embargo, la solución analítica para calcular $Y(\gamma)$ está dada por

$$Y_t(\gamma) = \gamma X_t + (1 - \gamma)(A_n + B_n t), \quad t = 1, \dots, n,$$

donde A_n y B_n son los coeficientes de mínimos cuadrados de una regresión lineal simple de X_1, \dots, X_n

sobre $1, \dots, n$ y están dados por

$$A_n = \frac{1}{n} \sum_{t=1}^n X_t - \frac{n+1}{2} B_n \quad \text{y} \quad B_n = \frac{6}{n^2-1} \left(\frac{2}{n} \sum_{t=1}^n tX_t - \frac{1+n}{n} \sum_{t=1}^n X_t \right).$$

Tal y como dijimos anteriormente, este modelo, al fin y al cabo, no resulta ser muy efectivo ya que es bastante complejo y los resultados son equivalentes a los obtenidos por el método de suavización exponencial simple.

La función `thetaf` del paquete `forecast` ajusta la serie por el modelo Theta de la forma expuesta en Hyndman y Billah (2003).

2.2.11. X13-ARIMA-SEATS

El método X-13ARIMA-SEATS es una versión mejorada del modelo X-11 creado por Shiskin et al. (1967). Este método fue creado por la Oficina del Censo de los Estados Unidos (US Census Bureau) para el ajuste estacional de series temporales, es decir, para hacer un análisis de su tendencia y hacer comparaciones entre momentos arbitrarios tras haber compensado los efectos estacionales. Una descripción completa de este método podemos encontrarla en Time Series Research Staff (2017).

Este método se basa en el ajuste de modelos regARIMA, que son modelos de regresión con errores ARIMA; de forma más precisa, son modelos en los que la función media de la serie de tiempo está descrita por una combinación lineal de regresores, y la estructura de la matriz de covarianzas de la serie es aquella dada por un proceso ARIMA. Están incluidos en el modelo regresores para detectar cambios en el calendario (fechas señaladas que cambian de un año a otro, como por ejemplo carnavales o la semana santa) y regresores para detectar valores atípicos o cambios repentinos en el valor de la serie que deberán ser temporalmente eliminados antes de estimar el ajuste estacional.

La ecuación del modelo RegARIMA es la siguiente

$$x_t = z_t \beta + k_t \tag{2.10}$$

donde x_t es la serie original, $z_t = (z_t^{(1)}, \dots, z_t^{(n)})$ vector con las variables de regresión (Outliers, efecto calendario, variables definidas por el usuario), $\beta = (\beta_1, \dots, \beta_n)'$ es un vector con los coeficientes de regresión y k_t son los errores de regresión que siguen un proceso ARIMA.

Así, la serie se descompone en una componente lineal determinista y una componente estocástica. La componente determinista puede contener los valores outliers, efectos de calendario y los efectos de regresión. La parte estocástica está definida por un modelo ARIMA. En este preajuste, la serie original se corrige ante cualquier efecto determinista.

Una vez calculado esto, la serie pre-ajustada se descompone en las siguientes componentes: ciclo de tendencia (t), componente estacional (s), componente irregular (i). Las posibles descomposiciones pueden ser:

aditiva $x = t + s + i$

multiplicativa $x = t * s * i$

log-aditiva $\log(x) = \log(t) + \log(s) + \log(i)$

pseudo-aditiva $x = t * (s + i - 1)$

El último ajuste estacional debería estar libre de estacionalidad y de movimientos de calendario.

En resumen, el método X13-ARIMA-SEATS consiste en un pre-ajuste de la serie original con un modelo RegARIMA, donde la serie original se corrige para cualquier efecto determinista o observaciones perdidas y luego se realiza una descomposición de la serie en las componentes de tendencia, estacionalidad e irregularidad. Ejemplos utilizando este método se pueden encontrar en Findley y Hood (1999).

Este método se encuentra bajo el nombre de la función `seas` en el paquete `seasonal` (Sax, 2018) y también bajo el nombre de la función `x13_def` en el paquete `RJDemetra` (Quartier-la-Tente et al., 2019). Para que se generen las predicciones por estos algoritmos automáticos, en el caso del paquete `seasonal` es necesario utilizar la función `series` especificándole la salida deseada (ver especificaciones en el apartado de la función `series` de Sax (2018)), ya que esta función contiene las tablas con las salidas del método. Lo mismo sucede para el paquete `RJDemetra` aunque, la función indicada anteriormente ajusta y calcula las predicciones automáticamente tomando las especificaciones que vienen por defecto. Si se desea cambiarle algunas especificaciones es posible hacerlo, para ello y para más información, se puede consultar Quartier-la-Tente et al. (2019)

Este método funciona muy bien en la práctica debido a todos los tipos de efectos y errores que tiene en cuenta a la hora de predecir.

Cabe mencionar que las funciones del paquete `seasonal`, aunque las hemos probado, no las utilizaremos, a pesar de que produzcan muy buenos resultados, porque produce muchos errores a la hora de predecir, y tiene ciertas limitaciones como no poder predecir para series de frecuencia 1 (anuales).

2.2.12. TRAMO-SEATS

El método TRAMO-SEATS lo crea el Banco de España para el análisis de series temporales (Maravall, 2008). Las principales aplicaciones del método TRAMO-SEATS son la previsión, el ajuste estacional, la estimación de la tendencia y el ciclo, la interpolación, la detección y corrección de outliers y la estimación de los efectos de calendario. Este método es muy similar al X-13ARIMA-SEATS ya que la primera parte de Regresión ARIMA y corrección del efecto determinista y observaciones perdidas, es equivalente a la parte TRAMO (Time Series Regression with ARIMA Noise, Missing Observations and Outliers) para el pre-ajuste de la serie. La principal diferencia es que en la segunda parte del método (SEATS, Signal Extraction in ARIMA Time Series), se realiza una descomposición aditiva óptima de la estructura de la serie basada en la salida del ARIMA (de la parte estocástica del modelo pre-ajustado) en torno a las componentes de tendencia, estacionalidad, transitoriedad (captura periodos de corta duración con comportamiento errático) y parte irregular (errores). Una vez tenemos esto, lo juntamos con la parte determinista para obtener las componentes finales que ajustan la serie al modelo.

En la Figura 2.2 mostramos un esquema del funcionamiento del método TRAMO-SEATS. Para más información sobre este método podemos consultar Maravall (2008).

Al igual que el método X13-ARIMA-SEATS, el método TRAMO-SEATS se encuentra implementado en el paquete `RJDemetra` en la función `tramoseats_def` y también genera predicciones con las especificaciones que tiene por defecto. Para ver todas las posibles especificaciones y más información sobre este paquete, ver Quartier-la-Tente et al. (2019).

Es necesario mencionar que para cada una de las funciones del paquete `forecast` (Hyndman, 2019) que indicamos en la explicación de cada método, utilizadas para la modelización automática de las

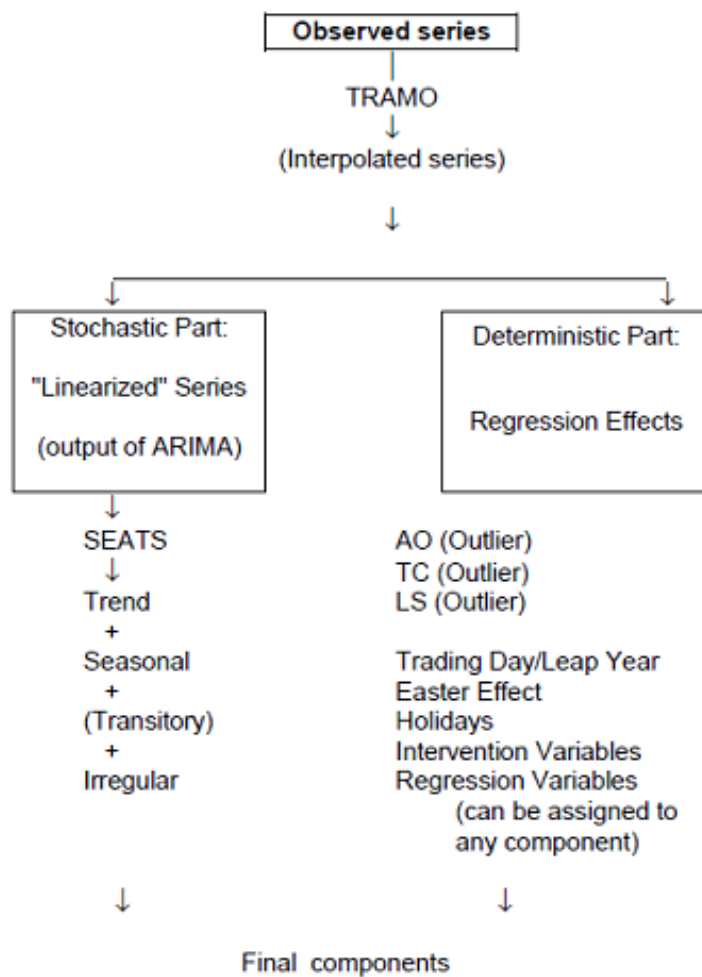


Figura 2.2: Esquema del método TRAMO-SEATS (Maravall, 2008).

series, debemos aplicarle después la función `forecast` para que se calculen las predicciones junto con sus intervalos de predicción. Sin embargo, las funciones del paquete `RJDemetra` no calculan automáticamente los intervalos de predicción, pero utilizando la especificación `preprocessing.model.y_ef` la función te devuelve el error estándar de predicción con el que podemos calcular los intervalos de la forma

$$\text{predicción} \pm K * \text{error estandar de predicción},$$

donde K es el multiplicador del error estándar tomado la tabla de la distribución normal. Como nosotros trabajamos con intervalos de predicción al 95% de confianza, $K = 1,96$.

Capítulo 3

Desarrollo de los métodos

Una primera idea que se nos ocurrió para abordar el problema de la coherencia del último dato, fue la de comprobar cual de los métodos de predicción automáticos existentes predice mejor, para así calcular las predicciones con sus intervalos de predicción siempre por dicho método. A esta idea que desarrollaremos y explicaremos en la tercera sección de este capítulo le llamaremos *Best Forecasting Methods* (BFM) por ser la traducción de "Mejores Métodos de Predicción" en inglés.

Los resultados generados por el BFM nos llevaron a pensar que dependiendo del tipo de serie temporal con la que estemos trabajando, unos métodos nos generan mejores predicciones que otros, descartando la primera idea que teníamos de predecir siempre con el mismo método. Sin embargo, todo este trabajo hecho hasta ese momento no resultó ser en vano, ya que con ese procedimiento descubrimos cuáles de los métodos de predicción automática existentes en \mathbb{R} producen mejores predicciones de forma general.

Una vez hecho todo esto, hemos centrado nuestra atención en estudiar el método Feature-based FOREcasting Model Averaging (FFORMA) de Montero-Manso et al. (2018) porque está programado en el software libre \mathbb{R} en su totalidad y porque los autores proponen una forma de solucionar el problema principal que nos había surgido tras ver los resultados del método BFM (dependiendo de las características de las series unos métodos nos generan mejores soluciones que otros), ya que, para predecir, después de un entrenamiento previo del algoritmo, utiliza una combinación lineal de los diferentes métodos de predicción utilizados basada en las características de las series, asignándole mayor peso a los métodos que estiman mejor ese tipo de serie.

Una vez entramos en detalle en el método FFORMA, nos hemos dado cuenta de que podíamos hacerle algunas modificaciones para que genere mejores predicciones de una forma más rápida. Una de estas modificaciones, por ejemplo, ha sido la de incorporar los métodos que mejores resultados nos habían dado en la aproximación de la tercera sección (BFM) al método FFORMA. Al método con las modificaciones realizadas le llamaremos Modified FFORMA (MFFORMA)

La estructura de este capítulo es de la siguiente forma: en la Sección 3.1 se exponen los trabajos preparatorios realizados a los conjuntos de datos proporcionados por el IGE para quedarnos únicamente con aquellas series que tienen información relevante, en la Sección 3.2 se definen las medidas de error utilizadas en la aplicación de los métodos a datos reales del Capítulo 4, en la Sección 3.3 se explica de forma detallada el algoritmo BFM, y por último, el método FFORMA y el MFFORMA se explican de forma rigurosa en la Sección 3.4.

3.1. Trabajo preparatorio

Antes de comenzar a predecir fue necesario realizar un trabajo preparatorio de los datos, para quedarnos únicamente con aquellas series que tenían información suficientemente relevante como para trabajar con ellas. Este trabajo preparatorio lo hemos programado para que se pueda utilizar para cualquier conjunto de datos existente en la base de datos del IGE.

Los pasos seguidos en este trabajo preparatorio, una vez cargados en memoria los conjuntos de datos, han sido los siguientes:

1. Encontrar las variables categóricas de cada conjunto de series, es decir, tomar aquellas variables que nos dan lugar a diferentes series de tiempo, para saber el número total de series con las que vamos a trabajar y poder identificarlas.
2. Una vez conocemos el número total de series las ordenamos de forma cronológica.
3. Creamos dos variables más para cada conjunto de series, una que nos indique el año y otra el mes, con el propósito convertir los datos luego a formato serie temporal con las fechas adecuadas.
4. Identificación y eliminación de aquellas series que no tengan ningún dato o menos de 24, ya que no nos proporcionan información suficiente como para trabajar con ellas.
5. Cálculo automático de la frecuencia de la serie.
6. Transformación automática de los datos a formato serie temporal para poder realizar las predicciones y el cálculo de los intervalos de predicción correspondientes.

3.2. Errores

La medida de error que utilizamos para medir la precisión de las predicciones a lo largo de la memoria es el error OWA (Overall Weighed Average).

Cabe mencionar que comenzamos utilizando el Error Cuadrático Medio (ECM) como medida de error, pero debido a la similitud de los resultados y a la mejor interpretación del error OWA nos decantamos por trabajar con éste último. Estas medidas de error están definidas de la siguiente forma:

- El Error Cuadrático Medio (ECM) como

$$ECM = \frac{1}{h} \sum_{t=1}^h (X_t - \hat{X}_t)^2, \quad (3.1)$$

donde X_t son las observaciones reales en el instante t , \hat{X}_t son las predicciones para el instante t y h es el horizonte de predicción.

- El error Overall Weighed Average (OWA) está basado en dos medidas de precisión que son el Error Escalado Absoluto Medio (Mean Absolute Scalated Error, MASE) y el Error de Porcentaje Absoluto Medio simétrico (symmetric Mean Absolute Percentage Error, sMAPE) que se calculan:

$$MASE = \frac{1}{h} \frac{\sum_{t=1}^h |X_t - \hat{X}_t|}{\frac{1}{n-m} \sum_{t=m+1}^n |X_t - X_{t-m}|}, \quad (3.2)$$

$$sMAPE = \frac{1}{h} \sum_{t=1}^h \frac{2|X_t - \hat{X}_t|}{|X_t| + |\hat{X}_t|}, \quad (3.3)$$

donde X_t son las observaciones reales en el instante t , \hat{X}_t son las predicciones para el instante t , h es el horizonte de predicción y m la frecuencia de la serie.

Sabiendo esto, el OWA se calcula como una media entre los errores MASE y sMAPE relativos, calculados tras dividir los errores originales entre el error obtenido por las predicciones del método naïve estacional.

La interpretación del error OWA proviene de que los errores MASE y sMAPE relativos se calculan a partir de dividir entre el error MASE y sMAPE de las predicciones obtenidas por el método naïve estacional, respectivamente. Por tanto, que el error OWA sea mayor que 1 significa que los errores MASE y sMAPE del método de estudio son mayores que los del método naïve estacional (siendo éste uno de los métodos de predicción más simples, tal y como dijimos en el Capítulo 2, pudiendo concluir que las predicciones no son muy precisas). Si el error OWA es menor que 1 significa que las predicciones obtenidas por ese método son mejores que las obtenidas por el método naïve estacional, y si son iguales a 1 significa que las predicciones tiene los mismos errores MASE y sMAPE que las predicciones del método naïve estacional, como era de esperar.

3.3. Best Forecasting Methods (BFM)

Con el algoritmo BFM vamos a obtener los métodos automáticos que mejor predicen de forma general, a pesar de que lo que pretendíamos conseguir era quedarnos con un único método que utilizar para predecir siempre, tal y como hemos dicho al principio del capítulo.

Para obtener estos resultados proponemos el siguiente método.

3.3.1. Metodología BFM

Para cada una de las series temporales con las que nos quedamos tras aplicar el trabajo preparatorio a los conjuntos de datos, la primera idea que tuvimos fue la de eliminarle el último año de las observaciones y hacer una predicción a futuro de este año (12 valores para las series mensuales y 4 para las trimestrales) por cada uno de los métodos automáticos que explicamos en la segunda sección del Capítulo 2, controlando los tiempos de ejecución de cada método. Una vez calculadas las predicciones calcular los errores OWA, definidos en la sección anterior, entre las predicciones y las observaciones reales como criterio para elegir el método que mejor predice (aquel que tenga menor error).

Los métodos utilizados en esta primera aproximación fueron los siguientes.

- Algoritmo ARIMA automático (`auto.arima, forecast`).
- Bagging (`baggedmodel, forecast`)
- BATS (`bats, forecast`)
- TBATS (`tbats, forecast`)
- Suavización exponencial (`ets, forecast`)

- Redes neuronales (`nnetar`, `forecast`)
- STL (`stlm`, `forecast`)
- Theta (`thetaf`, `forecast`)
- X13-ARIMA-SEATS (`seas`, `seasonal`)
- X13-ARIMA-SEATS (`x13_def`, `RJDemetra`)
- TRAMO-SEATS (`tramoseats_def`, `RJDemetra`)

Entre paréntesis se encuentra la función que ajusta ese método, junto al paquete de R en que se encuentra. Para todas las funciones se utilizan los argumentos que tiene por defecto, con la diferencia de que a las funciones del paquete `RJDemetra` tenemos que añadirle la especificación `preprocessing.model.y_ef` para obtener el error estándar de predicción y así poder calcular los intervalos de predicción.

Los resultados de esta primera aproximación nos condujeron a pensar que realizar la predicción para un único año no es un procedimiento consistente para verificar qué método es el que mejor predice, aunque nos ha servido para descartar algunos de los métodos de predicción utilizados y para añadir otros para la siguiente aproximación. Los métodos que no vamos a considerar por su largo tiempo de ejecución y por la cantidad de errores que nos producía a la hora de generar predicciones son el método Bagging y los métodos STL y X13-ARIMA-SEATS (del paquete `seasonal`), respectivamente. Sin embargo, trabajaremos con los métodos naïve (`naive`), naïve estacional (`snaive`), y paseo aleatorio (`rwf`), todos ellos pertenecientes al paquete `forecast`.

Para tener más datos con los que contrastar qué método predice mejor con la finalidad de poder captar más anomalías que estas presentan y ver como se comportan los métodos ante ellas, lo que vamos a hacer es dividir cada una de las series a la mitad e ir prediciendo con todos los métodos mencionados en el párrafo anterior, año a año hasta el final de la serie de forma recurrente, a partir de las observaciones reales. Una vez tenemos un número de predicciones igual a la mitad del tamaño de la serie, las comparamos con las observaciones reales mediante el error OWA para sacar las conclusiones oportunas. Los resultados obtenidos por esta última aproximación los expondremos en el siguiente capítulo.

3.4. FFORMA

El método Feature-based Forecast Model Averaging (FFORMA) creado por Montero-Manso et al. (2018), que alcanzó la segunda posición de la competición M4 (M4 Competitor's Guide, 2018) y en el cuál nos basaremos para la redacción de esta sección, surge de los numerosos estudios aplicados que muestran que una combinación lineal de predicciones es mejor que las predicciones obtenidas de cada método independientemente (ver Clemen (1989) o Timmermann (2006), por ejemplo). Esto es debido al riesgo existente de seleccionar el método equivocado a la hora de predecir. Para obtener una combinación de pesos para las predicciones de los métodos que sea adecuada, los autores utilizan un algoritmo de meta-aprendizaje basado en las características de las series. Además, Montero-Manso et al. (2018) también nos proporcionan una manera para calcular los intervalos de predicción que explicaremos en el segundo punto de esta sección.

Los elementos necesarios para utilizar este método (entradas del algoritmo de meta-aprendizaje) son: un conjunto de series temporales de gran tamaño para que pueda entrenar el algoritmo, un conjunto de métodos de predicción que calcule las predicciones y un conjunto de funciones que calculen las características de las series de tiempo.

El conjunto de series temporales con el que trabajan, proporcionado por la propia competición M4 y al que llaman conjunto de referencia $X = x_1, \dots, x_k$, está formado por 100,000 series de tiempo de todo tipo ($k=100.000$), anuales, cuatrimestrales, mensuales, semanales, diarias y horarias. Cada serie se divide en un periodo de entrenamiento y un periodo de prueba. La longitud de estos periodos para cada serie de tiempo está prefijada por la competición, siendo el periodo de prueba igual al horizonte de predicción y el periodo de entrenamiento la parte de la serie de tiempo restante. Series con menos de dos observaciones en el periodo de entrenamiento o series constantes, son eliminadas del conjunto de referencia.

Los diferentes métodos de predicción que se utilizan para generar las predicciones en el FFORMA son:

1. Naïve (`naive, forecast`)
2. Paseo aleatorio (`rwf, forecast`)
3. Naïve estacionario (`snaive, forecast`)
4. Theta (`thetaf, forecast`)
5. Algoritmo ARIMA automático (`auto.arima, forecast`).
6. Algoritmo de suavización exponencial automático (`ets, forecast`)
7. TBATS (`tbats, forecast`)
8. STL (`stlm, forecast`)

Las funciones de R que hacen referencia a los métodos están especificadas entre paréntesis junto con el paquete al que pertenecen (`forecast`, todas ellas). En caso de que no se puedan calcular las respectivas predicciones por cuestión de que se produzca algún error, se utilizan las predicciones obtenidas por la función `snaive` en su lugar. A este conjunto de métodos de predicción le llamaremos M para simplificar la notación.

El conjunto de características de series temporales que utiliza este método se encuentran en la Tabla 3.1. Dichas características están implementadas en el paquete de R, `tsfeatures`, creado por Hyndman et al. (2018b) para otros contextos de predicción de series temporales como Talagala et al. (2018). En este último artículo las características están descritas con más detalle. Al conjunto de funciones que calculan las características de las series lo denominaremos F . Cabe mencionar que las características relacionadas con la estacionalidad de las series se fijan a cero para series sin estacionalidad.

Las características de la Tabla 3.1 más comunes que se repiten con más frecuencia en las series temporales con las que trabajaremos en el siguiente capítulo son:

- Tendencia: comportamiento a largo plazo, que se produce en relación al nivel medio de la serie.
- Estacionalidad: comportamiento periódico de la serie.

- Autocorrelaciones: relación o dependencia de una variable consigo misma en instantes temporales diferentes.
- Linealidad: comportamiento lineal o estable en torno a la media.
- Curvatura: comportamiento curvilíneo de la serie
- Exponente de Hurst: valor entre 0 y 1 que expresa que estamos ante una serie tendencial si dicho valor es mayor que 0.5, que estamos ante una serie totalmente aleatoria si este valor es igual a 0.5, y que la serie tiene una dinámica de reversión a la media si el valor es menor que 0.5.
- Entropía: valor que cuantifica la regularidad de un sistema, de forma que cuanto más regular es una serie, más predecible y menos compleja será por lo que su entropía será menor.
- Estadístico ARCH.LM: procedimiento de detección de heterocedasticidad condicional autoregresiva, es decir, variabilidad de los residuos (errores) de la serie.

Ahora que conocemos estos elementos básicos que necesita el algoritmo para su funcionamiento, vamos a explicar la metodología del mismo.

3.4.1. Metodología

El método FFORMA consta de dos fases, una primera fase donde se entrena el modelo de aprendizaje, y una segunda fase donde se predice una nueva serie de tiempo.

Para la realización de la primera fase necesitamos darle como argumentos de entrada los elementos mencionados al principio de la Sección 3.4 (conjunto de referencia, M y F). Una vez dividido el conjunto de referencia en un periodo de entrenamiento y un periodo de prueba, se calculan para el periodo de entrenamiento, tanto las características de las series de tiempo con el conjunto de funciones F (como por ejemplo la longitud de la serie, la tendencia, la estacionalidad, linealidad, autocorrelaciones... ver Tabla 3.1), como las predicciones calculadas por cada uno de los métodos del conjunto M . A partir de las predicciones se calculan los errores de predicción OWA haciendo uso del periodo de prueba y se le asigna una combinación de pesos a cada método de forma que se minimice:

$$\arg \min_w \sum_{n=1}^N \sum_{m=1}^M w(f_n)_m L_{nm}, \quad (3.4)$$

donde $w(f_n)_m$ es el peso obtenido tras aplicar la transformación *softmax* (transformación que convierte valores denominados *logits* en probabilidades de la siguiente forma, $S(y_i) = \frac{\exp(y_i)}{\sum_j \exp(y_j)}$) a la salida del algoritmo de meta-aprendizaje para el método m y la serie temporal x_n del conjunto de referencia (ver Montero-Manso (2018)), f_n es el conjunto de características de la serie sobre el periodo de entrenamiento, L_{nm} es el conjunto de errores de predicción sobre el periodo de prueba, N es el número de series de tiempo del conjunto de referencia y M es el número de métodos de predicción empleados.

De esta forma, el modelo de meta-aprendizaje aprende a producir pesos para cada uno de los métodos utilizados, es decir, el modelo crea una función que a partir de las características de las series genera los pesos para cada método del conjunto M . La generación de los pesos hecha por el modelo de meta-aprendizaje utiliza técnicas aleatorias, con lo que el cálculo de predicciones para una misma

Característica	Descripción	Estacional	No estacional
1 T	longitud de la serie temporal	✓	✓
2 trend	fuerza de la tendencia	✓	✓
3 seasonality	fuerza de la estacionalidad	✓	-
4 linearity	linealidad	✓	✓
5 curvature	curvatura	✓	✓
6 spikiness	aspecto punzante	✓	✓
7 e_acf1	primer valor ACF de la serie restante	✓	✓
8 e_acf10	suma de cuadrados de los 10 primeros valores ACF de la serie restante	✓	✓
9 stability	estabilidad	✓	✓
10 lumpiness	aspecto abultado	✓	✓
11 entropy	entropía espectral	✓	✓
12 hurst	Exponente de Hurst	✓	✓
13 nonlinearity	no linealidad	✓	✓
14 alpha	ETS(A,A,N) $\hat{\alpha}$	✓	✓
15 beta	ETS(A,A,N) $\hat{\beta}$	✓	✓
16 hwalpha	ETS(A,A,A) $\hat{\beta}$	✓	-
17 hwbeta	ETS(A,A,A) $\hat{\alpha}$	✓	-
18 hwgamma	ETS(A,A,A) $\hat{\gamma}$	✓	-
19 ur_pp	estadístico basado en el test de Phillios-Perron	✓	✓
20 ur_kpss	estadístico basado en el test KPSS	✓	✓
21 y_acf1	primer valor ACF de la serie original	✓	✓
22 diff1y_acf1	primer valor ACF de la serie diferenciada	✓	✓
23 diff2y_acf1	primer valor ACF de la serie con 2 diferencias	✓	✓
24 y_acf10	suma de cuadrados de los 10 primeros valores ACF de la serie original	✓	✓
25 diff1y_acf10	suma de cuadrados de los 10 primeros valores ACF de la serie diferenciada	✓	✓
26 diff2y_acf10	suma de cuadrados de los 10 primeros valores ACF de la serie con 2 diferencias	✓	✓
27 seas_acf1	coeficiente de autocorrelación del primer retardo estacional	✓	-
28 sediff_acf1	primer valor ACF de la serie diferenciada estacionalmente	✓	-
29 y_pacf5	suma de cuadrados de los 5 primeros valores ACF de la serie original	✓	✓
30 diff1y_acf5	suma de cuadrados de los 5 primeros valores ACF de la serie diferenciada	✓	✓
31 diff2y_acf5	suma de cuadrados de los 5 primeros valores ACF de la serie con 2 diferencias	✓	✓
32 seas_pacf	coeficiente de autocorrelación parcial del primer retardo estacional	✓	✓
33 crossing_point	número de veces que la serie cruza la mediana	✓	✓
34 flat_spots	número de puntos planos	✓	✓
35 nperiods	número de periodos estacionales en la serie	✓	-
36 seasonal_period	longitud del periodo estacional	✓	-
37 peak	fuerza de los picos	✓	✓
38 trough	fuerza de los valles	✓	✓
39 ARCH.LM	estadístico ARCH.LM	✓	✓
40 arch_acf	suma de cuadrados de las 12 primeras autocorrelaciones de la serie preblanqueada	✓	✓
41 garch_acf	suma de cuadrados de las 12 primeras autocorrelaciones de los residuos de la serie preblanqueada	✓	✓
42 arch_r2	R^2 de un modelo AR aplicado a la serie preblanqueada	✓	✓
43 garch_r2	R^2 de un modelo AR aplicado a los residuos de la serie preblanqueada	✓	✓

Tabla 3.1: Características utilizadas en el método FFORMA (Montero-Manso et al, 2018).

serie puede ser ligeramente distinto. Se supone que las nuevas series provienen de un proceso generador similar al de algunas de las series que teníamos en el conjunto de referencia, según la similitud de sus características.

Los pesos producidos para cada método por el modelo de meta-aprendizaje representa la importancia que le asigna dicho modelo a cada uno de los métodos para cada serie. Cabe mencionar que la suma de los pesos debe dar 1, por lo tanto, valores asignados a un método próximos a 1 significa que dicho método genera buenas predicciones para esa serie en concreto, y valores cercanos a 0, que ese método produce malas predicciones para la serie.

La segunda fase, es la de predicción de una nueva serie de tiempo a partir del modelo de meta-aprendizaje pre-entrenado. En esta fase se calcula una nueva combinación de pesos a partir de las características de la nueva serie y a continuación se calcula sus predicciones. Para una explicación esquematizada sobre este algoritmo consultar la Sección 2.2 de Montero-Manso et al. (2018).

Cabe destacar que la implementación del modelo de meta-aprendizaje se basa en el modelo “*gradient tree boosting*” propuesto por Chen y Guestrin (2016), perteneciente a la librería *xgboost* de Chen et al. (2018). Dicha librería permite personalizar el modelo con respecto a la función objetivo. La función *xgboost* requiere del gradiente y de la hesiana de la función objetivo para ajustar el modelo. Para más información sobre el modelo de meta-aprendizaje consultar la Sección 3.5 de Montero-Manso et al. (2018). En la Figura 3.1 se muestra un esquema completo de todo el procedimiento que se sigue para obtener las predicciones por el método FFORMA.

3.4.2. Intervalos de predicción FFORMA

Los intervalos de predicción al 95 % de confianza propuestos por Montero-Manso et al. (2018) para las predicciones son de la siguiente forma:

Se toma la predicción obtenida por el método FFORMA como centro del intervalo de predicción, y los extremos de los intervalos con un 95 % de confianza se generan mediante el siguiente procedimiento:

1. Se divide el conjunto de datos de la competición M4 en dos partes, A y B. Se entrena el algoritmo de aprendizaje FFORMA utilizando los periodos de entrenamiento de las series de la parte A para generar predicciones sobre los periodos de entrenamiento de la parte B y viceversa.
2. Se calcula el radio de las predicciones al 95 % de confianza para los métodos naïve, Theta y naïve estacionario (distancia entre el extremos superior de cada intervalo y su punto medio, para cada uno de esos métodos).
3. Para cada horizonte de predicción se encuentran los coeficientes que minimizan el Mean Scaled Interval Score (MSIS) del intervalo, definido en M4 Competitor’s Guide (2018), tomando la predicción hecha por el método FFORMA como centro del intervalo y una combinación lineal de los radios de los métodos naïve, Theta y naïve estacionario como intervalo. La minimización se realizó mediante *gradient descent* (descenso por gradiente, que es un método general de minimización, que tras localizarse en un punto de una función, trata de descender a un punto mínimo usando la información de la primera derivada. Ver Qian, 1999)) sobre el periodo de prueba de las series.

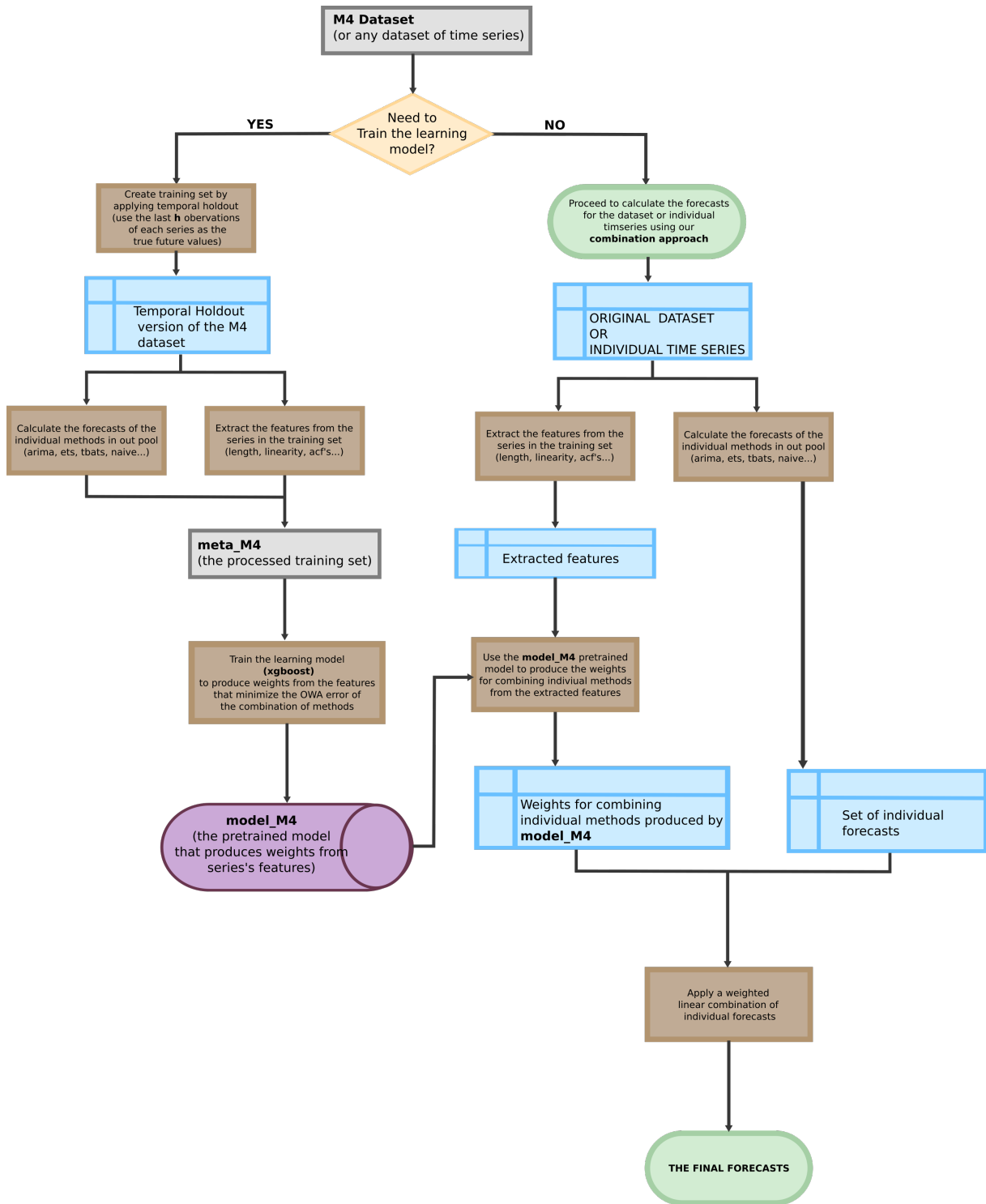


Figura 3.1: Esquema del procedimiento completo del método FFORMA (Hyndman, 2018b).

Como conclusiones generales que nos surgen después de trabajar con este método podemos decir que es un método muy eficiente para la predicción y cálculo de intervalos de predicción para todo tipo de series temporales, tal y como se demostró en la competición M4. Una vez entrenado el algoritmo, el cálculo de la combinación de los pesos para las nuevas series es muy rápida, sin embargo, el inconveniente que le vemos a este método es que para estas nuevas series para las que queremos predecir nuevos valores, se calculan las predicciones por todos los métodos para luego hacer una combinación lineal con las predicciones de todos ellos (aunque a algunos métodos les asigne un peso casi despreciable), siendo este un trabajo muy costoso computacionalmente. Así, las modificaciones que haremos para intentar mejorar el método, tanto en calidad de predicción como en tiempo computacional son:

1. Introducirle al método FFORMA los mejores métodos que obtuvimos al hacer la aproximación BFM.
2. Utilizar para predecir únicamente los métodos cuya suma de pesos, ordenados de mayor a menor, supere 0.8, calculando así las predicciones para una serie por 2 o 3 métodos, en lugar de por 10, reduciendo considerablemente el tiempo de computación, que era otro de nuestros objetivos principales del trabajo.

A continuación describiremos cada una de las modificaciones hechas al método FFORMA con el fin de crear un método de predicción que sea más preciso y más rápido.

3.4.3. Modified FFORMA (MFFORMA)

Para explicar las modificaciones que hemos hecho al método FFORMA, vamos a clasificarlas de acuerdo con el objetivo propuesto en: mejorar la precisión de las predicciones y aumentar la velocidad del cálculo de predicciones. Finalizaremos la sección explicando una forma alternativa para el cálculo de los intervalos de predicción que se adecúa más a las necesidades de nuestro trabajo, siendo a su vez una manera más rápida y sencilla de calcularlos.

El código de R creado para realizar todos estos cambios se incluirá en el anexo para aquellos usuarios que deseen probarlo.

Mejora en la precisión

Según los resultados obtenidos por el método BFM, llegamos a la conclusión de que los métodos individuales existentes en la literatura que mejores predicciones nos ofrecen de forma general (para un mayor número de series) y que no se encuentran ya en el conjunto de métodos M utilizados en el método FFORMA, son el método X13-ARIMA-SEATS y el método TRAMO-SEATS. Esta razón fue la que nos condujo a introducir estos métodos dentro del FFORMA para conseguir una mayor exactitud en las predicciones del método, ya que estos dos métodos son capaces de captar realmente bien el comportamiento del tipo de series temporales utilizadas en este trabajo. Así, el conjunto de métodos que utilizaremos será:

1. Naïve (`naive, forecast`).
2. Paseo aleatorio (`rwf, forecast`).
3. Naïve estacionario (`snaive, forecast`).

4. Theta (`thetaf, forecast`).
5. Algoritmo ARIMA automático (`auto.arima, forecast`).
6. Algoritmo de suavización exponencial automático (`ets, forecast`).
7. TBATS (`tbats, forecast`).
8. STL (`stlm, forecast`).
9. X13-ARIMA-SEATS (`x13_def, RJDemetra`).
10. TRAMO-SEATS (`tramoseats_def, RJDemetra`).

Entre paréntesis se encuentran las funciones de R que hacen referencia a dichos métodos junto al paquete al que pertenecen. Se sigue manteniendo la condición de que en el caso de que no se puedan calcular las predicciones por algún método a causa de que se produzca algún error, se utilicen las predicciones obtenidas por la función `snaive` en su lugar.

Mejora en la velocidad

Para aumentar la velocidad de procesamiento de las predicciones lo que haremos será calcular las predicciones únicamente con los métodos cuya suma de pesos (después de ser ordenados de forma decreciente), supere 0.8, para así evitar el cálculo de las predicciones por los métodos cuyos respectivos pesos sean muy pequeños. Como nos quedaríamos únicamente con el número mínimo de métodos que su suma de pesos sea mayor que 0.8, debemos hacer un re-escalado de los pesos para que la suma de los pesos vuelva a ser 1, ya que estos pesos nos indican la importancia que le da el algoritmo a cada método de predicción, siendo 1 el máximo y 0 el mínimo. Éste re-escalado de los pesos se hace dividiendo el peso asignado a cada método entre la suma de los pesos de los métodos utilizados. De esta forma conseguimos que nuestro método sea más rápido que el método FFORMA a la hora de calcular las predicciones finales.

3.4.4. Intervalos de predicción MFFORMA

El procedimiento que se utiliza para calcular los intervalos de predicción ha sido el más razonable e intuitivo con respecto a la forma en que calculamos las predicciones, ahorrándonos así el tener que entrenar el algoritmo también para el cálculo de los intervalos de predicción. Lo que hacemos es una combinación lineal de los intervalos de predicción de los métodos utilizados para predecir, siendo la combinación de pesos utilizada para el cálculo de los intervalos de predicción, la misma que para calcular las predicciones.

Para poder hacer esto, hemos tenido que crear y modificar un gran número de funciones del método FFORMA, para así poder calcular al mismo tiempo tanto las predicciones como los intervalos de predicción. Ésta información la guardamos junto con el resto de información que necesita el algoritmo sobre cada serie, para poder acceder a ella cuando lo necesitemos.

El conjunto de funciones que hemos tenido que crear o modificar son las siguientes:

1. Función que calcule los intervalos de predicción de las funciones del paquete `RJDemetra`.

2. Funciones que guarden en un *data frame*, tanto las predicciones como sus intervalos de predicción, para cada método de predicción.
3. Bucle que calcule aquellos métodos que vamos a utilizar para la predicción y los pesos re-escalados (ya utilizado en la Sección 3.4.3).
4. Función que llame a las anteriores para calcular las predicciones y los intervalos de predicción, únicamente por los métodos necesarios, es decir, por aquellos métodos cuya suma de pesos supera 0.8 (esta función ya la habíamos creado anteriormente para la Sección 3.4.3).
5. Funciones que calculen los intervalos de predicción finales como una combinación lineal de los intervalos de los métodos utilizados para la predicción.

En el IGE buscaban encontrar unos intervalos de predicción que tuviesen una cobertura del 95 %, que ajustase bien el valor de las predicciones, y que a su vez, las longitudes del intervalo, no fuesen tan pequeñas como para que se detecte un error cada vez que se introduzca un dato que tenga variabilidad.

Tras comprobar los resultados obtenidos y ver que respondían mejor a nuestras necesidades que los intervalos generados por el método FFORMA, nos decantamos por trabajar con ellos.

Capítulo 4

Aplicación a datos reales

En este capítulo se aplican los métodos explicados en el capítulo anterior a las bases de datos de interés para el IGE, y que se comentaron brevemente en la Introducción. Se trata de ilustrar con datos reales el funcionamiento de estos métodos en distintos contextos. El código de R desarrollado para realizar todos estos procedimientos y las tablas con los resultados globales se incluirán de forma estructurada en los anexos.

Para facilitar la lectura y comprensión de los procedimientos, a lo largo de este capítulo mostraremos las tablas de resultados y gráficas para dos series de cada conjunto de datos de los mencionados en la Introducción (10 series en total, que llamaremos conjunto de series de referencia). La idea es presentar estas series y hacer su seguimiento a lo largo de todo el capítulo. En primer lugar les haremos un análisis exploratorio, a continuación les aplicaremos el método BFM para comprobar empíricamente cuáles son los métodos que mejor predicen de forma general. Finalizamos el capítulo mostrando para estas series del conjunto de referencia una comparación del método FFORMA con el MFFORMA; teniendo en cuenta la precisión de las predicciones por medio del error OWA, el tiempo de ejecución, y la cobertura y longitud de los intervalos de predicción. Con esta muestra, se busca reflejar los resultados del conjunto global de series, es decir, informar del comportamiento global de los métodos.

4.1. Análisis exploratorio

Los conjuntos de datos con los que trabajamos, que mencionamos en el Capítulo de Introducción y que explicaremos a continuación, son todos muy heterogéneos y de diferente naturaleza. Con el fin de eliminar de cada uno de estos conjuntos las series que no disponían de datos suficientes como para trabajar con ellas, les aplicamos el trabajo preparatorio explicado en la Sección 3.1. Así, la Tabla 4.1 nos muestra, en la primera fila, el número de series que había inicialmente en cada conjunto y en la segunda fila, el número de series con que nos quedamos después de realizar el trabajo preparatorio.

A continuación expondremos el contenido de cada conjunto de datos y analizaremos las características y el comportamiento de las dos series que tomamos como referencia de cada conjunto.

Conjunto 1	Conjunto 2	Conjunto 3	Conjunto 4	Conjunto 5
108	72	55	42	162
84	60	50	42	90

Tabla 4.1: Número de series temporales que hay en cada conjunto de datos antes y después del trabajo preparatorio.

Conjunto 1:

Series de datos mensuales que contienen la información de los viajeros, noches y estancia media en establecimientos hoteleros y de turismo rural en España, Galicia y sus provincias.

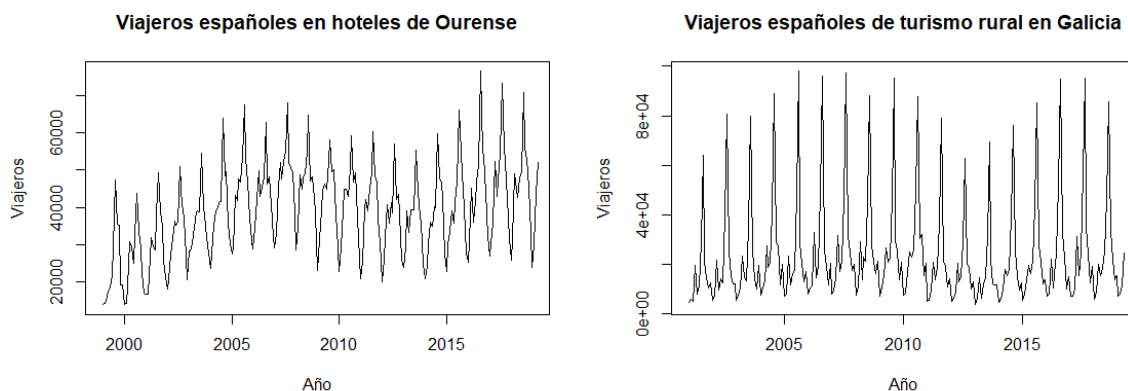


Figura 4.1: Series tomadas como muestra de referencia del Conjunto 1, de series de turismo. La gráfica de la izquierda hace referencia al número de turistas residentes en España hospedados en hoteles en Ourense y la segunda al número de turistas residentes en España haciendo turismo rural en Galicia.

En la Figura 4.1 se muestran las dos series que tomamos como referencia del Conjunto 1 y que denominaremos como Serie 1 y Serie 2, respectivamente. La gráfica de la izquierda (Serie 1) hace referencia al número de turistas residentes en España hospedados en hoteles en Ourense y tiene 244 observaciones. La serie de la derecha (Serie 2) hace referencia al número de turistas residentes en España haciendo turismo rural en Galicia y tiene 220 observaciones. Según las características extraídas por la función `calc_forecasts`, creada por Montero-Manso et al. (2018) para el método FFORMA y que también utilizaremos nosotros en el MFFORMA, la Serie 1 presenta en concreto: una fuerte estacionalidad, gran autocorrelación, tendencia, exponente de Hurst igual a 0.99, lo que nos indica que es una serie tendencial, y entropía igual a 0.76, lo que significa que es una serie irregular y compleja y la Serie 2: una fuerte estacionalidad, gran autocorrelación, exponente de Hurst igual a 0.79, lo que indica que es una serie tendencial, y entropía igual a 0.84, que significa que es una serie irregular.

Conjunto 2:

Series de datos trimestrales que tratan sobre la población de 16 y más años por sexo, grupos de edad y relación con la actividad económica en Galicia.

En la Figura 4.2 se muestra a la izquierda la Serie 3, que hace referencia al número total de personas ocupadas en Galicia y a la derecha, la Serie 4, que muestra el número total de personas paradas en Galicia. Ambas series tienen 93 observaciones. Las características más destacables que presenta la Serie 3 son: autocorrelación, tendencia, exponente de Hurst igual a 0.99, lo cual indica que es una serie tendencial, y entropía igual a 0.55, lo que significa que no es una serie demasiado irregular. La Serie 4 presenta características muy similares, autocorrelación, tendencia, exponente de Hurst igual a 0.99, lo cual indica que es una serie tendencial, y entropía igual a 0.57, lo que significa que tampoco es una serie demasiado irregular y compleja.

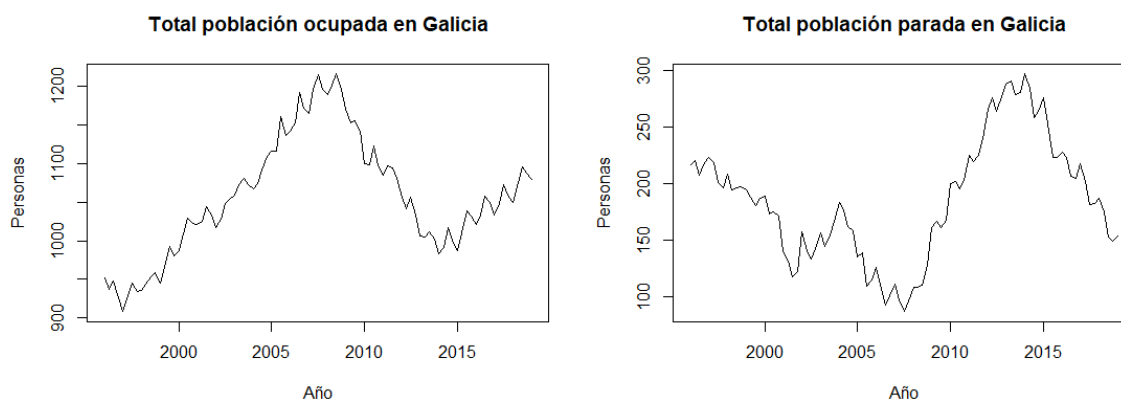


Figura 4.2: Series tomadas como muestra de referencia del Conjunto 2, de series de actividad en Galicia. La gráfica de la izquierda hace referencia al total de la población ocupada en Galicia y la segunda al total de la población parada en Galicia.

Conjunto 3:

Series de datos mensuales que tratan sobre los contratos registrados según la modalidad.

La Figura 4.3 nos muestra a la izquierda la Serie 5 que representa el número de contratos iniciales registrados en Galicia y a la derecha la Serie 6 que muestra el número de contratos indefinidos registrados en Galicia. Las dos series constan de 245 datos. La Serie 5 presenta las siguientes características: variabilidad de los residuos (errores), entropía, fuerte estacionalidad, tendencia, curvatura, exponente de Hurst igual a 0.97 que nos indica que es una serie tendencial y entropía igual a 0.77 lo que significa que es una serie bastante irregular. La Serie 6 presenta: Tendencia, inestabilidad, exponente de Hurst igual a 0.99, que nos indica que es una serie tendencial, y entropía igual a 0.63, lo que significa que es una serie irregular.

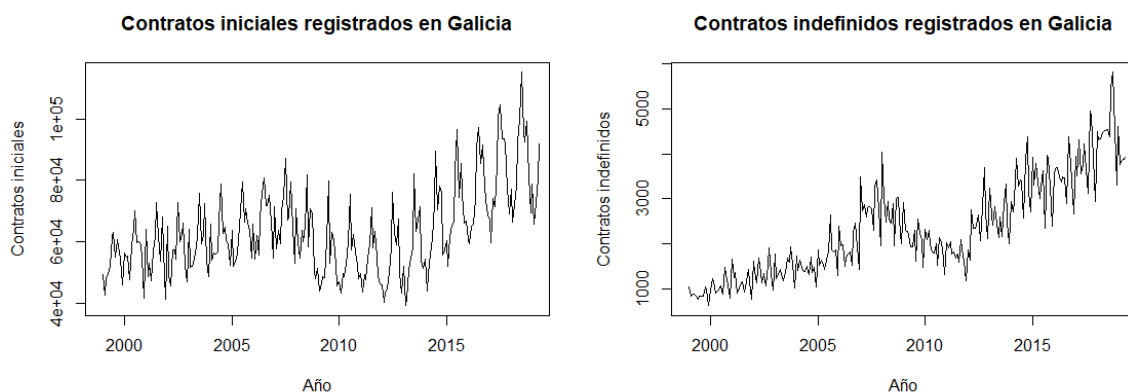


Figura 4.3: Series tomadas como muestra de referencia del Conjunto 3, de series de la modalidad de contratos registrados en Galicia. La gráfica de la izquierda hace referencia al número de contratos iniciales en Galicia y la de la derecha al número de contratos indefinidos en Galicia.

Conjunto 4:

Conjunto de series sobre el índice de producción industrial general y por destino económico de los bienes. Se trata de un conjunto compuesto por series de datos mensuales.

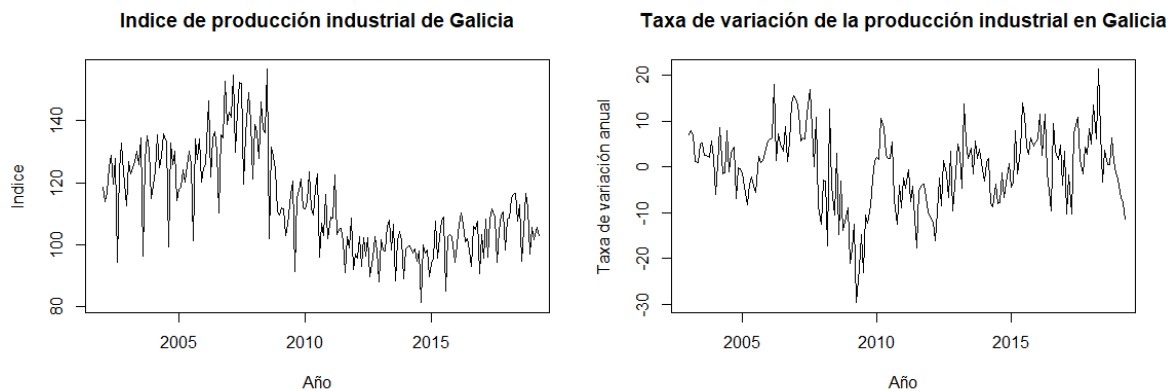


Figura 4.4: Series tomadas como muestra de referencia del Conjunto 4, de series producción industrial. La gráfica de la izquierda hace referencia al índice de producción industrial de Galicia y la de la derecha la taxa variación anual de la producción industrial en Galicia.

La Figura 4.4 nos muestra a la izquierda la Serie 7 que hace referencia al índice de producción industrial de Galicia y la de la derecha, la Serie 8, que muestra la taxa variación anual de la producción industrial en Galicia. Cada serie tiene 208 y 196 elementos, respectivamente. Las características más destacadas que presenta la Serie 7 son: tendencia, curvatura, estacionalidad, autocorrelaciones, exponente de Hurst igual a 0.89, que nos indica que es una serie tendencial, y entropía igual a 0.68, lo que significa que es una serie irregular; y la Serie 8: aparente linealidad a lo largo del tiempo y

exponente de Hurst igual a 0.94, que nos indica que es una serie tendencial, y entropía igual a 0.84, lo que significa que es una serie bastante irregular.

Conjunto 5:

Conjunto de series trimestrales de transacciones inmobiliarias por régimen y tipo de vivienda. Número, valor total y valor medio.

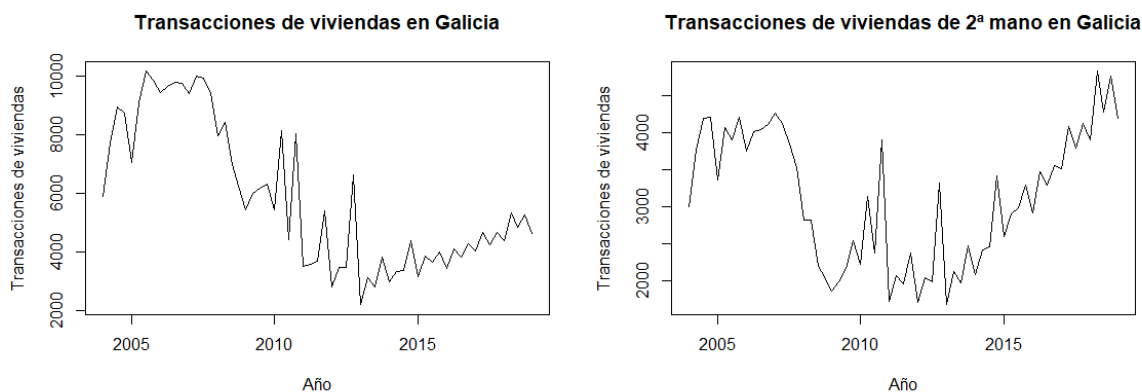


Figura 4.5: Series tomadas como muestra de referencia del Conjunto 5, de series de transacciones inmobiliarias. La gráfica de la izquierda hace referencia al número de transacciones de viviendas en Galicia y la de la derecha al número de transacciones de viviendas de segunda mano en Galicia.

En la Figura 4.5 aparece, en la gráfica de la izquierda, la Serie 9 que hace referencia al número total de transacciones de viviendas en Galicia y, en la gráfica de la derecha, la Serie 10 que presenta el número total de transacciones de viviendas de segunda mano en Galicia. Ambas series constan de 61 observaciones. Las características más destacables de la Serie 9 son: tendencia, estabilidad, autocorrelaciones, exponente de Hurst igual a 0.97, que nos indica que es una serie tendencial, y entropía igual a 0.71, lo que significa que es una serie bastante irregular. Las características de la Serie 10 son: tendencia, autocorrelación, exponente de Hurst igual a 0.96, que nos indica que es una serie tendencial y entropía igual a 0.79, lo que significa que es una serie bastante irregular.

En la Tabla 4.2 se exponen, en las dos primeras columnas, las series del conjunto de referencia con las que vamos a trabajar a lo largo de este capítulo para mostrar los resultados de nuestros métodos con su descripción, y en la tercera columna se muestra el conjunto de series proporcionado por el IGE al que pertenece.

Serie	Descripción	Conjunto
Serie 1	Viajeros españoles en hoteles de Ourense	Conjunto 1
Serie 2	Viajeros españoles haciendo turismo rural en Galicia	Conjunto 1
Serie 3	Población ocupada en Galicia	Conjunto 2
Serie 4	Población parada en Galicia	Conjunto 2
Serie 5	Contratos iniciales registrados en Galicia	Conjunto 3
Serie 6	Contratos indefinidos registrados en Galicia	Conjunto 3
Serie 7	Índice de producción industrial de Galicia	Conjunto 4
Serie 8	Taxa de variación interanual de la producción industrial en Galicia	Conjunto 4
Serie 9	Transacciones de viviendas en Galicia	Conjunto 5
Serie 10	Transacciones de viviendas de segunda mano en Galicia	Conjunto 5

Tabla 4.2: Tabla resumen que asocia las series de estudio al conjunto que pertenecen.

Ahora que tenemos el análisis exploratorio de las series de estudio hecho, podemos pasar a explicar los resultados obtenidos por los métodos utilizados por medio de tablas y gráficas representativas, que muestren el funcionamiento de los mismos de una manera sencilla e intuitiva.

4.2. BFM

Una vez hecho el análisis exploratorio, pasamos a la aproximación del método BFM en la que calculamos las predicciones de la mitad de cada serie en adelante año a año de forma recurrente con las observaciones reales. Para así, calcular posteriormente el error OWA cometido entre todas estas predicciones con las observaciones reales como medida de precisión de las predicciones.

Para la comprobación de este método utilizamos de cada conjunto, todas las series con las que podemos trabajar, es decir, el número de series obtenido después de realizar el trabajo preparatorio a cada conjunto de datos, que aparece reflejado en la Tabla 4.1.

En las Tablas 4.4 y 4.5 expondremos los resultados obtenidos por el método BFM. En la Tabla 4.4 se muestran los errores OWA obtenidos por cada método para las 10 series mencionadas anteriormente, con las que vamos a trabajar durante todo este capítulo. En la Tabla 4.5 se muestra para cada una de estas series, el método que mejor predijo esa serie (el método que menor error OWA tiene en la Tabla 4.4), el tiempo que tardó en realizar las predicciones y el error que nos genera.

Para simplificar la notación le hemos asignado a cada método (función de R) utilizado una notación formado por la letra “M” y un número de la forma mostrada en la Tabla 4.3.

auto.arima	naive	bats	tbats	ets	snaive	nnetar	rwf	thetaf	x13_def	tramoseats_def
M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11

Tabla 4.3: Asignación entre métodos y notación.

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
OWA Serie 1	0.95	2.69	0.99	1.02	1.08	1.00	1.07	2.69	0.98	1.00	1.02
OWA Serie 2	1.09	2.73	0.84	0.93	1.07	1.00	0.88	2.73	0.85	0.70	0.74
OWA Serie 3	0.60	0.70	0.51	0.52	0.62	1.00	0.86	0.70	0.68	0.66	0.65
OWA Serie 4	0.55	0.68	0.62	0.61	0.59	1.00	0.80	0.68	0.67	0.62	0.61
OWA Serie 5	0.92	1.95	1.14	1.23	1.25	1.00	0.97	1.95	1.12	0.81	0.88
OWA Serie 6	0.63	0.95	0.76	0.74	0.65	1.00	0.93	0.95	0.66	0.90	0.70
OWA Serie 7	0.67	0.92	0.82	0.82	0.82	1.00	0.84	0.92	0.82	0.72	0.74
OWA Serie 8	0.86	1.98	0.83	0.97	0.84	1.00	0.90	1.98	0.86	0.70	0.68
OWA Serie 9	0.85	1.50	1.03	1.00	1.16	1.00	1.45	1.50	1.04	0.98	1.04
OWA Serie 10	0.75	1.30	0.70	0.81	0.71	1.00	1.00	1.30	0.81	0.58	0.72

Tabla 4.4: Errores OWA producidos por las predicciones de la mitad de observaciones, generadas por cada método, para las 10 series tomadas como referencia.

En la Tabla 4.4 podemos observar que el error OWA obtenido con el método M6 es siempre 1, como era de esperar, por la definición de errores OWA, ya que relativiza los errores MASE y sMAPE de cada método, dividiendo entre el error MASE y sMAPE del método naïve estacional (M6), resultando un cociente entre elementos iguales. Que el error OWA sea mayor que 1 significa que los errores MASE y sMAPE de ese método son mayores que los del método M6 (siendo éste uno de los métodos de predicción más simples, tal y como se comentó en la Sección 2.2.4) y viceversa. Si el error OWA es menor que 1 significa que las predicciones obtenidas por ese método son mejores que las obtenidas por el método naïve estacional. Para más detalles ver la Sección 3.2.

De forma general podemos asumir que la mayoría de los métodos de predicción utilizados son mejores que el método M6 ya que sus errores OWA suelen ser menores que 1 (salvo en las series 1, 2 o 5, que si revisamos su análisis exploratorio, coinciden con series estacionales, que son el tipo de series que este método predice de mejor manera). De esta tabla también podemos deducir que los métodos M1, M4, M9, M10 y M11, son buenos métodos para la generación de predicciones ya que sus errores suelen ser bastante menores que 1, con lo que podríamos suponer que el método MFFORMA tomará,

en muchas ocasiones, estos métodos en la combinación lineal para la generación de predicciones.

	Mejor método	Tiempo (seg.)	Error OWA
Serie 1	M1	23.52	0.95
Serie 2	M10	2.80	0.70
Serie 3	M3	27.80	0.51
Serie 4	M1	37.77	0.55
Serie 5	M10	9.58	0.81
Serie 6	M1	38.55	0.63
Serie 7	M11	20.26	0.68
Serie 8	M1	101.75	0.67
Serie 9	M1	9.15	0.85
Serie 10	M10	7.14	0.58

Tabla 4.5: Tabla que muestra el modelo que mejores predicciones nos dió para cada serie con respecto al error OWA mostrado en la última columna, y el tiempo en segundos que tardó en generar esas predicciones dicho método en la segunda columna.

En la Tabla 4.5, la primera columna muestra la serie del conjunto de referencia con que estamos trabajando, la segunda columna muestra el método que mejor predijo esa serie, es decir, el método cuyas predicciones produjeron un menor error OWA (ver Tabla 4.4), la tercera columna es el tiempo en segundos que tardó ese método en generar todas las predicciones para esa serie y la última columna son los errores OWA producidos por las predicciones generadas por los respectivos métodos.

Es de interés recoger en una tabla el número de observaciones que tiene cada una de estas series, aunque ya lo hayamos mencionado en su análisis exploratorio, así como el número de predicciones hechas para cada una de ellas en esta aproximación del método BFM, ya que se predice un poco menos que media serie en lugar de un único año para trabajar con periodos más largos de las series con la finalidad de poder captar anomalías que estas presentan, y ver como se comportan los métodos ante ellas. Estos datos se recogen en la Tabla 4.6.

	Serie 1	Serie 2	Serie 3	Serie 4	Serie 5	Serie 6	Serie 7	Serie 8	Serie 9	Serie 10
Observaciones	244	220	93	93	245	245	208	196	61	61
Predicciones	119	107	43	43	119	119	83	83	27	27

Tabla 4.6: Número observaciones (primera fila) que tiene cada serie y número de predicciones (segunda fila) realizadas por cada método para cada serie.

Podemos observar en la Tabla 4.5 que aunque tengamos únicamente 10 series, los métodos que mejores predicciones producen son el M1, el M10 y el M11 ya que entre los 3 predicen el 90% de las series. Estos resultados eran los esperados desde un primer momento ya que estos dos últimos métodos son los más novedosos y complejos utilizados en múltiples entidades para la predicción de series financieras, ya que estos métodos tienen en cuenta todas las puntualidades explicadas en las Secciones 2.2.11 y 2.2.12.

En la Tabla 4.5 podemos observar además que el método M1 tiene un tiempo de ejecución considerablemente mayor que los otros dos. Esto se debe a que, según vimos en la Sección 2.2, el método M1, realiza una búsqueda entre todos los parámetros del modelo para quedarse con el que tiene un menor valor AIC, BIC o AICc, siendo este un proceso costoso, computacionalmente hablando.

En las siguientes gráficas mostramos para estas 10 series, las observaciones reales en color negro, las predicciones hechas con el método que nos da un menor error OWA (Tabla 4.5) en color rojo y sus intervalos de predicción en color azul.

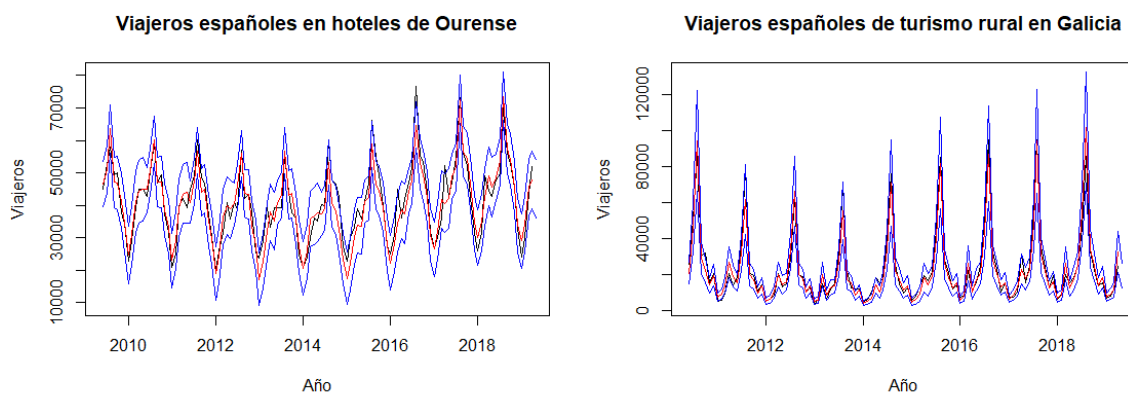


Figura 4.6: Series de referencia del conjunto de series de turismo en las que se muestran las predicciones obtenidas por el método que mejor predice esas series (rojo), junto con las observaciones (negro) y los intervalos de predicción (azul).

La Figura 4.6 representa, a la izquierda, las predicciones junto con los intervalos de predicción generadas por el método M1 para la Serie 1, mientras que a la derecha se muestran las predicciones y los intervalos de predicción calculados por el método M10 para la Serie 2. Las predicciones y los intervalos de predicción se calculan por esos métodos ya que son los que mejor predicen esas series en concreto. Podemos decir que las predicciones generadas por estos dos métodos para estas series son

buenas ya que las predicciones están bastante próximas a las observaciones reales y estas últimas no se salen de los intervalos de predicción calculados en ninguna ocasión.

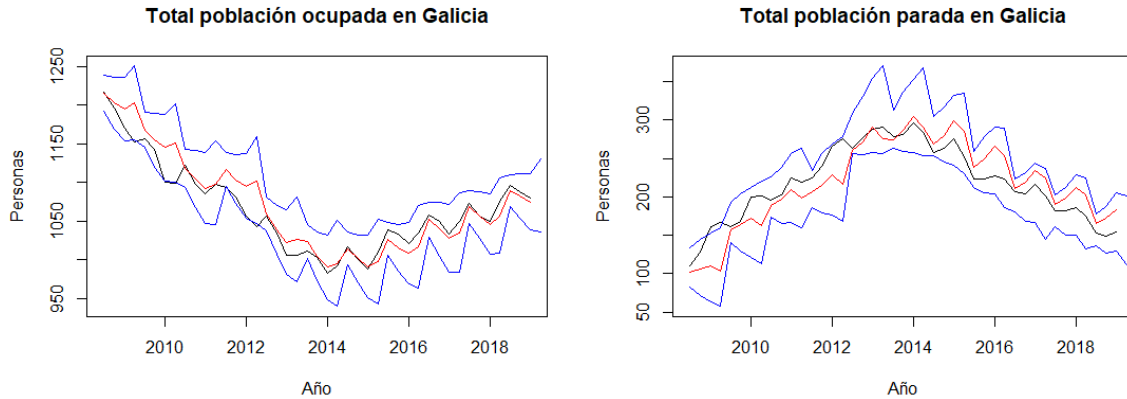


Figura 4.7: Series tomadas de referencia del conjunto de series de actividad económica en las que se muestran las predicciones obtenidas por el mejor método que predice esas series (rojo), junto con las observaciones (negro) y los intervalos de predicción (azul).

En la Figura 4.7 se muestra a la izquierda las predicciones calculadas por el método M3 para la Serie 3 y a la derecha las calculadas por el método M1 para la Serie 4. Las predicciones calculadas para estas dos series no son tan precisas ya que el comportamiento de estas series es más difícil de predecir debido a que es más irregular, tal y como vimos en su análisis exploratorio, provocando que las observaciones reales se escapen de los intervalos de predicción un par de veces en cada gráfica.

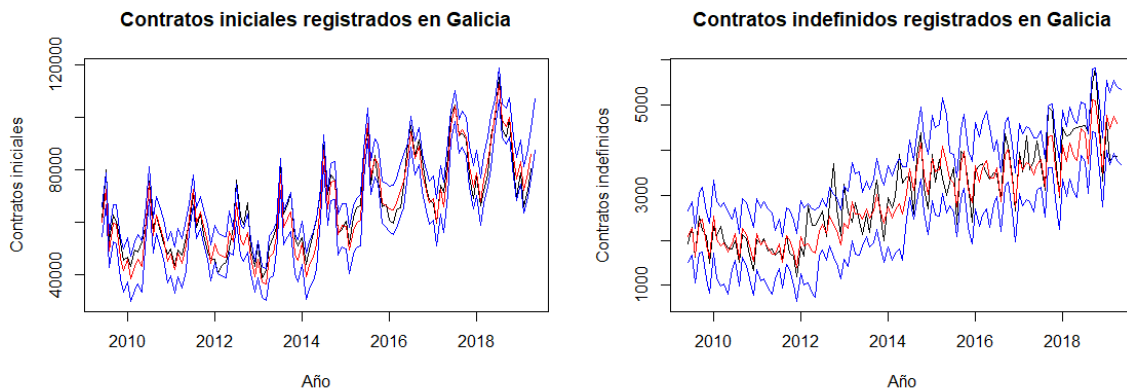


Figura 4.8: Series de referencia del conjunto de series de contratos registrados en las que se muestran las predicciones obtenidas por el método que mejor predice esas series (rojo), junto con las observaciones (negro) y los intervalos de predicción (azul).

En la Figura 4.8 se muestra a la izquierda las predicciones calculadas por el método M10 para la Serie 5 y a la derecha las calculadas por el método M1 para la Serie 6. El comportamiento de la Serie

5 es similar al de las series 1 y 2, mientras que la Serie 6 presenta un comportamiento más irregular, provocando que las predicciones no sean tan precisas y que las observaciones reales se escapen 2 veces de los intervalos de predicción calculados.

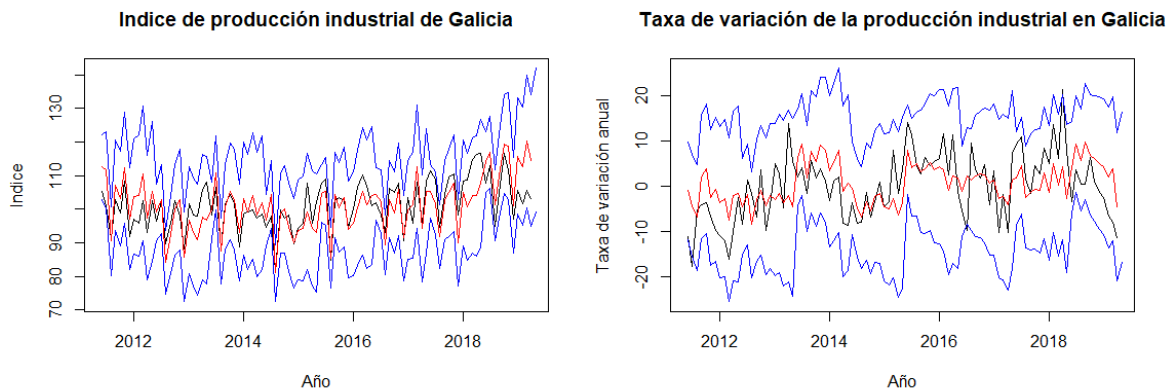


Figura 4.9: Series de referencia del conjunto de series de producción industrial en las que se muestran las predicciones obtenidas por el método que mejor predice esas series (rojo), junto con las observaciones (negro) y los intervalos de predicción (azul).

En la Figura 4.9 se muestra a la izquierda las predicciones calculadas por el método M11 para la Serie 7 y a la derecha las calculadas por el método M1 para la Serie 8. La linealidad de estas dos series facilita en cierta medida el cálculo de las predicciones aunque la variabilidad impide que estas sean demasiado precisas.

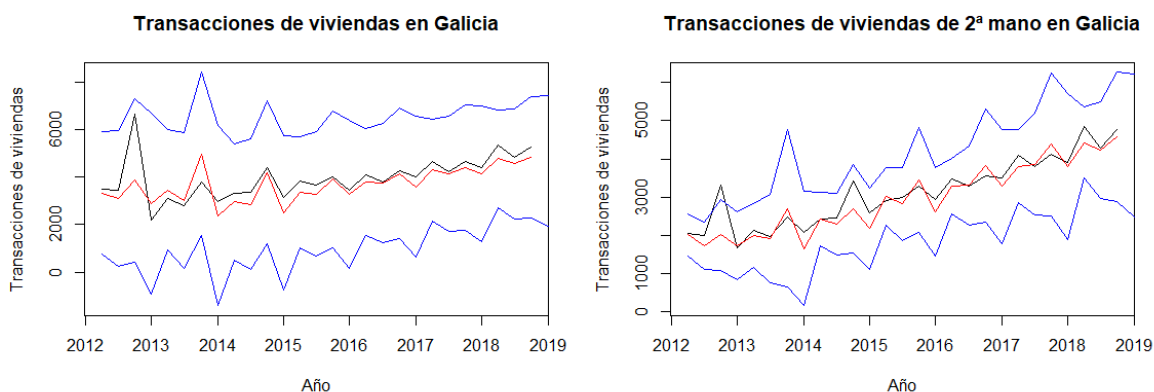


Figura 4.10: Series de referencia del conjunto de series de transacciones inmobiliarias en las que se muestran las predicciones obtenidas por el método que mejor predice esas series (rojo), junto con las observaciones (negro) y los intervalos de predicción (azul).

En la Figura 4.10 se muestra a la izquierda las predicciones calculadas por el método M1 para la Serie 9 y a la derecha las calculadas por el método M10 para la Serie 10. Podemos ver que las predicciones calculadas para estas series son buenas, salvo al principio, debido a su heterocedasticidad.

En las siguientes tablas mostraremos para cada conjunto de series el número de veces que cada método ha sido el que mejor predice basándonos en el error OWA. Que alguno de los métodos utilizados no esté en alguna de estas tablas significa que no ha conseguido predecir de mejor manera ninguna de las series del conjunto estudiado. Es conveniente consultar la Tabla 4.3 para ver las equivalencias entre los códigos y los métodos.

En la Tabla 4.7 podemos ver que de las 84 series temporales que tiene el Conjunto 1, el método M11 ha sido el que mejores predicciones ha generado para 37 series diferentes, y el método M10 ha sido el mejor para 24 de ellas. El número de veces que han predicho de mejor manera con respecto al error OWA los otros métodos podemos verlo en dicha tabla.

	M1	M3	M4	M5	M6	M7	M9	M10	M11
frecuencia	5	4	1	4	6	1	2	24	37

Tabla 4.7: Frecuencia de mejores predicciones de cada método para las series del Conjunto 1.

En la Tabla 4.8 vemos que los métodos que mejores predicciones nos han generado para las 60 series temporales del Conjunto 2 han sido el M1 12 veces, seguido del M10 11 veces y del M11 9 veces. Los demás valores los podemos ver en dicha tabla.

	M1	M2	M3	M4	M5	M6	M7	M9	M10	M11
frecuencia	12	6	4	4	6	2	1	5	11	9

Tabla 4.8: Frecuencia de mejores predicciones de cada método para las series del Conjunto 2.

En la Tabla 4.9 aparece reflejado el número de veces que cada método ha sido el que mejores predicciones ha generado para las 50 series del Conjunto 3. Los métodos que destacan son: el M11 con 18 y el M10 17, seguido más de lejos por las 7 veces del método M1.

	M1	M2	M3	M5	M6	M9	M10	M11
frecuencia	7	1	1	3	1	2	17	18

Tabla 4.9: Frecuencia de mejores predicciones de cada método para las series del Conjunto 3.

En la Tabla 4.10 se muestra el número de veces que cada método ha sido el que mejores predicciones ha generado para las 42 series del Conjunto 4. Los métodos que destacan son: M11 con 17 y M10 con 9.

	M1	M2	M3	M5	M6	M7	M9	M10	M11
frecuencia	5	1	3	1	1	2	3	9	17

Tabla 4.10: Frecuencia de mejores predicciones de cada método para las series del Conjunto 4.

En la Tabla 4.11 se muestra el número de veces que cada método ha sido el que mejores predicciones ha generado para las 90 series del Conjunto 5. Los métodos que destacan son: el M1 con 17, el M2 con 15 y el M4 y M10 con 12.

	M1	M2	M3	M4	M5	M6	M7	M9	M10	M11
frecuencia	17	15	5	12	7	6	3	7	12	6

Tabla 4.11: Frecuencia de mejores predicciones de cada método para las series del Conjunto 5.

Estas tablas nos muestran que los métodos automáticos que mejor predicen de forma general son el M1, el M10 y el M11.

4.3. Comparación del FFORMA con el MFFORMA

Una vez obtuvimos los resultados de la aproximación BFM y pasamos a trabajar con el método FFORMA, tuvimos que ampliar el trabajo preparatorio para poner nuestros datos de la forma que se requería en este método. Por esta razón tuvimos que crear una lista, donde cada elemento de la lista era otra lista con los siguientes elementos: la serie original completa, el número total de observaciones y el horizonte de predicción. Los elementos necesarios para calcular las predicciones para cada serie de tiempo por este método, como por ejemplo las predicciones hechas por cada método o las características de cada serie, se irán guardando como otros elementos de las sub-listas de la lista general.

Gracias a la aproximación calculada por el método BFM, proponemos como método alternativo incluir los métodos M10 y M11 a los que utilizaba el método FFORMA con el fin de refinar las predicciones. Tras seguir trabajando con este método, también nos dimos cuenta de que se podía mejorar el tiempo de ejecución del método sin que variase mucho la calidad de las predicciones, ya que tardaba mucho en generar predicciones para un número considerablemente pequeño de series (50), lo que nos llevó a aplicar el procedimiento explicado en la Sección 3.4.3 con el fin de reducir el tiempo que el FFORMA necesitaba para calcular las predicciones.

La Tabla 4.12 nos muestra el número de series que tomamos de cada conjunto para entrenar (llamadas conjunto de entrenamiento) y para probar el funcionamiento del algoritmo (llamado conjunto de prueba), respectivamente. En primer lugar queríamos tomar 50 series de cada conjunto para entrenar el algoritmo y 10 series diferentes de cada conjunto para probarlo, necesitando un total de 60 series de cada conjunto. Si recordamos la Tabla 4.1 podemos observar que después de aplicarle el trabajo preparatorio a cada conjunto, tenemos conjuntos con menos de 60 series, lo cual no hizo posible ese reparto e hizo decantarnos por la elección expresada en la Tabla 4.12, tomando un total de 222 series

para el conjunto de entrenamiento y 50 series para el conjunto de prueba. Cabe mencionar que las 10 series del conjunto de referencia están incluidas dentro del conjunto de prueba.

	Conjunto 1	Conjunto 2	Conjunto 3	Conjunto 4	Conjunto 5
Conjunto de entrenamiento	50	40	32	50	50
Conjunto de prueba	10	10	10	10	10

Tabla 4.12: Número de series tomadas de cada conjunto tanto para el conjunto de entrenamiento como para el conjunto de prueba.

En la siguiente tabla se muestran los métodos utilizados en el MFFORMA para obtener las predicciones finales como una combinación lineal de las predicciones de esos métodos indicados para las 10 series de referencia. Si comprobamos en la Sección 3.3.1 los métodos utilizados en la aproximación BFM y en la Sección 3.4.3 los métodos utilizados en el método MFFORMA, podemos comprobar que el método STL (función `stlm`) no se utiliza en el primero de ellos, pero sí en el segundo, por esta razón no está incluido en la Tabla 4.3, por lo que a partir de ahora lo denotaremos por M12.

Funciones utilizadas en la predicción final	
Serie 1	M10 y M9
Serie 2	M10 y M9
Serie 3	M11 y M9
Serie 4	M10 y M1
Serie 5	M5 y M6
Serie 6	M5, M1 y M4
Serie 7	M11 y M10
Serie 8	M12, M5, M10, M4 y M1
Serie 9	M11
Serie 10	M11

Tabla 4.13: Métodos utilizados para la combinación lineal de las predicciones del método MFFORMA.

Los métodos están ordenados por el nivel de importancia que le asigna el algoritmo de meta-aprendizaje, es decir, aparecen primero los métodos que tienen más importancia para producir las predicciones finales de cada serie. Se puede observar que el número de métodos utilizado para la predicción es muy inferior al número de métodos utilizados por el FFORMA que es siempre 9, por lo que el tiempo de ejecución necesario para obtener las predicciones finales es mucho menor del método MFFORMA. Este resultado lo podemos comprobar en las columnas 5 y 6 de la Tabla 4.14.

En la Tabla 4.14 mostramos la comparación del método FFORMA con el método MFFORMA a través del sus errores OWA, su cobertura (porcentaje de observaciones reales que están dentro de los

	OWAFF	OWAMF	CobFF	cobMF	TiempoFF	TiempoMF	LongFF	LongMF
Serie 1	0.95	0.99	83.33	100.00	34.50	0.30	10422.41	16625.32
Serie 2	0.52	0.60	83.33	100.00	34.50	0.31	11891.20	19135.54
Serie 3	0.68	0.60	100.00	100.00	11.50	0.53	45.99	71.65
Serie 4	0.21	0.10	100.00	100.00	11.50	7.64	41.41	76.20
Serie 5	0.95	0.94	100.00	100.00	34.50	0.83	12773.94	26636.30
Serie 6	0.62	0.75	75.00	100.00	34.50	79.48	1137.75	2370.05
Serie 7	1.04	1.74	91.67	100.00	34.50	0.53	24.03	32.19
Serie 8	0.95	1.02	83.33	100.00	34.50	38.34	26.38	31.01
Serie 9	0.84	0.32	100.00	100.00	11.50	0.24	3104.17	4477.05
Serie 10	0.54	0.67	100.00	100.00	11.50	0.25	1322.87	2312.09

Tabla 4.14: Tabla de comparación del método FFORMA con el MFFORMA

intervalos de predicción calculados), sus tiempos de computación en segundos y las longitudes de sus intervalos de predicción que varían mucho dependiendo del tipo de unidades de trabajo de cada una de las series, para las 10 series de referencia. “FF” y “MF” indican el método utilizado, FFORMA y MFFORMA, respectivamente, “OWA” los errores, “Cob” la cobertura, “Tiempo” el tiempo de ejecución en segundos y “Long” a la longitud de los intervalos de predicción en las unidades de trabajo. La tabla completa para todas las series del conjunto de prueba se puede encontrar en el Apéndice A.3.

Los tiempos respectivos al método FFORMA son una aproximación calculada haciendo un reparto proporcional, ya que no se podían calcular los tiempos para cada serie implícitamente. Hacer esta aproximación tiene sentido ya que el método FFORMA, para cada serie, calcula las predicciones por todos los métodos que utiliza y luego les aplica la combinación lineal calculada por el algoritmo de meta-aprendizaje, siendo el tiempo que tarda en calcular las predicciones para cada serie similar.

En la Tabla 4.14 podemos ver a través de los errores OWA que estos dos métodos son muy buenos, ya que de forma general los errores son menores que 1, lo que significa que las predicciones son más precisas que las generadas por el método M6. Las columnas que hacen referencia a la cobertura de los intervalos de predicción de cada método, podemos observar que la cobertura de nuestros intervalos de predicción es mucho mayor, es decir, que las observaciones reales, se encuentran dentro de los intervalos de predicción calculados un mayor número de veces, esto se debe a que la longitud de nuestros intervalos de predicción es mucho mayor, tal y como podemos ver en las dos últimas columnas de la tabla. La gran diferencia entre las longitudes de los intervalos entre unas series y otras se debe a las unidades de medida de cada serie, siendo las dos primeras de viajeros, las dos siguientes de personas activas y paradas, respectivamente. Las dos siguientes de contratos iniciales y contratos indefinidos registrados, respectivamente, las dos siguientes del índice de producción industrial y la tasa de variación anual de la

producción industrial, respectivamente, y las dos últimas de transacciones de viviendas. Las columnas que hacen referencia al tiempo de ejecución en segundos que empleó cada método en predecir esas series, nos muestran que el método MFFORMA es mucho más rápido, aunque en ocasiones, cuando este método toma las predicciones obtenidas por el método M1 en la combinación lineal de las predicciones finales, la diferencia de tiempo a cuando no lo toma es muy diferente, ya que este método tarda mucho tiempo en generar predicciones (compárense los tiempos de las Series 4, 6 y 8 con los demás).

Cabe mencionar que no es conveniente comparar estos errores OWA con los errores OWA obtenidos para estas series en la aproximación BFM (Tabla 4.5) ya que en la Tabla 4.14 aparece reflejado el error OWA que obtenemos tras predecir únicamente un año, mientras que en la otra tabla, aparece el error OWA obtenido de predecir casi media serie.

A continuación mostraremos 10 figuras, en las que se pueden comparar visualmente los resultados producidos por ambos métodos para las 10 series con las que trabajamos a lo largo de este capítulo. En cada una de las figuras se muestra tanto para el método FFORMA (izquierda) como para el método MFFORMA (derecha), una gráfica que incluye las observaciones reales (negro) de cada serie, las predicciones (rojo) y los intervalos de predicción (azul) calculados por estos dos métodos. Las gráficas se encuentran representadas en la misma escala para que se pueda hacer una comparación visual de los resultados de manera más sencilla.

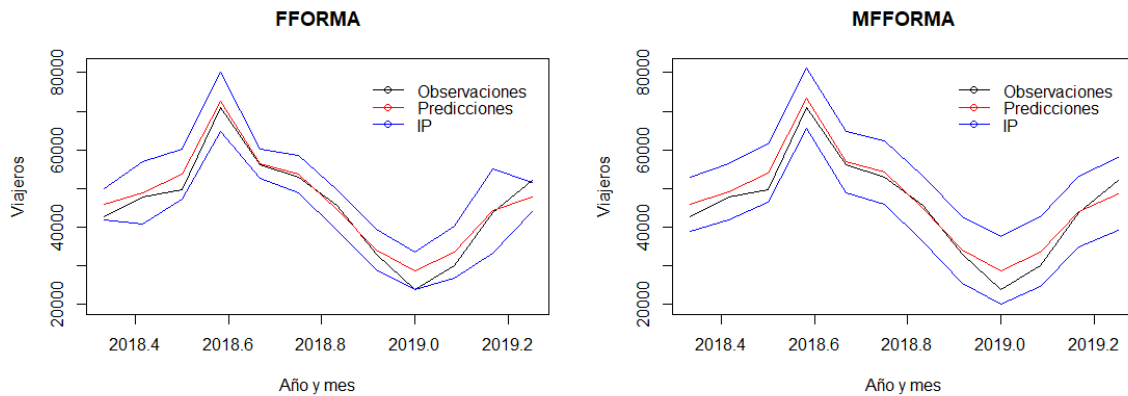


Figura 4.11: Serie 1 que hace referencia al número de viajeros estaños en hospedados en hoteles de Ourense. En ambas gráficas se muestran las predicciones para esa serie (rojo), las observaciones reales (negro) y los intervalos de predicción (azul) generados por los métodos FFORMA y MFFORMA, respectivamente.

Aparentemente, las predicciones generadas por ambos métodos para la Serie 1 son muy similares. En la Tabla 4.14 podemos ver que el error generado por el método FFORMA es un poco menor que el generado por el método MFFORMA, sin embargo, la cobertura de los intervalos de predicción es mejor la del MFFORMA ya que una observación se escapa de los intervalos de predicción generados por el método FFORMA de la manera explicada en la Sección 3.4.2, ya que estos son más estrechos, tal y como podemos ver en la Tabla 4.14. Con respecto al tiempo de predicción (en segundos), nuestro método de predicción es mucho más rápido, pues para la predicción únicamente utiliza una combinación lineal de los métodos M10 y M9 (Tabla 4.13).

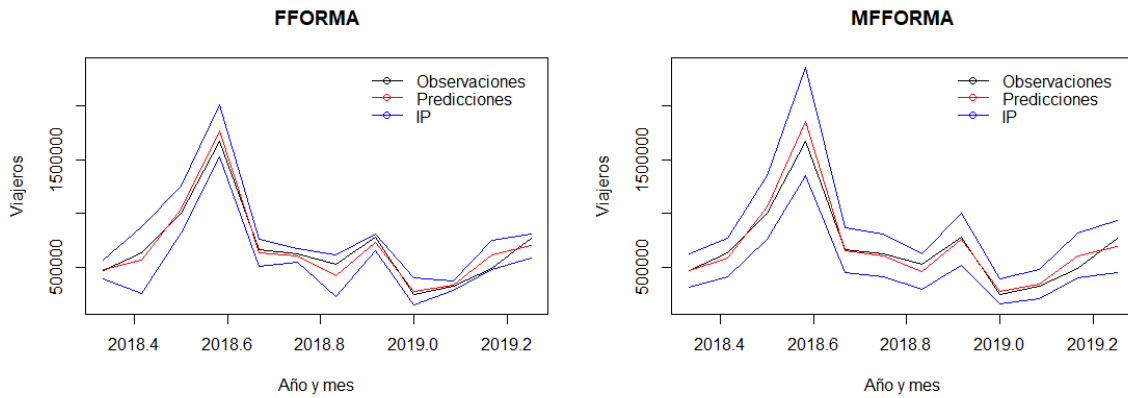


Figura 4.12: Serie 2 que hace referencia al número de viajeros españoles hospedados en establecimientos de turismo rural de Galicia. En ambas gráficas se muestran las predicciones para esa serie (rojo), las observaciones reales (negro) y los intervalos de predicción (azul) generados por los métodos FFORMA y MFFORMA, respectivamente.

Como las dos series tomadas del conjunto de referencia para el Conjunto 1 de series de turismo son muy similares, las predicciones generadas por ambos métodos para la Serie 2 vuelven a ser muy similares. En la Tabla 4.14 podemos ver que el error generado por el método FFORMA es un poco menor que el generado por el método MFFORMA, sin embargo, se repite lo sucedido para la Serie 1, la cobertura de los intervalos de predicción es mejor la del MFFORMA ya que una observación se escapa de los intervalos de predicción generados por el método FFORMA, tal y como podemos ver en la Tabla 4.14. Con respecto al tiempo de predicción (en segundos), vuelve a suceder lo mismo, nuestro método de predicción es mucho más rápido, ya que para la predicción únicamente utiliza una combinación lineal de los métodos M10 y M9 (Tabla 4.13).

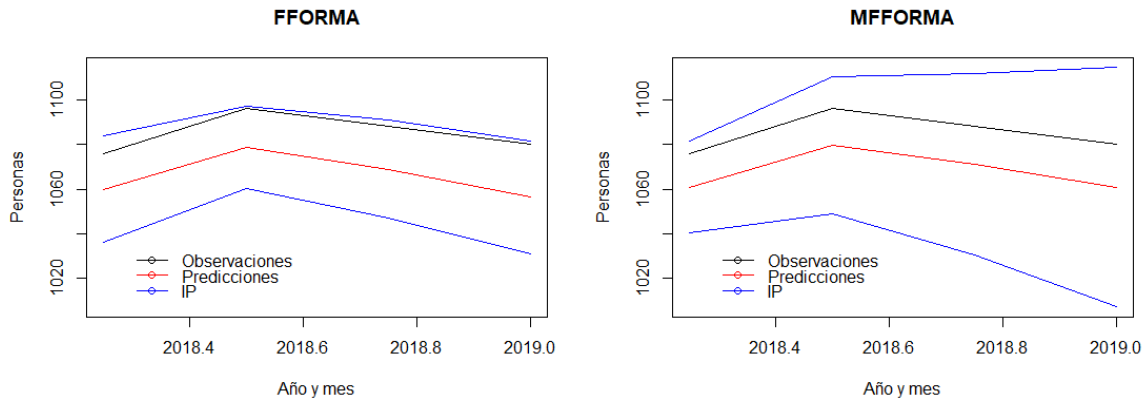


Figura 4.13: Serie 3 que hace referencia al total de la población ocupada en Galicia. En ambas gráficas se muestran las predicciones para esa serie (rojo), las observaciones reales (negro) y los intervalos de predicción (azul) generados por los métodos FFORMA y MFFORMA, respectivamente.

Las predicciones generadas por ambos métodos para la Serie 3 son muy similares, siendo las producidas por el método MFFORMA un poco mejores (ver el error de ambos métodos en la Tabla 4.14). En este caso, ninguna de las observaciones reales se escapa de los intervalos de predicción generados por ambos métodos, aunque en el método FFORMA está cerca, ya que sus intervalos son más estrechos. Con respecto al tiempo de ejecución (en segundos), sucede lo mismo que en los dos casos anteriores ya que el método MFFORMA únicamente utiliza una combinación lineal de los métodos M11 y M10.

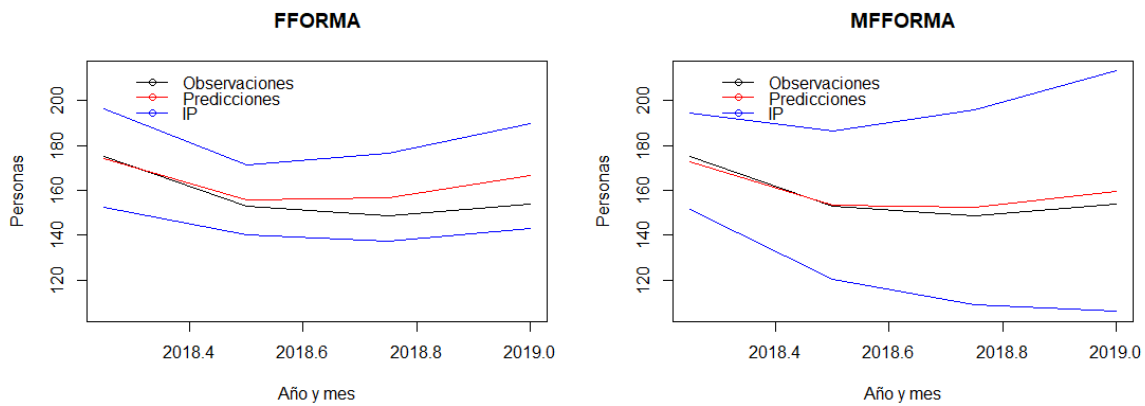


Figura 4.14: Serie 4 que hace referencia al total de la población parada en Galicia. En ambas gráficas se muestran las predicciones para esa serie (rojo), las observaciones reales (negro) y los intervalos de predicción (azul) generados por los métodos FFORMA y MFFORMA, respectivamente.

Para la Serie 4 a simple vista podemos ver que las predicciones generadas por el método MFFORMA son bastante mejores que las generadas por el FFORMA (ver además Tabla 4.14). Como en el caso anterior, ninguna de las observaciones reales se escapa de los intervalos de predicción generados por ambos métodos, aunque los del método FFORMA son más estrechos. Con respecto al tiempo de ejecución (en segundos), el del método MFFORMA es más bajo, aunque sin ser tan notable como en los otros casos, ya que el método MFFORMA utiliza una combinación lineal de los métodos M10 y el M1 que es el método más lento de todos los utilizados, aumentando el tiempo de ejecución del método MFFORMA.

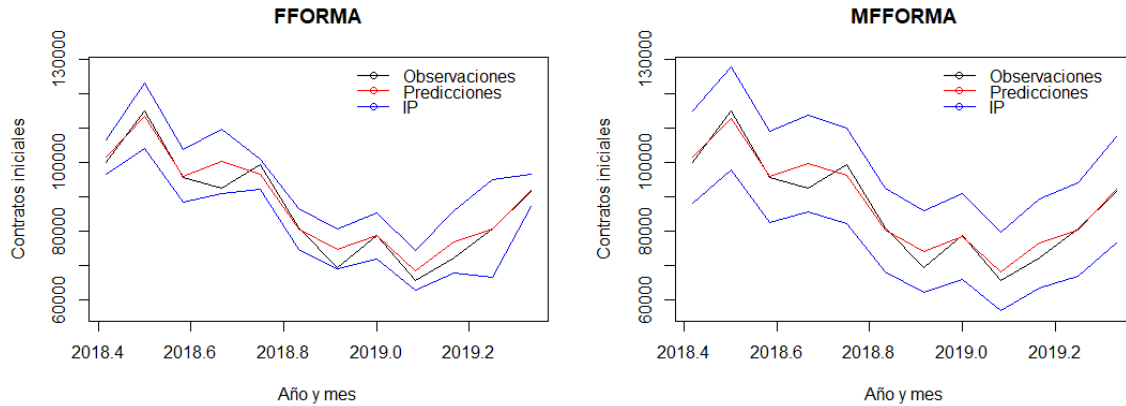


Figura 4.15: Serie 5 que hace referencia al número de contratos iniciales registrados en Galicia. En ambas gráficas se muestran las predicciones para esa serie (rojo), las observaciones reales (negro) y los intervalos de predicción (azul) generados por métodos FFORMA y el MFFORMA, respectivamente.

Para la serie 5 las predicciones obtenidas por ambos métodos son muy similares, pudiendo ver en la Tabla 4.14 que el método MFFORMA es un poco mejor. El porcentaje de cobertura de estos intervalos es del 100 %, siendo los calculados por el método MFFORMA más amplios. El tiempo de ejecución (en segundos) del método FFORMA es mucho mayor que el del MFFORMA, ya que este último únicamente utiliza para predecir los métodos de M5 y M6.

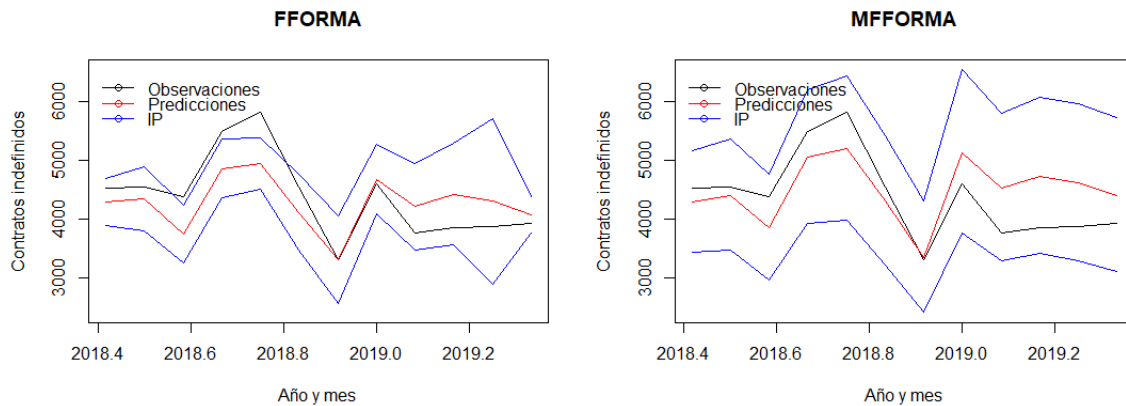


Figura 4.16: Serie 6 que hace referencia al número de contratos indefinidos registrados en Galicia. En ambas gráficas se muestran las predicciones para esa serie (rojo), las observaciones reales (negro) y los intervalos de predicción (azul) generados por los métodos FFORMA y MFFORMA, respectivamente.

A la vista de la Figura 4.16 puede parecer que las predicciones del método MFFORMA son mejores que las generadas por el método FFORMA debido a que las observaciones reales se salen de los intervalos de predicción, sin embargo, el error OWA generado por el método FFORMA es menor. La cobertura de los intervalos de predicción del método MFFORMA es mucho mejor pues todas las

observaciones están dentro de los intervalos de predicción, mientras que 3 observaciones se escapan de los intervalos de predicción del método FFORMA. La longitud de los intervalos de predicción es mucho mayor la del método MFFORMA y en este caso, el tiempo de ejecución (en segundos) del MFFORMA es mayor porque utiliza una combinación lineal de los métodos M5, M1 y M4, necesitando mucho tiempo el método M1 para obtener sus predicciones, tal y como podemos ver en la Tabla 4.14.

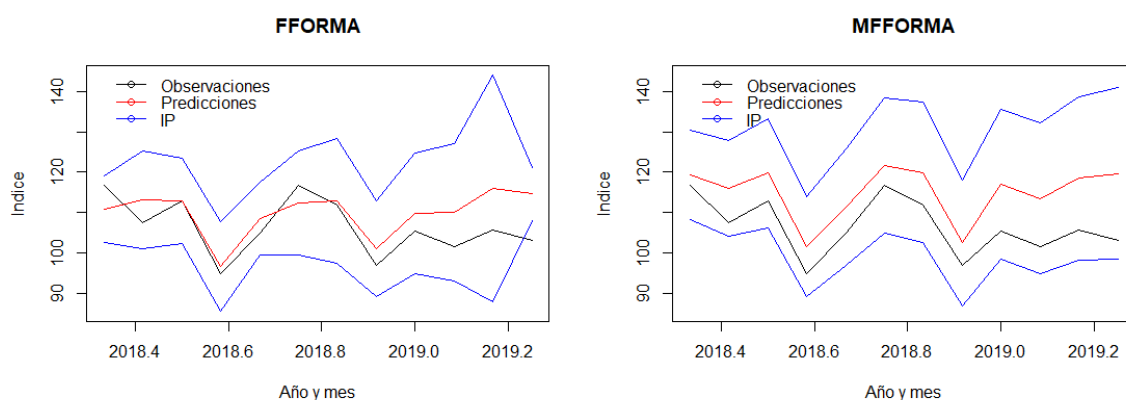


Figura 4.17: Serie 7 que hace referencia al índice de producción industrial de Galicia. En ambas gráficas se muestran las predicciones para esa serie (rojo), las observaciones reales (negro) y los intervalos de predicción (azul) generados por los métodos FFORMA y MFFORMA, respectivamente.

En la Figura 4.17, se puede observar que las predicciones calculadas por el método FFORMA son más precisas, sin embargo, la última observación se escapa del intervalo de predicción, mientras que en el método MFFORMA, todas se mantiene dentro de los intervalos. El tiempo de predicción (en segundos) del método MFFORMA es mucho menor, utilizando para predecir una combinación lineal de las predicciones calculadas por los métodos M11 y M10.

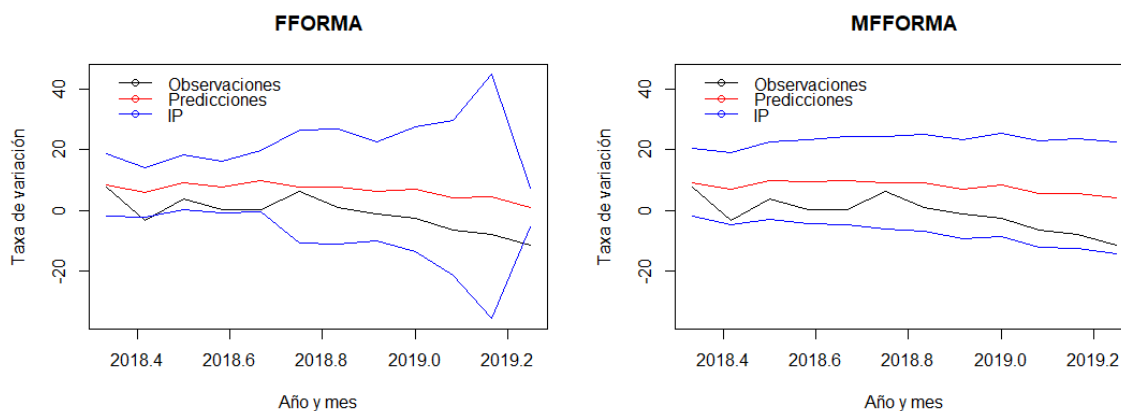


Figura 4.18: Serie 8 que hace referencia a la taxa de variación interanual de producción industrial en Galicia. En ambas gráficas se muestran las predicciones para esa serie (rojo), las observaciones reales (negro) y los intervalos de predicción (azul) generados por los métodos FFORMA y MFFORMA, respectivamente.

A la vista de la Figura 4.18, vemos que las predicciones calculadas por ambos métodos son muy similares, siendo un poco mejores las calculadas por el método FFORMA (ver Tabla 4.14), sin embargo, hay dos observaciones que se salen de los intervalos de predicción, mientras que en el método MFFORMA, todas se mantiene dentro de sus intervalos de predicción. El tiempo de predicción (en segundos) del método MFFORMA, en este caso es un poco mayor, ya que para la predicción utiliza los siguientes 5 métodos: M12, M5, M10, M4 y M1.

En la Figura 4.19 podemos ver a simple vista que las predicciones generadas por el método MFFORMA son más precisas que las obtenidas por el FFORMA. En este caso, todas las observaciones están dentro de los intervalos de predicción calculados por ambos métodos, y el tiempo de ejecución (en segundos) del MFFORMA es mucho menor ya que solo utiliza el método M11 para predecir.

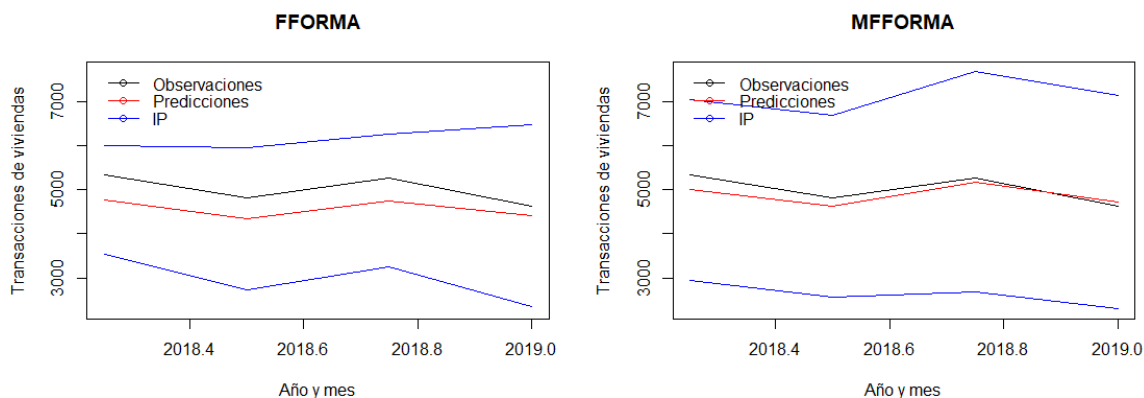


Figura 4.19: Serie 9 que hace referencia al número de transacciones de viviendas en Galicia. En ambas gráficas se muestran las predicciones para esa serie (rojo), las observaciones reales (negro) y los intervalos de predicción (azul) generados por los métodos FFORMA y MFFORMA, respectivamente.

En la Figura 4.20 podemos ver que las predicciones obtenidas por el método FFORMA son un poco mejores que las obtenidas por el MFFORMA, que todas las observaciones están dentro de los intervalos de predicción calculados por ambos métodos y que los intervalos de predicción son más amplios los del MFFORMA. Con respecto al tiempo de ejecución (en segundos) de ambos métodos, el del MFFORMA es mucho menor ya que solo utiliza el método M11 para predecir.

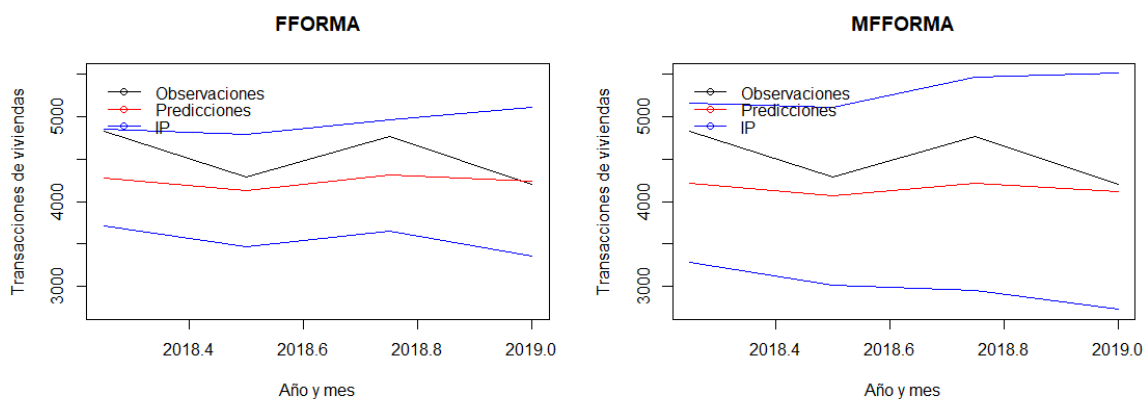


Figura 4.20: Serie 10 que hace referencia al número de transacciones de viviendas de segunda mano en Galicia. En ambas gráficas se muestran las predicciones para esa serie (rojo), las observaciones reales (negro) y los intervalos de predicción (azul) generados por los métodos FFORMA y MFFORMA, respectivamente.

Como resultados generales podemos decir que las predicciones obtenidas por el método MFFORMA son similares a las predicciones obtenidas por el método FFORMA, ya que de las 10 series, el método FFORMA fue mejor para 6 de ellas y el método MFFORMA para otras 4, siendo en varias ocasiones, difícil de reconocer a simple vista qué método generó mejores predicciones. En la Tabla A.12 podemos

ver que nuestro método mejora e iguala en 27 de las 50 series al método FFORMA.

La cobertura de nuestros intervalos es mayor, esto se debe a que la longitud de los intervalos de predicción calculados de la forma explicada en la Sección 3.4.2 es mayor, de forma acorde con nuestras necesidades. La cobertura global de los intervalos de predicción del método FFORMA es de un 92 % y para los del método MFFORMA es del 98 %.

Cabe mencionar que los métodos en los que el método FFORMA genera predicciones de una manera más rápida es cuando el método MFFORMA selecciona la función `auto.arima` dentro de la combinación lineal de métodos para calcular las predicciones finales. Ver las tablas A.12 y A.13 del anexo.

La siguiente tabla nos muestra los tiempos totales (en segundos) que tardó cada método en entrenar el algoritmo y en probarlo con el conjunto de prueba.

	Tiempo de entrenamiento	Tiempo de prueba
FFORMA	4532.04	1264.84
MFFORMA	10997.41	383.61

Tabla 4.15: Tabla de los tiempos de computación globales en segundos del conjunto de series de entrenamiento y del conjunto de prueba para cada método.

Como era de esperar, el tiempo de ejecución empleado por el método MFFORMA para el cálculo de las predicciones es 2.78 veces más rápido que el tiempo de ejecución que requiere el método FFORMA. Sin embargo, el tiempo necesario para el entrenamiento del algoritmo es menor el del método FFORMA debido a que utiliza dos métodos menos para la predicción y porque utiliza procesamiento en paralelo (realización de varias operaciones simultáneamente en los diferentes núcleos del ordenador) utilizando la librería `parallel` de R (McCallum y Weston, 2011) para obtener las predicciones de forma más rápida. Además el método MFFORMA calcula a su vez los intervalos de predicción de todos los métodos, mientras que el FFORMA los hace de forma independiente.

Recordamos que el modelo de meta-aprendizaje utiliza técnicas aleatorias (*boosting*) para el cálculo de los pesos, con lo que estos resultados podrían variar ligeramente si volvemos a ejecutar el código expuesto en el Apéndice B.

Capítulo 5

Conclusiones

La totalidad de la información estadística que se publica en el IGE tiene una base temporal, son datos referidos a un intervalo de tiempo y con una periodicidad determinados: anual, trimestral o mensual (en la mayor parte de los casos). Por tanto, como dimensión importante que es, es interesante estudiarla para controlar adecuadamente la calidad.

En el IGE, y en general en los institutos de estadística, se está continuamente añadiendo períodos de tiempo a la información existente. Por tanto, el primer control que se necesita es verificar *que el último dato añadido es coherente con los datos disponibles para períodos anteriores*. En un primer momento se planteó calcular la tasa de variación (interanual, intertrimestral, intermensual,...) para todas las posibles combinaciones y establecer un umbral que no debería superar. Esto ocasionó bastantes problemas con datos inferiores a 100 unidades, por ejemplo, que fluctúan mucho y tienen tasas de variación que superaban los umbrales considerados. Desechada esta posibilidad, se echó mano de la teoría de las series de tiempo.

Con este objetivo, a lo largo de este trabajo se ha expuesto una revisión bibliográfica de los métodos de predicción de series más utilizados en la práctica, seguido de la aproximación BFM utilizada para saber cuál de todos los métodos estudiados es el que mejores predicciones genera, utilizando como medida del error el OWA (Overall Weighing Average). Los resultados de esta aproximación han sugerido que dependiendo de las características de las series, unos métodos funcionan mejor que otros, con la posibilidad de que varios modelos puedan explicar bien el comportamiento de una misma serie. Este es un hecho que guarda relación directa con los estudios realizados por Montero-Manso et al. (2018) que muestran que un modelo de predicción mixto (que utiliza una combinación lineal de diferentes métodos de predicción) es mejor que un modelo de selección que se predice únicamente con un método. Por esta razón, se comenzó a trabajar con el método FFORMA.

Tras testear este método, aprovechamos los resultados obtenidos en la aproximación BFM para mejorar la precisión de las predicciones realizadas por el método FFORMA. También se redujo el tiempo de computación empleado para la obtención de las predicciones finales, sin que el valor de las predicciones se viese considerablemente modificado (reduciendo el número de métodos con el que predecimos).

Además, se propuso una alternativa para el cálculo de los intervalos de predicción, de forma más sencilla, intuitiva y rápida que produce muy buenos resultados en la práctica. Se trata de buscar un equilibrio entre la longitud del intervalo y el error que se puede cometer. Intervalos muy grandes

permiten que la serie no se revise continuamente pero podríamos estar cometiendo errores, y si el intervalo es muy ajustado esto nos garantiza errores pequeños pero sin embargo el IGE tendría alarmas de revisión de la serie de forma casi continua. Es en este punto es donde tenemos que conseguir un compromiso entre ambos intereses.

En el Capítulo 4 se muestra una comparación entre los métodos FFORMA y MFFORMA aplicados a 5 conjuntos de datos reales proporcionados por el IGE. En esta comparación podemos comprobar que las predicciones obtenidas por ambos métodos son similares, según lo visto en las figuras y tablas del Capítulo 4, sin embargo, el método MFFORMA mejora al FFORMA en la cobertura de los intervalos de predicción, debido a que los intervalos propuestos son más amplios y en el tiempo de ejecución, cumpliendo las expectativas y resolviendo el problema de la coherencia con el último dato de manera eficiente.

Cabe mencionar que ambos métodos fueron entrenados para el mismo número de series (222). Si el número de series fuese mayor, como el de la competición M4 (M4 Competitor's Guide, 2018), las predicciones generadas por ambos métodos posiblemente serían mucho mejores.

Futuras líneas de trabajo para mejorar y dar visibilidad al MFFORMA:

1. Comprobar si funciona para series con frecuencias diferentes a las trabajadas a lo largo de la memoria, ya que únicamente lo hemos probado para series mensuales y trimestrales.
2. Comprobar los resultados obtenidos tras entrenar el algoritmo con un mayor número de series.
3. Aumentar la velocidad de ejecución del método incorporando procesamiento en paralelo.
4. Subirlo a Git-Hub.
5. Crear una librería de R con el método FFORMA.

Apéndice A

Tablas

En esta parte del apéndice incluiremos las tablas completas de la aproximación BFM y de la comparación del método FFORMA con el MFFORMA que ya expusimos en el Capítulo 4 para las 10 series que tomamos de referencia.

Cada sección de este apéndice está organizada de la siguiente manera. En la primera mostraremos las tablas con los errores OWA obtenidos por cada método, para cada serie de cada conjunto. En la segunda pondremos las tablas que muestren para cada serie de cada conjunto el método que generó mejores predicciones, junto con su tiempo y su error obtenidos. Por último, en la última sección, pondremos la tabla en la que comparamos el método FFORMA con el método MFFORMA que incluye para cada serie del conjunto de prueba (50 elementos) sus errores OWA, su cobertura, su tiempo de ejecución en segundos y la longitud de sus intervalos de predicción, además de otra tabla que muestra para el método MFFORMA los métodos de predicción utilizados en la combinación lineal para obtener las predicciones finales, también para las 50 series del conjunto de referencia.

Cabe mencionar que las series con las que estuvimos trabajando en el Capítulo 4 son las series 53 y 56 para el Conjunto 1, 25 y 37 para el Conjunto 2, 1 y 6 para el Conjunto 3, 2 y 16 para el Conjunto 4, y 2 y 14 para el Conjunto 5.

A.1. Errores OWA

Tablas con los errores OWA obtenidos por cada método, para cada serie de cada conjunto, que hace referencia a la Tabla 4.4 del Capítulo 4. Los números de las columnas hacen referencia a los métodos utilizados de la misma forma que en 4.4, es decir:

auto.arima	naive	bats	tbats	ets	snaive	nnetar	rwf	thetaf	x13_def	tramoseats_def
M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11

Tabla A.1: Asignación entre métodos y números.

Conjunto 1											
Serie	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
1	0.67	5.46	0.72	0.93	0.83	1.00	0.91	5.46	0.71	0.53	0.48
2	0.91	4.41	1.02	1.00	0.95	1.00	0.95	4.41	0.99	0.64	0.65
3	0.82	3.81	1.13	1.03	0.98	1.00	1.10	3.81	1.03	0.75	0.77
4	1.03	3.88	1.05	1.20	1.03	1.00	0.99	3.88	1.32	0.82	0.72
5	0.85	2.29	0.84	0.92	0.92	1.00	1.05	2.29	0.85	0.91	0.92
6	0.97	4.58	0.97	1.02	0.89	1.00	0.92	4.58	0.92	0.83	0.90
7	0.83	2.44	0.91	1.03	0.89	1.00	0.92	2.44	0.86	0.64	0.61
8	1.06	3.07	1.03	1.14	0.91	1.00	1.01	3.07	0.99	0.70	0.72
9	1.05	2.90	1.13	1.17	0.86	1.00	1.03	2.90	0.90	0.86	0.81
10	0.98	2.71	0.88	0.87	0.95	1.00	0.93	2.71	0.88	0.89	0.78
11	0.93	1.59	0.84	0.77	0.76	1.00	0.91	1.59	0.74	0.70	0.73
12	1.04	2.67	1.14	1.15	1.06	1.00	0.96	2.67	1.11	0.85	0.88
13	0.83	7.46	0.73	1.02	0.81	1.00	0.97	7.46	0.75	0.70	0.68
14	0.88	4.93	0.98	1.02	0.90	1.00	0.97	4.93	0.96	0.72	0.73
15	0.94	4.12	1.07	1.06	1.02	1.00	1.05	4.12	1.03	0.77	0.77
16	1.03	4.00	1.08	1.11	1.04	1.00	1.09	4.00	1.20	0.83	0.78
17	0.90	2.78	0.95	1.02	1.04	1.00	1.03	2.78	0.94	0.91	0.89
18	1.00	5.38	1.02	1.07	1.00	1.00	1.04	5.38	1.04	1.09	1.05
19	0.98	3.07	0.88	1.09	0.84	1.00	1.03	3.07	0.84	0.66	0.76
20	1.11	3.16	0.93	1.02	1.09	1.00	0.99	3.16	0.93	0.71	0.72
21	0.99	2.86	0.95	1.07	0.85	1.00	0.95	2.86	0.86	0.82	0.66
22	1.08	2.49	0.84	0.92	0.88	1.00	0.96	2.49	0.89	0.90	0.82
23	0.98	1.81	0.80	0.80	0.72	1.00	0.95	1.81	0.73	0.61	0.73
24	1.10	2.91	1.04	1.01	0.98	1.00	0.99	2.91	0.94	0.82	0.66
25	1.15	4.47	1.29	1.20	1.22	1.00	1.23	4.47	1.20	1.15	1.14
26	1.10	2.56	1.29	1.19	1.30	1.00	1.26	2.56	1.26	1.16	1.10
27	1.20	1.82	1.03	1.05	1.05	1.00	1.29	1.82	1.01	1.06	1.14
28	0.92	2.61	1.01	1.00	0.94	1.00	1.14	2.61	1.05	0.89	0.88
29	1.09	1.44	0.88	0.91	0.96	1.00	1.07	1.44	0.95	0.98	1.02
30	0.93	2.83	1.00	1.08	1.02	1.00	1.06	2.83	1.03	1.06	0.90
31	1.22	3.74	1.21	1.47	1.31	1.00	1.29	3.74	1.26	1.14	1.10
32	0.96	2.17	0.90	0.94	0.92	1.00	0.95	2.17	0.86	0.79	0.74
33	1.05	1.77	0.89	0.89	1.06	1.00	1.00	1.77	0.90	0.87	0.90
34	0.72	1.90	0.75	0.78	0.70	1.00	0.96	1.90	0.69	0.65	0.63
35	0.85	1.02	0.80	0.82	0.79	1.00	0.96	1.02	0.79	0.69	0.62
36	0.83	1.86	0.87	0.83	0.82	1.00	0.88	1.86	0.79	0.71	0.70

Sigue en la página siguiente.

Serie	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
37	0.78	4.01	0.87	1.20	0.95	1.00	0.99	4.01	0.82	0.68	0.61
38	0.93	3.75	0.84	0.94	0.79	1.00	1.00	3.75	0.80	0.69	0.72
39	0.78	2.76	0.83	0.84	0.79	1.00	1.08	2.76	0.79	0.78	0.83
40	1.03	3.12	0.88	0.93	0.94	1.00	0.96	3.12	0.92	0.74	0.74
41	0.83	2.17	0.82	0.94	0.90	1.00	1.05	2.17	0.86	0.88	0.86
42	0.96	4.37	0.98	0.99	0.93	1.00	0.97	4.37	0.93	0.86	1.02
43	0.88	2.33	0.86	1.04	0.85	1.00	0.97	2.33	0.80	0.64	0.63
44	1.13	2.49	0.89	0.94	0.89	1.00	0.99	2.49	0.84	0.72	0.73
45	0.94	2.24	0.95	0.92	0.84	1.00	1.00	2.24	0.80	0.94	0.84
46	0.97	2.09	0.87	0.90	0.87	1.00	0.98	2.09	0.83	0.88	0.87
47	0.86	1.56	0.83	0.77	0.72	1.00	0.87	1.56	0.73	0.72	0.77
48	1.09	2.33	1.08	1.14	1.04	1.00	1.00	2.33	1.03	0.82	0.83
49	0.92	5.04	0.81	1.26	1.06	1.00	0.90	5.04	0.83	0.67	0.65
50	0.85	4.39	0.99	0.92	0.92	1.00	1.02	4.39	0.91	0.83	0.72
51	0.89	3.13	0.99	1.02	0.97	1.00	1.05	3.13	0.97	0.82	0.82
52	0.98	3.43	0.94	0.97	0.92	1.00	0.99	3.43	0.96	0.85	0.76
53	0.95	2.69	0.99	1.02	1.08	1.00	1.07	2.69	0.98	1.00	1.02
54	0.93	5.15	1.24	1.02	0.98	1.00	1.03	5.15	1.00	1.15	1.06
55	0.95	2.84	0.82	1.08	0.81	1.00	1.06	2.84	0.82	0.63	0.72
56	1.09	2.73	0.84	0.93	1.07	1.00	0.88	2.73	0.85	0.70	0.74
57	0.90	2.45	0.85	0.90	0.85	1.00	1.03	2.45	0.82	0.87	0.69
58	1.05	2.12	0.80	0.88	0.84	1.00	0.94	2.12	0.82	0.83	0.83
59	0.90	1.80	0.80	0.83	0.71	1.00	0.92	1.80	0.71	0.63	0.64
60	1.01	2.59	1.05	1.04	0.98	1.00	1.05	2.59	0.96	0.83	0.77
61	0.68	5.86	0.63	0.71	0.69	1.00	0.98	5.86	0.60	0.52	0.52
62	0.97	5.68	1.39	1.57	1.16	1.00	0.94	5.68	1.66	0.95	0.90
63	1.06	5.21	1.19	1.43	1.14	1.00	0.96	5.21	1.35	0.85	0.86
64	1.11	5.02	1.18	1.26	1.01	1.00	1.23	5.02	1.46	0.90	0.91
65	1.08	2.40	1.01	1.12	1.03	1.00	1.03	2.40	1.04	1.01	0.94
66	1.08	3.80	1.18	1.33	1.09	1.00	1.13	3.80	1.22	0.95	0.92
67	0.72	4.25	0.84	0.79	0.81	1.00	0.64	4.25	0.75	0.57	0.52
68	1.00	3.41	1.08	1.22	1.23	1.00	1.39	3.41	1.06	0.88	0.85
69	1.16	2.76	1.46	1.51	1.15	1.00	0.87	2.76	0.96	1.23	1.07
70	0.92	3.31	1.16	1.11	1.27	1.00	1.26	3.31	1.49	0.92	0.97
71	0.86	1.37	0.81	0.83	0.79	1.00	1.69	1.37	0.91	0.87	0.89
72	1.16	2.18	1.24	1.18	1.52	1.00	1.30	2.18	1.01	1.26	1.14
73	0.96	7.05	0.84	0.78	0.77	1.00	1.05	7.05	0.80	0.74	0.70
74	0.91	5.55	1.26	1.38	1.11	1.00	1.02	5.55	1.61	0.94	0.82

Sigue en la página siguiente.

Serie	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
75	0.94	5.02	1.24	1.25	1.00	1.00	1.05	5.02	1.26	0.86	0.90
76	1.07	4.90	1.33	1.24	1.08	1.00	1.16	4.90	1.72	1.01	0.99
77	0.94	1.72	0.89	0.87	0.96	1.00	1.13	1.72	0.93	0.88	0.97
78	0.95	3.82	1.10	1.23	1.09	1.00	1.00	3.82	1.12	0.95	1.00
79	1.02	4.54	0.96	1.04	0.92	1.00	0.92	4.54	0.90	0.82	0.75
80	1.05	3.02	1.12	1.55	1.28	1.00	1.33	3.02	1.13	1.16	0.96
81	0.93	2.68	1.34	1.36	1.43	1.00	1.16	2.68	1.05	1.10	1.14
82	1.00	3.07	1.14	1.11	1.28	1.00	1.01	3.07	1.74	0.91	0.95
83	0.84	1.40	0.79	0.78	0.75	1.00	1.24	1.40	0.90	0.97	0.94
84	1.02	2.14	1.01	1.21	1.26	1.00	1.21	2.14	1.10	1.38	1.34

Tabla A.2: Tabla donde se muestra para todas las series del Conjunto 1 los errores OWA producidos por cada método.

Conjunto 2											
Serie	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
1	0.26	0.60	0.37	0.33	0.26	1.00	0.41	0.60	0.84	0.35	0.36
2	0.28	0.60	0.38	0.41	0.23	1.00	0.39	0.60	0.83	0.32	0.37
3	0.26	0.60	0.40	0.46	0.28	1.00	0.41	0.60	0.85	0.30	0.35
4	0.07	0.61	0.09	0.09	0.08	1.00	0.13	0.61	0.25	0.10	0.10
5	0.08	0.61	0.10	0.10	0.08	1.00	0.12	0.61	0.26	0.09	0.09
6	0.07	0.61	0.08	0.08	0.08	1.00	0.11	0.61	0.24	0.08	0.06
7	0.10	0.62	0.13	0.14	0.11	1.00	0.16	0.62	0.79	0.10	0.11
8	0.09	0.62	0.19	0.19	0.11	1.00	0.17	0.62	0.78	0.11	0.12
9	0.10	0.62	0.13	0.12	0.11	1.00	0.15	0.62	0.80	0.12	0.12
10	0.11	0.67	0.10	0.10	0.11	1.00	0.18	0.67	0.33	0.15	0.14
11	0.09	0.66	0.10	0.10	0.09	1.00	0.17	0.66	0.33	0.10	0.13
12	0.12	0.67	0.12	0.12	0.13	1.00	0.20	0.67	0.33	0.11	0.12
13	0.81	0.76	0.85	0.80	0.85	1.00	0.93	0.76	0.96	1.00	0.95
14	0.67	0.73	0.79	0.74	0.78	1.00	1.00	0.73	0.89	0.88	0.81
15	1.06	0.90	0.92	0.92	0.94	1.00	1.16	0.90	1.01	1.26	1.14
16	0.69	0.96	0.79	0.77	0.78	1.00	0.85	0.96	0.60	0.57	0.61
17	0.70	0.95	0.79	0.90	0.95	1.00	0.88	0.95	0.73	0.65	0.63
18	0.79	0.88	0.78	0.86	0.77	1.00	1.05	0.88	0.70	0.78	0.81
19	0.90	0.69	0.82	0.82	0.70	1.00	0.92	0.69	0.97	0.91	1.01

Sigue en la página siguiente.

Serie	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
20	0.74	0.66	0.60	0.60	0.64	1.00	0.94	0.66	0.76	0.66	0.69
21	1.29	0.90	1.24	1.24	1.20	1.00	1.10	0.90	1.19	1.35	1.45
22	0.59	0.66	0.65	0.65	0.64	1.00	0.96	0.66	0.62	0.59	0.66
23	0.71	0.71	0.73	0.73	0.72	1.00	1.16	0.71	0.74	1.06	0.71
24	0.63	0.68	0.73	0.75	0.68	1.00	0.77	0.68	0.71	0.71	0.73
25	0.60	0.70	0.51	0.52	0.62	1.00	0.86	0.70	0.68	0.66	0.66
26	0.64	0.71	0.63	0.67	0.64	1.00	0.87	0.71	0.68	0.58	0.57
27	0.84	0.76	0.68	0.66	0.71	1.00	0.97	0.76	0.68	0.80	0.74
28	0.66	0.85	0.69	0.86	0.71	1.00	1.89	0.85	0.71	0.58	0.61
29	0.59	0.83	0.75	0.63	0.69	1.00	1.02	0.83	0.68	0.57	0.70
30	0.93	0.82	0.83	0.82	0.82	1.00	0.87	0.82	0.80	1.00	0.91
31	0.74	0.72	0.63	0.61	0.78	1.00	0.82	0.72	0.73	0.82	0.84
32	0.77	0.76	0.68	0.72	0.87	1.00	0.91	0.76	0.72	0.70	0.74
33	0.93	0.75	0.81	0.73	0.89	1.00	1.04	0.75	0.82	1.03	1.23
34	0.66	0.69	0.69	0.69	0.67	1.00	1.05	0.69	0.68	0.64	0.68
35	0.73	0.73	0.74	0.78	0.74	1.00	0.83	0.73	0.76	0.68	0.70
36	0.61	0.67	0.74	0.68	0.67	1.00	0.80	0.67	0.69	0.73	0.70
37	0.55	0.68	0.62	0.61	0.59	1.00	0.80	0.68	0.67	0.62	0.61
38	0.55	0.67	0.63	0.72	0.66	1.00	0.97	0.67	0.69	0.61	0.62
39	0.69	0.73	0.69	0.69	0.68	1.00	0.80	0.73	0.72	0.71	0.72
40	0.76	0.76	0.89	0.85	0.76	1.00	0.89	0.76	0.80	0.74	0.84
41	0.97	0.96	1.00	0.99	1.00	1.00	0.89	0.96	0.99	0.94	0.86
42	1.02	0.88	0.93	0.93	0.90	1.00	0.92	0.88	0.97	1.06	0.97
43	0.68	0.70	0.62	0.61	0.63	1.00	0.88	0.70	0.65	0.64	0.58
44	0.60	0.69	0.73	0.74	0.68	1.00	0.99	0.69	0.70	0.64	0.60
45	0.71	0.78	0.72	0.72	0.72	1.00	0.97	0.78	0.70	0.62	0.74
46	0.90	0.72	0.78	0.78	0.85	1.00	0.97	0.72	0.79	0.69	0.85
47	0.77	0.69	0.75	0.77	0.77	1.00	1.14	0.69	0.81	0.88	0.77
48	0.90	0.94	0.89	0.89	0.86	1.00	0.97	0.94	0.97	1.02	1.14
49	0.83	0.83	0.84	0.82	0.86	1.00	0.96	0.83	0.90	0.84	0.89
50	0.79	0.83	0.82	0.84	0.79	1.00	0.95	0.83	0.77	0.75	0.82
51	0.92	0.90	0.93	0.93	0.93	1.00	1.23	0.90	0.87	0.97	0.86
52	1.29	1.34	1.24	1.06	1.27	1.00	1.06	1.34	1.03	1.04	1.01
53	1.13	1.41	1.26	1.36	1.56	1.00	1.35	1.41	1.26	1.23	1.35
54	1.15	1.09	1.05	0.99	0.94	1.00	0.92	1.09	0.89	1.01	1.12
55	0.70	0.64	0.68	0.68	0.68	1.00	1.00	0.64	0.63	0.77	0.79
56	0.81	0.86	0.82	0.82	0.82	1.00	0.72	0.86	0.83	0.86	0.80
57	0.75	0.82	0.79	0.90	0.80	1.00	1.10	0.82	0.73	0.85	0.68

Sigue en la página siguiente.

Serie	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
58	0.83	0.92	0.89	0.89	0.78	1.00	1.01	0.92	0.73	0.89	0.82
59	0.51	0.83	0.76	0.76	0.78	1.00	0.73	0.83	0.71	0.76	0.50
60	1.00	0.80	0.80	0.78	0.77	1.00	0.99	0.80	0.84	1.01	0.99

Tabla A.3: Tabla donde se muestra para todas las series del Conjunto 2 los errores OWA producidos por cada método.

Conjunto 3											
Serie	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
1	0.92	1.95	1.14	1.23	1.25	1.00	0.97	1.95	1.12	0.81	0.88
2	0.92	1.71	1.11	1.24	1.18	1.00	0.98	1.71	1.05	0.93	0.93
3	1.02	2.04	1.04	1.07	1.01	1.00	1.02	2.04	1.24	1.04	0.90
4	1.00	2.10	1.22	1.25	1.30	1.00	1.00	2.10	1.09	0.90	0.81
5	1.03	1.94	1.27	1.27	1.30	1.00	1.09	1.94	1.09	0.84	0.90
6	0.63	0.95	0.76	0.74	0.65	1.00	0.93	0.95	0.66	0.90	0.70
7	0.73	0.97	0.70	0.77	0.79	1.00	0.97	0.97	0.71	0.70	0.72
8	0.85	1.03	0.81	0.83	0.83	1.00	0.95	1.03	0.81	0.83	0.79
9	0.84	0.90	0.77	0.81	0.92	1.00	0.94	0.90	0.77	0.70	0.68
10	0.67	0.96	0.77	0.82	0.77	1.00	0.93	0.96	0.67	0.76	0.67
11	0.90	2.20	1.08	1.27	1.14	1.00	1.01	2.20	1.31	0.97	0.96
12	0.95	1.94	1.02	1.27	1.01	1.00	1.02	1.94	1.20	0.98	0.97
13	0.99	1.99	1.12	1.13	1.07	1.00	1.48	1.99	1.20	1.04	0.97
14	1.12	2.20	0.96	1.07	0.95	1.00	1.05	2.20	1.20	0.94	0.94
15	0.92	1.85	1.07	1.20	1.12	1.00	1.08	1.85	1.21	0.94	0.94
16	0.90	1.53	0.91	0.97	0.95	1.00	1.11	1.53	0.98	0.81	0.84
17	0.98	1.40	1.18	1.05	1.01	1.00	1.21	1.40	1.08	0.83	0.76
18	0.86	1.56	0.93	0.93	0.88	1.00	0.94	1.56	0.91	0.88	0.81
19	0.88	1.23	0.97	0.95	0.94	1.00	0.95	1.23	0.88	0.67	0.63
20	0.96	1.62	1.03	0.96	0.92	1.00	0.98	1.62	0.95	0.83	0.81
21	0.99	1.77	1.03	1.04	1.01	1.00	0.98	1.77	1.05	0.94	0.89
22	1.04	1.88	1.02	1.01	1.06	1.00	1.03	1.88	1.09	1.00	1.00
23	0.89	1.66	0.91	0.90	0.90	1.00	0.97	1.66	1.10	0.94	1.03
24	0.98	1.51	0.89	0.91	0.90	1.00	0.96	1.51	0.99	0.90	0.95
25	1.15	1.48	0.96	0.95	1.09	1.00	1.31	1.48	1.04	0.95	0.89
26	0.90	1.22	0.84	0.84	0.77	1.00	1.12	1.22	0.82	0.72	0.79

Sigue en la página siguiente.

Serie	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
27	0.77	1.17	0.82	0.85	0.83	1.00	1.13	1.17	0.77	0.76	0.76
28	0.97	1.20	0.86	0.90	0.76	1.00	1.16	1.20	0.78	0.81	0.81
29	1.00	1.23	0.96	0.96	0.99	1.00	1.07	1.23	0.96	0.90	0.96
30	0.96	1.20	0.97	0.87	1.07	1.00	1.09	1.20	0.89	0.85	1.07
31	0.96	0.92	0.94	0.95	1.04	1.00	1.11	0.92	0.89	1.02	1.03
32	0.98	0.90	1.07	1.00	1.12	1.00	1.10	0.90	0.92	0.87	1.08
33	1.05	0.86	0.94	1.01	1.13	1.00	1.14	0.86	1.00	0.93	1.01
34	1.12	0.91	1.09	1.06	1.16	1.00	1.16	0.91	1.07	0.94	0.90
35	0.93	0.86	0.86	0.85	0.85	1.00	1.09	0.86	0.84	0.92	0.91
36	1.44	2.66	1.24	1.37	1.32	1.00	1.86	2.66	1.05	0.83	0.85
37	1.08	1.27	1.15	1.16	1.25	1.00	1.26	1.27	1.19	1.19	1.05
38	2.76	2.03	1.09	1.18	1.02	1.00	1.95	2.03	1.21	0.85	1.15
39	1.02	1.94	0.94	0.90	0.86	1.00	1.31	1.94	1.13	0.92	0.92
40	1.55	2.33	1.04	0.96	0.98	1.00	1.70	2.33	0.90	0.90	0.81
41	1.20	1.74	0.93	1.15	1.02	1.00	1.23	1.74	1.20	0.68	0.76
42	1.64	1.93	0.93	1.27	1.04	1.00	1.16	1.93	1.31	1.00	0.84
43	0.91	1.70	1.02	1.06	0.99	1.00	1.14	1.70	1.09	0.80	0.76
44	0.97	1.41	1.03	1.03	0.99	1.00	1.02	1.41	1.04	0.88	0.89
45	0.88	1.56	0.94	1.15	1.06	1.00	1.05	1.56	1.11	0.82	0.86
46	0.86	0.83	0.82	0.87	0.98	1.00	1.13	0.83	0.82	0.87	0.81
47	0.91	0.86	0.89	0.89	0.88	1.00	1.19	0.86	0.89	0.82	0.82
48	0.78	0.91	0.75	0.84	0.73	1.00	0.88	0.91	0.76	0.73	0.76
49	0.90	0.99	0.82	0.82	0.90	1.00	0.98	0.99	0.89	0.75	0.79
50	0.85	0.89	0.89	0.86	0.98	1.00	0.96	0.89	0.87	0.95	0.92

Tabla A.4: Tabla donde se muestra para todas las series del Conjunto 3 los errores OWA producidos por cada método.

Conjunto 4											
Serie	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
1	0.99	5.22	0.90	1.19	0.99	1.00	1.09	5.22	1.04	0.79	0.70
2	0.86	1.98	0.83	0.97	0.84	1.00	0.90	1.98	0.86	0.70	0.68
3	1.00	3.50	1.00	1.14	0.98	1.00	1.07	3.50	1.01	0.64	0.67
4	1.09	1.80	0.98	0.94	0.98	1.00	0.93	1.80	1.03	1.14	1.02
5	0.99	4.07	1.23	1.31	1.11	1.00	0.89	4.07	1.10	0.79	0.68

Sigue en la página siguiente.

Serie	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
6	0.81	1.30	0.84	0.81	0.90	1.00	0.99	1.30	0.81	0.83	0.78
7	1.04	1.98	0.86	0.99	0.90	1.00	0.99	1.98	0.91	0.59	0.55
8	0.90	1.82	0.92	0.94	0.91	1.00	0.87	1.82	0.95	0.91	0.81
9	0.91	1.84	0.73	0.88	0.77	1.00	1.19	1.84	0.81	0.83	0.82
10	1.01	1.11	0.75	0.93	0.85	1.00	0.98	1.11	0.76	0.77	0.67
11	1.15	1.73	0.78	0.94	0.81	1.00	1.06	1.73	0.81	0.65	0.61
12	0.91	1.26	0.74	0.78	0.73	1.00	1.07	1.26	0.73	0.86	0.68
13	1.38	2.95	1.34	1.28	1.46	1.00	1.16	2.95	1.50	1.33	1.37
14	0.90	1.37	0.86	0.97	1.03	1.00	1.08	1.37	1.06	0.91	0.84
15	0.82	0.82	0.82	0.82	0.81	1.00	0.95	0.82	0.80	0.73	0.71
16	0.67	0.92	0.81	0.81	0.82	1.00	0.84	0.92	0.82	0.72	0.74
17	0.86	0.85	0.82	0.82	0.88	1.00	0.85	0.85	0.86	0.66	0.65
18	0.82	0.90	0.79	0.79	0.79	1.00	0.88	0.90	0.78	0.74	0.77
19	1.03	0.87	1.00	1.00	1.06	1.00	0.95	0.87	0.89	0.89	0.91
20	0.68	0.82	0.81	0.81	0.81	1.00	0.80	0.82	0.81	0.71	0.72
21	0.77	0.83	0.86	0.86	0.92	1.00	0.85	0.83	0.90	0.65	0.61
22	0.79	0.88	0.81	0.81	0.81	1.00	0.84	0.88	0.80	0.71	0.74
23	0.69	0.78	0.78	0.78	0.87	1.00	0.78	0.78	0.79	0.69	0.68
24	0.58	0.82	0.72	0.72	0.75	1.00	1.00	0.82	0.75	0.55	0.72
25	0.79	0.73	0.71	0.71	0.66	1.00	0.89	0.73	0.74	0.74	0.72
26	0.73	0.77	0.71	0.71	0.71	1.00	1.08	0.77	0.71	0.72	0.64
27	0.82	1.09	0.88	0.88	0.93	1.00	0.82	1.09	0.93	0.85	0.88
28	0.70	0.89	0.74	0.74	0.82	1.00	0.77	0.89	3.70	0.70	0.82
29	0.63	0.44	0.44	0.44	0.44	1.00	0.74	0.44	0.43	0.60	0.49
30	0.57	0.54	0.54	0.54	0.54	1.00	0.53	0.54	0.53	0.60	0.57
31	0.67	0.60	0.63	0.63	0.63	1.00	0.69	0.60	0.63	0.57	0.56
32	0.62	0.64	0.65	0.65	0.65	1.00	0.77	0.64	0.64	0.54	0.62
33	0.94	0.61	0.59	0.59	0.59	1.00	1.02	0.61	0.58	0.69	0.62
34	0.73	0.76	0.72	0.72	0.76	1.00	1.01	0.76	0.75	0.85	0.80
35	0.60	0.63	0.63	0.63	0.66	1.00	0.65	0.63	0.65	0.57	0.66
36	0.64	0.62	0.63	0.63	0.63	1.00	0.74	0.62	0.62	0.55	0.59
37	0.47	0.57	0.58	0.58	0.58	1.00	0.62	0.57	0.58	0.41	0.56
38	0.59	0.62	0.62	0.62	0.62	1.00	0.79	0.62	0.61	0.59	0.59
39	0.53	0.53	0.53	0.53	0.53	1.00	1.01	0.53	0.52	0.54	0.53
40	0.54	0.53	0.63	0.63	0.53	1.00	0.85	0.53	0.54	0.48	0.50
41	0.60	0.66	0.57	0.57	0.66	1.00	0.66	0.66	0.67	0.61	0.68
42	0.51	0.62	0.56	0.56	0.62	1.00	0.64	0.62	0.62	0.73	0.59

Sigue en la página siguiente.

Serie	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
-------	----	----	----	----	----	----	----	----	----	-----	-----

Tabla A.5: Tabla donde se muestra para todas las series del Conjunto 4 los errores OWA producidos por cada método.

Conjunto 5											
Serie	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
1	0.78	1.37	0.69	0.88	0.68	1.00	1.13	1.37	1.00	0.91	0.87
2	0.85	1.50	1.03	1.00	1.16	1.00	1.45	1.50	1.04	0.98	1.04
3	0.99	1.39	1.00	1.07	0.97	1.00	1.61	1.39	1.03	1.16	0.95
4	0.74	1.35	1.05	1.12	1.10	1.00	1.97	1.35	1.08	1.28	1.10
5	0.91	1.37	1.03	1.01	0.92	1.00	1.34	1.37	0.96	1.39	1.15
6	0.85	1.45	0.99	1.00	1.03	1.00	1.31	1.45	1.09	0.80	0.91
7	1.09	1.21	1.66	1.36	1.31	1.00	1.75	1.21	1.28	1.45	1.16
8	1.00	1.05	1.47	0.81	1.11	1.00	1.44	1.05	0.96	1.55	1.08
9	1.08	0.97	1.43	1.14	1.11	1.00	1.40	0.97	1.03	1.51	1.15
10	0.96	1.30	1.98	1.92	1.32	1.00	1.22	1.30	0.99	1.21	1.27
11	0.92	0.95	1.14	0.89	0.94	1.00	1.18	0.95	0.88	1.06	0.95
12	1.25	1.36	1.97	1.46	1.40	1.00	0.97	1.36	1.42	1.49	1.40
13	0.63	1.24	0.76	0.90	0.72	1.00	1.09	1.24	0.92	0.87	0.97
14	0.75	1.30	0.70	0.81	0.71	1.00	1.00	1.30	0.81	0.58	0.72
15	0.69	1.30	0.69	0.80	0.79	1.00	1.40	1.30	0.87	0.83	0.73
16	0.84	1.03	0.89	0.86	0.92	1.00	1.06	1.03	0.94	0.99	1.14
17	1.05	1.52	1.06	1.14	1.13	1.00	1.19	1.52	1.08	0.99	0.95
18	0.76	1.25	0.75	0.98	0.80	1.00	0.98	1.25	0.84	0.85	0.78
19	0.89	1.42	0.66	0.81	0.65	1.00	1.02	1.42	1.06	0.78	0.86
20	0.89	1.51	1.01	1.01	1.08	1.00	1.41	1.51	1.00	0.88	0.95
21	1.00	1.44	1.02	1.06	1.08	1.00	1.38	1.44	1.00	1.06	0.95
22	0.76	1.37	1.02	1.02	1.10	1.00	1.82	1.37	1.03	1.04	0.95
23	0.88	1.35	1.01	0.97	0.95	1.00	1.16	1.35	0.90	1.31	1.35
24	0.86	1.43	0.95	0.98	1.04	1.00	1.44	1.43	1.04	0.71	0.94
25	1.03	1.31	1.51	1.21	1.31	1.00	1.48	1.31	1.37	1.73	1.36
26	1.03	1.12	1.47	0.87	0.78	1.00	1.47	1.12	0.93	1.49	1.05
27	1.02	1.00	1.40	0.88	0.90	1.00	1.32	1.00	1.00	1.58	1.15
28	1.06	1.36	1.60	1.55	1.03	1.00	1.21	1.36	1.01	1.37	1.35
29	0.90	0.98	1.06	0.88	1.03	1.00	1.48	0.98	0.88	1.00	0.97
30	1.83	1.27	2.10	1.79	1.56	1.00	1.19	1.27	1.33	1.64	1.30

Sigue en la página siguiente.

Serie	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
31	0.60	1.25	0.76	0.89	0.68	1.00	0.99	1.25	0.92	0.94	0.95
32	0.88	1.30	0.69	0.77	0.70	1.00	1.12	1.30	0.80	0.54	0.75
33	0.78	1.33	0.68	0.72	0.72	1.00	1.35	1.33	0.85	0.81	0.69
34	1.10	1.04	0.93	0.89	0.98	1.00	1.08	1.04	0.99	1.36	1.08
35	1.05	1.48	1.08	1.16	1.16	1.00	1.24	1.48	1.06	1.36	1.21
36	0.72	1.23	0.74	0.96	0.78	1.00	0.85	1.23	0.82	0.84	0.71
37	1.27	0.93	1.27	1.25	1.08	1.00	1.21	0.93	1.04	1.08	1.01
38	0.93	0.78	0.85	0.87	1.08	1.00	1.27	0.78	1.08	0.94	0.88
39	0.82	0.98	0.76	0.76	0.85	1.00	1.21	0.98	0.88	1.33	1.00
40	0.88	0.76	0.83	0.83	0.95	1.00	1.09	0.76	1.00	0.95	0.78
41	1.18	1.05	0.92	0.92	1.42	1.00	1.58	1.05	1.09	1.15	1.01
42	2.50	0.84	0.85	0.85	1.46	1.00	1.52	0.84	1.13	1.06	0.83
43	1.35	0.85	1.40	1.48	1.16	1.00	1.04	0.85	1.06	1.84	1.08
44	0.90	0.68	0.80	0.79	0.84	1.00	1.22	0.68	1.36	1.02	0.94
45	0.89	0.94	0.77	0.75	0.90	1.00	1.46	0.94	1.02	0.81	0.77
46	1.76	1.01	0.87	0.87	0.90	1.00	1.54	1.01	1.44	1.47	1.29
47	1.36	0.92	1.09	1.09	1.20	1.00	2.16	0.92	1.24	1.69	1.58
48	2.22	0.76	0.85	0.86	1.59	1.00	1.64	0.76	1.41	0.88	0.81
49	0.85	1.03	0.65	1.06	0.78	1.00	1.13	1.03	0.80	0.80	0.63
50	1.38	1.11	0.95	1.11	1.12	1.00	1.31	1.11	1.12	1.15	1.32
51	1.18	1.06	0.97	1.00	1.14	1.00	1.03	1.06	1.09	1.10	1.05
52	1.16	1.12	1.02	1.02	0.98	1.00	1.08	1.12	0.81	1.02	1.01
53	1.06	1.23	0.98	0.98	0.98	1.00	1.01	1.23	1.03	0.90	1.00
54	0.82	1.20	0.83	0.84	0.84	1.00	0.92	1.20	0.90	0.93	0.89
55	0.98	1.24	0.88	0.78	0.65	1.00	0.96	1.24	0.91	0.69	0.73
56	0.94	1.40	1.08	1.07	1.05	1.00	1.57	1.40	0.96	0.77	0.81
57	0.93	1.46	1.15	1.20	0.94	1.00	1.70	1.46	0.99	0.99	1.05
58	1.10	1.08	1.14	1.22	1.13	1.00	1.24	1.08	1.04	0.99	1.05
59	1.07	1.11	0.90	0.95	0.90	1.00	1.26	1.11	0.89	0.91	1.08
60	0.86	1.43	0.99	1.02	0.97	1.00	1.50	1.43	1.05	0.80	0.99
61	0.98	1.24	1.26	1.29	1.23	1.00	1.33	1.24	1.34	1.31	1.80
62	0.92	1.06	0.95	0.76	0.77	1.00	1.31	1.06	1.00	1.37	1.06
63	1.01	1.03	1.44	1.03	0.90	1.00	1.69	1.03	0.99	1.01	1.05
64	1.12	1.17	1.46	0.99	1.31	1.00	1.26	1.17	0.99	1.07	1.32
65	0.96	1.02	1.01	1.01	1.00	1.00	1.82	1.02	1.01	1.07	1.04
66	1.51	1.19	1.40	1.48	1.37	1.00	1.37	1.19	1.31	1.27	1.46
67	0.61	1.18	0.66	0.94	0.71	1.00	1.01	1.18	0.92	0.76	0.87
68	0.68	1.21	0.70	0.69	0.75	1.00	0.81	1.21	0.72	0.65	0.68

Sigue en la página siguiente.

Serie	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
69	0.86	1.36	0.78	1.00	0.74	1.00	0.89	1.36	0.85	0.90	0.86
70	1.13	1.11	0.86	1.08	1.00	1.00	1.21	1.11	1.01	1.09	0.94
71	1.20	1.18	1.32	1.32	1.13	1.00	1.22	1.18	1.10	0.96	1.07
72	0.83	1.15	0.74	0.86	0.68	1.00	0.68	1.15	0.72	0.75	0.76
73	0.62	0.66	0.62	0.74	0.66	1.00	0.78	0.66	0.59	0.69	1.14
74	0.86	0.64	0.74	0.76	0.71	1.00	0.68	0.64	0.66	0.85	0.93
75	1.01	0.86	0.86	0.84	0.94	1.00	1.26	0.86	0.92	1.00	1.00
76	1.30	0.77	1.19	1.19	1.13	1.00	2.92	0.77	0.90	1.04	0.84
77	0.69	0.76	0.85	0.85	0.87	1.00	0.83	0.76	0.75	0.69	0.77
78	1.26	0.86	0.97	0.97	0.95	1.00	1.04	0.86	0.80	1.32	0.80
79	0.58	0.78	0.44	0.42	0.49	1.00	0.91	0.78	0.78	0.68	0.60
80	0.89	0.71	0.85	0.85	0.84	1.00	0.88	0.71	0.72	0.80	0.86
81	0.93	0.81	0.79	0.80	0.82	1.00	0.79	0.81	0.84	1.05	0.88
82	0.96	0.83	1.39	1.36	1.24	1.00	1.01	0.83	0.97	1.01	0.97
83	1.11	1.13	0.86	0.86	1.00	1.00	0.84	1.13	0.97	1.05	1.00
84	0.92	0.90	0.96	0.91	0.79	1.00	0.94	0.90	0.78	0.85	0.97
85	0.43	0.78	0.97	1.01	0.82	1.00	0.74	0.78	0.76	0.96	0.88
86	0.94	0.81	0.92	0.92	0.86	1.00	0.89	0.81	0.83	1.04	0.89
87	1.27	0.91	1.01	1.01	1.25	1.00	1.11	0.91	0.93	1.20	1.01
88	1.07	1.10	1.14	1.14	1.00	1.00	0.96	1.10	1.06	0.71	1.00
89	0.97	1.21	1.00	0.96	0.94	1.00	0.90	1.21	0.84	0.88	0.97
90	0.89	0.80	1.07	0.97	1.05	1.00	1.15	0.80	0.97	1.21	0.93

Tabla A.6: Tabla donde se muestra para todas las series del Conjunto 5 los errores OWA producidos por cada método.

A.2. Mejor modelo, tiempo y OWA

Tablas que muestran para cada serie de cada conjunto el método que generó mejores predicciones, junto con su tiempo y su error obtenidos. Estas tablas hacen referencia a la Tabla 4.5 del Capítulo 4.

Conjunto 1			
Serie	Mejor modelo	Tiempo	OWA
1	M11	22.54	0.48
2	M10	75.57	0.65

Sigue en la página siguiente.

Serie	Mejor modelo	Tiempo	OWA
3	M10	98.24	0.75
4	M11	20.96	0.72
5	M3	33.84	0.84
6	M10	60.17	0.83
7	M11	215.01	0.61
8	M10	60.36	0.70
9	M11	20.60	0.81
10	M11	9.76	0.78
11	M10	56.87	0.70
12	M10	60.43	0.85
13	M11	8.14	0.68
14	M10	142.85	0.73
15	M11	9.43	0.77
16	M11	5.97	0.78
17	M11	4.32	0.89
18	M5	11.22	1.00
19	M10	4.92	0.66
20	M10	4.45	0.71
21	M11	4.75	0.66
22	M11	5.02	0.82
23	M10	4.58	0.61
24	M11	4.40	0.66
25	M6	0.08	1.00
26	M6	0.13	1.00
27	M6	0.07	1.00
28	M11	4.45	0.88
29	M3	95.23	0.88
30	M11	4.02	0.89
31	M6	0.03	1.00
32	M11	4.65	0.74
33	M10	4.19	0.87
34	M11	4.56	0.63
35	M11	4.40	0.62
36	M11	4.54	0.70
37	M11	4.89	0.61
38	M10	5.45	0.69
39	M1	170.90	0.78
40	M11	4.15	0.74

Sigue en la página siguiente.

Serie	Mejor modelo	Tiempo	OWA
41	M3	17.98	0.82
42	M10	3.95	0.86
43	M11	3.90	0.63
44	M10	3.42	0.72
45	M9	0.14	0.80
46	M9	0.13	0.83
47	M5	12.89	0.72
48	M10	3.60	0.82
49	M11	3.99	0.65
50	M11	3.74	0.72
51	M10	3.16	0.82
52	M11	3.07	0.76
53	M1	23.52	0.95
54	M1	363.84	0.93
55	M10	2.96	0.63
56	M10	2.80	0.70
57	M11	3.23	0.69
58	M3	14.99	0.80
59	M10	2.84	0.63
60	M11	3.17	0.77
61	M11	2.95	0.52
62	M11	3.28	0.90
63	M10	2.77	0.85
64	M10	3.14	0.90
65	M11	2.95	0.94
66	M11	2.95	0.93
67	M11	3.29	0.52
68	M11	3.03	0.85
69	M7	25.03	0.87
70	M1	39.05	0.92
71	M5	5.00	0.79
72	M6	0.03	1.00
73	M11	3.54	0.70
74	M11	2.98	0.82
75	M10	2.86	0.86
76	M11	2.96	0.99
77	M4	48.54	0.87
78	M10	3.44	0.95

Sigue en la página siguiente.

Serie	Mejor modelo	Tiempo	OWA
79	M11	3.06	0.75
80	M11	3.17	0.96
81	M1	82.42	0.93
82	M10	3.32	0.91
83	M5	4.92	0.75
84	M6	0.05	1.00

Tabla A.7: Tabla donde se muestra para cada serie del Conjunto 1 el método que mejores predicciones obtuvo, el tiempo que tardó en generar las predicciones y el error OWA obtenido.

Conjunto 2			
Serie	Mejor modelo	Tiempo	OWA
1	M5	10.35	0.25
2	M5	10.78	0.23
3	M1	38.10	0.26
4	M1	6.74	0.08
5	M5	10.55	0.08
6	M11	23.57	0.06
7	M10	24.33	0.10
8	M1	10.32	0.09
9	M1	9.26	0.10
10	M3	94.21	0.10
11	M1	34.59	0.09
12	M10	11.34	0.11
13	M2	0.16	0.76
14	M1	20.20	0.67
15	M2	0.18	0.90
16	M10	10.34	0.57
17	M11	18.39	0.63
18	M9	0.28	0.70
19	M2	0.17	0.69
20	M3	25.50	0.60
21	M2	0.13	0.90
22	M1	4.72	0.59
23	M1	4.85	0.71

Sigue en la página siguiente.

Serie	Mejor modelo	Tiempo	OWA
24	M1	5.54	0.63
25	M3	27.80	0.51
26	M11	7.14	0.57
27	M4	36.96	0.66
28	M10	8.94	0.59
29	M10	6.94	0.57
30	M9	0.31	0.80
31	M4	62.44	0.61
32	M3	23.87	0.68
33	M4	35.48	0.73
34	M10	29.40	0.64
35	M10	46.64	0.68
36	M1	4.18	0.61
37	M1	37.77	0.55
38	M1	14.01	0.55
39	M5	2.75	0.68
40	M10	5.07	0.74
41	M11	4.96	0.86
42	M2	0.08	0.88
43	M11	4.21	0.58
44	M11	5.67	0.60
45	M10	4.72	0.62
46	M10	6.04	0.69
47	M2	0.07	0.69
48	M5	4.49	0.86
49	M4	25.50	0.82
50	M10	5.37	0.75
51	M11	18.17	0.86
52	M6	0.10	1.00
53	M6	0.10	1.00
54	M9	0.23	0.89
55	M9	0.30	0.63
56	M7	21.58	0.72
57	M11	4.67	0.68
58	M9	0.16	0.73
59	M11	14.01	0.50
60	M5	5.34	0.77

Sigue en la página siguiente.

Serie	Mejor modelo	Tiempo	OWA
-------	--------------	--------	-----

Tabla A.8: Tabla donde se muestra para cada serie del Conjunto 2 el método que mejores predicciones obtuvo, el tiempo que tardó en generar las predicciones y el error OWA obtenido.

Conjunto 3			
Serie	Mejor modelo	Tiempo	OWA
1	M10	9.58	0.81
2	M1	195.83	0.92
3	M11	4.29	0.90
4	M11	4.62	0.81
5	M10	3.72	0.84
6	M1	38.55	0.63
7	M10	6.66	0.70
8	M11	6.15	0.79
9	M11	7.58	0.68
10	M11	3.82	0.67
11	M1	76.07	0.90
12	M1	120.62	0.95
13	M11	4.97	0.97
14	M10	4.09	0.94
15	M1	57.20	0.92
16	M10	4.02	0.81
17	M11	3.88	0.76
18	M11	3.76	0.81
19	M11	4.10	0.63
20	M11	4.27	0.81
21	M11	4.49	0.89
22	M11	4.00	1.00
23	M1	60.72	0.89
24	M3	29.21	0.89
25	M11	5.03	0.89
26	M10	5.40	0.72
27	M10	4.50	0.76
28	M5	12.91	0.76
29	M10	3.30	0.90
30	M10	3.80	0.85

Sigue en la página siguiente.

Serie	Mejor modelo	Tiempo	OWA
31	M9	0.19	0.89
32	M10	4.88	0.87
33	M2	0.06	0.86
34	M11	3.73	0.90
35	M9	0.18	0.84
36	M10	3.86	0.83
37	M6	0.04	1.00
38	M10	4.25	0.85
39	M5	13.95	0.86
40	M11	4.12	0.81
41	M10	4.13	0.68
42	M11	4.07	0.84
43	M11	3.74	0.76
44	M10	4.02	0.88
45	M10	3.81	0.82
46	M11	3.97	0.81
47	M10	3.98	0.82
48	M5	12.04	0.73
49	M10	3.35	0.75
50	M1	38.42	0.85

Tabla A.9: Tabla donde se muestra para cada serie del Conjunto 3 el método que mejores predicciones obtuvo, el tiempo que tardó en generar las predicciones y el error OWA obtenido.

Conjunto 4			
Serie	Mejor modelo	Tiempo	OWA
1	M11	10.37	0.71
2	M11	20.26	0.68
3	M10	15.32	0.64
4	M7	61.10	0.93
5	M11	13.10	0.68
6	M11	12.36	0.78
7	M11	42.56	0.55
8	M11	27.22	0.81
9	M3	96.18	0.73

Sigue en la página siguiente.

Serie	Mejor modelo	Tiempo	OWA
10	M11	38.76	0.67
11	M11	31.91	0.61
12	M11	11.22	0.69
13	M6	0.09	1.00
14	M11	29.98	0.84
15	M11	12.22	0.71
16	M1	101.75	0.67
17	M11	29.92	0.65
18	M10	24.28	0.74
19	M2	0.11	0.87
20	M1	27.22	0.68
21	M11	54.67	0.61
22	M10	225.53	0.71
23	M11	8.20	0.68
24	M10	21.23	0.55
25	M5	3.58	0.67
26	M11	5.46	0.64
27	M1	53.75	0.82
28	M1	48.13	0.70
29	M9	0.12	0.43
30	M7	31.78	0.53
31	M11	4.67	0.56
32	M10	7.43	0.54
33	M9	0.08	0.58
34	M3	4.99	0.72
35	M10	13.35	0.57
36	M10	15.11	0.55
37	M10	24.49	0.41
38	M11	4.75	0.59
39	M9	0.16	0.52
40	M10	5.53	0.48
41	M3	2.82	0.57
42	M1	13.91	0.51

Tabla A.10: Tabla donde se muestra para cada serie del Conjunto 4 el método que mejores predicciones obtuvo, el tiempo que tardó en generar las predicciones y el error OWA obtenido.

Conjunto 5			
Serie	Mejor modelo	Tiempo	OWA
1	M5	3.37	0.69
2	M1	9.15	0.85
3	M11	6.49	0.95
4	M1	6.35	0.74
5	M1	14.45	0.91
6	M10	5.57	0.80
7	M6	0.14	1.00
8	M4	46.25	0.81
9	M2	0.07	0.97
10	M1	6.67	0.96
11	M9	0.24	0.88
12	M7	21.37	0.97
13	M1	10.34	0.63
14	M10	7.14	0.58
15	M1	12.98	0.69
16	M1	7.85	0.84
17	M11	15.92	0.95
18	M3	15.59	0.75
19	M5	2.96	0.65
20	M10	7.32	0.88
21	M11	7.31	0.95
22	M1	10.32	0.76
23	M1	8.74	0.88
24	M10	7.12	0.71
25	M6	0.08	1.00
26	M5	2.32	0.77
27	M4	25.37	0.87
28	M6	0.11	1.00
29	M4	21.95	0.87
30	M6	0.06	1.00
31	M1	9.18	0.60
32	M10	5.31	0.54
33	M3	16.26	0.68
34	M4	42.66	0.89
35	M6	0.07	1.00
36	M11	6.24	0.71

Sigue en la página siguiente.

Serie	Mejor modelo	Tiempo	OWA
37	M2	0.09	0.93
38	M2	0.13	0.78
39	M4	29.80	0.76
40	M2	0.04	0.76
41	M4	32.43	0.92
42	M11	6.21	0.83
43	M2	0.12	0.85
44	M2	0.08	0.68
45	M4	27.53	0.75
46	M4	9.68	0.87
47	M2	0.04	0.92
48	M2	0.09	0.76
49	M11	7.26	0.63
50	M3	14.12	0.95
51	M3	13.96	0.97
52	M9	0.10	0.81
53	M10	4.31	0.90
54	M1	6.37	0.83
55	M5	2.39	0.65
56	M10	8.54	0.77
57	M1	5.86	0.93
58	M10	3.40	0.99
59	M9	0.13	0.89
60	M10	4.86	0.80
61	M1	5.65	0.98
62	M4	33.29	0.76
63	M5	2.10	0.90
64	M4	18.40	1.00
65	M1	4.26	0.96
66	M6	0.07	1.00
67	M1	5.17	0.61
68	M10	20.83	0.65
69	M5	2.28	0.74
70	M3	15.14	0.86
71	M10	3.86	0.96
72	M5	2.04	0.68
73	M9	0.21	0.59
74	M2	0.08	0.64
75	M4	19.70	0.84

Sigue en la página siguiente.

Serie	Mejor modelo	Tiempo	OWA
76	M2	0.08	0.77
77	M1	3.71	0.69
78	M9	0.12	0.80
79	M4	15.67	0.42
80	M2	0.08	0.71
81	M7	9.49	0.79
82	M2	0.08	0.83
83	M7	10.13	0.84
84	M9	0.13	0.78
85	M1	9.22	0.43
86	M2	0.04	0.81
87	M2	0.06	0.91
88	M10	14.13	0.71
89	M9	0.10	0.84
90	M2	0.08	0.80

Tabla A.11: Tabla donde se muestra para cada serie del Conjunto 5 el método que mejores predicciones obtuvo, el tiempo que tardó en generar las predicciones y el error OWA obtenido.

A.3. Comparación de métodos

Tabla que muestra para cada serie del conjunto de prueba una comparación de los método FFORMA y MFFORMA a partir de sus errores OWA, su cobertura, sus tiempos de ejecución y las longitudes de sus intervalos de predicción.

Conjunto de Prueba								
Serie	OWAFF	OWAMF	cobFF	cobMF	TiempoFF	TiempoMF	LongFF	LongMF
1	0.90	0.66	100.00	100.00	34.50	0.63	61425.00	107107.49
2	0.84	0.78	83.33	100.00	34.50	0.28	15002.33	34828.55
3	0.95	0.99	83.33	100.00	34.50	0.30	10422.41	16625.32
4	0.63	0.59	100.00	100.00	34.50	0.68	57906.82	92502.39
5	0.97	0.92	100.00	100.00	34.50	1.24	251807.95	405089.70
6	0.52	0.60	83.33	100.00	34.50	0.31	11891.20	19135.54
7	0.73	0.71	100.00	100.00	34.50	69.02	4632.97	7660.24
8	0.64	0.63	100.00	91.67	34.50	0.01	3807.95	6658.14
9	0.36	0.35	100.00	100.00	34.50	1.22	2168.54	3137.11

Sigue en la página siguiente.

Serie	OWAFF	OWAMF	cobFF	cobMF	TiempoFF	TiempoMF	LongFF	LongMF
10	1.06	1.09	100.00	83.33	34.50	0.00	4006.15	7139.99
11	0.68	0.60	100.00	100.00	11.50	0.53	45.99	71.65
12	0.21	0.10	100.00	100.00	11.50	7.64	41.41	76.20
13	0.91	0.99	100.00	100.00	11.50	0.47	7.70	12.47
14	1.05	1.15	75.00	100.00	11.50	8.36	9.62	14.21
15	1.08	1.06	100.00	100.00	11.50	5.31	23.67	36.11
16	0.69	0.90	100.00	100.00	11.50	1.55	10.19	19.92
17	0.56	0.57	100.00	100.00	11.50	1.87	18.38	24.86
18	0.63	0.68	100.00	100.00	11.50	4.79	14.41	20.55
19	0.45	0.45	100.00	100.00	11.50	8.28	9.05	15.11
20	1.10	0.93	100.00	100.00	11.50	4.23	8.81	14.87
21	0.95	0.94	100.00	100.00	34.50	0.83	12773.94	26636.30
22	0.62	0.75	75.00	100.00	34.50	79.48	1137.75	2370.05
23	1.26	0.57	91.67	100.00	34.50	1.57	279.57	456.00
24	1.08	0.77	66.67	100.00	34.50	1.83	298.61	556.44
25	0.76	0.85	83.33	100.00	34.50	38.28	1540.24	2862.61
26	0.44	0.67	91.67	100.00	34.50	6.64	1985.53	4218.41
27	0.57	0.56	91.67	100.00	34.50	8.97	925.32	2613.29
28	0.60	0.64	91.67	100.00	34.50	49.77	251.72	601.60
29	0.70	0.93	66.67	100.00	34.50	8.06	194.12	383.20
30	0.43	0.79	100.00	100.00	34.50	0.28	744.25	2244.96
31	1.04	1.74	91.67	100.00	34.50	0.53	24.03	32.19
32	0.95	1.02	83.33	100.00	34.50	38.34	26.38	31.01
33	1.33	1.31	100.00	100.00	34.50	1.50	10.69	15.03
34	0.37	0.37	100.00	100.00	34.50	1.65	17.68	23.46
35	0.80	0.82	100.00	100.00	34.50	0.01	24.70	26.57
36	1.23	1.23	100.00	100.00	34.50	0.00	39.36	38.38
37	0.82	0.84	100.00	100.00	34.50	0.02	23.75	25.20
38	0.60	0.62	83.33	66.67	34.50	0.01	21.36	25.05
39	0.84	0.82	91.67	100.00	34.50	0.02	11.45	12.86
40	0.40	0.35	75.00	83.33	34.50	0.01	137.99	43.05
41	0.84	0.32	100.00	100.00	11.50	0.24	3104.17	4477.05
42	0.54	0.67	100.00	100.00	11.50	0.25	1322.87	2312.09
43	1.00	0.97	75.00	100.00	11.50	5.92	11.02	20.23
44	0.98	1.01	75.00	100.00	11.50	0.25	26.70	40.26
45	0.69	0.36	100.00	100.00	11.50	0.47	10852517.75	19524397.02
46	0.73	0.54	100.00	100.00	11.50	0.64	411321.06	499069.02
47	0.79	0.78	100.00	100.00	11.50	3.80	203861.15	239259.40
48	0.36	0.53	100.00	100.00	11.50	4.09	54806.94	103711.55

Sigue en la página siguiente.

Serie	OWAFF	OWAMF	cobFF	cobMF	TiempoFF	TiempoMF	LongFF	LongMF
49	0.78	0.82	100.00	100.00	11.50	0.23	41787.71	56210.55
50	0.73	0.64	100.00	100.00	11.50	0.46	132419.61	168910.29

Tabla A.12: Tabla de comparación del método FFORMA con el MFFORMA

Tabla donde se muestran los métodos utilizados para la combinación lineal del método MFFORMA.

Conjunto de Prueba	
Serie	Funciones utilizadas en la predicción final
1	M1, M9 y M12
2	M9 y M1
3	M1 y M9
4	M1 y M11
5	M1, M6 y M5
6	M1 y M9
7	M9, M1, M12 y M5
8	M9
9	M9 y M5
10	M9
11	M11 y M1
12	M1 y M1
13	M1, M6 y M11
14	M1 y M1
15	M4, M9, M11, M1 y M6
16	M9, M11 y M4
17	M11, M4, M6, M9 y M5
18	M4, M1 y M9
19	M9, M11, M1, M4 y M1
20	M1, M4, M1 y M9
21	M5 y M6
22	M5, M1 y M4
23	M11, M5 y M6
24	M1, M5, M12 y M9
25	x13_de, M11, M1 y M5
26	M11 y M4
27	M4, M1, M9 y M2

Sigue en la página siguiente.

Serie	Funciones utilizadas en la predicción final
28	M4, M1, M11 y M2
29	M11, M12, M4 y M1
30	M11
31	M11 y M1
32	M12, M5, M1, M4 y M1
33	M9, M5 y M4
34	M9, M4 y M5
35	M12
36	M12
37	M12
38	M12 y M9
39	M12 y M9
40	M12
41	M11
42	M11
43	M1, M11 y M2
44	M11
45	M1 y M11
46	M11 y M1
47	M1 y M4
48	M2, M11, M1 y M1
49	M11
50	M11 y M1

Tabla A.13: Métodos utilizados para la combinación lineal de las predicciones del método MFFORMA

Apéndice B

Código de R

B.1. Trabajo preparatorio

En esta sección pondremos el código con el que realizamos el trabajo preparatorio de las series proporcionadas por el IGE necesario para poder aplicarle los métodos BFM, FFORMA y MFFORMA.

Extracción de los conjuntos de series de la base de datos del IGE

```
datos1=read.csv("http://www.ige.eu/igebdt/igeapi/datos/3476") # Conjunto 1
datos2=read.csv("http://www.ige.eu/igebdt/igeapi/datos/6356") # Conjunto 2
datos3=read.csv("http://www.ige.eu/igebdt/igeapi/datos/308") # Conjunto 3
datos4=read.csv("http://www.ige.eu/igebdt/igeapi/datos/9048") # Conjunto 4
datos5=read.csv("http://www.ige.eu/igebdt/igeapi/datos/4052") # Conjunto 5
datos <- list(datos1,datos2,datos3,datos4,datos5)
```

Trabajo preparatorio para todos los conjuntos de series a la vez

```
# Identificación de las variables no factoriales
varcat <- list()
unicos <- list()
a1 <- list()
a <- list()
menor24 <- list()
zeros <- list()
men24=numeric()
ceros=numeric()
seri <- list()
for (j in 1:length(datos)) {
  u1 <- which(names(datos[[j]])=="CodTempo")
  u2 <- which(names(datos[[j]])=="Tempo")
  u3 <- which(names(datos[[j]])=="Espazo")
  u4 <- which(names(datos[[j]])=="DatoN")
  u5 <- which(names(datos[[j]])=="DatoT")
  varcat[[j]] <- (1:length(datos[[j]]))[-c(u1,u2,u3,u4,u5)]

# Extracción de series
unicos[[j]]=unique(datos[[j]][,varcat[[j]])
for (i in 1:nrow(unicos[[j]])){
  b=merge(unicos[[j]][i,],datos[[j]],all.x=TRUE)
  b=b[order(b$CodTempo),]
```

```

b$ano=as.numeric(substring(b$CodTempo,1,4))
b$mes=as.numeric(substring(b$CodTempo,5,6))
a1[[i]]=b
a1[[i]]=subset(a1[[i]],!(is.na(mes)))
}

#identificar las series con menos de 24 datos y con mas de 24 ceros, y eliminarlas
for (i in 1:nrow(unicos[[j]]) ) {
  men24[i]=sum(!is.na(a[[j]][[i]]$DatoN))<25
  ceros[i]=sum((a[[j]][[i]]$DatoN)==0)>24
}
menor24[[j]] <- which(men24==1)
zeros[[j]] <- which(ceros==1)
seri[[j]] <- unique(sort(c(zeros[[j]],menor24[[j]])))
if(sum(seri[[j]])==0){unicos[[j]]=unicos[[j]]}else {unicos[[j]] <- unicos[[j]][-seri[[j]],]}
}

# Extracción de series
a1 <- list()
a <- list()
for (j in 1:length(datos)) {
  for (i in 1:nrow(unicos[[j]]) ){
    b=merge(unicos[[j]][i,],datos[[j]],all.x=TRUE)
    b=b[order(b$CodTempo),]
    b$ano=as.numeric(substring(b$CodTempo,1,4))
    b$mes=as.numeric(substring(b$CodTempo,5,6))
    a1[[i]]=b
    a1[[i]]=subset(a1[[i]],!(is.na(DatoN)))
    a1[[i]]=subset(a1[[i]],!(is.na(mes)))
    if(a1[[i]]$mes>12){a1[[i]]$mes=a1[[i]]$mes-12}else {a1[[i]]$mes=a1[[i]]$mes}
  }
  a[[j]]=a1
}

# Cálculo de la frecuencia
frec <- list()
h <- list()
for (j in 1:length(datos)) {
  frec[[j]]=length(which(a[[j]][[1]]$ano==(a[[j]][[1]]$ano[1]+1)))
  if (frec[[j]]==12) {h[[j]]=12} else if (frec[[j]]==4) {h[[j]]=4} else if (frec[[j]]==3) {h[[j]]=3} else {h[[j]]=1}
}

# Transformación de los datos a formato serie temporal
seri=list()
series=list()
for (j in 1:length(datos)) {
  for (i in 1:nrow(unicos[[j]]) ) {
    seri[[i]]=ts(a[[j]][[i]]$DatoN,start = c(a[[j]][[i]]$ano[1],a[[j]][[i]]$mes[1]),
    frequency=frec[[j]])
  }
  series[[j]]=seri
}

```

Bucle que pone las series en el formato utilizado en el FFORMA

```

dat=list()
data=list()
k=1
for (j in 1:length(datos)) {
  for (i in 1:nrow(unicos[[j]])) {
    dat[[i]] <- list(x=series[[j]][[i]],n=length(series[[j]][[i]]),h=h[[j]])
    data[[k]]=dat[[i]]
    k=k+1
  }
}

```

B.2. Método BFM

Cargamos las librerías

```

library(M4metalearning)
library(tsfeatures)
library(M4comp2018)
library(xgboost)
library(RJDemetra)
library(x13binary)
library(rJava)
library(forecast)
library(tictoc)

```

Cálculo de las predicciones por el método BFM

```

serieoriginal=list()
for (i in 1:nrow(unicos)) {
  serieoriginal[[i]] <- series[[i]]
}

serie=list()
seriecheck=list()
seriepredic=list()
seriecheckfinal=list()
errores=list()
predicciones=list()
lista1=list()
lista2=list()
lista3=list()
lista4=list()
lista5=list()
lista6=list()
lista7=list()
lista8=list()
lista9=list()
lista10=list()
lista11=list()
timings=list()
tiempos=list()
for (i in 1:nrow(unicos)) {

```

```

for (k in 0:(length(serieoriginal[[i]]) %% (2*h)-1)) {
# Bucle que va añadiendo un año a la mitad de la serie una vez se calculan las
predicciones
seriepredic[[i]] <- window(serieoriginal[[i]], end=start(serieoriginal[[i]])+c((
length(serieoriginal[[i]]) %% (2*h)),length(serieoriginal[[i]]) %% (2*h)))
+c(k,0))
seriepredic[[i]] <- ts(seriepredic[[i]],start =start(serieoriginal[[i]]),
frequency = h)
# Mitad en adelante de las observaciones para calcular luego el error de las
predicciones
seriecheckfinal[[i]] <- window(serieoriginal[[i]], start=end(serie[[i]])+c(1,-h
+1))
seriecheckfinal[[i]] <- ts(seriecheckfinal[[i]],start =(end(serie[[i]])+c(1,-h
+1)),frequency = h)

tic.clearlog()

tic()
lista1[[k+1]]=as.data.frame(forecast(auto.arima(seriepredic[[i]]),h=h))[,1] #
Método ARIMA automático
toc(log = TRUE, quiet = TRUE)

tic()
lista2[[k+1]]=as.data.frame(naive(seriepredic[[i]],h=h))[,1] # Método naive
toc(log = TRUE, quiet = TRUE)

tic()
lista3[[k+1]]=as.data.frame(forecast(bats(seriepredic[[i]]),h=h))[,1] # Método
BATS
toc(log = TRUE, quiet = TRUE)

tic()
lista4[[k+1]]=as.data.frame(forecast(tbats(seriepredic[[i]]),h=h))[,1] # Método
TBATS
toc(log = TRUE, quiet = TRUE)

tic()
lista5[[k+1]]=as.data.frame(forecast(ets(seriepredic[[i]]),h=h))[,1] # Métodos
suavización exponencial
toc(log = TRUE, quiet = TRUE)

tic()
lista6[[k+1]]=as.data.frame(forecast(snaive(seriepredic[[i]]),h=h))[,1] # Método
naive estacional
toc(log = TRUE, quiet = TRUE)

tic()
lista7[[k+1]]=as.data.frame(forecast(nnetar(seriepredic[[i]], size=2),h=h, PI=
TRUE, npaths=100))[,1] # Modelo de red neuronal
toc(log = TRUE, quiet = TRUE)

tic()
lista8[[k+1]]=as.data.frame(rwf(seriepredic[[i]],h=h))[,1] # Método paseo
aleatorio

```



```

    toc(log = TRUE, quiet = TRUE)

    tic()
    lista9[[k+1]]=as.data.frame(forecast(thetaf(seriepredic[[i]]),h=h))[,1] # Método
    Theta
    toc(log = TRUE, quiet = TRUE)

    tic()
    lista10[[k+1]]=as.vector(x13_def(seriepredic[[i]])$final$forecasts[,1]) # Método
    X13-ARIMA-SEATS
    toc(log = TRUE, quiet = TRUE)

    tic()
    lista11[[k+1]]=as.vector(tramoseats_def(seriepredic[[i]])$final$forecasts[,1]) #
    Método TRAMO-SEATS
    toc(log = TRUE, quiet = TRUE)

    log.txt <- tic.log(format = TRUE)
    log.lst <- tic.log(format = FALSE)
    tic.clearlog()

# tiempos de predicción de cada método en cada año
    timings[[k+1]] <- unlist(lapply(log.lst, function(x) x$toc - x$tic))

    print(k)
}

pred1 <- as.numeric(do.call(cbind,lista1))
pred2 <- as.numeric(do.call(cbind,lista2))
pred3 <- as.numeric(do.call(cbind,lista3))
pred4 <- as.numeric(do.call(cbind,lista4))
pred5 <- as.numeric(do.call(cbind,lista5))
pred6 <- as.numeric(do.call(cbind,lista6))
pred7 <- as.numeric(do.call(cbind,lista7))
pred8 <- as.numeric(do.call(cbind,lista8))
pred9 <- as.numeric(do.call(cbind,lista9))
pred10 <- as.numeric(do.call(cbind,lista10))
pred11 <- as.numeric(do.call(cbind,lista11))

# Predicciones de la mitad de las series para cada método
    predicciones[[i]]=list(pred1,pred2,pred3,pred4,pred5,pred6,pred7,pred8,pred9,
    pred10,pred11)

# Tiempos de cada serie y cada método
    tiempos[[i]] <- do.call(cbind,timings)

    print(i)
}

```

Función que calcula el error OWA

```

error <- function(x,y,z){
    smape <- mean((2*abs(x-y))/(abs(x)+abs(y)))

    frq <- stats::frequency(x)

```

```

mase <- (mean(abs(x-y))/(mean(abs(matrix(head(z, -frq)) - matrix(tail(z, -frq))))
)

smape_snaive <- mean((2*abs(x-predicciones[[i]][[6]])/(abs(x)+abs(pred
icciones[[i]][[6]])))
mase_snaive <- (mean(abs(x-predicciones[[i]][[6]]))/(mean(abs(matrix( head
(z, -frq)) - matrix(tail(z, -frq))))))

smape_rel <- smape/smape_snaive
mase_rel <- mase/mase_snaive

0.5*(smape_rel+mase_rel)
}

```

Cálculo de los errores

```

# Métodos
modelos=c("autoarima","naive", "bats", "tbats", "ets", "snaive", "Neuralnetwork", "
rwf", "theta", "seas", "x13", "tramoseats")

# Transformación de las predicciones a formato serie de tiempo
for (i in 1:nrow(unicos)) {
  for (j in 1:length(modelos)) {
    predicciones[[i]][[j]] <- ts(predicciones[[i]][[j]],start =start(seriecheckfinal
[[i]],frequency = h)
    predicciones[[i]][[j]] <- window(predicciones[[i]][[j]], end=end(seriecheckfinal
[[i]]))
  }
}

# Cálculo del error OWA de las predicciones
errores=numeric()
OWA=list()
for (i in 1:nrow(unicos)) {
  for (j in 1:length(modelos)) {
    errores[j]=error(as.numeric(seriecheckfinal[[i]]),as.numeric(predicciones[[i]][[
j]]),as.numeric(serie[[i]]))
  }
  OWA[[i]]=errores
}

```

Cálculo del mejor modelo

```

mejormodeloOWA=vector()
for (i in 1:nrow(unicos)) {
  mejormodeloOWA[i] = modelos[which.min(OWA[[i]])]
}

```

B.3. Método MFFORMA

Funciones y bucles necesarios para llevar a cabo el método MFFORMA.

Función que calcula los intervalos de predicción para los métodos del paquete RJDemetra (X13-ARIMA-SEATS y TRAMO-SEATS).

```
IPRJD <- function(x,y){
  upper <- x+1.96*y
  lower <- x-1.96*y
  data.frame(forecast=x,lower,upper)
}
```

Funciones que calculan las predicciones junto con los intervalos de predicción de cada método

```
snaive_forec <- function(x,h) {
  model <- forecast::snaive(x, h=length(x))
  forecast::forecast(model, h=h)
}

naive_forec <- function(x,h) {
  model <- forecast::naive(x, h=length(x))
  forecast::forecast(model, h=h)
}

auto_arima_forec <- function(x, h) {
  model <- forecast::auto.arima(x, stepwise=FALSE, approximation=FALSE)
  forecast::forecast(model, h=h)
}

ets_forec <- function(x, h) {
  model <- forecast::ets(x)
  forecast::forecast(model, h=h)
}

tbats_forec <- function(x, h) {
  model <- forecast::tbats(x, use.parallel=FALSE)
  forecast::forecast(model, h=h)
}

stlm_ar_forec <- function(x, h) {
  model <- tryCatch({
    forecast::stlm(x, modelfunction = stats::ar)
  }, error = function(e) forecast::auto.arima(x, d=0,D=0))
  forecast::forecast(model, h=h)
}

rw_drift_forec <- function(x, h) {
  model <- forecast::rwf(x, drift=TRUE, h=length(x))
  forecast::forecast(model, h=h)
}

thetaf_forec <- function(x, h) {
  forecast::forecast(thetaf(x, h=h))
}

x13_forec <- function(x,h) {
  x13_model <- RJDemetra::x13_def(x, userdefined = "preprocessing.model.y_ef")
  x <- x13_model$final$forecasts[,1]
  y <- x13_model$user_defined$preprocessing.model.y_ef[1:h]
  IPRJD(x,y)
}
```

```

tramoseats_forec <- function(x,h) {
  ts_model <- RJDemetra::tramoseats_def(x, userdefined = "preprocessing.model.y_ef")
  x <- ts_model$final$forecasts[,1]
  y <- ts_model$user_defined$preprocessing.model.y_ef[1:h]
  IPRJD(x,y)
}

```

Función que llama a las funciones anteriores para calcular las predicciones

```

predics <- function(x,h){
  rbind(auto_arima_forec(x,h)$mean,ets_forec(x,h)$mean,tbats_forec(x,h)$mean,stlm_ar
_forec(x,h)$mean,rw_drift_forec(x,h)$mean,thetaf_forec(x,h)$mean,naive_forec(x
,h)$mean,snaive_forec(x,h)$mean,x13_forec(x,h)[,1],tramoseats_forec(x,h)[,1])
}

```

Función que crea una lista con los métodos utilizados

```

forec_methods <- function() {
methods_list <- list("auto_arima_forec")
methods_list <- append(methods_list, "ets_forec")
methods_list <- append(methods_list, "tbats_forec")
methods_list <- append(methods_list, "stlm_ar_forec")
methods_list <- append(methods_list, "rw_drift_forec")
methods_list <- append(methods_list, "thetaf_forec")
methods_list <- append(methods_list, "naive_forec")
methods_list <- append(methods_list, "snaive_forec")
methods_list <- append(methods_list, "x13_forec")
methods_list <- append(methods_list, "tramoseats_forec")

methods_list
}

```

Función que calcula las predicciones únicamente con los métodos elegidos por el MFFORMA

```

toppreds <- function(x,y,h){
  if(any(x==1)){salidas=auto_arima_forec(y,h)
  p1=salidas$mean
  psup1=salidas$upper[,2]
  pinf1=salidas$lower[,2]}else{p1=psup1=pinf1=rep(0,h)}
  if(any(x==2)){salidas=ets_forec(y,h)
  p2=salidas$mean
  psup2=salidas$upper[,2]
  pinf2=salidas$lower[,2]}else{p2=psup2=pinf2=rep(0,h)}
  if(any(x==3)){salidas=tbats_forec(y,h)
  p3=salidas$mean
  psup3=salidas$upper[,2]
  pinf3=salidas$lower[,2]}else{p3=psup3=pinf3=rep(0,h)}
  if(any(x==4)){salidas=stlm_ar_forec(y,h)
  p4=salidas$mean
  psup4=salidas$upper[,2]
  pinf4=salidas$lower[,2]}else{p4=psup4=pinf4=rep(0,h)}
  if(any(x==5)){salidas=rw_drift_forec(y,h)
  p5=salidas$mean
  psup5=salidas$upper[,2]
  pinf5=salidas$lower[,2]}else{p5=psup5=pinf5=rep(0,h)}
  if(any(x==6)){salidas=thetaf_forec(y,h)

```

```

    p6=salidas$mean
    psup6=salidas$upper[,2]
    pinf6=salidas$lower[,2]}else{p6=psup6=pinf6=rep(0,h)}
if(any(x==7)){salidas=naive_forec(y,h)
    p7=salidas$mean
    psup7=salidas$upper[,2]
    pinf7=salidas$lower[,2]}else{p7=psup7=pinf7=rep(0,h)}
if(any(x==8)){salidas=snaive_forec(y,h)
    p8=salidas$mean
    psup8=salidas$upper[,2]
    pinf8=salidas$lower[,2]}else{p8=psup8=pinf8=rep(0,h)}
if(any(x==9)){salidas=x13_forec(y,h)
    p9=salidas[,1]
    psup9=salidas[,3]
    pinf9=salidas[,2]}else{p9=psup9=pinf9=rep(0,h)}
if(any(x==10)){salidas=tramoseats_forec(y,h)
    p10=salidas[,1]
    psup10=salidas[,3]
    pinf10=salidas[,2]}else{p10=psup10=pinf10=rep(0,h)}

p <- rbind(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10)
psup <- rbind(psup1,psup2,psup3,psup4,psup5,psup6,psup7,psup8,psup9,psup10)
pinf <- rbind(pinf1,pinf2,pinf3,pinf4,pinf5,pinf6,pinf7,pinf8,pinf9,pinf10)

list(forecasts=p,lower=pinf,upper=psup)
}

```

Función que calcula el intervalo de predicción interior para las predicciones finales

```

ensemble_ilower <- function (predictions, dataset, clamp_zero = TRUE){
  for (i in 1:length(dataset)) {
    weighted_ff <- as.vector(t(predictions[i, ]) %*% dataset[[i]]$iinf)
    dataset[[i]]$lower <- weighted_ff
  }
  dataset
}

```

Función que calcula el intervalo de predicción superior para las predicciones finales

```

ensemble_iupper <- function (predictions, dataset, clamp_zero = TRUE){
  for (i in 1:length(dataset)) {
    weighted_ff <- as.vector(t(predictions[i, ]) %*% dataset[[i]]$isup)
    dataset[[i]]$upper <- weighted_ff
  }
  dataset
}

```

Método MFFORMA (periodo de entrenamiento y periodo de prueba)

```

# Series tomadas para el periodo de entrenamiento
d <- c
  (1:50,85:108,110:120,122:136,146:149,151:186,195,197:209,211:228,237,239:249,251:288)

datos_train <- data[d]

# Series tomadas para el conjunto de prueba
e <- c(51:60,109,121,137:144,145,150,187:194,196,210,229:236,238,250,289:296)

```

```

datos_test <- data[e]

# Extracción del último año a la serie para calcular luego los errores
datos_train <- temp_holdout(datos_train)
datos_test <- temp_holdout(datos_test)

# PERIODO DE ENTRENAMIENTO
tic("Tiempo de entrenamiento MFFORMA")

# Cálculo de las predicciones por cada método
for (i in 1:length(datos_train)) {
  datos_train[[i]]$ff <- predic(datos_train[[i]]$x, datos_train[[i]]$h)
}

# Cálculo de los errores
datos_train <- calc_errors(datos_train)

# Cálculo de las características
datos_train <- THA_features(datos_train)

# Organización los datos en una lista para que pueda entrenar el algoritmo
train_data <- create_feat_classif_problem(datos_train)

# Entrenamiento del modelo
meta_model <- train_selection_ensemble(train_data$data, train_data$errors)

toc(log = TRUE)
#Tiempo de entrenamiento MFFORMA
train_time_MF<- tic.log(format = TRUE)

# PERIODO DE PRUEBA

tic("tiempo de prueba MFFORMA")

# Cálculo de las características
datos_test <- THA_features(datos_test, n.cores=1)

# Cálculo de los pesos con el modelo de meta-aprendizaje
test_data <- create_feat_classif_problem(datos_test)
preds <- predict_selection_ensemble(meta_model, test_data$data) # pesos

# métodos de predicción utilizados por el MFFORMA
modelos <- c("auto.arima", "ets", "nnetar", "tbats", "stlm", "rwf", "thetaf", "naive", "
  snaive", "x13", "tramoseats")

# Bucle que calcula los métodos que utilizar para el cálculo de las predicciones
  finales
pesos <- list()
orden <- list()
pesos_ord <- list()
nmin_meth <- numeric()
sum_pesos <- numeric()
mod_opt <- list()
peso <- list()

```

```

orden_final=list()
pesos_reesc=list()
for (i in 1:length(datos_test)) {
  pesos[[i]] <- preds[i,] #pesos en una lista
  orden[[i]] <- order(pesos[[i]], decreasing = T) #indices de los pesos de mayor a
  menor
  pesos_ord[[i]] <- sort(pesos[[i]],decreasing = T) # pesos ordenados
  nmin_meth[i] <- which(cumsum(pesos_ord[[i]])>0.8)[1] # numero minimo de modelos a
  utilizar
  sum_pesos[i] <- cumsum(pesos_ord[[i]])[nmin_meth[i]] #suma de los peso que superan
  0.8
  mod_opt[[i]] <- modelos[orden[[i]][1:nmin_meth[i]]] # modelos a utilizar
  peso[[i]] <- c(pesos_ord[[i]][1:nmin_meth[i]]/sum_pesos[i],rep(0,10-nmin_meth[i]))
  #pesos re-escalados de mayor a menor
  orden_final[[i]] <- order(orden[[i]]) # indices del orden correcto
  pesos_reesc[[i]] <- peso[[i]][orden_final[[i]]] #pesos re-escalados en el orden
  correcto
}
pesos_reesc <- do.call(rbind,pesos_reesc)

# Cálculo de las predicciones y los intervalos de predicción por los métodos
seleccionados
tic.clearlog()
for (i in 1:length(datos_test)) {
  tic(i)
  salidas <- toppreds(orden[[i]][1:nmin_meth[i]],datos_test[[i]]$x,datos_test[[i]]$
  h)
  datos_test[[i]]$ff <- salidas$forecasts
  datos_test[[i]]$iinf <- salidas$lower
  datos_test[[i]]$isup <- salidas$upper
  toc(log = TRUE, quiet = TRUE)
}

log.txt <- tic.log(format = TRUE)
log.lst <- tic.log(format = FALSE)
tic.clearlog()

# tiempo que el método tardó en predecir cada serie
timings <- unlist(lapply(log.lst, function(x) x$toc - x$tic))
mean(timings)

# Cálculo de las predicciones finales
datos_test <- ensemble_forecast(pesos_reesc, datos_test)

# Cálculo de los extremos inferiores de los intervalos de predicción finales
datos_test <- ensemble_ilower(pesos_reesc, datos_test)

# Cálculo de los extremos superiores de los intervalos de predicción finales
datos_test <- ensemble_iupper(pesos_reesc, datos_test)
toc(log = TRUE)
times_MF<- tic.log(format = TRUE)

# Resultados para la serie 3 del conjunto de series de prueba
datos_test[[3]]$y_hat # Predicciones

```

```
datos_test[[3]]$upper # Intervalo superior del IP
datos_test[[3]]$lower # Intervalo inferior del IP
```


Bibliografía

- [1] Assimakopoulos V, Nikolopoulos K (2000). The theta model: a decomposition approach to forecasting. *International Journal of Forecasting* **16**, 521-530.
- [2] Aznar A, Trivez FJ (1993). *Métodos de predicción en economía II. Análisis de series temporales*. Ariel Economía.
- [3] Bergmeir C, Hyndman RJ, Benitez JM (2016). Bagging Exponential Smoothing Methods using STL Decomposition and Box-Cox Transformation. *International Journal of Forecasting*. **32**, 303-312.
- [4] Box GEP, Jenkins GM (1970). *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco.
- [5] Brown RG (1956). *Exponential Smoothing for Predicting Demand*. Cambridge, Massachusetts: Arthur D. Little Inc. p. 15.
- [6] Brown RG (1963). *Smoothing Forecasting and Prediction of Discrete Time Series*. Englewood Cliffs, NJ: Prentice-Hall.
- [7] Burnham KP, Anderson DR (2004). Multimodel inference: understanding AIC and BIC in Model Selection. *Sociological Methods and Research* **33**, 261-304.
- [8] Chen T, Guestrin C (2016) XGBoost: A Scalable Tree Boosting System, *22nd SIGKDD Conference on Knowledge Discovery and Data Mining*, <https://arxiv.org/abs/1603.02754>.
- [9] Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H, Chen K, Mitchell R, Cano I, Zhou T, Li M, Xie J, Lin M, Geng Y, Li Y. (2018). *xgboost: Extreme Gradient Boosting*. R package version 0.71.2. <https://CRAN.R-project.org/package=xgboost>.
- [10] Cleveland RB, Cleveland WS, McRae JE, and Terpenning IJ (1990). STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, **6(1)**, 3-33.
- [11] Izrailev S (2016) *Functions for timing R scripts, as well as implementations of Stack and List structures*. R package version 1.0 <https://cran.r-project.org/web/packages/tictoc/tictoc.pdf>.
- [12] De Livera, Alysha M (2010) Automatic forecasting with a modified exponential smoothing state space framework. *Monash Econometrics and Business Statistics Working Papers*. **10**, no. 10.
- [13] De Livera AM, Hyndman RJ, Snyder RD (2011), Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*. **106(496)**, 1513-1527.

- [14] Gardner Jr ES (1985). Exponential Smoothing: The State of the Art. *Journal of Forecasting*. **4**, 1-28.
- [15] Gómez, J. (2018). Tablas multidimensionales en R. *XX Jornadas de Estadística de las Comunidades Autónomas*. Logroño.
- [16] Qian N (1999). On the momentum term in gradient descent learning algorithms. *Neural Networks*. **12(1)**: 145-151.
- [17] Holt CC (1957). Forecasting Trends and Seasonal by Exponentially Weighted Averages. *Office of Naval Research Memorandum*, Carnegie Institute of Technology **52**.
- [18] Hyndman RJ, Koehler AB, Snyder RD, Grose S (2002). A state space framework for automatic forecasting using exponential smoothing methods. *International J. Forecasting*. **18(3)**, 439-454.
- [19] Hyndman RJ, Billah B. (2003) Unmasking the Theta method. *International Journal of Forecasting* **19**, 287-290
- [20] Hyndman RJ, Koehler, Ord-Snyder (2008) *expsmooth: Data Sets from "Forecasting with Exponential Smoothing"*. R package version 1.11 <http://CRAN.R-project.org/package=forecasting>.
- [21] Hyndman RJ, Koehler AB, Ord JK, Snyder RD (2008b). *Forecasting with Exponential Smoothing: The State Space Approach*. Springer-Verlag. URL <http://www.exponentialsMOOTHING.net/>.
- [22] Hyndman RJ, Khandakar Y. (2008) Automatic time series forecasting: The forecast package for R, *Journal of Statistical Software*, **26(3)**.
- [23] Hyndman RJ, Wang E, Kang Y, Talagala T (2018). *tsfeatures: Time Series Feature Extraction*. R package version 0.1. <https://github.com/robjhyndman/tsfeatures>.
- [24] Hyndman RJ (2018b). Explanation of the FFORMA method for the M4 competition (FFORMA). <https://github.com/robjhyndman/M4metalearning>
- [25] Hyndman RJ (2019). *Forecasting Functions for Time Series and Linear Models*. R package version 8.5 <http://pkg.robjhyndman.com/forecast>.
- [26] M4 competitor's guide. (2018). <https://www.m4.unic.ac.cy/wp-content/uploads/2018/03/M4-Competitors-Guide.pdf>. Accessed: 2018-09-26.
- [27] Maravall A (2008). *Seats part. Signal extraction in ARIMA time series. Notes on programs Tramo and Seats*. Bank of Spain.
- [28] McCallum Q, Weston S (2011). *Parallel R. Data Analysis in the Distributed World*. O'Reilly Media.
- [29] Michlek A, Quartier-la-Tente (2019). *R interface to JDemetra+*. <https://github.com/nbbird/rjdemetra>.
- [30] Montero-Manso P (2019) Quen predí mellor o futuro? Os resultados da competición internacional M4. INFORMEST, *Boletín da SGAPEIO*, Marzo de 2019, **51**.

- [31] Montero-Manso P, Athanasopoulos G, Hyndman RJ, S Talagala T (2018) FFORMA: Featured-based Forecast Model Averaging. *Monash business School*, <http://business.monash.edu/econometrics-and-business-statistics/research/publications>.
- [32] Quartier-la-Tente A, Michalek A, Palate J, Baeyens R (2019). *Interface to 'JDemetra+' Seasonal Adjustment Software*. R package version 0.1.1 <https://cran.r-project.org/web/packages/RJDemetra/RJDemetra.pdf>.
- [33] Sax C (2017). *seasonal: R Interface to X-13-ARIMA-SEATS*. R package version 1.7.0. <https://CRAN.R-project.org/package=seasonal>.
- [34] Shiskin J, Young AH, and Musgrave JC (1967). The X-11 variant of the Census Method II seasonal adjustment program. *U.S. Government Printing Office*. Technical Paper No. **15**, U. S. Census Bureau.
- [35] Smyl S, Ranganathan J, Pasqua A (2018). *Introducing a New Hybrid ES-RNN Model*. Uber Engineering. <https://eng.uber.com/m4-forecasting-competition/>.
- [36] Sutcliffe A (1993). X11 time series decomposition and sampling errors. *Australian Bureau of Statistics Working paper*. No. **93**/2.
- [37] Talagala TS, Hyndman RJ, Athanasopoulos G (2018). Meta-learning how to forecast time series. *Monash Econometrics and Business Statistics Working Papers*, **6**/18, Monash University.
- [38] Taylor JW (2003). Exponential Smoothing with a Damped Multiplicative Trend. *International Journal of Forecasting*, **19**, 715-725.
- [39] Time Series Research Staff (2017). *X-13ARIMA-SEATS Reference Manual*. Center for Statistical Research and Methodology. U.S. Census Bureau. <https://www.census.gov/srd/www/x13as/>
- [40] Winters PR (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, **6**, 324-342.
- [41] Woodward WA, Gray HL, Elliott AC (2017). *Applied time series analysis, with R*. Second edition. Boca Raton: Taylor & Francis.