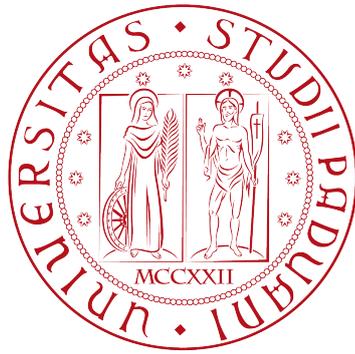


Università degli Studi di Padova

Dipartimento di Scienze Statistiche

Corso di Laurea Triennale in
Statistica per l'Economia e l'Impresa



Relazione finale

**Procedure automatiche di identificazione
di serie storiche e performance previsive**

Relatore: Prof. Tommaso Di Fonzo
Dipartimento di Scienze Statistiche

Laureanda: Contisciani Martina
Matricola N. 1101605

Anno Accademico 2016/2017

Indice

Introduzione	1
1 Procedure automatiche di previsione	3
1.1 Metodi di analisi delle previsioni	3
1.1.1 Indici di accuratezza delle previsioni	5
2 Software dedicati	9
2.1 Forecast	9
2.1.1 Modelli ARIMA	9
2.2 JDemetra+	12
2.2.1 Seasonal Adjustment	14
2.2.2 TRAMO-SEATS	15
2.3 Commenti	19
3 Applicazione	21
3.1 Le M-Competitions	21
3.1.1 Storia degli studi sull'accuratezza e le competizioni	22
3.1.2 M3-Competition	23
3.2 Esperimento e risultati	24
3.2.1 Identificazione automatica	25
3.2.2 Performance previsive	28
4 Conclusioni	33
Bibliografia	35

Introduzione

L'analisi statistica di una serie storica, cioè di un insieme di dati numerici ordinati in maniera sequenziale nel tempo, si propone di chiarire il meccanismo causale che l'ha generata, o per dare una descrizione succinta delle caratteristiche della serie, oppure per prevedere l'evoluzione del fenomeno osservato, di cui è nota la storia passata. In altre parole, tutta l'analisi si traduce nel problema della costruzione di un modello che approssimi adeguatamente il processo generatore dei dati.

Una procedura adatta a questo scopo è stata proposta da G.Box e G.Jenkins (1976) e consta di tre fasi fondamentali: identificazione, stima dei parametri, controllo diagnostico. Si tratta di un approccio artigianale, rivelatosi particolarmente efficace alla luce dei risultati ottenuti nel corso del tempo, che procede passo-passo con approfondimenti, modifiche, analisi grafiche, ripensamenti e calcoli, talvolta faticosi, da fare e rifare con scarsa interazione. Ne deriva che la procedura Box-Jenkins (BJ), effettuata con questo iter, è spesso lenta e dispersiva, nonché soggetta a molte scelte soggettive col rischio di cadere nell'arbitrarietà, e richiede importanti competenze specifiche nell'ambito delle serie storiche.

È chiaro, quindi, che questa non sia la via più appropriata quando si ha l'esigenza di trattare tantissime serie storiche relative a fenomeni distinti, in tempi ristretti e -come spesso accade in contesti quali gli istituti produttori di statistiche ufficiali- con professionalità solo parzialmente disponibili.

Ecco allora che entrano in gioco le procedure automatiche, in cui il ruolo dell'utente è ridotto in termini di impegno 'esecutivo', ma è rilevante per gli aspetti di indirizzo e controllo. Queste si implementano utilizzando opportuni software, i quali sottopongono a stima e verifica automatica moltissimi modelli alternativi per ciascuna serie storica. Tra questi si chiede al software di evidenziarne solo alcuni, avendone il tempo, per una analisi di dettaglio. Se serve, si può procedere a selezionare il migliore, impiegandolo quindi nelle previsioni. A questo approccio si è arrivati grazie all'aumento delle attività nell'ambito delle serie

storiche, che non solo ha velocizzato le tecniche esistenti ed agevolato il lavoro di chi già operava in questo campo, ma ha ampliato l'opportunità di usare la procedura BJ anche a molte altre discipline.

Scopo di questo lavoro è analizzare e confrontare due importanti strumenti di *automatic forecasting*: il software statistico open source R (pacchetto **forecast**), e lo strumento utilizzato dalla procedura di destagionalizzazione per le serie storiche usato da Eurostat (**JDemetra+**). Sebbene quest'ultimo nasca per la destagionalizzazione, è noto che tali tecniche devono passare attraverso la fase della previsione, che ne costituisce uno step imprescindibile e la cui qualità è condizione essenziale per la qualità della destagionalizzazione. È, quindi, del tutto ragionevole far riferimento a quest'ultimo come strumento di forecasting, scegliendolo anche come termine di paragone.

Per confrontare le due procedure, ovvero per capire quale procedura fornisca previsioni migliori, si è deciso di usare un esperimento di **one-step-ahead rolling forecast**. Una volta ottenute le previsioni a un passo e calcolati gli errori di previsione, abbiamo confrontato gli indici di accuratezza **MASE** e tratto una serie di conclusioni.

L'elaborato si articola in tre capitoli, il cui contenuto è schematizzato qui di seguito:

- Il primo capitolo descrive brevemente le procedure automatiche di previsione presentando anche la cross-validation e l'esperimento di rolling forecast. Il capitolo introduce, inoltre, le principali misure di accuratezza impiegate per l'analisi delle previsioni, soffermando l'attenzione sul MAE, RMSE, MAPE e il MASE e giustificando la scelta di usare il MASE.
- Il secondo capitolo presenta i software utilizzati, quindi R con il pacchetto **forecast** e **JDemetra+**, e i principali algoritmi di identificazione automatica da loro utilizzati.
- Il terzo e ultimo capitolo è dedicato alla descrizione dell'applicazione, svolta sulle serie della M3-Competition.

Capitolo 1

Procedure automatiche di previsione

Le previsioni automatiche di un gran numero di serie storiche univariate sono spesso necessari in vari ambiti, tra cui quello aziendale. È, infatti, molto comune avere molte (anche più di mille) linee di produzione che necessitano di previsioni almeno una volta al mese. Inoltre, anche quando è richiesto un numero contenuto di previsioni, c'è bisogno di personale dotato di adeguate competenze nell'uso dei modelli per serie storiche per produrre tali previsioni. In queste circostanze, una procedura automatica di previsione diventa uno strumento essenziale.

Gli algoritmi automatici di previsione hanno il compito di specificare un appropriato modello, di stimare adeguatamente i parametri e di calcolare le previsioni. Tali procedure devono quindi essere robuste rispetto a possibili anomalie presenti nei dati e applicabili ad un largo numero di serie storiche senza richiedere l'intervento dell'utente.

1.1 Metodi di analisi delle previsioni

Ci sono diversi modi per misurare l'accuratezza di una previsione, e questi dipendono da quando la previsione è stata fatta, quale indice di accuratezza è stato usato, e con quale data set è stata calcolata tale misura.

Per prima cosa bisogna considerare che non basta guardare quanto un modello si adatta bene ai dati passati; l'accuratezza delle previsioni può essere

determinata solamente considerando quanto un modello è adeguato per i nuovi dati, i quali non devono essere usati nella fase di stima.

Quando si va alla ricerca di un modello, è comune usare una porzione dei dati per identificare e stimare (*training set*) e usare il resto per verificare (*test set*). Quindi il *test set* verrà utilizzato per misurare quanto un modello si adatta ai dati futuri. Generalmente questo è costituito dal 20% del totale delle osservazioni.

Per serie storiche aventi un numero limitato di osservazioni, si cerca di evitare una divisione netta in training e test set. Inoltre, se il test set è piccolo, le conclusioni dedotte dagli indici di accuratezza potrebbero essere non affidabili. Una soluzione a questi problemi è quella di usare la ‘*cross-validation*’.

Con questo approccio si vanno a considerare differenti training set, ognuno di volta in volta allungato di una osservazione rispetto al precedente. Le misure di accuratezza per le previsioni sono calcolate su ogni test set e successivamente viene fatta una media dei risultati (aggiustando per le differenti dimensioni).

Una variante di questo approccio si focalizza su un singolo orizzonte di previsione per ogni test set ($h = 1$). La Figura 1.1 mostra una serie di test set contenenti solo una osservazione. Il calcolo delle misure di accuratezza è relativo alle previsioni a *un passo* piuttosto che il risultato della media per più orizzonti temporali. Questa procedura è nota anche come *one-step-ahead rolling forecast*.

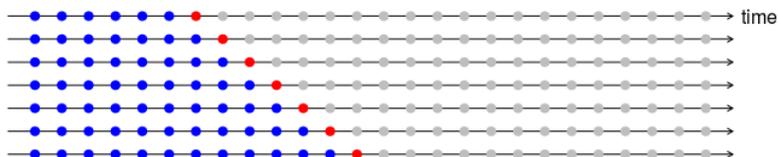


Figura 1.1: One-step-ahead rolling forecast. I punti blu sono i training set, i punti rossi sono i test set e i punti grigi sono ignorati.

Supponendo che k sia il numero di osservazioni usato nel training set, un esperimento di one-step-ahead rolling forecast è così strutturato:

1. Si seleziona l'osservazione al tempo $k + 1$ per il test set, e si usano le osservazioni ai tempi $1, 2, \dots, k$ per stimare il modello e per calcolare la previsione relativa al tempo $k + 1$.
Si calcola l'errore di previsione per il tempo $k + 1$.
2. Si ripetono i passi precedenti incrementando di volta in volta k di una unità, fino ad arrivare a $k = T - 1$, dove T è il numero totale di osservazioni.
3. Si calcolano le misure di accuratezza basandosi sugli errori ottenuti.

1.1.1 Indici di accuratezza delle previsioni

Dividiamo la serie storica osservata y_1, \dots, y_T in training set (y_1, \dots, y_N) e test set (y_{N+1}, \dots, y_T) . Per controllare l'accuratezza delle previsioni fornite dal modello prescelto, si stimeranno i parametri usando il training set, e verranno predette le $T - N$ osservazioni. Queste saranno poi confrontate con il test set.

Le previsioni h passi in avanti, possono essere scritte $\hat{y}_{N+h|N}$; dove la notazione con il cappello indica proprio la stima rispetto al valore osservato e i valori a pedice stanno ad indicare che la stima al tempo $N + h$ è stata calcolata usando tutte le osservazioni fino al tempo N .

Gli errori di previsione sono dati dalla differenza tra i valori osservati del test set e le previsioni calcolate usando le osservazioni del training set

$$e_t = y_t - \hat{y}_{t|N}, \text{ per } t = N + 1, \dots, T.$$

Misure di accuratezza dipendenti dall'unità di misura

Queste misure si basano direttamente su e_t quindi dipendono dall'unità di misura dei dati della serie storica; pertanto, non possono essere usate se si vuole paragonare metodi previsionali su serie storiche misurate su scala differente.

Le due misure in questo senso più comuni sono:

$$\text{Mean Absolute Error: MAE} = \text{mean}(|e_t|) = \frac{1}{T - N} \sum_{t=N+1}^T |e_t|$$

$$\text{Root Mean Squared Error: RMSE} = \sqrt{\text{mean}(e_t^2)} = \sqrt{\frac{1}{T - N} \sum_{t=N+1}^T e_t^2}.$$

Quando si confrontano metodi di previsioni su un singolo data set, il MAE è più comune, in quanto più facile da interpretare e da calcolare.

Misure di accuratezza basate sull'errore percentuale

L'errore percentuale è definito come $p_t = 100e_t/y_t$. Gli errori percentuali hanno il vantaggio di non dipendere dall'unità di misura, e per questo sono usati molto frequentemente per confrontare le performance previsive su dataset differenti. La misura più comune è:

$$\text{Mean Absolute Percentage Error: MAPE} = \text{mean}(|p_t|) = \frac{1}{T - N} \sum_{t=N+1}^T |p_t|$$

Le misure basate sugli errori percentuali hanno lo svantaggio di essere infinite o indefinite se $y_t = 0$ per qualche osservazione del test set, o di avere valori estremi per valori di y_t vicini allo zero.

Misure di accuratezza basate sull'errore scalato

Gli errori scalati furono proposti da Hyndman e Koehler (2006) come un'alternativa all'uso degli errori percentuali per confrontare l'accuratezza delle previsioni tra serie con unità di misura differenti.

Un errore scalato è dato da $q_t = e_t/Q$, dove Q è una statistica scalata calcolata sul training set. Il valore di Q è calcolato usando il training set perché è importante per fornire una misura stabile dell'unità di misura dei dati. Il training set è in genere molto più ampio del test set, e questo permette una migliore stima della statistica scalata.

Per serie storiche non stagionali, un comodo modo per definire tale statistica è calcolarla come la media delle differenze assolute tra osservazioni successive:

$$Q = \frac{1}{N-1} \sum_{j=2}^N |y_j - y_{j-1}|.$$

Per serie storiche stagionali, invece, la statistica scalata può essere così definita:

$$Q = \frac{1}{N-m} \sum_{j=m+1}^N |y_j - y_{j-m}|.$$

Allora,

$$\begin{aligned} & \text{Mean Absolute Scaled Error: MASE} \\ & = \text{mean}(|q_j|) = \text{MAE}/Q = \frac{\frac{1}{T-N} \sum_{t=N+1}^T |e_t|}{\frac{1}{N-m} \sum_{j=m+1}^N |y_j - y_{j-m}|}. \end{aligned}$$

Dal momento che il numeratore e il denominatore operano su valori la cui unità di misura è quella dei dati originali, q_t è indipendente dalla scala dei dati. Un $\text{MASE} < 1$ indica che la previsione è migliore di quella ottenuta con il metodo naïve ($\hat{y}_t = y_{t-1}$). Al contrario, un valore maggiore di 1 indica che il metodo proposto fornisce, in media, errori che sono maggiori degli errori a un passo ottenuti con il metodo naïve.

Hyndman e Koehler (2006) ritengono questo indice il migliore tra quelli esistenti: è il più affidabile, il più versatile e costruito in maniera tale da non soffrire di

problemi degenerativi. Infatti, il MASE è infinito o indefinito solo nel caso in cui tutte le osservazioni del passato sono uguali tra loro. Un altro punto di forza di questa misura di accuratezza, è l'essere poco sensibile agli outlier e godere di una facile interpretazione.

Capitolo 2

Software dedicati

In questo capitolo vengono presentate le due procedure utilizzate nel lavoro. Da un parte troviamo **forecast**, un pacchetto applicativo di R, dall'altra **JDemetra+**, un software di cui è stata utilizzata la versione 2.1.0.

2.1 Forecast

Con riferimento all'articolo *Automatic Time Series Forecasting: The forecast Package for R* di Hyndman e Khandakar (2008), in questo capitolo viene presentata l'implementazione di un metodo automatico di previsione per serie univariate presente nel pacchetto **forecast** di R. Questo contiene al suo interno altri pacchetti, tra cui **fma**, **expsmooth** e **Mcomp** che contiene le 1001 serie della M-Competition e le 3003 serie della M3-Competition (si veda il paragrafo 3.1).

Il pacchetto **forecast** implementa previsioni automatiche usando il liscio esponenziale, i modelli ARIMA, il metodo Theta, cubic splines e altri metodi di previsione di uso comune. In questo elaborato ci si limita a considerare i modelli ARIMA per generare previsioni in modalità automatica.

2.1.1 Modelli ARIMA

Un elemento di complessità nell'uso di modelli *Autoregressive Integrated Moving Average* (ARIMA) per le previsioni, è che il processo di selezione degli ordini del modello (*identificazione*) è usualmente considerato soggettivo e difficile da applicare. Per la verità, ciò non è vero, e nel corso del tempo ci sono stati

parecchi tentativi per automatizzare la modellazione ARIMA. Tra tutti vanno ricordati:

- Hannan e Rissanen (1982) che hanno proposto un metodo per identificare gli ordini di un modello ARIMA per serie stazionarie.
- Gómez (1998) che ha esteso il metodo precedente per includere l'identificazione di un modello ARIMA per serie stagionali; Gómez e Maravall implementarono questa procedura nel software **TRAMO** e **SEATS** (paragrafo 2.2.2).
- Liu (1989) che ha proposto una procedura per l'identificazione di modelli ARIMA stagionali usando un metodo di filtraggio e particolari regole euristiche. Tale procedura è usata nel software **SCA-Expert**.
- Mélard e Pasteels (2000), la cui procedura per modelli ARIMA univariati è implementata nel software **TSE-AX**.
- Altre procedure di identificazione automatica sono **Forecast Pro** (Goedrich 2000) e **Autobox** (Reilly 2000).

Scelta degli ordini del modello ARIMA

Il principale obiettivo di un algoritmo automatico di previsione è la selezione di un appropriato modello, cioè l'identificazione degli ordini p, q, P, Q, d, D . Se d e D sono noti, si possono selezionare gli ordini p, q, P e Q attraverso il criterio di informazione AIC:

$$AIC = -2\log(L) + 2(p + q + P + Q + k)$$

dove $k = 1$ se nel modello è presente la costante e 0 altrimenti, e L è il valore della funzione di verosimiglianza del modello stimato sulla serie differenziata.

Un aspetto da tenere in considerazione riguarda l'over-differenziazione, che porterebbe a previsioni di scarsa qualità e ad intervalli di previsione più ampi. Per ovviare a questo problema, nelle procedure automatiche gli ordini d e D vengono tipicamente identificati usando test per radici unitarie piuttosto che l'AIC.

Per serie non stagionali, si considerano modelli ARIMA(p, d, q) dove d è selezionato basandoci su successivi test KPSS (Kwiatkowski et al. 1992).

Per serie stagionali, noi consideriamo modelli ARIMA(p, d, q)(P, D, Q) $_m$ dove m è la frequenza stagionale e $D = 0$ o $D = 1$ a seconda del risultato del test Canova-Hansen (Canova and Hansen 1995). Dopo aver selezionato D (numero

di differenziazioni stagionali), si sceglie d applicando successivi test per radici unitarie KPSS alla serie differenziata ($D = 1$) o alla serie originale ($D = 0$).

Una volta che d (e eventualmente D) sono selezionati, si procede con l'identificazione dei valori di p, q, P e Q minimizzando l'AIC. Si ammette una costante diversa da zero per quei modelli in cui $d + D < 2$.

Un altro criterio di informazione in base al quale si potrebbero scegliere gli ordini p, q, P, Q , è il criterio di Schwarz (Bayesian Information Criterion, BIC)

$$BIC = -2\log(L) + (p + q + P + Q + k)\log(n).$$

Il criterio del minimo BIC è consistente e impone un termine di penalità che, all'aumentare del numero di osservazioni n , decresce più lentamente del termine di penalizzazione usato da AIC. Per questo motivo, BIC conduce di solito all'identificazione di un modello più parsimonioso quanto a numero di parametri da stimare. Tuttavia, per n non elevato, la penalizzazione è eccessiva e BIC tende a selezionare un modello leggermente sottoriparametrizzato.

Procedura step-wise

Supponiamo di avere una serie stagionale e consideriamo un modello ARIMA $(p, d, q)(P, D, Q)_m$ dove p e q possono avere valori tra 0 e 3, e P e Q valori tra 0 e 1. In assenza della costante ($c = 0$) ci sono 288 modelli possibili, e quando $c \neq 0$ ce ne sono altri 192, per un totale di 490 modelli. Ovviamente, se i valori dei coefficienti variano in un range maggiore, il numero dei possibili modelli cresce rapidamente. Pertanto, è spesso non fattibile esaminare tutti i possibili modelli e scegliere così quello con il minore AIC (o qualsiasi altro criterio di informazione). Quindi, si ha bisogno di un modo che attraversi efficientemente lo spazio dei modelli possibili per arrivare a quello con un basso AIC, evitando però di testarli tutti.

Hyndman e Khandakar (2008) hanno proposto un algoritmo step-wise che, a parte in situazioni particolari dettagliate dagli autori, identifica un modello valido in quanto lo spazio dei possibili modelli è finito e almeno uno dei modelli iniziali sarà accettato. Il modello così selezionato sarà poi usato per produrre previsioni.

Implementazione dell'algoritmo automatico ARIMA

Gli algoritmi per la previsione automatica di serie storiche univariate sono implementati nel pacchetto **forecast** di R. L'algoritmo step-wise sopra citato è

implementato dalla funzione `auto.arima()` e restituisce un oggetto della classe `Arima`. Per maggiori dettagli si rimanda al ‘Package forecast’¹.

2.2 JDemetra+

JDemetra+ (JD+) è un programma open source orientato alla soluzione di problemi di Seasonal Adjustment (SA) e adatto a trattare altri problemi relativi alle serie storiche (benchmarking e riconciliazione di sistemi di serie storiche), sviluppato dalla Banca nazionale del Belgio, con il sostegno finanziario di Eurostat. JD+ è di fatto una reingegnerizzazione in Java delle versioni più recenti della procedura TRAMO-SEATS e della procedura X12-ARIMA/X13-ARIMA-SEATS, e consente l’implementazione delle indicazioni presenti in ‘*ESS Guidelines on Seasonal Adjustment*’² fornite da Eurostat nel 2015.

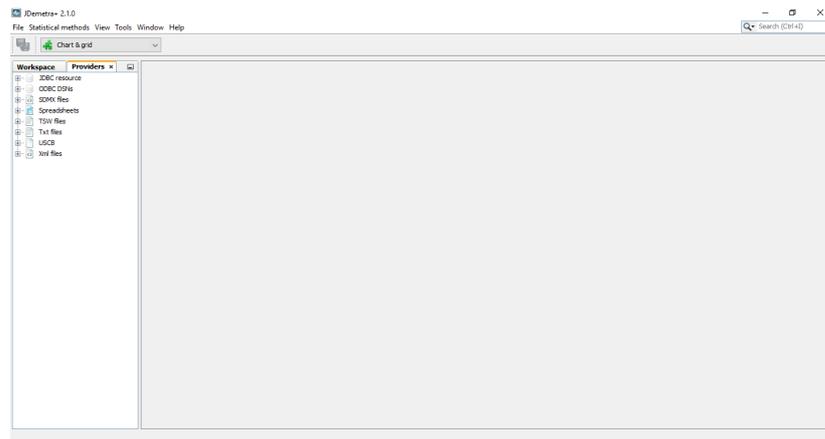


Figura 2.1: Schermata iniziale di JDemetra+ versione 2.1.0

Il programma JD+ comprende un’avanzata interfaccia grafica per l’uso delle due procedure di destagionalizzazione, ed include un insieme aggiuntivo di librerie che possono essere utilizzate per effettuare specifiche analisi (come la destagionalizzazione di dataset di grandi dimensioni, il benchmarking e la disaggregazione temporale di serie storiche e l’analisi del ciclo economico). Le sue funzionalità, inoltre, possono essere estese mediante lo sviluppo di plug-in aggiuntivi, richiamabili dalle procedure di destagionalizzazione. Infine, JD+ fornisce strumenti di analisi e output standardizzati per i diversi approcci alla destagionalizzazione, consentendo un confronto più agevole dei risultati ad essi associati. Tra le nu-

¹<https://cran.r-project.org/web/packages/forecast/forecast.pdf>

²<http://ec.europa.eu/eurostat/documents/3859598/6830795/KS-GQ-15-001-EN-N.pdf>

merose e importanti funzionalità implementate in JDemetra+, vanno evidenziati i seguenti strumenti³:

- RegARIMA modelling (che usano concetti sviluppati in TRAMO e in X-12-ARIMA);
- Residuals analysis (principalmente TRAMO-like);
- Seasonality tests (TRAMO e X-12-ARIMA-like);
- Spectral analysis (secondo l'implementazione di X-12-ARIMA);
- Sliding spans (X-12-ARIMA);
- Analisi di Wiener-Kolmogorov (per modelli ARIMA a componenti non osservate, SEATS-like).

Accanto alle procedure statistiche, JD+ fornisce anche una serie di servizi 'collaterali'. I più importanti sono i seguenti:

- *Dynamic access to various 'time series providers'*: JDemetra+ offre diversi moduli per maneggiare serie storiche salvate in formati differenti: Excel, XML, TXT, database (JDBC), ...
- *Common XML formatting*: il processo di seasonal adjustment può essere salvato in un file XML, che può essere utilizzato anche per creare servizi WEB.

Riassumendo, le principali caratteristiche e punti di forza di JD+ risultano essere:

- **Flessibilità**
 - Include i principali algoritmi per il SA
 - Potrebbe evolversi indipendentemente se dovessero rendersi disponibili nuovi metodi o miglioramenti di quelli esistenti
- **Versatilità**
 - Interfaccia grafica intuitiva e 'ricca'
 - Può essere integrato in altri ambienti di calcolo/gestione informatizzati
- **Riusabilità dei moduli**
 - JEcotrim (per benchmarking e disaggregazioni temporali)

³https://ec.europa.eu/eurostat/cros/content/documentation_en.

- Uso di altre librerie di JDemetra+ per l'individuazione degli outlier
- Stima dei valori mancanti, delle previsioni ARIMA, degli effetti di calendario
- **Estensibilità**
 - Plug-in e moduli aggiuntivi non cambiano il core engine
- **Risultati efficienti su grandi dataset**
 - JWSAcruncher, sistema per il trattamento batch di un gran numero di serie storiche
 - servizi Web
 - chiamata diretta alle librerie Java
- **Open source**
 - Contribuisce ad aumentare la trasparenza del calcolo statistico e a condividere maggiormente le conoscenze statistiche
 - Dà la possibilità a diversi gruppi di sviluppatori di usare il software e contribuire così al suo miglioramento
 - Plug-in sviluppati dagli esperti del SA possono essere condivisi tra tutti i membri della comunità

2.2.1 Seasonal Adjustment

Il *Seasonal Adjustment* (processo di destagionalizzazione) è una fase importante del processo di trattamento statistico delle serie storiche. Si tratta di una tecnica utilizzata per stimare e rimuovere i movimenti stagionali e i relativi effetti di calendario da serie storiche infrannuali, che rappresentano molto spesso fenomeni di natura economica. Per queste ragioni, Eurostat ha preso parte in varie attività aventi lo scopo di promuovere, sviluppare e mantenere un software accessibile a tutti.

Il primo obiettivo del processo di destagionalizzazione è quello di rimuovere le fluttuazioni stagionali dalla serie storica. Per raggiungere questo, i metodi di destagionalizzazione decompongono la serie originale nelle componenti che descrivono gli specifici movimenti, ossia: trend-ciclo, stagionalità, errore.

La *stagionalità*, nella dinamica di una serie storica, è quella componente che si ripete ad intervalli regolari ogni anno, con variazioni di intensità più o meno analoga nello stesso periodo (mese, trimestre, semestre, etc.) di anni successivi e di intensità diversa nel corso di uno stesso anno. Nell'ultimo ventennio la

richiesta di dati più agevolmente interpretabili e comparabili, in particolare di indicatori congiunturali presentati in forma destagionalizzata (cioè al netto di fluttuazioni stagionali), è andata crescendo. In questo quadro, numerosi sono stati gli sforzi compiuti a livello internazionale ed europeo per giungere ad una maggiore armonizzazione delle metodologie e delle pratiche impiegate per la destagionalizzazione. In particolare, Eurostat e la Banca centrale europea hanno contribuito a produrre le linee guida sulla destagionalizzazione (Eurostat, 2015), allo scopo di promuovere, nell'ambito del Sistema statistico europeo (SSE), le migliori pratiche sulla destagionalizzazione e sulla correzione per gli effetti di calendario.

In entrambe le metodologie, la fase vera e propria di destagionalizzazione, cioè di stima e di conseguente rimozione della componente stagionale, è preceduta da una fase di pre-trattamento dei dati, sostanzialmente comune ai due approcci, in cui avviene la scelta dello schema di scomposizione che lega le diverse componenti della serie storica (additiva, moltiplicativa, log-additiva, ecc.) e sono identificati ed eliminati una serie di effetti, quali i valori anomali (outlier) e quelli legati agli effetti di calendario. È su questa serie corretta preliminarmente che viene condotta la fase successiva di destagionalizzazione.

2.2.2 TRAMO-SEATS

TRAMO-SEATS (*Time series Regression with Arima noise, Missing observations and Outliers e Signal Extraction in Arima Time Series - TS*), sviluppato da Gómez e Maravall (1996), è un metodo di destagionalizzazione che segue un approccio AMB (*Arima Model Based*), cioè si basa sulla costruzione di un particolare modello statistico per ciascuna serie storica analizzata.

La procedura TS si compone di due parti integrate. Con la prima, TRAMO, si sceglie il tipo di scomposizione, si interpolano eventuali dati mancanti e si eliminano i cosiddetti effetti deterministici (effetti di calendario e valori anomali); inoltre si procede all'identificazione e alla stima di un idoneo modello ARIMA per la serie depurata dagli effetti deterministici, cioè per la parte stocastica della serie originaria (nota anche come *serie 'linearizzata'*). La seconda parte della procedura, SEATS, effettua, sulla base di opportune ipotesi, l'individuazione della componente stagionale e la sua rimozione.

Maggiori informazioni sul metodo TRAMO/SEATS e sul relativo software possono essere trovate nella sezione dedicata presente nel sito di Banco de España⁴.

⁴www.bde.es, Services → Professionals → Statistical and Econometric Software

Nel seguito vengono descritte brevemente le funzioni principali dei due moduli TRAMO e SEATS di cui si compone la procedura TS. Dal punto di vista operativo, le fasi essenziali di TRAMO possono essere così riassunte:

1. FASE PRELIMINARE (pre-trattamento della serie storica osservata) costituita, nell'ordine, da due passi:
 - (a) SCOMPOSIZIONE. Viene valutata l'opportunità di ricorrere ad una trasformazione dei valori originari della serie, cioè viene individuato lo schema di aggregazione delle componenti non osservabili. La scelta del tipo di scomposizione avviene tra due tipologie: additiva o log-additiva (relazione moltiplicativa tra le componenti).
 - (b) EFFETTI DETERMINISTICI. Vengono individuati e rimossi eventuali effetti deterministici la cui presenza può influire negativamente sulla corretta identificazione della componente stagionale nella parte successiva. Tra questi, i principali sono rappresentati da *effetti di calendario*, *outlier* (additivi, temporanei o permanenti) e *altri effetti deterministici* la cui rimozione può favorire una più corretta identificazione e stima del modello ARIMA. La presenza di *outlier*, ossia di osservazioni anomale nella serie storica causate da eventi straordinari, può determinare significative distorsioni nell'identificazione dell'ordine del modello reg-ARIMA e/o nella stima dei suoi coefficienti.

Alla fine di questa prima fase si avrà la serie linearizzata Z_t ottenuta usando modelli di regressione con errori ARIMA.

2. IDENTIFICAZIONE del modello $ARIMA(p, d, q)(P, D, Q)_m$ (che in JD+ viene indicato come $ARIMA(P, D, Q)(BP, BD, BQ)$) per la componente stocastica Z_t , che consente di specificarne gli ordini dei polinomi AR e MA per la parte non stagionale (p e q) e per la parte stagionale (P e Q) e l'eventuale rispettivo grado di integrazione (d e D) del modello ARIMA.

Nei metodi AMB, come TS, è fondamentale che il modello stimato ARIMA $(p, d, q)(P, D, Q)_m$ per la serie Z_t , sulla quale verrà successivamente stimata la componente stagionale, sia correttamente specificato, in quanto questo concorre direttamente alla stima della componente stagionale della serie. Le principali procedure disponibili per la destagionalizzazione sono dotate di routine per l'identificazione automatica, sia dei principali effetti deterministici che del modello ARIMA. Questa procedura è basata sul confronto di vari modelli utilizzando criteri standard sviluppati in letteratura (in particolare, valutando le statistiche espresse dagli indici BIC e

AIC). Tale funzionalità è molto utile soprattutto nei casi in cui si debba trattare un elevato numero di serie storiche, ma è consigliabile analizzarne attentamente i risultati e intervenire se necessario per modificarli (si veda la sezione 2.2.2).

3. STIMA del modello ARIMA identificato, utilizzando il metodo della massima verosimiglianza.
4. VALIDAZIONE DEL MODELLO, basata sull'elaborazione di test statistici per verificare se il modello individuato sia una buona rappresentazione del processo stocastico che ha generato la serie osservata. Le tre fasi (identificazione, stima e controllo diagnostico) di cui consta la costruzione di un modello ARIMA (Box e Jenkins, 1976), possono essere iterate finché non si perviene al modello ritenuto migliore.

Le serie così trattate vengono, quindi, sottoposte a SEATS in cui, in estrema sintesi, vengono compiute le seguenti operazioni:

1. VERIFICA del modello individuato da TRAMO ed eventuale nuova identificazione e stima del modello ARIMA.
2. IDENTIFICAZIONE dei modelli ARIMA per le componenti della serie linearizzata Z_t , coerenti con quello seguito dalla serie Z_t .
3. STIMA delle componenti e ANALISI DEGLI ERRORI

Identificati i modelli ARIMA per ciascuna componente, si calcolano le relative stime delle componenti applicando un opportuno filtro lineare alla serie Z_t . SEATS utilizza, in particolare, il filtro di Wiener-Kolmogorov, che è infinito, simmetrico e convergente. Sottolineiamo ancora come, nell'approccio TS, la scelta del filtro ottimale si riconduca all'identificazione ottimale del modello ARIMA. Ciò conferma che la corretta specificazione, operata nella fase di pre-trattamento, è un'operazione fondamentale, in quanto condiziona in maniera determinante le fasi successive.

4. STIMA FINALE che prevede l'allocazione, tra le varie componenti, delle parti deterministiche individuate da TRAMO.

SEATS offre l'opportunità di valutare i risultati raggiunti sia attraverso una diagnostica molto ampia, sia tramite un'analisi grafica delle componenti non osservabili.

Procedura di identificazione automatica del modello in TRAMO

L'algoritmo per l'identificazione automatica del modello, implementato in TRAMO, è contraddistinto dall'acronimo AMI (*Automatic Model Identification in*

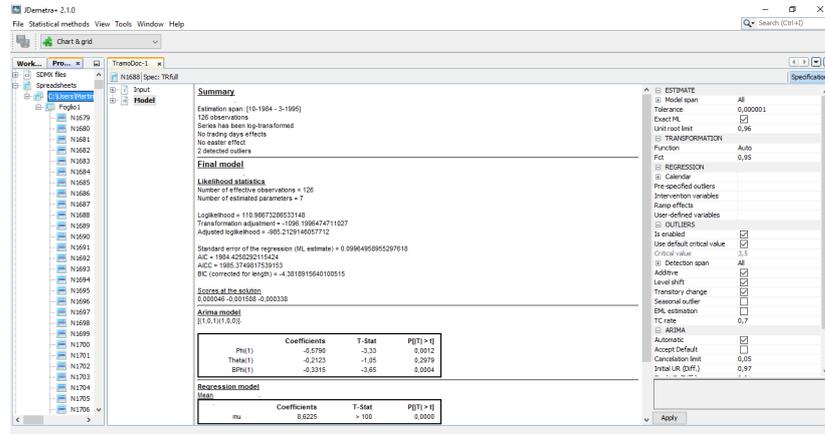


Figura 2.2: Identificazione automatica di un modello in JDemetra+

the Presence of Outliers). Questo iterata i due step: individua e corregge gli outlier e identifica automaticamente il modello. L'algoritmo parte identificando il modello da usare come default e inizia una procedura di test, dove al primo passo viene esaminata la trasformazione logaritmica. Questo test si basa sulla stima della massima verosimiglianza del parametro λ , che nella trasformazione di Box-Cox 'governa' la trasformazione logaritmica. Per prima cosa, poi, due modelli Airline (ARIMA(0, 1, 1)(0, 1, 1)) con costante vengono adattati alle serie storiche: uno alla serie trasformata ($\lambda = 0$) e uno alla serie non trasformata ($\lambda = 1$). Successivamente, viene eseguito il test per gli effetti di calendario con i regressori usando il modello di default per l'errore e , se il modello è sostanzialmente cambiato, il test viene eseguito nuovamente. Per le serie stagionali il modello di default è un Airline (ARIMA(0, 1, 1)(0, 1, 1)) mentre per le serie non stagionali un ARIMA(0, 1, 1)(0, 0, 0) con costante. Il modello di default, che verrà usato anche come benchmark nei prossimi step, è il risultato di pre-test fatti per la possibile presenza di stagionalità.

Una volta eseguiti i pre-test, la procedura provvede a identificare gli ordini delle differenziazioni e valuta se è opportuno considerare la media oppure no. L'identificazione del modello ARMA, cioè degli ordini di $\phi(B)$ e $\theta(B)$, è eseguita con il metodo Hannan-Rissanen⁵ minimizzando il criterio di informazione Bayesiano (BIC) imponendo qualche vincolo con lo scopo di aumentare la parsimonia e favorendo modelli bilanciati.

Questa procedura produce i valori iniziali per i parametri di un modello ARIMA. Inoltre quando le radici stimate dei processi AR e MA sono vicine tra loro,

⁵L'algoritmo di Hannan-Rissanen è un metodo di penalizzazione basato sul criterio d'informazione BIC. Poiché tale procedura opera su una serie stazionaria, devono essere identificati solo i parametri della parte AR e MA.

gli ordini del modello ARIMA possono essere ridotti. L'identificazione e la stima del modello è fatta usando la Massima Verosimiglianza esatta (EML) o la tecnica Unconditional Least Squares (ULS).

Se nel modello di default sono stati identificati gli effetti di calendario, questi dovranno essere inclusi nel nuovo modello ARIMA. I residui stimati del modello ARIMA modificato con le stime dei parametri fissati e la deviazione mediana assoluta della deviazione standard di questi residui sono usati nella procedura di individuazione degli outlier. Per ogni osservazione, viene calcolato il t -test per tutti i tipi di outlier considerati nella procedura automatica, seguendo la procedura proposta da Chen e Liu (1993). Dopodichè questi vengono confrontati con i valori critici determinati dalla lunghezza della serie. Se ci sono outlier, cioè osservazioni con un t -test maggiore del valore soglia, viene selezionato quello con il valore del t -test maggiore. Dopo aver corretto tale osservazione, il processo ricomincia per verificare se ci sono altri valori anomali. La procedura si ripete fino a che nessuno dei potenziali outlier ha un valore della statistica t che eccede il valore critico.

Se sono stati rilevati outlier, si effettua una regressione multipla usando il filtro di Kalman o l'algoritmo QR per evitare, per quanto possibile, effetti nascosti e correggere la distorsione delle stime ottenuta.

Se ci sono outlier i cui valori assoluti delle statistiche t sono maggiori dei valori critici, quello con valore assoluto maggiore del t -test è selezionato e l'algoritmo procede nell'iterazione con la stima dei parametri del modello ARMA. Altrimenti l'algoritmo si arresta.

I residui stimati del modello finale ARIMA sono posti a confronto per l'adeguatezza del modello con i residui prodotti dal modello bilanciato. Il modello finale identificato dalla procedura AMI deve mostrare qualche miglioramento rispetto al modello di default in queste diagnostiche dei residui; altrimenti, la procedura accetterà il modello di default.

2.3 Commenti

Come detto nei paragrafi precedenti, le due procedure automatiche si basano su 'filosofie' parecchio differenti, ciascuna coi suoi vantaggi e con le sue debolezze. A priori potremmo dire che il processo di linearizzazione presente in JDemetra+ rappresenta un punto di forza rispetto a forecast parte Arima, che appare una procedura meno flessibile, in particolare per la parte relativa a possibili trasformazioni delle serie in input. Ma d'altra parte, forecast permette all'utente di 'entrare' meglio nella fase di gestione dei parametri di input, mentre JD+ è chiaramente pensato come uno strumento di produzione, volto a sgravare l'utente

di buona parte delle scelte tipiche in un'analisi 'artigianale' di una serie storica. Quindi, obiettivo primario di questo elaborato è capire come due procedure piuttosto sofisticate, molto robuste e particolarmente utilizzate, differiscono lavorando sullo stesso dataset; e quindi qual è il contributo che l'utente, dotato di una qualche capacità di comprensione del problema, può dare nell'interpretare questi risultati.

Capitolo 3

Applicazione

Nei capitoli precedenti si è introdotto il problema e sono stati discussi gli aspetti teorici con riferimento a diverse fonti. Per concludere tale discussione, in questo capitolo viene presentato l'esperimento avente l'obiettivo di capire le peculiarità delle procedure presentate e confrontarle.

Lo studio è basato interamente sull'analisi di serie storiche appartenenti al dataset della M3-Competition, di cui qui di seguito si dà una rapida descrizione, assieme ad un'analisi dei risultati ottenuti.

3.1 Le M-Competitions

Le *M-Competitions* sono studi empirici volti a confrontare la performance di un gran numero di tecniche di analisi di serie storiche con lo scopo di comprendere quale metodo previsionale sia maggiormente performante. Esperti riconosciuti forniscono le previsioni usando i metodi che preferiscono, e una volta ottenute tutte le previsioni da ogni esperto, queste sono state valutate e confrontate con quelle degli altri e con quelle ottenute mediante alcuni semplici metodi usati come termini di paragone. L'idea sottostante a questo modo di operare è che queste competizioni basate sulle previsioni assicurino l'oggettività nel valutare le competenze degli esperti.

Le *M-Competitions* si riferiscono principalmente a serie storiche economiche, ma i risultati e le conclusioni ottenuti potrebbero essere importanti anche per altre discipline. Va inoltre sottolineato che in seguito ai risultati di questi studi, molti studiosi compresero che prima di individuare il migliore metodo previsionale,

era necessario spostare l'attenzione sulle misure di accuratezza utilizzate per valutarli.

3.1.1 Storia degli studi sull'accuratezza e le competizioni

Il primo studio in cui si cercò di comparare nella maniera più oggettiva possibile vari metodi di previsione temporale fu pubblicato nel 1979 da Makridakis e Hibon (*"Accuracy of forecasting: an empirical investigation"*). Le analisi si basarono su un campione di 111 serie storiche tratte da situazioni reali relative ad un ampio range di applicazioni.

Da questa analisi Makridakis e Hibon (1979) arrivarono ad alcune conclusioni che si scostavano molto dai risultati precedentemente ottenuti da studi simili. Primo fra tutti, provarono che metodi semplici, come il liscio esponenziale, davano spesso risultati migliori di metodi sofisticati. Inoltre l'ordine di accuratezza dei vari metodi predittivi può variare a seconda della misura di accuratezza utilizzata. Infine, per alcuni metodi predittivi l'accuratezza migliora all'aumentare delle osservazioni raccolte.

All'epoca, questi risultati vennero contestati da parecchi studiosi, che mettevano in dubbio la reale oggettività del lavoro, basato solamente sulla correttezza dei calcoli di due singoli studiosi. Per rispondere a queste critiche e accogliendo i consigli di vari commentatori, Makridakis lanciò la *M-Competition* (Makridakis et al. 1982).

Nella *M-Competition* il numero di serie utilizzate aumentò a 1001 e il numero di metodi a 15 (con incluse altre 9 varianti di questi). Inoltre vennero utilizzate cinque misure di accuratezza, due delle quali erano già state utilizzate precedentemente. La più importante innovazione fu che chiunque poteva fare i propri studi su questi dati e che studiosi in grado di proporre nuovi metodi potevano usarlo nella competizione. Ogni esperto, così, forniva le sue previsioni che venivano poi confrontate con i valori reali non utilizzati per sviluppare il modello. Questi errori di previsione venivano poi usati per calcolare le varie misure di accuratezza.

I risultati della *M-Competition* furono simili a quelli del precedente studio di Makridakis e Hibon (1979) e possono essere sintetizzati come segue:

- Metodi statisticamente sofisticati e complessi non necessariamente forniscono previsioni più accurate di metodi più semplici.
- La classifica delle performance dei vari modelli, varia a seconda della misura di accuratezza usata.
- L'accuratezza quando vari metodi vengono combinati insieme, in media, supera quella ottenuta con metodi usati singolarmente.

- L'accuratezza dei vari metodi dipende dalla lunghezza dell'orizzonte temporale di previsione usato.

La *M2-Competition* (Makridakis et al. 1993) rappresentò una sorta di 'prolungamento' della precedente iniziativa, volta a capire meglio i fattori che influenzano l'accuratezza delle previsioni. Anche questa competizione era aperta a chiunque volesse partecipare, ed è stata organizzata in collaborazione con quattro aziende. Queste avevano il compito sia di fornire i dati ai partecipanti, sia quello di rispondere a qualsiasi domanda che veniva posta loro.

Tale competizione chiedeva di analizzare 29 serie storiche e confrontare i risultati con tre misure di accuratezza. I risultati ottenuti sono praticamente identici a quelli della *M-Competition*, prima elencati.

3.1.2 M3-Competition

La *M3-Competition* (Makridakis e Hibon, 2000) è il tentativo finale degli autori per definire il problema di accuratezza dei vari metodi usati per la previsione delle serie storiche. Il principale scopo è stato quello di replicare e estendere le *M-* e *M2-Competitions*. L'estensione consisteva nell'inclusione di più metodi e più serie e la replica era indirizzata a determinare se le principali conclusioni della *M-Competition* continuavano a rimanere valide su un insieme più numeroso (3003) di serie storiche o meno. La tabella 3.1 mostra la classificazione delle 3003 serie considerando le varie aree di interesse e i differenti intervalli temporali.

Time interval between successive observations	Types of time series data						Total
	Micro	Industry	Macro	Finance	Demographic	Other	
Yearly	146	102	83	58	245	11	645
Quarterly	204	83	336	76	57		756
Monthly	474	334	312	145	111	52	1428
Other	4			29		141	174
Total	828	519	731	308	413	204	3003

Tabella 3.1: Classificazione delle 3003 serie storiche usate nella M3-Competition
Fonte: Makridakis e Hibon (2000)

Le serie utilizzate erano tutte strettamente positive e con un numero di osservazioni tale da non creare problemi previsionali. In questa competizione, i metodi utilizzati furono 24 e le misure di accuratezza tornarono ad essere cinque. I risultati ottenuti confermano le quattro conclusioni della *M-Competition* usando però un nuovo e più ampio dataset. In più, si è dimostrato, ancora una volta, che metodi semplici funzionano bene e, a volte, meglio di quelli statisticamente sofisticati. Inoltre, la *M3-Competition* ha fornito tre nuovi risultati che hanno bisogno di ulteriori conferme. Primo fra tutti, si è trovato che il metodo Theta (vincitore della competizione) offre risultati decisamente buoni, sebbene si

basi su un impianto teorico piuttosto semplice dal punto di vista statistico. In secondo luogo, anche il software Forecast Pro sembra dare buoni risultati. Infine, questa competizione ha mostrato che uno specifico metodo (*Robust-Trend*) dà risultati migliori di tutti gli altri quando si usano dati annuali. Tutto ciò porta a pensare che le risultanze offerte da simili competizioni non debbano essere considerate come definitive, ma piuttosto indicative del livello di qualità raggiungibile con le tecniche ‘statistiche dell’arte’, che può (e deve) essere rivalutato in epoche successive per tenere conto degli avanzamenti fatti registrare dalla teoria degli studi empirici e dagli strumenti di calcolo.

3.2 Esperimento e risultati

L’elaborato tratta il tema dell’identificazione automatica di un modello per una serie storica osservata e delle previsioni che ne conseguono. Lo scopo dell’applicazione è quello di mettere a confronto le due procedure di automatic forecasting precedentemente spiegate, usando come metro di paragone la performance previsiva. Il procedimento può quindi essere diviso in due parti:

1. Si procede con l’**identificazione automatica** del modello per tutte le serie storiche del dataset analizzato;
2. Si effettua un **esperimento di one-step-ahead rolling forecast** sulle serie storiche per cui le due tecniche identificano modelli differenti.

Come precedentemente accennato, per l’analisi sono state analizzate le serie storiche del dataset della *M3-Competition*, scaricabile gratuitamente all’indirizzo <https://forecasters.org/resources/time-series-data/m3-competition/>. Delle 3003 serie, però, sono state utilizzate solo quelle con una esplicita cadenza infrannuale (Monthly, Quarterly); escludendo così le serie annuali e le undated/other. Un’altra scelta effettuata, è stata quella di considerare solo le serie storiche che avessero un numero di osservazioni tale da non creare problemi in termini previsionali. Vale a dire: per le serie mensili si è considerata una numerosità minima pari a 72 osservazioni; per quelle trimestrali il numero minimo scelto è pari a 52. Procedendo in questo modo, per entrambe le tipologie di serie, le osservazioni di 10 anni venivano usate per la stima e quelle dei restanti 3 anni per la validazione. L’ultimo accorgimento adoperato, è stato quello di ‘ridatare’ le serie antecedenti al 1900, assegnando il 1 gennaio del 1900 come data d’inizio serie, per evitare problemi nella creazione dei file Excel da usare come input per JDemetra+. Così facendo è stato ottenuto un campione di 1506 serie storiche, di cui 1108 mensili e 398 trimestrali.

	Serie Mensili		Serie Trimestrali	
	Stesso Modello	Totale	Stesso Modello	Totale
Default	80	1108	79	398
Massimali degli ordini	15	1028	1	319

Tabella 3.2: Risultati del confronto tra i modelli specificati con le due procedure utilizzando le diverse opzioni.

3.2.1 Identificazione automatica

Per prima cosa si richiedeva la specificazione automatica del modello per ogni serie storica trattata, focalizzando l'attenzione sul modello ARIMA, perché questa è l'unica classe di modelli considerata da `JDemetra+`.

Per tale scopo, le due procedure utilizzano approcci differenti: il pacchetto `forecast` di R usa la funzione `auto.arima()`, mentre `JD+` sfrutta il `Multiprocessing` del `Seasonal Adjustment` per specificare gli ordini p, d, q, P, D, Q . In entrambi i casi, si è identificato automaticamente il modello *once-for-all*, vale a dire una volta per tutte sull'80% delle osservazioni. Per le serie mensili, mediamente, il training set è composto da 105 osservazioni e, nel peggiore dei casi, da 46 valori. Analogamente, per le serie trimestrali si ha un numero medio di osservazioni pari a 45, mentre nel caso peggiore il training set ne conta 42.

Per concludere questa prima fase, e quindi capire per quali serie storiche è stato identificato un modello uguale o diverso, sono stati confrontati gli ordini p, d, q, P, D, Q , nonché i valori delle due variabili 'indicatrici' ottenute in questa fase, *mean* e *seasonal*, che sono pari a 1 se, rispettivamente, il modello è dotato di costante diversa da zero e risulta stagionale, e uguali a 0 altrimenti.

Inizialmente si è deciso di identificare i vari modelli sfruttando le opzioni di default delle due procedure, così come si ritiene che operi la maggior parte degli utenti. I risultati sono mostrati nella tabella 3.2. Le serie storiche per cui le due tecniche danno lo stesso risultato vengono escluse da successive analisi in quanto, al netto dell'algoritmo di stima dei parametri che potrebbe dar luogo a valori numerici diversi, le previsioni sarebbero le stesse. Successivamente, è stato considerato che eventuali differenze potevano essere dovute ai massimali degli ordini p, d, q, P, D, Q imposti dalle due procedure come default. Quindi abbiamo posto `forecast` sullo stesso piano di `JD+`, fissando i massimali degli ordini come quelli della procedura concorrente¹. Come riportato nella tabella 3.2, questo approccio ha fornito dei risultati poco significativi e, per questo, si è deciso di non eliminare le serie storiche in questione.

¹`auto.arima(trainData, max.p=3, max.d=2, max.q=3, max.P=1, max.D=1, max.Q=1, start.p=0, start.q=0, start.P=0, start.Q=0)`

Una tra le più importanti peculiarità di JDemetra+ è la valutazione, tramite un test automatico, dell'opportunità di usare o meno la trasformata logaritmica. In `forecast` manca un'opzione automatica di questo tipo e per ovviare a questa 'debolezza strutturale' si è deciso, per quelle serie per cui JD+ suggerisce la trasformazione e per le quali non è stato identificato uno stesso modello con le opzioni di default, di stimare in `forecast` un modello sia nei livelli, sia nella trasformata logaritmica². Considerando che, in generale, JDemetra+ suggerisce di trasformare 918 serie mensili e 294 serie trimestrali e sapendo che per 73 serie mensili e 54 serie trimestrali è stato identificato uno stesso modello nella prima fase, ora sono state considerate solamente le restanti 845 mensili e 240 trimestrali. La tabella 3.3 mostra che, considerando i modelli identificati da `forecast` nella trasformata logaritmica, le due procedure forniscono lo stesso risultato per altre 27 serie mensili e 17 trimestrali. Queste ultime 44 serie storiche, non verranno sottoposte all'esperimento di rolling forecast.

	Serie Mensili		Serie Trimestrali	
	Stesso Modello	Totale	Stesso Modello	Totale
Default	73	918	54	294
$\lambda = 0$	27	845	17	240

Tabella 3.3: Risultati del confronto tra i modelli specificati con le due procedure per le serie storiche che secondo JD+ hanno bisogno della trasformazione logaritmica.

Nei commenti conclusivi del Capitolo 2, si è detto che l'elemento principale che contraddistingue le due procedure è il processo di linearizzazione inglobato in TRAMO, che vuol dire valutare l'utilità della trasformata logaritmica e individuare eventuali outlier. Ciò induce a pensare che tale peculiarità di JDemetra+ sia uno dei motivi per cui le due procedure automatiche di identificazione danno risultati diversi, la cui qualità verrà valutata in chiave di performance previsiva, appunto tramite un esperimento di rolling forecast.

Nel passo precedente, abbiamo provato a porre rimedio a questa debolezza di `forecast` inserendo un elemento semi-automatico quale $\lambda = 0$, estendendo a `forecast` la scelta della trasformata logaritmica fatta da JD+. Ma, in realtà, c'è un elemento che non consente di confrontare totalmente i due strumenti, perché `forecast` non ha al suo interno nessuna procedura di identificazione automatica degli outlier.

In questa applicazione abbiamo tralasciato tale aspetto, sia perché è probabile che gran parte degli utenti che utilizzano questo strumento non si pongono il problema di identificare gli outlier, ma anche perché questo avrebbe richiesto

²`auto.arima(trainData, lambda=0)`; λ è il parametro della trasformata di Box-Cox. Se posto uguale a zero, si effettua una trasformazione logaritmica prima che il modello venga stimato.

	Serie Mensili		Serie Trimestrali	
	Stesso Modello	Totale	Stesso Modello	Totale
Default	60	1028	42	319

Tabella 3.4: Risultati del confronto tra i modelli specificati con le due procedure utilizzando l'opzione di default sulle **serie linearizzate**.

un ulteriore approfondimento, volto a valutare l'efficacia delle procedure di linearizzazione ed il loro impatto sulla qualità delle previsioni.

Si è tuttavia provveduto ad usare `forecast` sulle serie linearizzate da `JDemetra+`, nella speranza che il numero di serie con lo stesso modello aumentasse. I risultati riportati nella tabella 3.4 sembrano confermare questa ipotesi.

Pur tenendo queste serie nell'esperimento, possiamo dire che, se `forecast` fosse dotato di una procedura di individuazione degli outlier e di altre non-linearità, è ragionevole ritenere che otterrebbe gli stessi risultati di `JDemetra+` in circa 60 serie mensili e 40 serie trimestrali. Quindi eventuali differenze nelle performance previsive per queste serie a favore di `JDemetra+` sono meno significative, perché di fatto sappiamo già a cosa sono dovute.

Il lavoro di pre-processing discusso finora, è servito ad eliminare il superfluo, ossia i casi per cui `JDemetra+` e `forecast` danno lo stesso risultato. Sommando i risultati delle tabelle 3.2 e 3.3 possiamo concludere che, con le dovute accortezze, le due procedure identificano lo stesso modello (o per la serie nei livelli o per quella trasformata coi logaritmi) per 107 serie mensili (9.66%) e 96 serie trimestrali (24.12%). Come si è già detto, queste serie storiche vengono escluse dall'esperimento di rolling forecast perché, al netto dell'algoritmo di stima dei parametri, porterebbero ad avere le stesse previsioni. Come nota finale, e per confermare quanto detto precedentemente, nella tabella 3.5 emerge che pur avendo identificato lo stesso modello, a causa delle diverse procedure di calcolo, le stime dei coefficienti differiscono tra le due procedure. Questo implica che ci siano delle discrepanze anche nelle previsioni perché, essendo la procedura di stima una procedura non lineare, essa dipende dall'algoritmo che si usa. Si precisa inoltre che il segno invertito del coefficiente AR nella tabella 3.5 si spiega col modo diverso con cui `forecast` e `JD+` formulano i polinomi AR del modello.

Serie	Modello	FORECAST				JDEMETRA+			
		p	q	P	Q	p	q	P	Q
N 2109	ARIMA(1,0,0)(0,1,1) ₁₂	0.7236			-0.7763	-0.7011			-0.7118
N 853	ARIMA(0,1,1)(0,1,1) ₄		-0.7884		-0.5894		-0.5722		-0.5365

Tabella 3.5: Stime dei coefficienti per alcune serie storiche per cui è stato identificato lo stesso modello.

3.2.2 Performance previsive

Lo scopo della seconda fase dell'applicazione è valutare, con riferimento alle serie storiche per cui è stato identificato un modello diverso, la bontà dei modelli proposti e capire quale tra le due procedure fornisca una specificazione migliore. Per giudicare i risultati ottenuti con i due strumenti di automatic forecasting abbiamo confrontato le performance previsive tramite l'indice di accuratezza MASE. L'assenza in JDemetra+ di una procedura iterativa in grado di gestire sessioni ripetute di one-step-ahead forecast, ha suggerito di applicare l'esperimento di rolling forecast solo alle serie trimestrali. Più precisamente, sono state trattate le 302 serie storiche per cui si è identificato un modello diverso nella fase precedente, con una avvertenza: per le 223 serie per cui JDemetra+ usa i logaritmi, vengono considerati entrambi i modelli proposti da forecast (uno nei livelli e uno nei logaritmi).

Data quindi la natura delle serie e simulando un atteggiamento tipico dell'utente che analizza le serie storiche, si è costruito un *esperimento di one-step-ahead rolling forecast* privilegiando così il breve periodo. In questa applicazione, l'identificazione del modello è stata fatta solo all'inizio sull'80% delle osservazioni, mentre ad ogni passo, per il restante 20%, venivano ristimati solo i coefficienti. Inoltre, con un esperimento di questo tipo e utilizzando modelli ARIMA si ha che, se il modello è stimato correttamente ed è di buona qualità, gli errori di previsione (*one-step-ahead forecast error*) non si discostano significativamente dalla realizzazione di un *White Noise* ($e_i \sim WN(0, \sigma_\epsilon)$). Quindi, come strumento preliminare per valutare l'accuratezza delle previsioni, si è utilizzato un test di casualità sui residui. Controllando i p-value del test di Ljung-Box per le 302 serie storiche trimestrali sottoposte a rolling forecast, si ha che l'87.75% delle serie dei residui sono WN mentre per il restante 12.25% si rifiuta la casualità dei risultati.

Messo da parte il problema della casualità, i risultati sono stati ottenuti confrontando i diversi valori dei MASE.

Per prima cosa si è calcolato quante volte il MASE derivante da JDemetra+ è minore rispetto a quello di forecast:

- Con riferimento alle 79 serie per cui si va sui livelli il MASE ottenuto con le previsioni calcolate con il modello identificato da JD+ è *minore o uguale* al concorrente nel 45.57% dei casi;
- Per le restanti 223 serie si è confrontato, in prima battuta, il valore ottenuto con i modelli identificati da forecast sui livelli. In questa situazione JD+ lavora meglio di forecast nel 47.09% dei casi.

Industry	Macro	Finance	Demographic	Totale
50.00%	44.88%	65.00%	40.00%	46.69%

Tabella 3.6: Percentuali, per tipologia delle serie, in cui il MASE calcolato con JD+ è minore o uguale a quello calcolato con forecast.

Contrariamente alle supposizioni iniziali, sembra che, nonostante la mancanza di una procedura di linearizzazione, forecast fornisca previsioni migliori. Focalizzando però l'attenzione sulle diverse categorie delle serie, vediamo che forecast *parte Arima* dà previsioni scadenti per le *serie storiche finanziarie*. Questo risultato conferma che tale strumento, essendo naturalmente meglio disposto per le serie che sono intrinsecamente lineari, non è adatto per analizzare fenomeni caratterizzati da una marcata volatilità. Dei risultati più robusti si potrebbero avere utilizzando l'*exponential smoothing* con filtro di kalman, che in forecast è implementato dalla funzione `ets()`. In JDemetra+, invece, tali non linearità vengono trattate precedentemente da TRAMO, permettendo così ottenere previsioni migliori (vedi tabella 3.6).

Una cosa che ci si aspetta, è che le performance previsive nei logaritmi, laddove JD+ lo richiede, siano migliori delle performance date da forecast con un modello identificato nei livelli. In effetti, accade che il MASE di JD+ risulta *minore o uguale* a quello di forecast per il 43.50% dei casi e, confrontato con quello precedente (47,09%), porta a dire che c'è stato effettivamente un cambiamento favorevole. Andando in dettaglio e confrontando uno ad uno i valori dei MASE ottenuti con i modelli identificati nei livelli e quelli nei logaritmi, vediamo che c'è un miglioramento (quindi una diminuzione del MASE) per l'83.86% dei casi. Questo risultato suggerisce che, se la procedura `auto.arima()` inglobasse al proprio interno una procedura di identificazione automatica della trasformata logaritmica, porterebbe presumibilmente a dei risultati migliori.

Considerando che un valore del MASE > 1 indica che il metodo proposto fornisce, in media, errori che sono maggiori degli errori a un passo ottenuti con il metodo naïve, abbiamo quindi posto l'attenzione su quelle serie storiche che presentavano un indice maggiore di 1 per entrambe le procedure. Analizzando le 67 serie risultanti e confrontando i diversi *Warnings* segnalati da JDemetra+ già in fase di identificazione del modello, si è visto che per queste serie c'erano problemi tipo: *non decomposable model*; *parameters cut off*; *no seasonal peak in the original differenced series*; *severe quality*.

Le figure 3.1 e 3.2 mostrano esempi di serie storiche per cui entrambe le procedure forniscono stime di scarsa qualità.

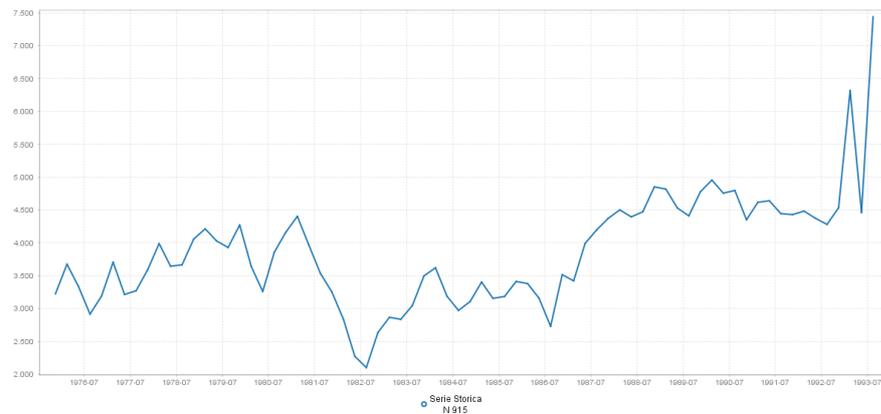


Figura 3.1: Serie Storica Trimestrale N 915

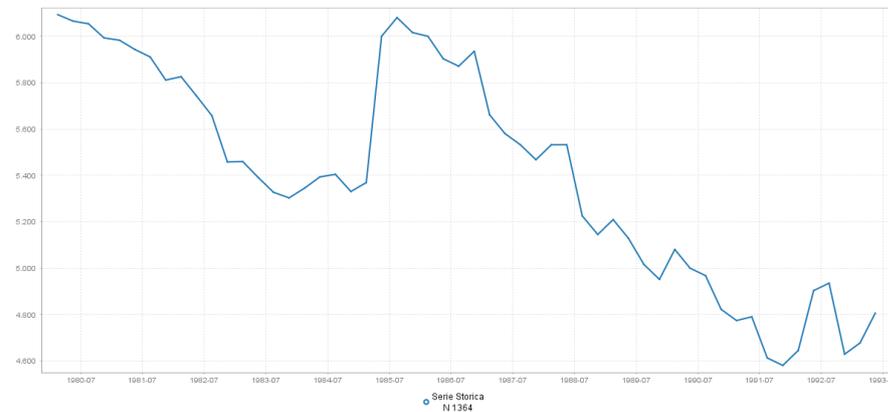


Figura 3.2: Serie Storica Trimestrale N 1364

Per poter commentare i casi limite, cioè quelli in cui una delle due procedure fornisce risultati particolarmente deludenti, abbiamo usato come metro di valutazione il rapporto tra il MASE ottenuto con $JDemetra+$ e quello fornito da *forecast*. Più ci si allontana dal valore di parità 1, più uno dei due strumenti prevale sull'altro. Ponendo l'attenzione su quelle serie che hanno un rapporto maggiore o uguale a 1.45 (in favore di *forecast*) o minore o uguale a 0.55 per $JD+$, sono state individuate rispettivamente 29 serie storiche in cui $JD+$ produce risultati di scarsa qualità e 5 in cui invece sembra funzionare meglio di *forecast*. Analizzando queste ultime serie, si è visto che per tutte $JDemetra+$ proponeva la trasformata logaritmica. Nella tabella 3.7 vengono riportati i valori dei MASE ottenuti con i vari modelli (sia quelli identificati nei livelli che quelli nei logaritmi).

Da un rapido confronto, si può dire che per 2 serie su 5 c'è un miglioramento

quando si considera un MASE ottenuto da `forecast` con il modello sui logaritmi, per le restanti 3 il problema potrebbe essere causato dalla diversa specificazione del modello (in particolare `forecast` identifica una parte stagionale mentre `JD+` no).

Serie	MASE proposto da Forecast con un modello sui livelli	MASE proposto da Forecast con un modello sui logaritmi	MASE proposto da JDemetra+
N 877	0.7160	0.4067	0.1789
N 965	0.5095	0.5095	0.2652
N 968	1.2541	0.4130	0.3038
N 1113	0.5106	0.5106	0.2662
N 1382	1.5524	1.9969	0.7173

	MODELLO proposto da Forecast sui LIVELLI	MODELLO proposto da Forecast sui LOGARITMI	MODELLO proposto da JDEMETRA+
N 877	ARIMA(1,0,0)(2,1,0) ₄	ARIMA(1,0,0)(0,1,1) ₄ with drift	ARIMA(3,1,0)(0,1,1) ₄
N 965	ARIMA(0,1,1)(0,0,1) ₄ with drift	ARIMA(0,1,1)(0,0,1) ₄ with drift	ARIMA(0,1,1) with drift
N 968	ARIMA(0,1,0) with drift	ARIMA(2,1,0) with drift	ARIMA(1,1,0) with drift
N 1113	ARIMA(0,1,1)(0,0,1) ₄ with drift	ARIMA(0,1,1)(0,0,1) ₄ with drift	ARIMA(0,1,1) with drift
N 1382	ARIMA(2,1,1)(0,1,0) ₄	ARIMA(1,2,1)	ARIMA(0,1,1)

Tabella 3.7: Confronti per le serie per cui JDemetra+ fornisce previsioni migliori rispetto a Forecast (rapporto tra i MASE < 0.55).

Capitolo 4

Conclusioni

Come considerazione finale va detto che l'analisi messa in atto in questo elaborato è esclusivamente descrittiva; cioè non è stata valutata l'equivalenza o le difformità delle capacità previsive dei modelli proposti, ma si è fornita solo una valutazione descrittiva su un buon numero di serie secondo la logica delle forecast competition. Per avere una valutazione statisticamente fondata, sarebbe stato necessario usare una procedura di verifica dell'accuratezza predittiva di due tecniche. Un approccio inferenziale di questo tipo si basa sul test di Diebold e Mariano (1995) e sugli sviluppi che ne sono derivati.

In questo lavoro, inoltre, abbiamo considerato solo due delle tante procedure automatiche disponibili. Pertanto, potrebbe essere interessante confrontare i risultati ottenuti anche con quelli che si avrebbero con procedure, tipo X11-ARIMA, che usano tecniche di identificazione automatica leggermente diverse da quelle usate da JD+ e forecast.

Sulla base dei risultati della *M3-Competition*, inoltre, pensiamo che un ulteriore approfondimento si potrebbe avere considerando orizzonti di previsione più lunghi. Infatti, per questioni di spazio e di tempo, in questo elaborato ci siamo soffermati solo su un esperimento di one-step-ahead rolling forecast ($h = 1$). Inoltre, va anche precisato che quello condotto è uno *pseudo rolling forecast experiment* (*rfe*). Infatti, un *rfe* 'puro' andrebbe condotto in tempo reale (*true real time rolling forecast experiment*), cioè usando i dati osservati di volta in volta resi disponibili dalla fonte che li produce. Questo modo di operare, che incorpora nel validation set anche un eventuale processo di revisione dei dati, avrebbe una natura molto più complessa e darebbe luogo a dataset di dimensio-

ni non trascurabili, con non banali problemi di modellazione e di gestione delle informazioni.

Bibliografía

- [1] Box G.E.P., Jenkins G.M. (1976), *Time Series Analysis, Forecasting and Control*, Holden-Day, San Francisco.
- [2] Canova F., Hansen B.E. (1995), *Are Seasonal Patterns Constant Over Time? A Test for Seasonal Stability*, *Journal of Business and Economic Statistics*, 13, 237-252.
- [3] Chen B.C., Liu L.M (1993), *Joint Estimation of Model Parameters and Outlier Effects in Time Series*, *Journal of the American Statistical Association*, 88, 284-297.
- [4] Diebold F., Mariano R. (1995), *Comparing Predictive Accuracy*, *Journal of Business and Economic Statistics*, 13, 253-263.
- [5] (2015), *ESS Guidelines on Seasonal Adjustment*, Eurostat Methodological Working Papers, <http://ec.europa.eu/eurostat/documents/3859598/6830795/KS-GQ-15-001-EN-N.pdf>.
- [6] Grudkowska S. (2015), *JDemetra+ Reference Manual*, Eurostat, https://ec.europa.eu/eurostat/cros/content/documentation_en.
- [7] Gómez V. (1998), *Automatic Model Identification in the Presence of Missing Observations and Outliers*, Working paper D-98009, Ministerio de Economía y Hacienda, Dirección General de Análisis y Programación Presupuestaria.
- [8] Gómez V., Maravall A. (1996), *Programs TRAMO and SEATS; Instructions for the Users*, Working Paper 9628, Servicio de Estudios, Banco de España.
- [9] Goodrich R.L. (2000), *The Forecast Pro Methodology*, *International Journal of Forecasting*, 16(4), 533-535.
- [10] Hannan E.J., Rissanen J. (1982). *Recursive Estimation of Mixed Autoregressive-Moving Average Order*, *Biometrika*, 69(1), 81-94.

-
- [11] Hyndman R.J., Koehler A.B. (2006), *Another look at measures of forecast accuracy*, International Journal of Forecasting, 22 (2006), 679-688.
- [12] Hyndman R.J., Khandakar Y. (2008), *Automatic Time Series Forecasting: The forecast Package for R*, Journal of Statistical Software, 27, Issue 3, July 2008.
- [13] Hyndman R.J., O'Hara-Wild M., Bergmeir C., Razbash S., Wang E. (2017), *Package 'forecast'*, <https://cran.r-project.org/web/packages/forecast/forecast.pdf>.
- [14] Kwiatkowski D., Phillips P.C., Schmidt P., Shin Y. (1992), *Testing the Null Hypothesis of Stationarity Against the Alternative of a Unit Root*, Journal of Econometrics, 54, 159-178.
- [15] Liu L.M. (1989), *Identification of Seasonal Arima Models Using a Filtering Method*, Communications in Statistics: Theory & Methods, 18, 2279-2288.
- [16] Makridakis S., Hibon M. (1979), *Accuracy of forecasting: an empirical investigation (with discussion)*, Journal of the Royal Statistical Society A 142, 97-145.
- [17] Makridakis S., Andersen A., Carbone R., Fildes R., Hibon M., Lewandowski R., Newton J., Parzen E., Winkler R. (1982), *The accuracy of extrapolation (time series) methods: results of a forecasting competition*, Journal of Forecasting, 1, 111-153.
- [18] Makridakis S., Chatfield C., Hibon M., Lawrence M., Mills T., Ord K., Simmons L.F. (1993), *The M-2 Competition: a real-time judgmentally based forecasting study*, International Journal of Forecasting, 9, 5-23.
- [19] Makridakis S., Hibon M. (2000), *The M3-Competition: results, conclusions and implications*. Francia: International Journal of Forecasting, 16, 451-476, 2000.
- [20] M elard G., Pasteels J.M. (2000), *Automatic ARIMA Modeling Including Intervention, Using Time Series Expert Software*, International Journal of Forecasting, 16, 497-508.
- [21] Reilly D. (2000), *The Autobox System*, International Journal of Forecasting, 16(4), 531-533.