

UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

---

Scuola di Economia e Statistica

Corso di laurea Magistrale in Scienze statistiche ed economiche

Tesi di Laurea Magistrale

**LODE: un metodo per  
l'individuazione e la stima di valori  
anomali e di cambiamenti strutturali,  
nelle serie storiche**

Un nuovo metodo per irrobustire il filtro di Kalman nei modelli a  
componenti non osservabili



**Relatore:**

Matteo M. Pelagatti

**Correlatore:**

Antonio Candelieri

**Laureando:**

Simone Frigerio  
Matr. N. 862410

---

Luglio 2021

# Indice

<b>1</b>	<b>Metodo LODE</b>	2
1.1	Introduzione ai modelli a componenti non osservabili . . . . .	2
1.2	Idea . . . . .	3
1.3	Interpretazione Bayesiana . . . . .	6
1.4	Ricerca del minimo . . . . .	6
1.4.1	Derivate . . . . .	7
1.4.2	Valori iniziali . . . . .	8
1.4.3	Algoritmi di minimizzazione . . . . .	12
1.4.4	Scelta dei $\lambda$ . . . . .	14
1.4.5	Riscalamento della serie . . . . .	21
1.4.6	Definizione dei gradi di libertà . . . . .	21
<b>2</b>	<b>Metodo basato sui residui ausiliari</b>	25
2.1	Funzionamento di auxres . . . . .	25
2.2	Software STAMP . . . . .	26
<b>3</b>	<b>Applicazione e risultati</b>	28
3.1	Simulazione delle serie storiche . . . . .	28
3.2	Considerazioni sulla stima del trend della serie . . . . .	30
3.3	Risultati utili per comprendere i risultati . . . . .	30
3.3.1	Esempio di una stima del trend . . . . .	30
3.3.2	Serie gaussiane pure . . . . .	31
3.3.3	Individuazione degli outlier . . . . .	33
3.3.4	Stima dimensione cambio di livello . . . . .	35
3.4	Stima del trend della serie . . . . .	39
3.4.1	Random walk . . . . .	39
3.4.2	RW con $\sigma_\eta = 0$ . . . . .	43
3.4.3	LLT più stagionalità . . . . .	45
3.4.4	Cambio di pendenza . . . . .	47
3.4.5	Cambio di stagionalità . . . . .	49
3.4.6	Errata ricerca di outlier . . . . .	51
3.4.7	Generazione disturbi da t-student . . . . .	52
3.4.8	Media tra LODE, auxres e KFS . . . . .	52
3.5	Unione( <i>LODE</i> , <i>auxres</i> ) . . . . .	54
3.6	Andamento delle varianze aggiuntive . . . . .	55

3.7	Confronto con STAMP e SAS . . . . .	57
3.8	Comportamento su serie storiche reali . . . . .	58
3.8.1	Nilo . . . . .	59
3.8.2	Incidenti stradali UK . . . . .	59
3.8.3	Inventario contabile USA . . . . .	60
3.8.4	Matrimoni in Inghilterra e Galles . . . . .	61
3.9	Conclusioni . . . . .	61
<b>A</b>	<b>Calcolo derivate e gradi di libertà per altre tipologie di modelli</b>	63
A.1	Calcolo delle derivate . . . . .	63
A.1.1	LLT con stagionalità . . . . .	63
A.1.2	RW integrato con outlier sullo slope . . . . .	64
A.1.3	RW con outlier su stagionalità . . . . .	65
A.2	Calcolo dei gradi di libertà . . . . .	65
<b>B</b>	<b>Ulteriori dettagli su <i>auxres</i></b>	66
	<b>Riferimenti bibliografici</b>	68

# Introduzione

Le serie storiche sono una tipologia di dati che al giorno d'oggi vengono frequentemente studiati ed utilizzati. Questo è dovuto al fatto che possono essere applicate in vari ambiti, da quello economico a quello medico. Numerosi sono i modelli che si sono sviluppati con l'obiettivo di fare previsione o di spiegare l'andamento della serie.

Generalmente i vari approcci hanno solide basi teoriche che sovente si basano sull'ipotesi di gaussianità o comunque di "regolarità" della serie storica; purtroppo i dati spesso non sono così belli come vorremmo, a causa di svariati motivi, come eventi inaspettati, che possono provocare valori anomali o anche cambiamenti strutturali<sup>1</sup> nella serie storica, rendendo difficoltosa l'analisi.

Perciò si sono sviluppati metodi che hanno lo scopo di individuare e/o modellare tali outlier, per riuscire a stimare in modo corretto l'andamento dei dati. Analisi di questo tipo sono anche utilizzate per valutare l'impatto di una particolare legge o fenomeno sui dati.

Nei modelli a componenti non osservabili si utilizza il filtro di Kalman<sup>2</sup> per stimare le varie componenti del modello. Anch'esso si basa sull'ipotesi di gaussianità della serie storica, perciò quando vi sono outlier è necessario irrobustire la procedura per renderla ottimale.

In questa tesi proponiamo un metodo innovativo per irrobustire il filtro di Kalman applicato ai modelli a componenti non osservabili. Il nuovo approccio lo abbiamo chiamato LODE (LASSO based Outlier Detection and Estimation), in quanto basato su una regolarizzazione simile a quella presente nella regressione LASSO e ha l'obiettivo di individuare e stimare correttamente sia gli outlier additivi sia i cambiamenti strutturali.

---

<sup>1</sup>i valori anomali, o outlier additivi, influenzano un unico istante temporale, mentre i cambiamenti strutturali modificano permanentemente una delle varie componenti della serie.

<sup>2</sup>il filtro di Kalman è un algoritmo in grado di ricostruire il segnale di una serie storica soggetta a rumore.

# Capitolo 1

## Metodo LODE

### 1.1 Introduzione ai modelli a componenti non osservabili

Al giorno d'oggi sono molteplici i metodi per analizzare le serie storiche, uno di questi si basa sulle componenti non osservabili (Pelagatti 2016). Esso si fonda sull'idea che i dati osservati siano scomponibili in più componenti a se stanti: il *trend*  $\mu_t$ , la *stagionalità*  $S_t$  e l'*errore*  $\epsilon_t$ .

$$y_t = \mu_t + S_t + \epsilon_t$$

Ogni elemento cattura una parte importante della serie storica. Andiamo ora a descrivere com'è composta ogni singola componente e come evolve nel tempo.

**Trend** Il trend della serie storica,  $\mu_t$ , è la componente più importante, quella adibita a cogliere il segnale. Essa viene modellata come un local linear trend (LLT), ovvero

$$\begin{aligned} \mu_{t+1} &= \mu_t + \beta_t + \eta_t & \eta_t &\sim NID(0, \sigma_\eta^2) \\ \beta_{t+1} &= \beta_t + \zeta_t & \zeta_t &\sim NID(0, \sigma_\zeta^2) \end{aligned}$$

Se per esempio  $\beta_t = 0 \forall t$ , ovvero è nulla la parte di slope, allora  $\mu_t$  è un semplice random walk (RW). Mentre se  $\sigma_\zeta = 0$ , allora è un RW con drift. Se invece  $\sigma_\eta = 0$  si parla di RW integrato.

**Stagionalità** Alcune serie storiche posso presentare una stagionalità, ovvero una periodicità nell'andamento dei dati. Per cogliere questa parte della serie storica aggiungiamo al nostro modello la componente  $S_t$ , una stagionalità a dummy stocastiche

$$S_{t+1} = - \sum_{i=1}^{T-1} S_{t+1-i} + \omega_t \quad \omega_t \sim NID(0, \sigma_\omega^2)$$

ove  $T$  è il periodo della stagionalità. Essa è la somma dei  $T - 1$  istanti precedenti cambiati di segno, in modo che la somma della stagionalità in un periodo sia pari a zero, o quasi, visto la presenza degli  $\omega_t$ , che permettono l'evoluzione temporale.

**Errore** L’ultima componente della nostra serie storica è  $\epsilon_t$ , il termine di errore, il disturbo additivo, e semplicemente si ipotizza

$$\epsilon_t \sim NID(0, \sigma_\epsilon^2)$$

La prima cosa che notiamo dopo aver descritto le varie componenti è che tutti i disturbi,  $\eta_t, \zeta_t, \omega_t$  e  $\epsilon_t$ , che permettono l’evoluzione delle componenti nel tempo, presentato l’ipotesi di gaussianità, si ipotizza che siano NID (Normal Identical Distributed).

Una volta definito il modello, per calcolare effettivamente le componenti, è necessario stimare le varianze dei disturbi, ovvero  $\sigma_\eta^2, \sigma_\zeta^2, \sigma_\omega^2$  e  $\sigma_\epsilon^2$ . Esse non sono note a priori, dunque si ottengono andando a massimizzare la verosimiglianza del modello, che è facilmente calcolabile dato che abbiamo supposto la gaussianità dei vari disturbi. Questo appena descritto è un passaggio fondamentale, da tener bene a mente per comprendere il funzionamento del metodo che andiamo a proporre.

Successivamente, possiamo sfruttare il fatto che tutti i modelli a componenti non osservabili possono essere anche formulati come state space model e dunque possiamo usare il filtro/smooth<sup>1</sup> di Kalman per calcolare le varie componenti.

Ricapitolando, quando si studia una serie storica tramite un modello a componenti non osservabili è necessario:

1. definire il modello
2. stimare le varianze dei disturbi massimizzando la verosimiglianza
3. calcolare le componenti con lo smoother di Kalman

Queste sono solo le nozioni basi riguardanti i modelli a componenti non osservabili, essenziali per capire ciò verrà detto.

## 1.2 Idea

Andiamo ora a presentare l’idea di questo nuovo approccio. Per semplificare, al momento non consideriamo la stagionalità, mentre il trend lo ipotizziamo pari ad un random walk.

L’ipotesi su cui si basa il filtro di Kalman è la gaussianità di  $\epsilon$  e  $\eta$ , i disturbi. Sotto di essa l’algoritmo che permette di stimare  $\mu_t$  è ottimo, ovvero ridà la migliore stima del trend possibile. Tale ipotesi non è sempre verificata, per esempio possiamo immaginare la presenza di outlier nella serie, ovvero valori molto grandi rispetto agli altri disturbi  $\epsilon$  e  $\eta$ , che quindi non possono esser considerati provenienti dalla medesima distribuzione di probabilità di  $\epsilon$  e  $\eta$ . Per un esempio si veda la Figura 1.1. Per semplificare ancora concentriamoci unicamente su outlier su  $\mu$ , il trend della serie. Per esempio immaginiamo di trattare la serie storica del Nilo, ove la costruzione della diga di Assuan nel 1899 ha provocato un’importante abbassamento del livello delle acque, si veda Figura 1.2.

---

<sup>1</sup>nel prosieguo della tesi useremo le parole filtro e smoother come sinonimi, anche se non lo sono. Lo smoother utilizza tutti i dati della serie storica per effettuare la stima al tempo  $t$ , invece il filtro usa unicamente le osservazioni fino al tempo  $t$  e non quelle successive.

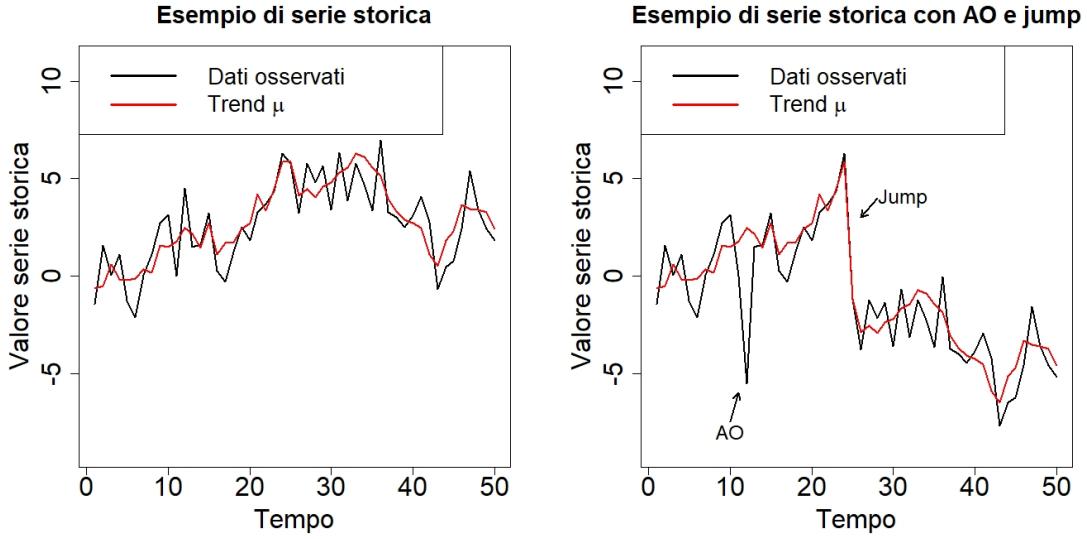


Figura 1.1: Esempio di serie storica. A sinistra coi disturbi puramente gaussiani, a destra con l’aggiunta di un outlier additivo e di un jump.

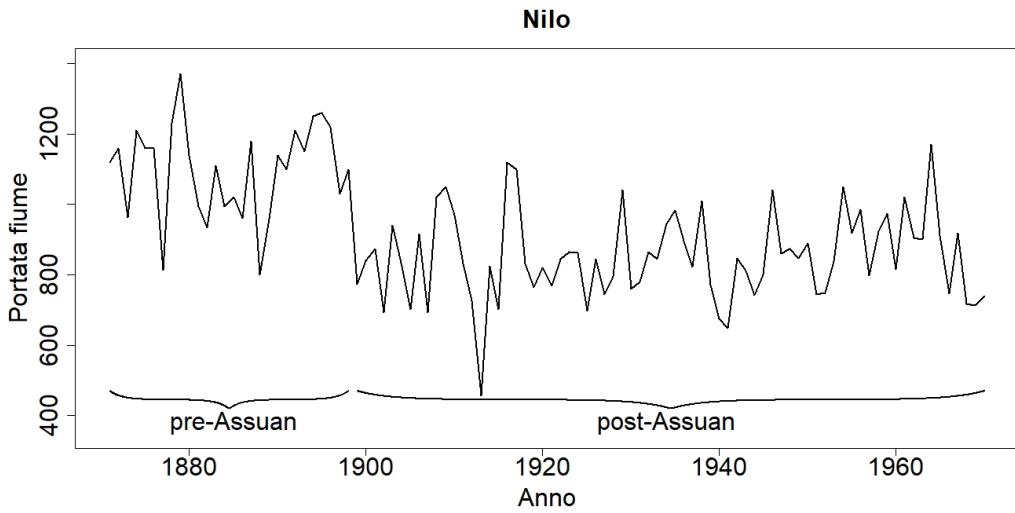


Figura 1.2: Serie storica della portata annuale del Nilo. Nel 1899 la costruzione della diga di Assuan ha provocato un notevole abbassamento delle acque.

In un modello come quello appena descritto nella Sezione 1.1,  $\sigma_\eta^2$ , ovvero la varianza della gaussiana che genera  $\eta$ , è una costante temporale e viene stimata massimizzando la verosimiglianza. In questa serie si calcolerà una varianza maggiore di quella reale, perché bisogna tenere conto anche del level shift (LS), detto anche cambio di livello o jump, causato dalla diga di Assuan. Ciò si rifletterà su una errata stima del trend, in quanto un valore di  $\sigma_\eta$  più alto implica valori di  $\eta$  più alti, ciò significa che il trend ha una miglior capacità di

variare nel tempo e perciò seguirà maggiormente l'andamento dei dati osservati. Inoltre, in genere, l'aumento nella stima della varianza non è tale da permettere di seguire il repentino cambio di livello, che dunque verrà "smussato".

Possiamo concludere come una varianza costante non sia la scelta migliore, perciò ipotizziamo di renderla variabile nel tempo. Per la precisione consideriamo  $\sigma_{\eta_t}$  costante tranne che nell'istante  $t_0$ , in cui avviene il jump, ove la consideriamo molto più grande, in modo da assecondare il cambio di livello

$$\sigma_{\eta_t} = \begin{cases} c & \text{se } t \neq t_0 \\ k \gg c & \text{se } t = t_0 \end{cases}$$

In questo modo l'algoritmo è in grado di stimare correttamente la serie in tutti punti, infatti dove si presenta il salto, la varianza maggiorata permette al trend stimato di seguire l'andamento dei dati, mentre negli altri istanti la varianza è calcolata correttamente e dunque si ha una stima corretta del trend.

Il problema che si pone ora è andare ad individuare questo istante  $t_0$  dove porre l'aumento della varianza. Senza informazioni esterne ogni punto è un buon candidato, ma considerarli tutti è impossibile, in quanto avremmo una varianza diversa per ogni punto campionario, che porterebbe ad un sovradattamento ai dati nella stima del trend. Bisogna individuare i punti in cui aumentare  $\sigma_{\eta_t}$  porta ad un reale miglioramento nella stima. Andiamo formalizzare questo metodo considerando

$$\sigma_{\eta_t}^2 = \sigma_{\eta_t}^2 + \dot{\sigma}_{\eta_t}^2 \quad \forall t,$$

ovvero una varianza composta da una costante ed un termine aggiuntivo, diverso per ogni istante temporale, che chiameremo varianza aggiuntiva,  $\dot{\sigma}_{\eta_t}^2$ . Dato  $n$  la lunghezza della serie storica, avremo  $n + 1$  parametri da stimare, per quanto riguarda la sola componente di trend. Generalmente andiamo a massimizzare la verosimiglianza, o equivalentemente, minimizzare

$$-\log(\text{verosimiglianza}),$$

ma in questo caso fare ciò è un errore, in quanto cadremmo in una sorta di *overfitting*, per questo bisogna selezionare solo le varianze aggiuntive rilevanti. Facciamo ciò andando a penalizzare i vari parametri  $\dot{\sigma}_{\eta_t}$ , cercando di renderli pari a zero se non necessari. La funzione finale è dunque del tipo:

$$-\log(\text{verosimiglianza}) + \lambda \sum_{t=1}^n |\dot{\sigma}_{\eta_t}|$$

una penalità simile a quella presente nella regressione LASSO, che rende i due modelli molto simili. Per questo motivo il nuovo approccio che proponiamo lo abbiamo chiamato LODE (LASSO based Outlier Detection and Estimation). L'obiettivo è dunque andare a minimizzare tale funzione, ottenendo così una varianza  $\sigma_{\eta_t}$  costante tranne che nei punti in cui un valore di  $\dot{\sigma}_{\eta_t} \neq 0$  comporta un miglioramento nella verosimiglianza maggiore della penalità. In quest'ultima non andiamo a mettere  $\sigma_{\eta_t}$  poiché si produrrebbe una stima distorta di essa. Abbiamo scelto una penalità con norma L1 poiché è un grado di porre a zero i coefficienti non necessari. Norme come L2 o maggiori non hanno questa proprietà, mentre una norma L0 renderebbe il problema di minimizzazione non continuo.

In generale andiamo a considerare anche la possibilità di outlier additivi (AO), dunque ripetiamo il medesimo ragionamento anche per  $\sigma_\epsilon$ . Ora la funzione da minimizzare è più complessa in quanto abbiamo:

$$\mathcal{L}_{\lambda_\epsilon, \lambda_\eta}(\theta) = -\log(f(y, \theta)) + \lambda_\epsilon \sum_{t=1}^n |\dot{\sigma}_{\epsilon_t}| + \lambda_\eta \sum_{t=1}^n |\dot{\sigma}_{\eta_t}| \quad (1.1)$$

con  $\log(f(y, \theta))$  il logaritmo della funzione di verosimiglianza,  $y$  la serie storica in analisi e  $\theta = (\sigma_\epsilon, \dot{\sigma}_{\epsilon_1}, \dots, \dot{\sigma}_{\epsilon_n}, \sigma_\eta, \dot{\sigma}_{\eta_1}, \dots, \dot{\sigma}_{\eta_n})$ . Scegliendo dei parametri  $\lambda_\epsilon$  e  $\lambda_\eta$  opportuni e ottimizzando tale funzione rispetto a  $\theta$ , otteniamo il valore stimato di tutte le varianze, in tal modo possiamo successivamente applicare lo smoother di Kalman per costruire le stime delle varie componenti. Rifacendoci alla scaletta presentata nella Sezione 1.1, sottolineiamo come il nostro nuovo approccio vada a modificare solo i punti 1 e 2. L'ultimo punto rimane invariato e per questo a volte non verrà esplicitamente descritto.

### 1.3 Interpretazione Bayesiana

La funzione da minimizzare (1.1), può per comodità essere cambiata di segno e essere vista come una funzione da massimizzare:

$$+ \log(f(y, \theta)) - \lambda_\epsilon \sum_{t=1}^n \dot{\sigma}_{\epsilon_t} - \lambda_\eta \sum_{t=1}^n \dot{\sigma}_{\eta_t} \quad (1.2)$$

dove consideriamo  $\dot{\sigma}_{\eta_t} > 0$  e  $\dot{\sigma}_{\epsilon_t} > 0$ . Nell'interpretazione Bayesiana andiamo a porre una prior pari ad una esponenziale sui parametri aggiuntivi ed una uniforme sui parametri  $\sigma_\epsilon$  e  $\sigma_\eta$ , come 1 o  $1/c$ , a seconda del dominio che gli poniamo. Ipotizzando l'indipendenza di tutti i parametri, la prior si può scrivere come:

$$\pi(\theta) = \pi(\sigma_\epsilon, \dot{\sigma}_{\epsilon_1}, \dots, \dot{\sigma}_{\epsilon_n}, \sigma_\eta, \dot{\sigma}_{\eta_1}, \dots, \dot{\sigma}_{\eta_n}) = 1 \cdot \prod_t b e^{-b\dot{\sigma}_{\epsilon_t}} \cdot 1 \cdot \prod_t d e^{-d\dot{\sigma}_{\eta_t}}.$$

A noi interessa la moda della posterior, o più semplicemente la moda della log posterior, tanto è medesima:

$$\begin{aligned} \log(f \cdot \pi) &= +\log(f) + \log(\pi) = \\ &= +\log(f) + \log(b) + \sum_t -b\dot{\sigma}_{\epsilon_t} + \log(d) + \sum_t -d\dot{\sigma}_{\eta_t} = \\ &= +\log(f) + \log(b) - b \sum_t \dot{\sigma}_{\epsilon_t} + \log(d) - d \sum_t \dot{\sigma}_{\eta_t}. \end{aligned}$$

Nel calcolo del massimo  $\log(b)$  e  $\log(d)$  non giocano alcun ruolo. Quindi ponendo  $\lambda_\epsilon = b$  e  $\lambda_\eta = d$  si ottiene l'equazione (1.2).

### 1.4 Ricerca del minimo

Ora che è chiara l'idea dietro il metodo vediamo come realizzarla nella pratica. Consideriamo sempre il modello del random walk, il nostro obiettivo è minimizzare la funzione (1.1).

Subito al primo impatto si osserva come non sappiamo nulla riguardo al valore di  $\lambda_\eta$  e  $\lambda_\epsilon$ . Essi sono molto importanti, influenzano i valori ottimali delle varianze aggiuntive. Porre  $\lambda_\eta = \lambda_\epsilon = 0$  produrrebbe un sovradattamento ai dati osservati, mentre porre  $\lambda_\eta = \lambda_\epsilon = \infty$  equivale ad una stima con tutti i parametri aggiuntivi pari a zero. Per ora ci limitiamo a considerarli noti, nella Sezione 1.4.4 andremo ad affrontare anche tale problematica. Purtroppo le varie parti del metodo sono state sviluppate in contemporanea, dunque ci dispiace se alcune volte saremo costretti a riferirci a sezioni non ancora presentate.

La nostra funzione presenta un gran numero di parametri da stimare, se  $n$  sono il numero di istanti temporali, allora avremo,  $2n + 2$  parametri e ciò non rende a prima vista agevole il calcolo. Soprattutto immaginiamo che ciò comporti la presenza di gran numero di minimi locali. Di conseguenza bisogna usare particolare accortezza nel applicare algoritmi di minimizzazione locale, basati sulle derivate, come quelli presenti nella funzione *optim* del linguaggio di programmazione *R*. Algoritmi di minimizzazione globale, come quelli genetici, soffrono di meno il problema dei minimi locali, tuttavia richiedono un gran dispendio di tempo computazionale. Proprio per questi motivi ci concentriamo sull'ottimizzare le performance degli algoritmi più veloci, confrontandoli poi con algoritmi di minimizzazione globale per cercare il miglior compromesso tempo-precisione.

Nel prosieguo della tesi capiterà spesso di parlare di varianze  $\sigma^2$  e soprattutto di radice di varianze  $\sqrt{\sigma^2} = |\sigma|$ . Delle volte sarà sotto inteso il valore assoluto e si considereranno come quantità positive.

#### 1.4.1 Derivate

Per prima cosa ci accorgiamo che siamo in grado di calcolare le derivate analitiche della funzione rispetto ai singoli parametri.

Iniziamo con il riportare i calcoli per il modello più semplice, il random walk, ove cerchiamo sia outlier additivi sia level shift:

$$\begin{aligned} y_t &= \mu_t + \epsilon_t & \epsilon_t &\sim NID(0, \sigma_\epsilon^2 + \dot{\sigma}_{\epsilon_t}^2) \\ \mu_{t+1} &= \mu_t + \eta_t & \eta_t &\sim NID(0, \sigma_\eta^2 + \dot{\sigma}_{\eta_t}^2). \end{aligned} \quad (1.3)$$

La funzione da minimizzare è:

$$\mathcal{L}_{\lambda_\epsilon, \lambda_\eta}(\theta) = -\log(f(y, \theta)) + \lambda_\epsilon \sum_{t=1}^n |\dot{\sigma}_{\epsilon_t}| + \lambda_\eta \sum_{t=1}^n |\dot{\sigma}_{\eta_t}|$$

con  $\theta = (\sigma_\epsilon, \dot{\sigma}_{\epsilon_1}, \dots, \dot{\sigma}_{\epsilon_n}, \sigma_\eta, \dot{\sigma}_{\eta_1}, \dots, \dot{\sigma}_{\eta_n})$ .

Le derivate sono state calcolate in (Koopman e Shepard 1992), per cui ci limitiamo ad

applicare la formula 3.2 dell’articolo:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \sigma_\epsilon} &= -\frac{1}{2} \sum_{t=1}^n (e_t e'_t - D_t) \sigma_\epsilon \\ \frac{\partial \mathcal{L}}{\partial \sigma_\eta} &= -\sum_{t=1}^n (r_{t-1} r'_{t-1} - N_{t-1}) \sigma_\eta \\ \frac{\partial \mathcal{L}}{\partial \dot{\sigma}_{\epsilon_t}} &= -(e_t e'_t - D_t) \dot{\sigma}_{\epsilon_t} + \lambda_\epsilon \text{sign}(\dot{\sigma}_{\epsilon_t}) \\ \frac{\partial \mathcal{L}}{\partial \dot{\sigma}_\eta} &= -(r_{t-1} r'_{t-1} - N_{t-1}) \dot{\sigma}_\eta + \lambda_\eta \text{sign}(\dot{\sigma}_\eta)\end{aligned}$$

dove  $e_t$ ,  $D_t$ ,  $r_t$  e  $N_t$  sono quantità complicate senza un preciso significato logico. Per una loro rigorosa definizione matematica si veda la formula 3.1 nell’articolo (Koopman e Shepard 1992).

Per il calcolo delle derivate degli altri modelli, con anche l’individuazione di outlier su altre componenti, si veda l’Appendice A.1.

La conoscenza analitica delle derivate è molto importante perché permette di ridurre drasticamente i tempi degli algoritmi di minimizzazione, che altrimenti dovrebbero calcolarle numericamente.

#### 1.4.2 Valori iniziali

Un altro input importante che le funzioni richiedono sono i punti di partenza per i parametri. Nelle prime fasi di analisi, non ci è sembrata troppo significativa la definizione dei valori iniziali, in quanto, studiando serie simulate gli algoritmi di minimizzazione globale provati, riuscivano in ogni caso a fornire una stima diversa da quella del filtro di Kalman classico, selezionando opportunamente le variazioni aggiuntive rilevanti. Tuttavia questo non accade con algoritmi locali in quanto soggetti al problema dei minimi locali. Perciò provando valori casuali, o banali, come porre tutte le varianze aggiuntive circa zero, non si ottengono grossi risultati, causa minimi locali. Per ovviare al problema è chiaro che è necessaria una strategia per definire i valori iniziali. Ne sono state individuate due. Le presentiamo come sempre per il caso più semplice, l’analisi di un random walk con outlier additivi e cambi di livello.

Il primo approccio sfrutta le informazioni che possono essere ricavate osservando la serie delle differenze prime della serie storica. La presenza di un jump o di un outlier additivo si manifesta in una grande differenza tra due istanti temporali successivi, mentre se essa è quasi nulla, è improbabile che siano verificati particolari fenomeni. Non possiamo comunque avere la certezza di questo, in quanto la somma di un jump negativo e di un outlier additivo positivo nel medesimo istante temporale possono portare ad un effetto totale nullo. Noi consideriamo questa eventualità molto rara e dunque la trascuriamo, in quanto comunque sarebbe impossibile da rilevare basandosi unicamente sui dati della serie storica.

Il vettore delle differenze  $\Delta_t = y_{t+1} - y_t$ , con  $t = 1, \dots, n-1$ , può essere usato per definire i valori iniziali delle varianze aggiuntive. Notiamo come il vettore sia di lunghezza  $n-1$ , manca il dato per la prima varianza aggiuntiva sulla parte additiva e l’ultima per la parte

di trend. Se guardiamo il modello (1.3) capiamo il perché di questo fatto.  $\epsilon_t$  influenza  $y_t$ , mentre  $\eta_t$  influenza  $y_{t+1}$ .  $\dot{\sigma}_{\eta_n}$  è zero, in quanto andrebbe ad influenzare l'istante  $n + 1$ , che non osserviamo, dunque è inutile. Mentre per  $\dot{\sigma}_{\epsilon_1}$  optiamo per usare  $2sd(y)$  ovvero un valore "grande" da permettere la presenza di un outlier in tale istante. Per quanto riguarda  $\sigma_\epsilon$  e  $\sigma_\eta$ , ovvie scelte riguardano quelle stimate dall'algoritmo classico, che chiameremo  $\hat{\sigma}_{\epsilon KFS}$  e  $\hat{\sigma}_{\eta KFS}$ <sup>2</sup>. Dunque l'idea è utilizzare per i valori iniziali un vettore del tipo

$$I = (\sigma_\epsilon, \dot{\sigma}_{\epsilon_1}, \dot{\sigma}_{\epsilon_{2:n}}, \sigma_\eta, \dot{\sigma}_{\eta_{1:(n-1)}}, \dot{\sigma}_{\eta_n}) = (\hat{\sigma}_{\epsilon KFS}, 2sd(y), \Delta_{1:(n-1)}, \hat{\sigma}_{\eta KFS}, \Delta_{1:(n-1)}, 0). \quad (1.4)$$

I valori ottimi dei parametri ci aspettiamo siano minori di questi presentati, in quanto,  $\hat{\sigma}_{\epsilon KFS}, \hat{\sigma}_{\eta KFS}$  sono sovrastime della varianza e  $\Delta$  racchiude sia la parte additiva che di trend. Dunque abbiamo pensato che l'opzione migliore potesse essere una frazione di questo vettore.

Per valutare le differenze di performance abbiamo preso a campione 5 serie storiche e abbiamo calcolato il minimo della funzione con l'algoritmo *BFGS*, per alcuni  $\lambda$  selezionati tramite una griglia, dando in input diversi valori iniziali:  $\frac{I}{3}, \frac{I}{2}, \frac{2I}{3}, \frac{3I}{4}$ . I dettagli sulla scelta dell'algoritmo di minimizzazione verranno forniti nei prossimi paragrafi.

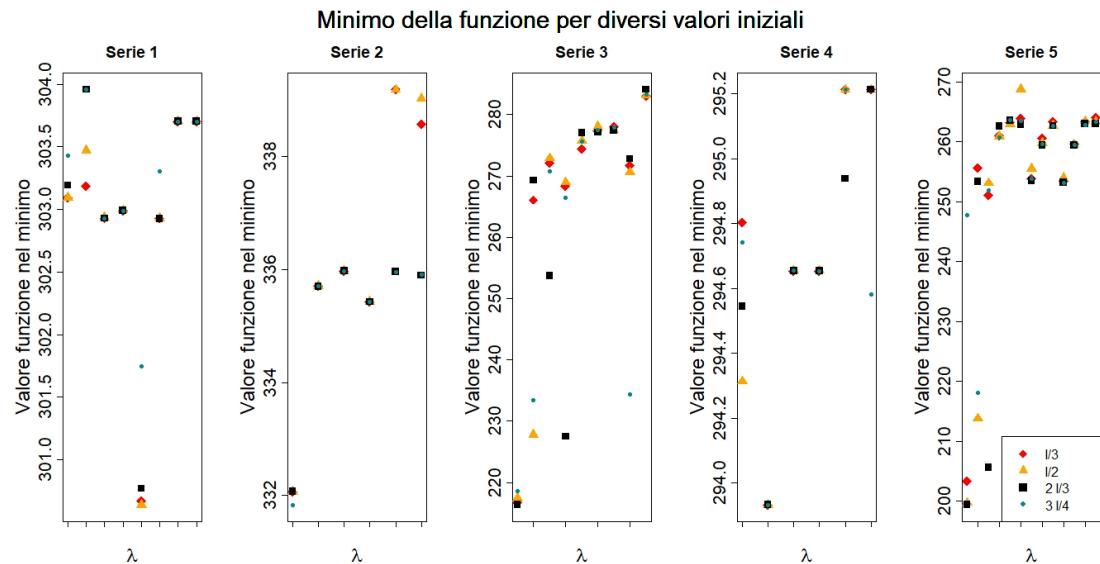


Figura 1.3: Valore del minimo della funzione  $\mathcal{L}$  per 5 serie storiche con svariati  $\lambda$ . I diversi valori iniziali, tutti basati sul vettore  $\Delta_t$ , sono rappresentati dalle diverse tipologie di punti. La legenda è stata riportata solo nel grafico della serie 5, in quanto identica per ognuno di esse. Sulle ascisse non sono riportati i specifici valori di  $\lambda$ , in quanto non di nostro interesse. Il numero di  $\lambda$  riportati non è identico per le diverse serie a causa di un errore numerico, che sarà discusso nella Sezione 1.4.3.

<sup>2</sup>la sigla KFS sta per *Kalman filter smoother*, noi la useremo per indicare l'approccio classico, ove gli outlier non vengono trattati.

Valore medio	
$\frac{I}{3}$	284.4
$\frac{I}{2}$	282.7
$\frac{2I}{3}$	281.9
$\frac{3I}{4}$	282.9

Tabella 1.1: Media dei valori di minimo della funzione da minimizzare per i diversi  $\lambda$  e per le diverse serie storiche, presentate nella Figura 1.3. I risultati sono riportati per le diverse scelte di valori iniziali.

I risultati elencati nella Figura 1.3 e nella Tabella 1.1 ci portano a scegliere  $\frac{2I}{3}$ , in quanto si ottengono valori di minimo inferiori, anche se gli altri punti iniziali hanno risultati simili. Per dire simili o meno, oltre a guardare la figura è necessario fare un passo avanti, che tratteremo meglio in seguito, nella Sezione 1.4.4. Il valore del minimo è molto legato alla verosimiglianza, che entra in gioco nel calcolo dei criteri di informazione, che useremo per scegliere i valori ottimali dei  $\lambda$ . Per calcolare il BIC<sup>3</sup>, il criterio che usiamo noi, bisogna valutare meno due volte la verosimiglianza, dunque la differenza tra i valori di minimo trovata, grosso modo si raddoppia nel calcolo del BIC. Si noterà come variazioni del BIC inferiori all’unità non siano così influenti per le performance del modello, dunque considerando setting simili per la simulazione delle serie storiche, valori di differenza di minimo attorno allo 0.5 possono essere considerati molto simili.

Un’altra strategia individuata per scegliere i valori iniziali si basa sullo *disturbance smoother*. Esso stima i valori di  $\hat{\epsilon}_t$  e  $\hat{\eta}_t$  necessari a ricostruire la serie. Proprio questi valori possono essere utilizzati come valori iniziali per i parametri aggiuntivi. A differenza del caso precedente essi non sono per forza una sovrastima dei parametri ottimali, però comunque si possono applicare altre trasformate per ricercare le migliori performance. Ne abbiamo provate svariate, ma ne riportiamo solo alcune come esempi:

$$\begin{aligned} I1 &= (\sigma_\epsilon, \dot{\sigma}_{\epsilon_{1:n}}, \sigma_\eta, \dot{\sigma}_{\eta_{1:n}}) = (0.66\hat{\sigma}_{\epsilon KFS}, \hat{\epsilon}_{1:n}, 0.66\hat{\sigma}_{\eta KFS}, \hat{\eta}_{1:n}) \\ I2 &= (0.66\hat{\sigma}_{\epsilon KFS}, 2\hat{\epsilon}_{1:n}, 0.66\hat{\sigma}_{\eta KFS}, 2\hat{\eta}_{1:n}) \\ I3 &= (0.66\hat{\sigma}_{\epsilon KFS}, \tilde{\epsilon}_{1:n}, 0.66\hat{\sigma}_{\eta KFS}, \tilde{\eta}_{1:n}) \end{aligned}$$

con

$$\tilde{\epsilon}_t = \begin{cases} \hat{\epsilon}_t & \text{se } \hat{\epsilon}_t > \hat{\sigma}_{\epsilon KFS} \\ \frac{\hat{\epsilon}_t}{100} & \text{se } \hat{\epsilon}_t \leq \hat{\sigma}_{\epsilon KFS} \end{cases} \quad \tilde{\eta}_t = \begin{cases} \hat{\eta}_t & \text{se } \hat{\eta}_t > \hat{\sigma}_{\eta KFS} \\ \frac{\hat{\eta}_t}{100} & \text{se } \hat{\eta}_t \leq \hat{\sigma}_{\eta KFS} \end{cases}$$

ed infine consideriamo quelli trovati precedentemente, come benchmark

$$I4 = (\sigma_\epsilon, \dot{\sigma}_{\epsilon_1}, \dot{\sigma}_{\epsilon_{2:n}}, \sigma_\eta, \dot{\sigma}_{\eta_{1:(n-1)}}, \dot{\sigma}_{\eta_n}) = (0.66\hat{\sigma}_{\epsilon KFS}, sd(y), \Delta_{1:(n-1)}, 0.66\hat{\sigma}_{\eta KFS}, \Delta_{1:(n-1)}, 0).$$

<sup>3</sup>ricordiamo che il BIC è pari a  $-2\log(f(y, \theta)) + \log(n) \cdot df$ , con  $f$  la verosimiglianza,  $n$  il numero di istanti temporali della serie e  $df$  i gradi di libertà

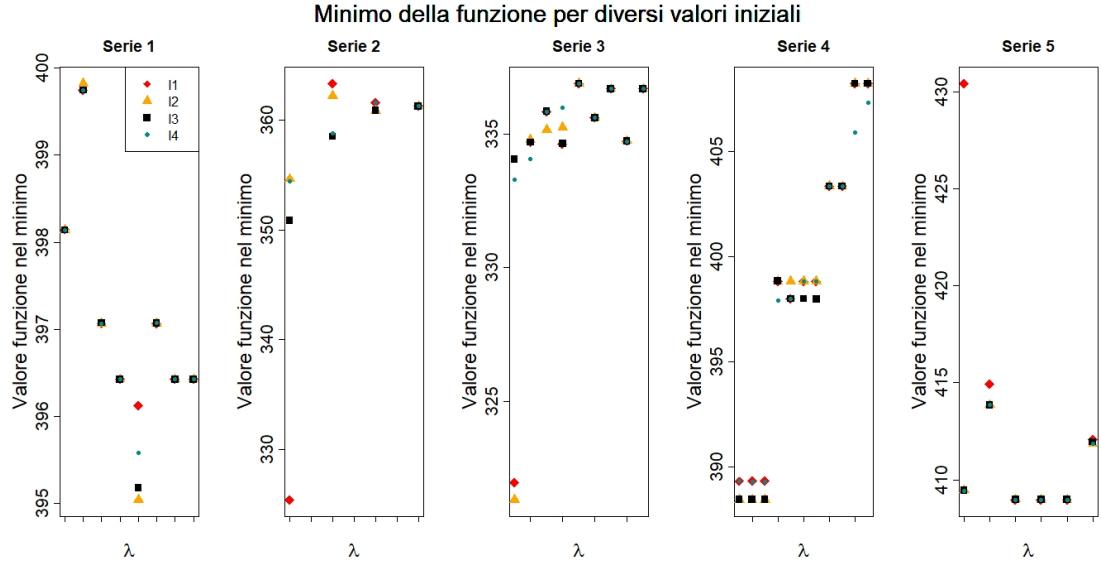


Figura 1.4: Valore del minimo della funzione  $\mathcal{L}$  per 5 serie storiche con svariati  $\lambda$ . I diversi valori iniziali, basati sullo *disturbance smoother*, sono rappresentati dalle diverse tipologie di punti. La legenda è stata riportata solo nel grafico della serie 1, in quanto identica per ognuno di esse. Sulle ascisse non sono riportati i specifici valori di  $\lambda$ , in quanto non di nostro interesse. Il numero di  $\lambda$  riportati non è identico per le diverse serie a causa di un errore numerico, che sarà discusso nella Sezione 1.4.3.

Valore medio	
I1	380.9
I2	380.9
I3	381.1
I4	381.2

Tabella 1.2: Media dei valori di minimo della funzione da minimizzare per i diversi  $\lambda$  e per le diverse serie storiche, presentate nella Figura 1.4. I risultati sono riportati per le diverse scelte di valori iniziali.

Come si vede dalla Figura 1.4 i risultati sono molto simili, nessun vettore è migliore uniformemente. Questo secondo metodo è facilmente utilizzabile anche in modelli con altre componenti, come slope o stagionalità. Per quello precedente è necessario sottrarre alla serie originale le stime di slope e/o stagionalità ridate dallo smoother classico, per riottenere un modello paragonabile ad un random walk e poi applicare le differenze. Se tuttavia siamo alla ricerca di outlier sullo slope o sulla stagionalità, il metodo delle differenze non funziona più così facilmente, bisognerebbe utilizzare differenze seconde e/o stagionali, mentre con questo nuovo approccio si possono utilizzare facilmente i disturbi calcolati per le rispettive

componenti. Per uniformità di decisione tra le varie serie storiche, decidiamo di usare proprio quest'ultimo metodo presentato, in particolare *I1*, la scelta più semplice, senza trasformazione dei parametri.

### 1.4.3 Algoritmi di minimizzazione

Nelle fasi precedenti non abbiamo approfondito il discorso riguardo alla scelta dell'algoritmo di minimizzazione, in questa sezione descriveremo il processo che ha portato a scegliere *BFGS*. Algoritmi di minimizzazione locale sono più veloci, ma più soggetti al problema dei minimi locali, per questo teniamo sempre come riferimento un algoritmo globale, che è presumibilmente più preciso. Tra i vari metodi disponibili sceglieremo quello implementato dalla funzione *genoud* (*Package rgenoud*), che combina algoritmi genetici con minimizzazione locale, ottimo per noi in quanto in grado di sfruttare le derivate. Avendo numerosi parametri da minimizzare, è stato necessario fissare il numero massimo di generazioni a 20 per limitare il tempo di esecuzione.

Per quanto riguarda algoritmi locali abbiamo a disposizione le diverse opzioni fornite dalla funzione *optim*, come *BFGS*, *CG* e *L-BFGS-B*. Quest'ultimo presenta anche l'opportunità di definire un dominio. In tal caso è stato fissato pari a otto volte i valori iniziali e simmetricamente allo zero:

$$D = (\sigma_\epsilon, \dot{\sigma}_{\epsilon_1}, \dot{\sigma}_{\epsilon_{2:n}}, \sigma_\eta, \dot{\sigma}_{\eta_{1:(n-1)}}, \dot{\sigma}_{\eta_n}) = 8 \cdot (0.66\hat{\sigma}_{\epsilon KFS}, \hat{\epsilon}_{1:n}, 0.66\hat{\sigma}_{\eta KFS}, \hat{\eta}_{1:n}).$$

Il dominio è stato fissato in modo che fosse sufficiente ampio per comprendere sempre i valori ottimali dei parametri. Forse può sembrare troppo grande, ma si è visto che ciò non ha molta influenza sui risultati.

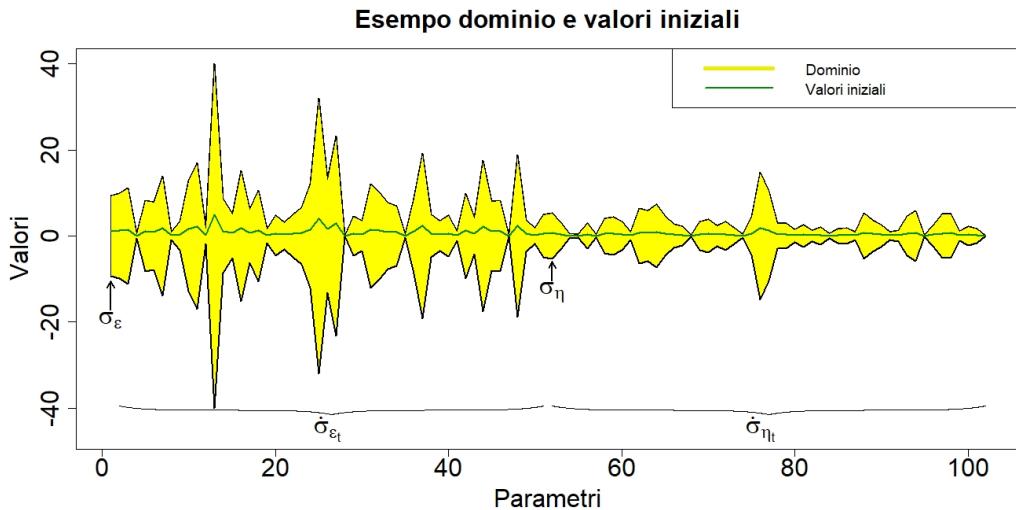


Figura 1.5: Dominio e valori iniziali per l'esempio di serie storica mostrato nella parte di destra della Figura 1.1. Si può notare come vi siano valori iniziali, o in egual modo, parametri il cui dominio è molto grande. Sono gli istanti in cui sono presenti gli outlier.

Sottolineiamo come per tutti gli algoritmi di *optim* abbiamo portato il numero massimo di iterazioni a 100 volte il numero di parametri, per assicurarci che lo stop dell'algoritmo fosse dovuto alla convergenza e non al raggiungimento del numero massimo di iterazioni.

Come primo confronto abbiamo usato l'algoritmo *BFGS*, dunque non fissando alcun dominio e utilizzando i valori iniziali descritti trovati con il vettore delle differenze. Per *genoud*, invece abbiamo provato sia quest'ultimi, sia un vettore composto solo da 0.01. I risultati sono molto simili, riportiamo solo quest'ultimo caso. Come serie storiche per questi test abbiamo sempre usato random walk con outlier additivi e cambi di livelli. I valori di lambda sono stati fissati attraverso una griglia. Nelle immagini riportate il numero di minimi trovati per serie storica non è costante, come è già successo nelle immagini precedenti. Questo poiché alcune volte il programma restituiva errore, dunque tali valori dei lambda non sono stati riportati. L'errore restituito dalle funzioni presenti nel pacchetto *KFAS* (*Package KFAS*) è il seguente:

```
1 Error in is.SSModel(model, na.check = TRUE, return.logical =
  FALSE) :
2 System matrices (excluding Z) contain NA or infinite values,
covariance matrices contain values larger than 1e+07
```

Non si sa bene da cosa sia causato tale errore, ma ha provocato diversi problemi per tutta la durata della tesi.

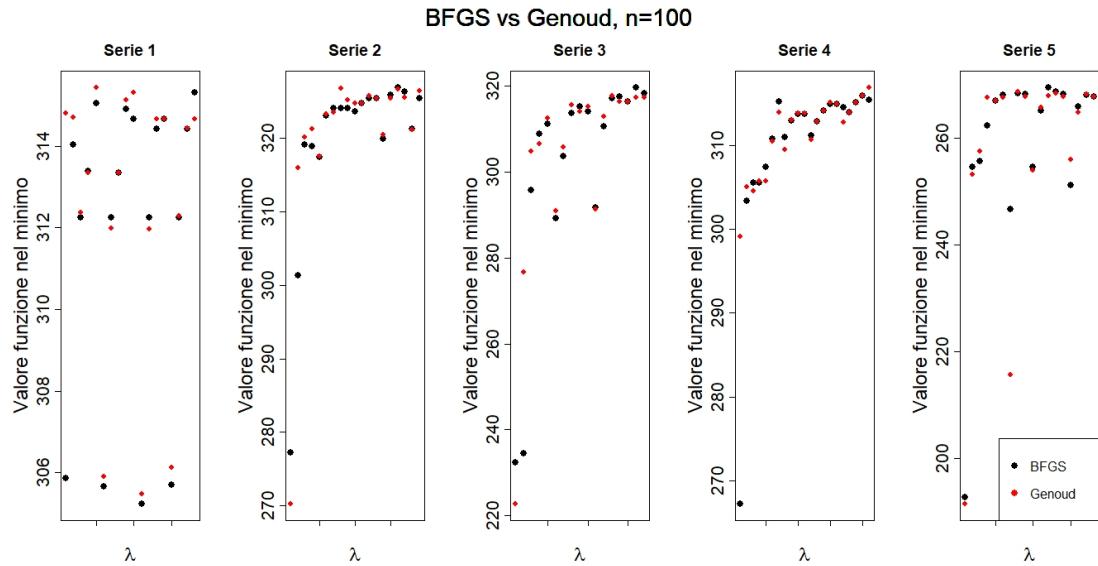


Figura 1.6: Valori di minimo ottenuti con *BFGS* e *genoud*, per 5 serie storiche e diversi valori di  $\lambda$ , nei casi in cui il numero di istanti temporali sia pari a 100. La legenda è stata riportata solo nel grafico della serie 5, in quanto identica per ognuno di esse. Sulle ascisse non sono riportati i specifici valori di  $\lambda$ , in quanto non di nostro interesse.

Dalle figure 1.6 e 1.7 notiamo come nessuno dei due algoritmi porti sempre a valori di minimo più bassi, in generale sono molto simili. Dunque *BFGS* non sembra troppo soggetto al problema dei minimi locali, se definiamo in modo opportuno i valori iniziali. Ovviamente

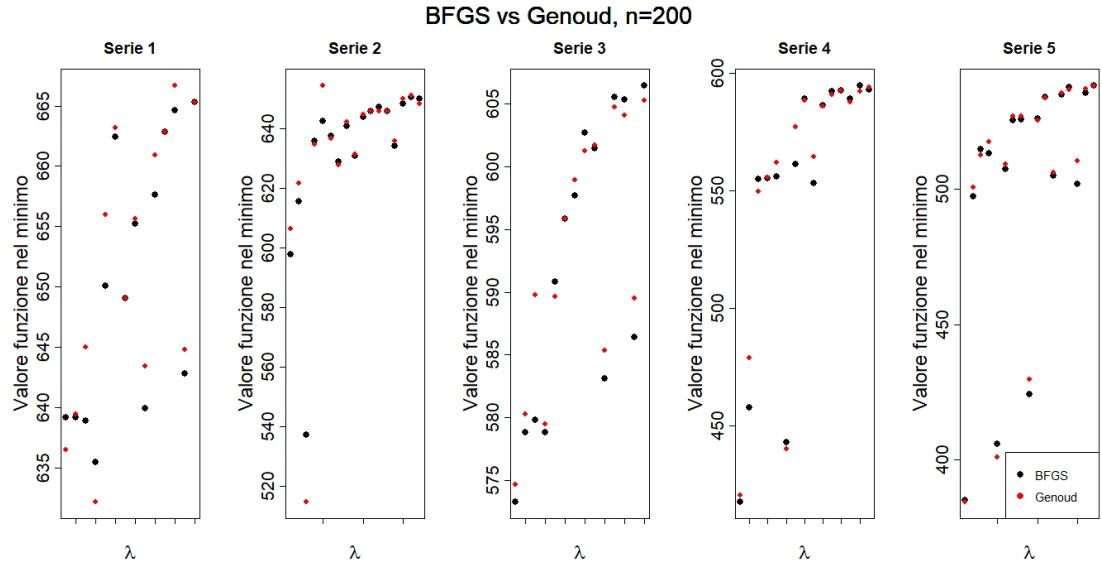


Figura 1.7: Valori di minimo ottenuti con *BFGS* e *genoud*, per 5 serie storiche e diversi valori di  $\lambda$ , nei casi in cui il numero di istanti temporali sia pari a 200. La legenda è stata riportata solo nel grafico della serie 5, in quanto identica per ognuno di esse. Sulle ascisse non sono riportati i specifici valori di  $\lambda$ , in quanto non di nostro interesse.

trai due *BFGS* è molto più efficiente, in quanto impiega dai 2 ai 10 secondi, invece *genoud* qualche minuto.

Un’ulteriore test riguardo l’algoritmo L-BFGS-B, che abbiamo usato fissando il dominio. Anche in questo caso abbiamo usato le modalità descritte sopra per effettuare il confronto.

Dalla Figura 1.8 risulta evidente come *BFGS* sia migliore nella maggior parte dei casi. Tuttavia L-BFGS-B, non è da scartare completamente. Non fissando alcun dominio vi è il problema numerico descritto poco fa, invece fissando il dominio dei parametri questo non accade più, dunque L-BFGS-B, può essere utile nel caso in cui *BFGS* restituisca errore. In ogni caso anche usando senza fissare il dominio ridava molto raramente il messaggio di errore, dunque è stato considerato per far convergere il filtro di kalman classico, ove non ci sono particolari problemi di minimo locali.

Un ultimo confronto è stato effettuato con l’algoritmo *CG*. I risultati sono molto simili a *BFGS*, tuttavia il metodo del gradiente coniugato presenta lo svantaggio della lentezza, non è veloce quanto *BFGS*, impiega circa 10-40 secondi. Per questo motivo non lo consideriamo.

Per quanto detto in tutta questa sezione abbiamo scelto di usare l’algoritmo *BFGS*, tranne quando restituisce errore, in tal caso fissiamo il dominio ed usiamo *L-BFGS-B*.

#### 1.4.4 Scelta dei $\lambda$

Nella funzione da minimizzare  $\mathcal{L}$  (1.1) sono presenti i parametri  $\lambda_e$  e  $\lambda_\eta$  che fino ad ora abbiamo considerato noti. Tuttavia non lo sono e tale ignoranza porta problemi sia a livello di tempo computazionale sia di ottimizzazione delle performance del modello. A seconda di quali tipologie di outlier andiamo a cercare, avremo un diverso numero di parametri  $\lambda$  da

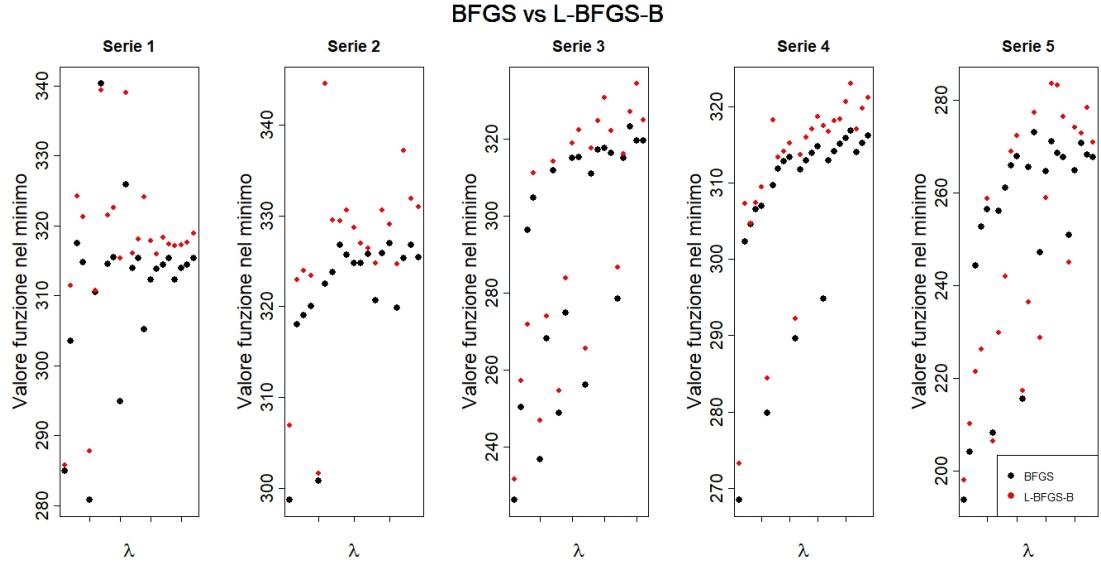


Figura 1.8: Valori di minimo ottenuti con BFGS e L-BFGS-B, per 5 serie storiche con 100 istanti temporali e diversi valori di  $\lambda$ . La legenda è stata riportata solo nel grafico della serie 5, in quanto identica per ognuno di esse. Sulle ascisse non sono riportati i specifici valori di  $\lambda$ , in quanto non di nostro interesse.

scegliere. Se ci focalizziamo ancora una volta sul caso del random walk con outlier additivi e shift, allora saranno due,  $\lambda_\epsilon$  e  $\lambda_\eta$ .

La risoluzione delle varie problematica riguardanti i  $\lambda$  si divide in due fasi:

- decidere l’insieme di valori per cui cercare il minimo di  $\mathcal{L}_\lambda$
- definire un criterio per stabilire il migliore valore.

Partiamo prima da quest’ultimo, ipotizzando di avere già trovato il minimo della funzione (1.1) per diversi valori dei  $\lambda$ . Ciò che si fa generalmente nella regressione LASSO, per stabilire il miglior parametro, è una cross validazione e si sceglie quello con cui si ottiene un errore minore. Adottare una strategia simile nel nostro caso, ovvero eliminare delle osservazioni, stimare il modello e predire tali dati, è assai oneroso, poiché ci si impiega circa 10 secondi per trovare il minimo, ed andrebbe fatto svariate volte e dunque ripetuta per i diversi valori di  $\lambda$ . Per cui optiamo per un altro metodo, molto più veloce: i criteri di informazione. Noi utilizziamo il BIC, il più parsimonioso, semplicemente perché si ottengono risultati migliori nelle simulazioni. Usare tale criterio però porta ad un problema teorico, la definizione dei gradi di libertà, che affronteremo nella Sezione 1.4.6.

Ora concentriamoci sul primo punto, la scelta dei  $\lambda$  sui cui valutare la nostra funzione  $\mathcal{L}_\lambda$  da minimizzare. Un approccio base consiste nel definire una griglia di possibili valori e valutare il modello per ognuno di essi. Il principale problema di questa tecnica è il grave impegno computazionale, infatti considerando 8 valori per ogni  $\lambda$  avremo un griglia dimensione  $8 \times 8 = 64$ . Considerando che per una serie di 100 istanti temporali si impiega circa 10 secondi per cercare un minimo, si comprende come il tempo totale sia elevato. Per

ovviare a tale problematica utilizziamo l’ottimizzazione bayesiana (BO) (Frazier 2018), la quale offre l’opportunità di trovare valori dei  $\lambda$  ottimi in un minor numero di iterazioni.

**Ottimizzazione bayesiana** L’ottimizzazione bayesiana è un approccio che permette di trovare il minimo di una funzione che impiega molto tempo ad essere valutata, come nel nostro caso è  $\mathcal{L}_\lambda$ . Data una funzione continua, possibilmente concava, di cui non si hanno a disposizione le derivate, la BO è in grado di ottimizzare i parametri, all’interno di un iper rettangolo. Nel nostro caso i parametri sono  $\lambda_e$  e  $\lambda_\eta$ . Il vantaggio di utilizzare l’ottimizzazione bayesiana è quello di unire i due punti elencati in precedenza. Per ogni vettore  $\lambda$  troviamo il minimo di  $\mathcal{L}_\lambda$ , ed applichiamo lo smoother di Kalman con il valore delle varianze appena trovate. Ora siamo in grado di calcolare il BIC relativo ad ogni  $\lambda$ , così da avere una funzione  $BIC(\lambda)$ , che sarà la nostra funzione obiettivo da minimizzare.

Si inizia valutando la funzione  $BIC(\lambda)$  in alcuni punti scelti casualmente,  $\lambda_{1:n}$ , nel nostro caso generati secondo un Latin Hypercube Sampling (LHS) (Carnell 2020), per capire come funzioni quest’ultimo si guardi la Figura 1.9. Nel nostro caso specifico abbiamo utilizzato

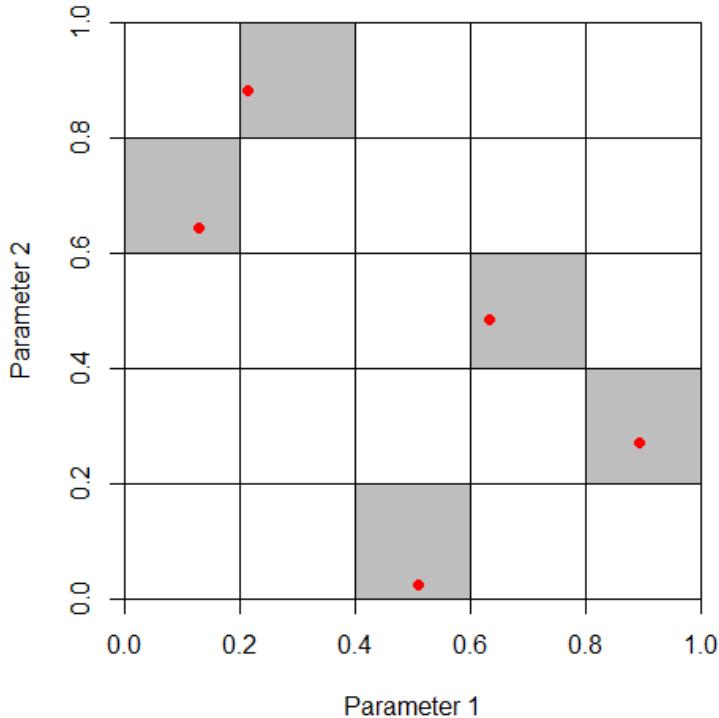


Figura 1.9: Esempio di punti generati secondo LHS. Si divide il rettangolo in una griglia e poi si genera casualmente all’interno dei rettangolini formatosi, facendo in modo che per ogni riga e colonna vi sia un unico punto generato, similmente ai numeri nel sudoku. Immagine presa da (Carnell 2020).

la funzione *maxminlhs*, la quale dopo aver generato la griglia, sceglie i punti all’interno dei rettangolini massimizzando la distanza minima tra i punti.

Dopo questa fase iniziale è necessario modellare la funzione obiettivo, ovvero si ipotizza

$$BIC(\lambda_{1:n}) \sim Normale(\mu(\lambda_{1:n}), \Sigma(\lambda_{1:n}, \lambda_{1:n}))$$

con  $\Sigma$  la funzione kernel. Noi usiamo un kernel gaussiano, quindi

$$\Sigma(\lambda_1, \lambda_2) = s^2 \cdot e^{-\frac{1}{2} \frac{|\lambda_1 - \lambda_2|^2}{r}} + Nu \cdot Id(\lambda_1, \lambda_2) \quad (1.5)$$

con  $s$  e  $r$  dei parametri ottimizzati ad ogni iterazione.  $Nu$  è il parametro di nugget, infatti noi ipotizziamo che la funzione  $BIC(\lambda)$  sia "rumorosa", per far sì che anche punti vicini tra loro possano essere valutati.  $Id(\lambda_1, \lambda_2)$  è la funzione indicatrice, che vale uno se  $\lambda_1 = \lambda_2$  e zero altrimenti.  $\mu$  è la media della normale, a cui assegniamo come prior un vettore di zeri e il suo aggiornamento è dato da

$$\mu(\lambda) = \Sigma(\lambda, \lambda_{1:n})[\Sigma(\lambda_{1:n}, \lambda_{1:n}) + Nu^2 I]^{-1} BIC(\lambda_{1:n}).$$

A questo punto è necessario scegliere il prossimo punto da valutare, per farlo ci basiamo su una funzione di acquisizione, nel nostro caso Lower Confidence Bound:

$$LCB(\lambda) = \mu(\lambda) - \sqrt{\Sigma(\lambda)}.$$

Minimizzando tale funzione si ottiene il punto candidato. Calcoliamo il BIC in esso, così abbiamo aggiunto un ulteriore punto in cui la funzione è stata valutata, quindi se prima avevamo  $\lambda_{1:n}$  ora abbiamo  $\lambda_{1:(n+1)}$ . Ristimiamo la funzione delle medie ed il kernel con il punto aggiuntivo, riminimizziamo la funzione di acquisizione e così via, fino al raggiungimento del numero di iterazioni prefissate.

Prima di applicare l'ottimizzazione bayesiana è importante valutare la superficie su cui andremo a minimizzare il BIC. Infatti essa deve essere "liscia" per ottenere ottimi risultati, altrimenti ci si fermerebbe spesso nei minimi locali, ed una scelta tramite griglia dei parametri sarebbe più efficacie. Inoltre è utile anche confrontare la superficie del BIC con quella del MSE relativo<sup>4</sup>, per valutare l'efficacia del criterio di informazione nel scegliere i migliori valori dei  $\lambda$ . Dalla Figura 1.10 risulta chiaro come non ci siano problemi di minimo locale, dunque l'ottimizzazione bayesiana è un'ottima strada da percorrere, perciò abbiamo implementato tale via.

Sono molteplici le scelte da effettuare, in primis decidere cosa fare nel caso in cui l'algoritmo *BFGS* restituisca errore. Sono state applicate due strategie: la prima consiste nel provare a prevedere le aree in cui non è possibile computare la funzione (Candelieri 2021), attraverso un algoritmo di classificazione basato sulle *support vector machine*, tuttavia in generale, esse sono troppo poche e casuali, quindi non si riesce ad individuarle. La soluzione più ovvia è dunque scartare tali valori. Una seconda strategia adottata consiste nell'applicare l'algoritmo *L-BFGS-B* con dominio fissato nei casi in cui si verifichi l'errore, in modo da restituire comunque un valore del BIC.

Come si evince dalle immagini in Figura 1.11 quest'ultimo metodo ridà i risultati migliori, in quanto si ottiene un BIC minore, per cui decidiamo di usarlo. Nella figura sono

---

<sup>4</sup>il MSE relativo indica il MSE del metodo LODE diviso il MSE di KFS nella stima del trend della serie. Esso sarà descritto per bene nella Sezione 3.2.

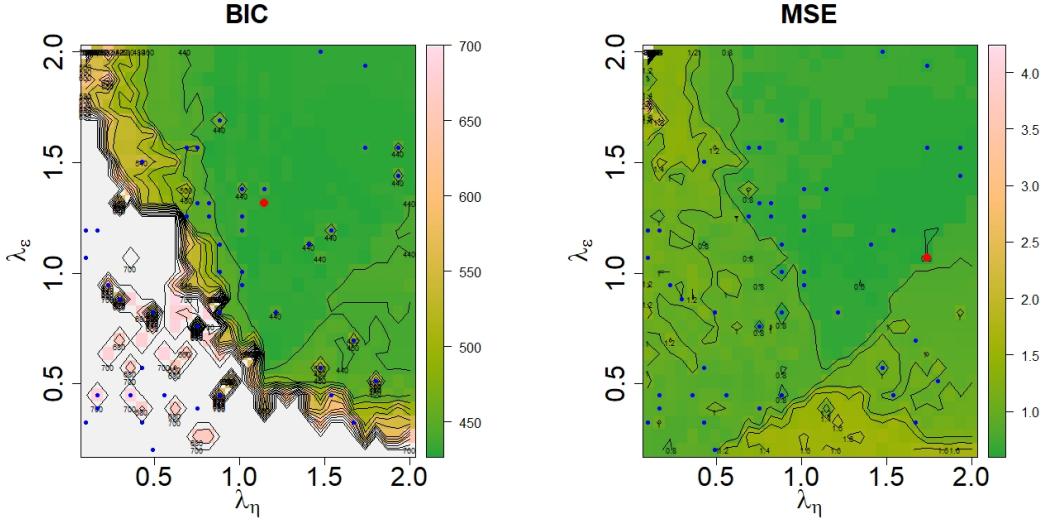


Figura 1.10: Superficie del BIC e del MSE al variare di  $\lambda_\epsilon$  e  $\lambda_\eta$  per una serie storica con un local linear trend. In rosso i punti di minimo delle due superfici, mentre in blu i punti in cui è stato necessario utilizzare *L-BFGS-B*. Nella figura di sinistra si è dovuto limitare superiormente la scala del BIC, in quanto vi erano valori troppo grandi. Le superfici riportate sono state ottenute con una griglia 30x30 di valori per  $\lambda_\epsilon$  e  $\lambda_\eta$ , ricercando dunque outlier additivi e level shift.

riportati gli esiti per diverse funzioni kernel e si nota come siano tutte abbastanza simili. Perciò la scelta ricade sul kernel forse più comune, quello gaussiano, descritto nell’equazione (1.5).

Per usare l’ottimizzazione bayesiana è necessario definire un iperrettangolo che contenga i nostri parametri. Abbiamo deciso di usare sempre valori dei  $\lambda$  compresi tra 0.1 e 2, per i diversi modelli e per i diversi tipi di outlier ricercati, tranne che per outlier sulla stagionalità, riscalando la serie storica in esame in modo opportuno, affinché i valori ottimi di  $\lambda$  cadano nell’iperrettangolo definito, si veda la Sezione 1.4.5 per maggiori dettagli. Ora concentriamoci su altre due scelte da effettuare: il numero di iterazioni totali ed il numero di valori iniziali da generare con LHS. Per quanto riguarda la prima scelta è evidente come più si facciano iterazioni più i risultati saranno ottimi, tuttavia a noi interessa il compromesso prestazioni/tempo. In generale si dice che l’ottimizzazione bayesiana converga dopo circa 15-20 iterazioni per numero di parametri, dunque 30-40 se si considerano outlier additivi e level shift, come faremo ora. I risultati confermano tale teoria, abbiamo notato come andare oltre le 50 iterazioni non è efficiente, in quanto il guadagno è minimo. Dunque ogni valore  $\leq 50$  è il benvenuto. Noi in linea di massima considereremo, oltre a 50, anche 30 iterazioni, un valore che porta ad un buon compromesso in termini di costo computazionale, portando a risultati spesso identici a quelli ottenuti con 50 iterazioni. La differenza di BIC ottenuto risiede solo nel alcune serie storiche, circa il 25%, nelle altre il valore restituito è identico. Nelle nostre simulazioni andremo a considerare anche il caso di ricerca di un unico tipo di outlier, come cambio di livello o pendenza o stagionalità, dunque useremo un unico  $\lambda$ . Ciò

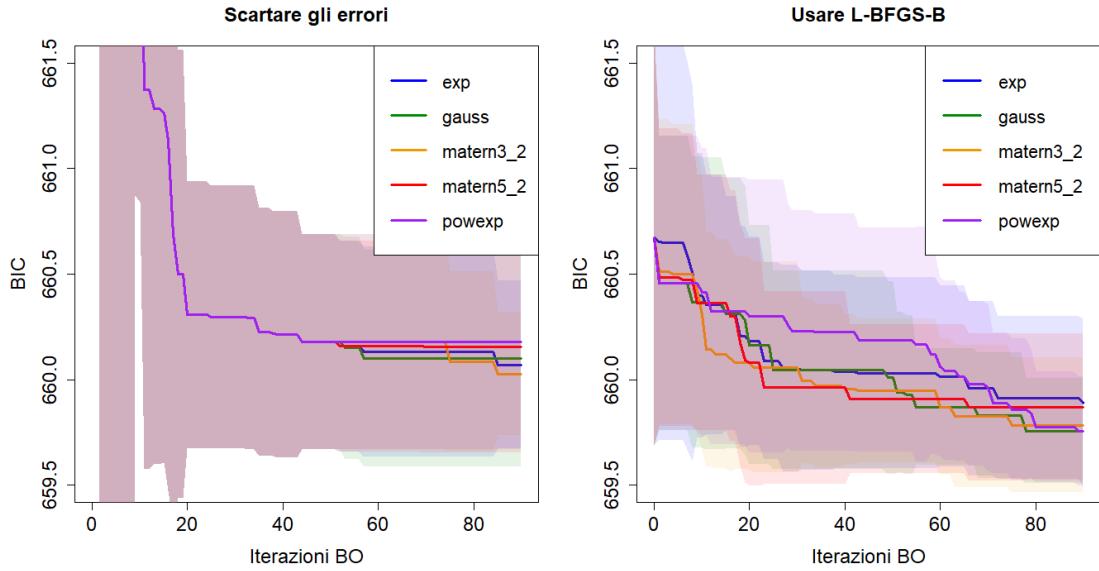


Figura 1.11: Risultati dell’ottimizzazione bayesiana per un esempio di serie storica. A sinistra prevedendo e scartando le aree ove l’algoritmo *BFGS* restituisce errore, a destra usando *L-BFGS-B* in quei casi. Sono riportati gli esiti ottenuti usando diversi kernel e l’area colorata rappresenta la media più o meno la deviazione standard delle stime che è stata ottenuta ripetendo più volte l’analisi, usando diversi valori iniziali.

riduce enormemente il costo computazionale, avendo un solo parametro da ottimizzare. In questi casi 15 iterazioni sono più che sufficienti.

Un’altra importante scelta da effettuare riguarda il numero di valori iniziali da generare con LHS. Considerando ancora il caso con due  $\lambda$  abbiamo notato come un numero abbastanza alto, come 15, portasse ai risultati migliori. Valori limitrofi portano comunque a risultati molto simili. Si è deciso di utilizzare sempre  $50/3$  valori iniziali, arrotondati a 17, sia con 50 che 30 iterazioni, poiché si è notato come 17 portasse a risultati apparentemente un poco migliori rispetto a  $30/3=10$ . Inoltre in futuro quando riporteremo i risultati per 30 iterazioni, essi sono stati quasi sempre ottenuti considerando i primi 30 cicli di una ottimizzazione che si sarebbe fermata a 50, questo per ridurre i tempi computazionali delle simulazioni, evitando di lanciare due volte l’ottimizzazione bayesiana. In questa parte finale del paragrafo non abbiamo riportato precisi risultati di simulazioni, questo perché l’analisi è stata effettuata su più simulazioni, in contemporanea con l’intero sviluppo dell’algoritmo.

**BO vs BO vs Griglia** La BO è un metodo di ottimizzazione stocastico, in quanto dipende da come sono stati generati i punti iniziali, perciò abbiamo controllato come variano i risultati applicando più volte il medesimo algoritmo. Inoltre l’ottimizzazione bayesiana è un approccio che dovrebbe ridurre il tempo computazionale rispetto ad una ricerca tramite griglia, in questo paragrafo andiamo a verificarlo.

Abbiamo preso in considerazione 10 serie storiche con outlier additivi e cambi di livello ed abbiamo applicato l’ottimizzazione bayesiana 10 volte per ciascuna serie, variando

i punti iniziali e poi confrontato i risultati con la griglia. Per la BO abbiamo usato 30 e 50 iterazioni, mentre per la griglia  $8 \times 8 = 64$ . Non abbiamo usato lo stesso numero di iterazioni per svariati motivi: il nostro parametro di interesse è il rapporto precisione/tempo, dunque è vero che numero di iterazioni e tempo sono fortemente correlati, ma ci sono altri fattori che li influenzano. L’ottimizzazione bayesiana ad ogni ciclo deve ottimizzare la funzione di acquisizione, ciò richiede tempo, che seppur poco, ripetuto per ogni iterazione diventa rilevante. Questo soprattutto se il numero di instanti temporali è basso, per esempio 100, come nelle serie che abbiamo testato, poiché più è lunga la serie, più parametri da ottimizzare ci sono e dunque la ricerca del minimo della funzione  $\mathcal{L}$  è più lenta, mentre rimane costante il tempo per ottimizzare la funzione di acquisizione. Nella ricerca tramite griglia, quando l’algoritmo *BFGS* ridava errore, ci siamo limitati a scartare quel punto, dunque non abbiamo utilizzato *L-BFGS-B* come nella BO, ciò riduce il numero di iterazioni realmente effettuate. Come ultimo parametro da tenere in considerazione vi è il calcolo in parallelo. Un ricerca tramite griglia può essere effettuata interamente in parallelo su più core del computer, mentre per la BO ciò può essere fatto sono per la valutazione dei punti iniziali, generati con LHS, dopo diventa un processo sequenziale. Perciò a seconda di come si usa il calcolo parallelo e della potenza computazionale a disposizione, varia il tempo dei due approcci, rendono difficile un confronto generale. Sottolineiamo che nel caso in cui si volesse abbattere il tempo computazionale e si avessero a disposizione numerosi core sul proprio computer, allora il calcolo parallelo aumenta notevolmente le prestazioni, di conseguenza una ricerca tramite griglia con numerose iterazioni è comunque molto più veloce. Tuttavia, volendo ciò è molto simile ad usare una BO ove poniamo il numero di valori iniziali, pari al numero di iterazioni, così da effettuare una random search interamente in calcolo parallelo, in modo tale da ridurre il tempo computazionale.

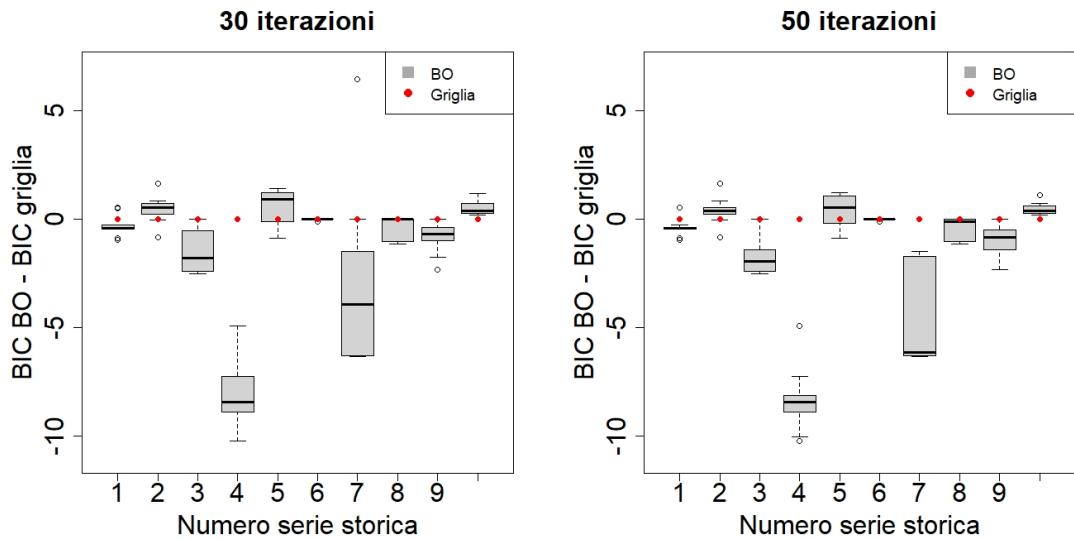


Figura 1.12: Risultati dell’ottimizzazione bayesiana, ripetuta 10 volte, rispetto ad una ricerca tramite griglia. Sulle ordinate la differenza di BIC tra la BO e la griglia. In grigio i boxplot della BO ed in rosso i BIC ottenuti con la griglia, che ovviamente sono tutti pari a zero.

Nella Figura 1.12 sono rappresentati i box plot dei BIC per le varie serie storiche. Si nota come in certe serie storiche siano quasi una riga piatta, indice del fatto che diversi punti iniziali per la BO portano al medesimo risultato. Ciò è confortante, in quanto mostra come possiamo stare tranquilli ad utilizzare un unico lancio dell’ottimizzazione bayesiana. Tuttavia nella serie 4 e 7, dove vi è la maggior differenza in termini di BIC rispetto alla griglia, vi è anche la maggior variabilità. Notiamo come i box plot siano molto allungati, indice del fatto che queste serie sono influenzate dai differenti punti iniziali della BO.

Concentriamoci ora sulla griglia, sui punti rossi, si nota come i risultati siano un poco peggiori, anche rispetto ad una BO con 30 iterazioni, infatti in media si ha una differenza di BIC di 1.2, comunque non trascurabili se si vuole ottenere la migliore performance.

#### 1.4.5 Riscalamento della serie

Come già anticipato per fare sì che i valori ottimi dei  $\lambda$  cadano sempre nell’intervallo 0.1-2 è necessario riscalare la serie in analisi. Nelle prime simulazioni si è sempre considerato un valore di  $\sigma_\epsilon$  compreso tra 1 e 5, generando le altre componenti su questo ordine di grandezza. Si è verificato che con tale impostazione 0.1-2 comprendeva sempre i valori ottimi dei  $\lambda$ , ovvero raramente veniva selezionato  $\lambda = 0.1$  o peggio  $\lambda = 2$  con alcuni parametri diversi da zero, infatti ricordiamo che è bene che il nostro algoritmo provi sempre il modello con le varianze aggiuntive tutte pari a zero, e per fare ciò è necessario tentare con un  $\lambda$  sufficientemente grande. Per tutti i motivi sopra elencati abbiamo deciso di utilizzare il seguente schema per riscalare le serie storiche:

$$y'_t = \frac{(y_t - y_1) \cdot 5}{\hat{\sigma}_{\epsilon KFS}}$$

con  $\hat{\sigma}_{\epsilon KFS}$  la radice della varianza della parte additiva stimata dal filtro di kalman classico. Nel riscalamento si è deciso di porre  $y'_1 = 0$  per evitare problemi nella scelta del valore di inizializzazione del primo punto. A volte però capita di ottenere  $\hat{\sigma}_{\epsilon KFS} \approx 0$ , ovvero la serie storica viene stimata come pure segnale. Questo crea degli ovvi problemi, per cui in questi casi abbiamo usato  $\hat{\sigma}_{\eta KFS}$  al denominatore.

Nella Figura 1.13 si nota come a valori di  $\lambda$  più bassi sono associati un maggior numero di varianze aggiuntive diverse da zero, mentre quando  $\lambda \approx 2$  le varianze sono poste tutte pari a zero. La trasperanza dei punti ci permette di osservare come vi siano punti sovrapposti, ciò è dovuta al fatto che quei valori di  $\lambda$  sono quelli scelti nella fase iniziale di generazione punti casuale secondo LHS, e poiché si fissa un seme generatore, i valori di  $\lambda$  generati sono sempre i medesimi.

#### 1.4.6 Definizione dei gradi di libertà

Per usare un qualunque criterio di informazione è necessario andare a definire i gradi di libertà. La teoria usata nella LASSO, descritta in (Zou, Hastie e Tibshirani 2007), si basa su dati indipendenti, noi però trattiamo serie storiche, quindi ciò non è valido, per questo non possiamo utilizzare tale teoria. Purtroppo non ne esiste ancora una che tratti tale problematica, per cui abbiamo definito delle quantità sensate da un punto di vista logico, rifacendoci alla teoria delle regressioni, e che fossero ottime anche dal punto di vista numerico nelle simulazioni.

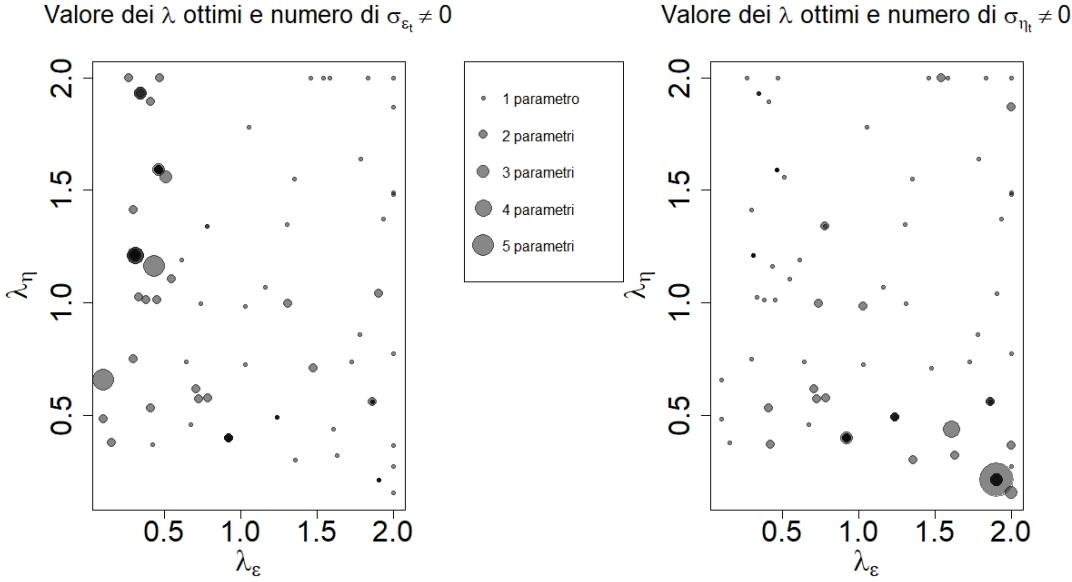


Figura 1.13: Distribuzione dei valori di  $\lambda_\epsilon$  e  $\lambda_\eta$ , per alcune serie storiche dopo il riscalamento. La dimensione dei punti è proporzionale al numero di varianze aggiuntive poste diverse da zero, per le singole componenti, a sinistra il numero di outlier additivi individuati, a destra il numero di cambi di livello. I valori dei  $\lambda$  rappresentati sono quelli ottimali ottenuti nella simulazione della Sezione 3.3.3.

Se la minimizzazione non fosse vincolata i gradi di libertà sarebbero pari al numero di parametri, tuttavia nel nostro caso la penalità è molto forte. I nostri gradi di libertà saranno pari al numero di parametri "diversi da zero", ovvero quelli che hanno una qualche influenza nei calcoli delle varianze totali,  $\sigma_{\epsilon_t}^2$  e  $\sigma_{\eta_t}^2$ . Esse sono la somma di una varianza costante e dei nostri parametri aggiuntivi. Se per esempio  $\dot{\sigma}_{\eta_{t_0}} \approx 0$ , allora varrà  $\sigma_{\eta_{t_0}}^2 = \sigma_\eta^2 + \dot{\sigma}_{\eta_{t_0}}^2 \approx \sigma_\eta^2$ . Dunque possiamo non considerare come gradi di libertà quei parametri per cui vale  $\dot{\sigma}_{\eta_{t_0}}^2 \ll \sigma_\eta^2$ . Come soglia noi consideriamo  $\dot{\sigma}_{\eta_{t_0}} = \frac{\sigma_\eta}{100}$ , ovvero consideriamo il parametro come grado di libertà se maggiore di questa soglia. Le somme nella matrice sono in quadratura, quindi avremo

$$\sigma_{\eta_{t_0}}^2 = \sigma_\eta^2 + \dot{\sigma}_{\eta_{t_0}}^2 = \sigma_\eta^2 + \frac{1}{10^4} \sigma_\eta^2 \approx \sigma_\eta^2.$$

Si è notato che non vi sono grosse differenze se si sostituisce il 100 con valori più grandi o più piccoli, in quanto in genere i parametri o sono maggiori di  $\sigma_\eta$  oppure sono molto vicini allo zero. La formula per il calcolo dei gradi di libertà è dunque:

$$df = \sum_t \left( \dot{\sigma}_{\eta_t} > \frac{\sigma_\eta}{100} \right) + \sum_t \left( \dot{\sigma}_{\epsilon_t} > \frac{\sigma_\epsilon}{100} \right) + \left( \sigma_\eta > \frac{\sigma_\eta}{100} \right) + \left( \sigma_\epsilon > \frac{\sigma_\epsilon}{100} \right). \quad (1.6)$$

Un problema che sorge subito spontaneo è quando  $\sigma_\eta \approx 0$ , come nella serie storica del Nilo. In questo caso alcuni parametri aggiuntivi, che seppur sono molto vicini allo zero

potrebbero essere contati come gradi di libertà. Inoltre usando la soglia appena descritta risulterà che  $\sigma_\eta$  è sempre conteggiato come parametro. Ciò è corretto, in quanto esso non rientra nella penalizzazione, dunque è libero di variare, tuttavia non esistendo una teoria affidabile a cui rifarsi, possiamo provare a non fissarlo come grado di libertà per vedere cosa accade.

Per questi motivi proponiamo alcune varianti della formula descritta in precedenza. Bisogna identificare un'altra soglia da usare quando  $\sigma_\eta$  è "piccolo". La stima di  $\hat{\sigma}_\eta$  ridata dal filtro di Kalman classico (KFS), che indicheremo con  $\hat{\sigma}_{\eta KFS}$ , può tornare utile, infatti essa sarà sempre diversa da zero, nei casi da noi analizzati, altrimenti la serie sarebbe puro rumore, se parliamo di random walk. Utilizzare una piccola frazione di essa può essere un'idea, come  $\hat{\sigma}_{\eta KFS}/1000$ , o anche fratto 100,10000,..., come in precedenza una scelta o l'altra non influenza molto i risultati. Detto ciò possiamo proporre due varianti della formula precedente:

- andare a cambiare soglia per tutti i parametri e quindi utilizzare

$$df = \sum_t \left( \dot{\sigma}_{\eta t} > \max \left( \frac{\sigma_\eta}{100}, \frac{\hat{\sigma}_{\eta KFS}}{1000} \right) \right) + \sum_t \left( \dot{\sigma}_{\epsilon t} > \frac{\sigma_\epsilon}{100} \right) + \\ + \left( \sigma_\eta > \max \left( \frac{\sigma_\eta}{100}, \frac{\hat{\sigma}_{\eta KFS}}{1000} \right) \right) + \left( \sigma_\epsilon > \frac{\sigma_\epsilon}{100} \right).$$

- cambiare la soglia solo per  $\sigma_\eta$

$$df = \sum_t \left( \dot{\sigma}_{\eta t} > \frac{\sigma_\eta}{100} \right) + \sum_t \left( \dot{\sigma}_{\epsilon t} > \frac{\sigma_\epsilon}{100} \right) + \\ + \left( \sigma_\eta > \max \left( \frac{\sigma_\eta}{100}, \frac{\hat{\sigma}_{\eta KFS}}{1000} \right) \right) + \left( \sigma_\epsilon > \frac{\sigma_\epsilon}{100} \right). \quad (1.7)$$

Le due scelte risultano molto simili da un punto di vista dei risultati. In quanto, come già detto, i parametri  $\dot{\sigma}_{\eta t}$  non influenti nella stima della serie sono sempre molto vicini allo zero, più piccoli di  $\hat{\sigma}_{\eta KFS}/1000$ . Tutt'altro discorso per  $\sigma_\eta$ , esso può benissimo essere sia inferiore che maggiore di  $\hat{\sigma}_{\eta KFS}/1000$ , dunque la soglia diversa può influenzare i risultati. Non imponendolo come grado di libertà si avvantaggia una descrizione di un trend a gradini, in cui  $\hat{\sigma}_\eta = 0$ . Ciò porta ad ovvi vantaggi quando si analizzano serie di questo tipo, dove il modello reale è proprio fatto così, tuttavia quando si analizzano serie con  $\sigma_\eta \neq 0$  il criterio corretto può portare con una maggioranza facilità a scegliere un modello a gradini anche quando non ce n'è bisogno.

La formula (1.7) per calcolare i gradi di libertà porta alla definizione di un BIC corretto, che chiameremo BICC, mentre quando parleremo di BIC ci staremo riferendo alla formula (1.6). Poiché da un punto di vista teorico è più corretto imporre sempre  $\sigma_\eta$  come grado di libertà ed essendo i miglioramenti dei risultati non uniformi su tutti i tipi di serie storiche, decidiamo di usare in via definitiva la formula (1.6), quella senza correzioni.

Le formule per il calcolo dei gradi di libertà quando andiamo alla ricerca di cambi di pendenza o di stagionalità possono essere trovate nell'Appendice A.2.

**In sintesi** Il nostro obiettivo è individuare valori anomali nelle serie storiche, principalmente outlier additivi e cambi di livello. Per fare questo poniamo delle varianze aggiuntive in ogni istante temporale e poi selezioniamo le più importanti con una sorta di regressione LASSO, il cui ottimo è trovato dall'algoritmo *BFGS*. Il parametro  $\lambda$  è scelto minimizzando il BIC attraverso l'ottimizzazione bayesiana.

# Capitolo 2

## Metodo basato sui residui ausiliari

Quando si sviluppa un nuovo metodo, è bene confrontarlo sempre con quelli già esistenti. Abbiamo deciso di prendere come benchmark un modello che basa l'individuazione degli outlier sui residui ausiliari. Lo chiameremo *auxres*.

### 2.1 Funzionamento di auxres

Prima di poter confrontare il nostro metodo con auxres è necessario comprendere come funziona quest'ultimo. Esso si basa sull'analisi dei residui standardizzati e riprende grosso modo il metodo illustrato in (Harvey e Koopman 1992). Grazie al filtro di Kalman, in particolare allo *disturbance smoother* siamo in grado di ricostruire i valori dei disturbi  $\epsilon_t, \eta_t$ , ecc... con le rispettive varianze. Di conseguenza siamo in grado di standardizzare i disturbi, o residui, com'è meglio chiamarli in questo caso. Su di essi pende l'ipotesi di gaussianità, di conseguenza una volta normalizzati essi si dovrebbero distribuire secondo una  $N(0,1)$ . Un'analisi della loro distribuzione può essere utile per diversi scopi. Il primo riguarda la costruzione di test per la normalità della serie. In secondo luogo essi sono indicatori della presenza di disturbi nella serie, valori troppi grandi a volte possono essere spiegati considerando fenomeni esterni. Per una descrizione più formale dei residui standardizzati si può consultare pag. 154 di (Pelagatti 2016).

Per la costruzione dell'algoritmo di auxres abbiamo proceduto nel seguente modo. Illustriamo la procedura per un modello random walk, ma è analogo negli altri casi. Andando alla ricerca di outlier sia sulla parte additiva che sul trend, fissiamo in prima battuta un valore soglia, oltre il quale consideriamo che è necessario introdurre una variabile esterna per spiegare il fenomeno. Prendiamo il 99.5esimo percentile della normale,  $\approx 2.5$ , in modo da considerare un livello di confidenza del 99% per i nostri residui, oltre il quale sono considerati non spiegabili dal modello. Sottolineiamo che qua stiamo considerando sempre il valore assoluto dei residui. Il valore soglia è stato scelto per ottenere un giusto compromesso tra false e true positive rate.

Una volta individuati i residui anomali andiamo a costruire delle variabili esterne, della forma più opportuna, per spiegare meglio il fenomeno. Se si tratta di un level shift poniamo

una variabile a gradino, se è un cambio di pendenza aggiungiamo al modello una rampa, se si tratta di un outlier additivo poniamo una variabile ad impulso. La Figura 2.1 illustra quanto detto. Per ulteriori informazioni si veda pag.76 di (Pelagatti 2016).

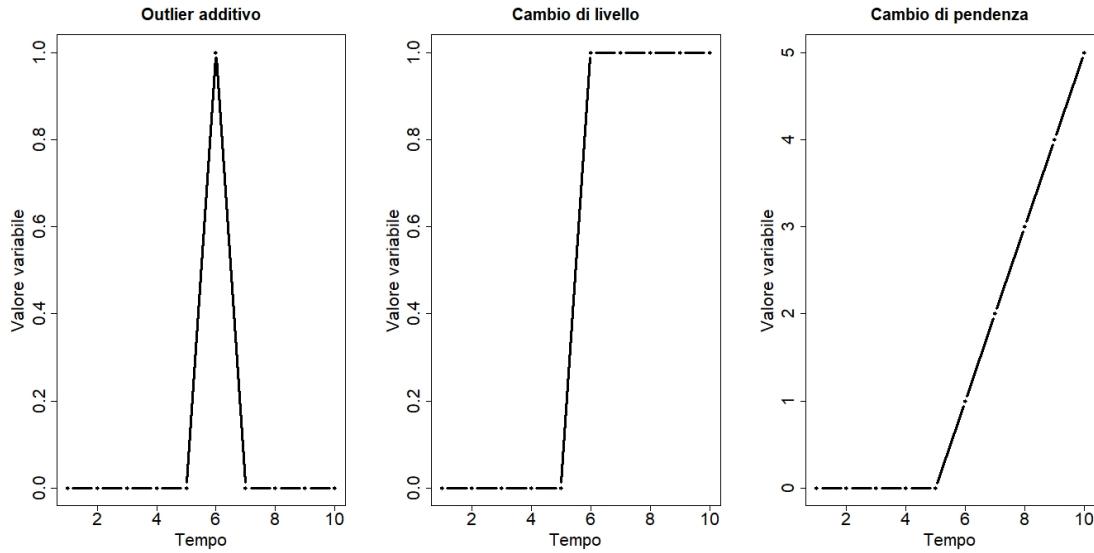


Figura 2.1: Forma delle variabili esterne per differenti tipi di outlier.

Un problema presente nell’analisi dei residui riguarda la forte correlazione tra di essi, in quanto si utilizzano i medesimi dati per costruirli. Dunque può capitare che in presenza di un jump, più residui contigui presentino valori elevati, anche se l’outlier è solo in un istante. Tale problema raggiunge il suo apice quando ricerchiamo cambi di pendenza. Per risolvere questa problematica possiamo considerare solo l’istante temporale con il residuo più grande, per un esempio si veda la Figura 2.2. Non ripetiamo l’operazione per la parte additiva, in quanto è plausibile la presenza di più outlier additivi contigui. Un’altra problematica riguarda la correlazione tra i residui della parte additiva e del trend. Capita a volte che un jump provochi la presenza di valori anomali anche nei residui additivi, e viceversa. Non potendo a priori dire su quale componente sia presente l’outlier, poniamo entrambe le variabili esterne.

Queste presentate sono le caratteristiche base del metodo, per ulteriori informazioni si può consultare l’Appendice B.

## 2.2 Software STAMP

Il software STAMP(Structural Time Series Analyser, Modeler and Predictor) è un programma ad interfaccia grafica appartenente alla grande famiglia di *OxMetrics* (Mendelsohn 2011). Esso ha una grande varietà di utilizzi poiché permette la definizione di un gran numero di modelli. Noi andremo ad analizzarlo nel nostro ambito, ovvero nella robustificazione del filtro di Kalman. Lo trattiamo poiché contiene una rilevazione automatica degli outlier, con cui confrontare il nostro metodo. Koopman, il creatore del software STAMP, ha per primo proposto di ricercare outlier tramite i residui ausiliari (Harvey e Koopman 1992),

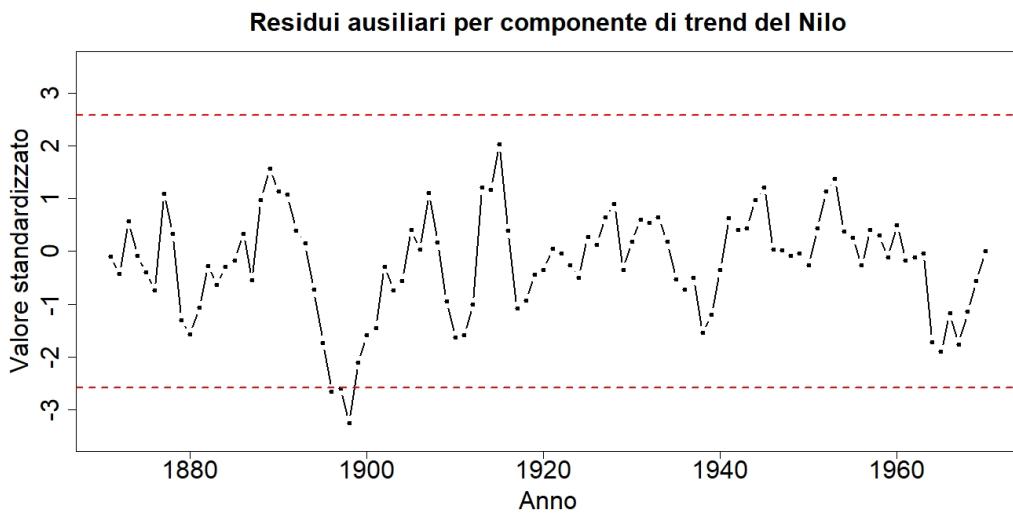


Figura 2.2: Residui ausiliari per la componente di trend nella serie storica del Nilo. Si può notare come in questo caso vi siano ben tre residui contigui al di fuori delle bande di confidenza del 99%, provocati da un unico cambio di livello. Si considera solo il terzo come outlier, in quanto è il più grande.

per questo motivo, molto probabilmente anche l'algoritmo implementato in STAMP si basa su di essi. Tuttavia l'algoritmo esatto non è reso noto per motivi commerciali. Comunque si è verificato che *auxres* fornisce una discreta approssimazione di ciò che fa, ma andremo in ogni caso ad analizzare serie storiche direttamente con il software, in un'apposita sezione.

# Capitolo 3

## Applicazione e risultati

In questo capitolo tratteremo tutte le varie applicazioni ed i risultati che sono stati ottenuti. Sono state provate numerose serie storiche di differenti tipologie, applicando differenti metriche per la valutazione delle performance, per cui si è scelto di descrivere simulazione dopo simulazione illustrando subito i risultati.

### 3.1 Simulazione delle serie storiche

A noi ovviamente non interessa più di tanto analizzare serie puramente gaussiani, ma dobbiamo simulare anche i valori anomali, sia sul trend che sulla parte additiva. Per fare questo ci sono molto metodi, per completezza ne abbiamo provati diversi, come simulare da mistura di gaussiane, da t-student, oppure impostare posizione e grandezza fissate a priori.

Abbiamo generato diversi set di serie storiche e le abbiamo analizzate sia con LODE sia con *auxres*. Il principale metodo usato per la simulazione di outlier è la mistura di gaussiane. In linea di massima, quando non abbiamo definito, o non definiremo in seguito il metodo di generazione delle serie è sottinteso che è stato utilizzato proprio questo approccio, almeno per quanto riguarda outlier additivi e cambi di livello, non per cambi di pendenza e di stagionalità.

Per simulare outlier da una mistura di gaussiane generiamo  $n$  numeri  $\epsilon_i$  tali che

$$\epsilon_i \sim N(0, \sigma_\epsilon^2)$$

e altri  $n$  papabili outlier da

$$\tilde{\epsilon}_i \sim N(0, \sigma_\epsilon^2 \cdot c^2)$$

con  $c \sim Uniforme(2,6)$ . La distribuzione di  $c$  è molto importante, perché descrive quanto sono grandi gli outlier e ciò influenza i risultati finali. Noi abbiamo optato per una tale distribuzione al fine di rendere il più realistiche possibile le serie simulate, dunque evitando di generare outlier giganteschi, visibili a occhio nudo, o minuscoli ed impossibili da individuare.

Dopodiché si seleziona un solo disturbo per istante temporale trai due generati, con probabilità 0.9 si seleziona  $\epsilon_i$  e con 0.1  $\tilde{\epsilon}_i$ . In questo modo circa il 10% dei disturbi possono essere outlier, diciamo "possono" perché non sempre generando da una gaussiana con ampia varianza, si estraggono valori "grandi", considerabili come outlier. Lo stesso si fa per la parte

di trend,  $\eta$ . Nelle serie simulate il rapporto segnale rumore, ovvero  $\sigma_\epsilon/\sigma_\eta$  è sempre compreso tra 0.5 e 2, più precisamente

$$\frac{\sigma_\epsilon}{\sigma_\eta} \sim \frac{U(1,4)}{2}.$$

Infine aggiungiamo al primo istante del trend un valore casuale, in modo che la serie non oscilli sempre attorno allo zero.

Quando presenti senza outlier, le componenti di slope e stagionalità le abbiamo generate nel seguente modo:

$$\begin{aligned}\beta_0 &\sim U(-5,5) \cdot \sigma_\epsilon / 3 \\ \beta_t &= \beta_{t-1} + \zeta_i \cdot \sigma_\epsilon / 3 \quad t = 1, \dots, n\end{aligned}$$

con  $\zeta_i \sim N(0, \sigma_\zeta^2)$  e  $\sigma_\zeta \sim U(0.2, 0.8)$ .

Per la stagionalità, chiamando  $T$  il periodo abbiamo generato:

$$\begin{aligned}S_t &\sim U(-5,5) \cdot \sigma_\omega \quad t = 1, \dots, T-1 \\ S_t &= \sum_{i=1}^{T-1} S_{t-i} + \omega_t \cdot \sigma_\omega \quad t > T-1\end{aligned}$$

con  $\omega_t \sim N(0, \sigma_\omega^2)$  e  $\sigma_\omega \sim U(0.2, 0.8)$ .

Nelle prossime pagine saranno mostrati esempi di serie simulate, ora riportiamo quanto descritto tradotto in codice R riproducibile, trovabile anche al seguente link [Sim\\_serie](#)<sup>1</sup>.

---

```

1 n<-100
2 sig_eps <- runif(1,1,100)
3 sig_eta <- runif(1,1,4)*sig_eps/2
4 c_eps <- runif(1,2,6)
5 c_eta <- runif(1,2,6)
6 disturbi_eps<-cbind(rnorm(n,0,sd=sig_eps),rnorm(n,0,sd=c_eps*sig_eps))
7 eps<-apply(disturbi_eps, 1, sample, prob=c(0.9,0.1),size=1)
8 distrurbi_eta<-cbind(rnorm(n,0,sd=sig_eta),rnorm(n,0,sd=c_eta*sig_eta))
9 eta<-apply(distrurbi_eta, 1, sample, prob=c(0.9,0.1),size=1)
10 sig_beta<-runif(1,0.2,0.8)
11 b<-cumsum(rnorm(n, sd = sig_beta))+runif(1,-5,5)
12 b<-b*sig_eps/3
13 periodicit<-4
14 stg<-numeric()
15 sig_stg<-runif(1,0.2,0.8)
16 stg[1:(periodicit-1)]<-runif(periodicit-1,-5,5)
17 for(i in periodicit:n){
18   stg[i]<-sum(stg[(i-periodicit+1):(i-1)])+rnorm(1,0,sig_stg)
19 }
20 stg<-stg*sig_eps
21 mu <- cumsum(eta+b)+runif(1,-100*sig_eps,100*sig_eps)
22 y <- mu + eps + stg

```

---

<sup>1</sup>[https://github.com/simone-frigerio/LODE/blob/main/sim\\_llt\\_stg.R](https://github.com/simone-frigerio/LODE/blob/main/sim_llt_stg.R)

In generale, alcuni codici utilizzati si possono trovare sulla pagina [github](#) di Simone Frigerio, nella cartella **LODE**: <https://github.com/simone-frigerio/LODE>.

## 3.2 Considerazioni sulla stima del trend della serie

L'applicazione più importante del metodo LODE riguarda la stima attraverso lo smoother del trend della serie.

Per tutta questa parte la metrica di valutazione è stata l'errore quadratico medio (MSE), tra il vero trend della serie e la sua stima ottenuta coi vari metodi. Questi ultimi sono tre: LODE, *auxres* e il filtro di kalman classico (KFS). Quest'ultimo è stato preso come benchmark delle performance. Perciò non si è quasi mai andato a valutare il MSE in generale, ma sempre quello relativo al metodo classico, ovvero MSE ottenuto diviso MSE classico, in tal modo il MSE relativo di KFS è sempre pari ad uno. Questo è stato fatto per rendere più facile la media tra le varie serie simulate, altrimenti ci sarebbe stato un problema di dimensionalità. Inoltre il MSE relativo è di facile interpretazione, un valore minore di uno è ottimo, mentre se maggiore di uno significa che non migliora le performance del metodo classico. Noi parleremo di MSE e MSE relativo in egual modo, riferendoci sempre a quest'ultimo.

Per le varie simulazioni sono stati considerati sottoinsiemi diversi delle serie generate, per valutare sotto diversi aspetti i modelli: in primo luogo si sono considerate tutte le serie generate nella medesima simulazione e si è calcolata la media degli MSE relativi. Indicheremo tutte le serie con il termine *tutte*.

In secondo luogo abbiamo considerato solo le serie ove sia il nostro metodo, sia *auxres* hanno un  $MSE < 0.95$ , ovvero entrambi i metodi performano bene. Questo per concentrarci sulle serie ove i metodi di irrobustimento andrebbero realmente applicati. Le serie che soddisfano questo requisito sono state chiamate  $MSE < 0.95$ .

In una vera applicazione pratica, prima di applicare un metodo di irrobustimento si fa una attenta analisi preliminare per stabilire se la serie presenta o meno outlier. Per simulare velocemente questo fatto, abbiamo applicato il filtro classico e svolto il test di Shapiro-Wilk (SW) sui residui standardizzati delle varie componenti su cui cerchiamo outlier, per capire se hanno una distribuzione normale, come da teoria, o meno, come accade nel caso di presenza di outlier. Una volta effettuato il test abbiamo selezionato le serie ove almeno uno dei p-value dei test era minore del 5%. Chiameremo questo gruppo di serie  $SW < 5\%$ . Questa è forse la categoria più importante tra le tre, in quanto, quando si simulano un gran numero di serie con outlier, non è detto che in tutte vi siano realmente presenti. Fare una semplice analisi preliminare di questo tipo permette di scartare la maggior parte di esse, per evitare una contaminazione da serie senza outlier, nei risultati.

## 3.3 Risultati utili per comprendere i risultati

Per comprendere al meglio tutti i vari risultati riguardanti la stima del trend della serie è necessario osservare altri risultati, che possono essere meno importanti da un punto di vista applicativo, ma essenziali per comprendere il funzionamento del metodo LODE.

### 3.3.1 Esempio di una stima del trend

Andiamo ad analizzare un esempio di trend stimato con il metodo LODE, confrontandolo con la stima ottenuta dal filtro classico, per capire in che modo modellizzare gli outlier

possa aiutare a ridurre il MSE. Consideriamo una serie storica di 100 istanti con un outlier additivo e un jump.

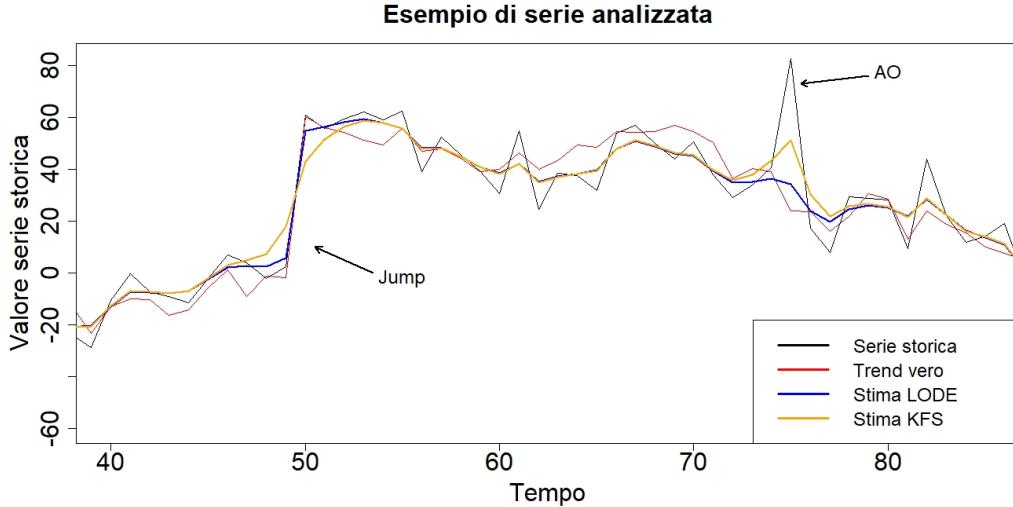


Figura 3.1: Esempio di serie analizzata con il metodo LODE. L’immagine è limitata alla parte della serie ove vi sono outlier, nel resto le stime di LODE e KFS sono molto simili.

La Figura 3.1 mette in risalto il confronto tra le stime nella zona interessata, ove ci sono gli outlier. Concentriamoci sull’outlier additivo (AO). Il trend stimato dal filtro classico asseconda molto l’andamento della serie storica, distanziandosi dal trend vero e provocando un notevole errore nella stima. Invece il metodo LODE capisce che si tratta di un valore anomalo e la sua stima non è influenzata da esso. Osserviamo ora il jump. KFS tende a lasciare il salto, "smussando" gli angoli, invece LODE individua il jump e riesce a seguire il reale andamento del trend. Capiamo dunque in che modo individuare e modellare gli outlier porta ad un errore inferiore nella stima del trend. Per esempio, nella serie raffigurata il MSE relativo del metodo LODE è di 0.684, ovvero

$$MSE_{relativo} = \frac{MSE_{LODE}}{MSE_{KFS}} = 0.684.$$

### 3.3.2 Serie gaussiane pure

Le prime serie che andiamo ad analizzare sono paradossalmente quelle dove non è necessario applicare il metodo, serie storiche puramente gaussiane. Dalla teoria si dimostra che il filtro di kalman classico è il miglior stimatore possibile per il trend e le altre componenti, nel caso in cui la serie soddisfi le ipotesi di gaussianità dei disturbi. Tuttavia non si potrà mai avere la certezza di ciò, dunque può essere interessante vedere le prestazioni applicate ad una serie a cui non si dovrebbe applicare il metodo, in quanto non vi sono outlier, ma per nostro errore lo facciamo comunque. Perciò sono stati creati RW puramente gaussiani, ed analizzati con i vari metodi.

Il modello che andremo ad utilizzare è il seguente:

$$\begin{aligned} y_t &= \mu_t + \epsilon_t & \epsilon_t &\sim NID(0, \sigma_\epsilon^2 + \dot{\sigma}_{\epsilon_t}^2) \\ \mu_{t+1} &= \mu_t + \eta_t & \eta_t &\sim NID(0, \sigma_\eta^2 + \dot{\sigma}_{\eta_t}^2). \end{aligned} \quad (3.1)$$

Mentre la funzione da minimizzare è:

$$\mathcal{L}_{\lambda_\epsilon, \lambda_\eta}(\theta) = -\log(f(y, \theta)) + \lambda_\epsilon \sum_{t=1}^n |\dot{\sigma}_{\epsilon_t}| + \lambda_\eta \sum_{t=1}^n |\dot{\sigma}_{\eta_t}|$$

con  $\theta = (\sigma_\epsilon, \dot{\sigma}_{\epsilon_1}, \dots, \dot{\sigma}_{\epsilon_n}, \sigma_\eta, \dot{\sigma}_{\eta_1}, \dots, \dot{\sigma}_{\eta_n})$ .

Sono state generate 100 serie, tuttavia due sono state scartate per problemi di convergenza del filtro classico, poiché andava a porre  $\hat{\sigma}_{\epsilon KFS} = 0$ , ciò ovviamente porta ad un grande MSE, oltre a rendere impossibile il calcolo dei residui ausiliari additivi. Questi sono solo problemi di convergenza, in quanto, sulle medesime serie, applicando il nostro algoritmo, andava a porre a zero tutte le varianze aggiuntive, nonostante ciò si aveva un  $\hat{\sigma}_{\epsilon KFS} \neq 0$ . In altre serie si è notato un fenomeno simile, ovvero che nonostante le varianze aggiuntive fossero tutte zero, la nostra stima era diversa rispetto a quella del filtro classico. Ciò è dovuto sia a problemi di minimo locali, partendo da punti iniziali diversi, si trovano valori ottimi diversi, sia al fatto che le varianze aggiuntive non vengono mai poste esattamente pari a zero, dunque una loro influenza, seppur piccola nei calcoli è presente. Inoltre per il metodo LODE vengono utilizzate le derivate analitiche, mentre il filtro classico sfrutta quella numeriche. Per trasparenza riportiamo l'istogramma dei valori del MSE relativo, riguardanti queste serie, Figura 3.2. La maggior parte sono molto vicini ad uno, come deve essere.

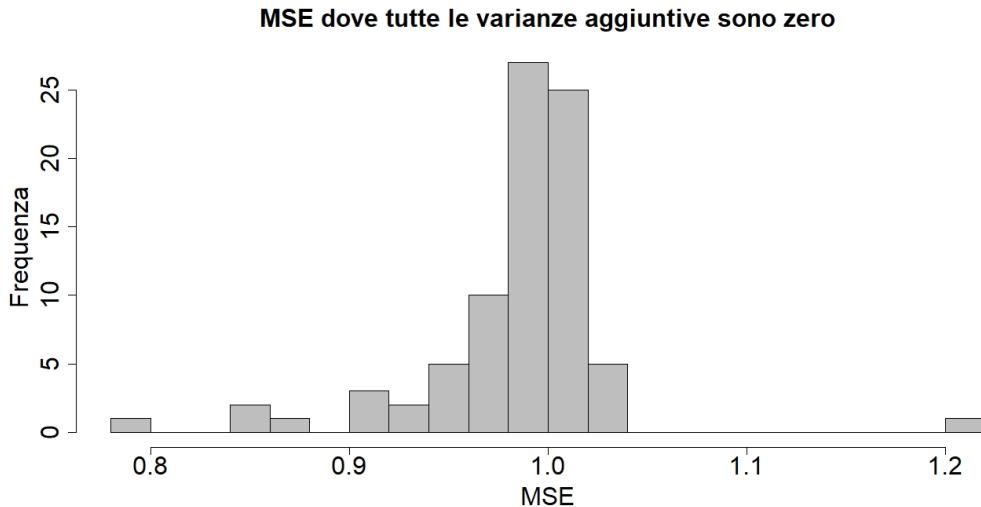


Figura 3.2: Istogramma dei valori del MSE relativo per le serie in cui LODE non pone alcuna varianza aggiuntiva diversa da zero.

Già da queste prime considerazioni possiamo comprendere uno degli aspetti che ha caratterizzato l'intero percorso della tesi, i problemi numerici degli algoritmi. I valori presenti

nell’istogramma 3.2 forse sono più variabili di quanto ci si aspetterebbe, purtroppo il processo di minimizzazione è complicato, vi sono tanti fattori che lo influenzano, ed è davvero difficile capire il perché di alcuni risultati o di alcuni problemi che sono capitati.

Per queste simulazioni il metodo LODE è stato applicato con 30 iterazioni, per velocizzare le simulazioni. Concentriamoci ora sui valori di MSE riscontrati sulle serie storiche simulate.

MSE: RW gaussiani			
	Numerosità	LODE30	<i>auxres</i>
<i>Tutte</i>	98	0.993	1.109
<i>SW&lt;5%</i>	12	1.010	1.048

Tabella 3.1: MSE medio ottenuto su random walk puramente gaussiani.

Soffermiamoci a capire per bene la tabella, in quanto ne saranno presentate diverse di questo tipo. Le righe rappresentano i diversi sottogruppi di serie simulate, mentre sulle colonne abbiamo la numerosità di questi sottogruppi ed i valori medi del MSE relativo per i vari approcci, LODE e *auxres*. Per il filtro classico ovviamente non sono riportati i risultati, in quanto ha un MSE relativo sempre pari ad 1, su ogni serie.

I risultati sono in linea con quanto ci si potrebbe aspettare, *auxres*, peggiora le prestazioni del filtro classico, in quanto pone outlier dove non dovrebbe. Con il metodo LODE si ha un leggero miglioramento, probabilmente dovuto al caso, o per quanto detto prima riguardo al problema dei minimi locali anche quando le varianze sono tutte poste pari a zero. Interessante in questo è anche valutare quante sono le serie storiche appartenenti a *SW<5%*, ovvero quando i residui ausiliari non sono risultati gaussiani ad un livello di confidenza del 95%. Sono 12 in totale queste serie, 7 residui additivi non sono risultati gaussiani e lo stesso per quelli riguardanti il livello, un numero conforme a quello atteso, ovvero  $98 \cdot 0.05 \approx 5$ , con 98 il numero di serie analizzate.

Questa è una sezione molto importante per comprendere al meglio l’efficienza del nostro metodo. Da un punto di vista puramente teorico, esso ha la capacità di capire che non è necessario introdurre alcun parametro aggiuntivo, in quanto aumenterebbe il BIC, e dunque restituire la stima classica. Un qualunque metodo basato sui residui ausiliari, invece sarà molto più propenso ad aggiungere variabili esterne, in quanto considerando un CL del 99%, con residui puramente gaussiani, mi aspetto che su 100 residui uno possa essere sopra la soglia considerata, e dunque individuare erroneamente un potenziale outlier. Si comprende facilmente come il nostro metodo abbia l’abilità di non sbagliare, a differenza di *auxres*. A sostegno di questa teoria osserviamo il numero di volte in cui i due metodi non hanno posto alcuna varianza o dummy aggiuntiva. LODE si è astenuto lo 84% delle volte, contro il 17% di *auxres*.

### 3.3.3 Individuazione degli outlier

Per stimare correttamente il trend di una serie storica è necessario individuare gli outlier e stimare la loro dimensione. In questa sezione ci concentriamo sul primo punto, capire

quanti outlier sono individuati. Analizziamo 100 serie storiche di 100 dati con un jump in 50 ed un AO in 75, con una dimensione che varia uniformemente tra 2.5 e 7.5 volte la rispettiva  $\sigma$ , come spiegato nella Sezione 3.4.1.

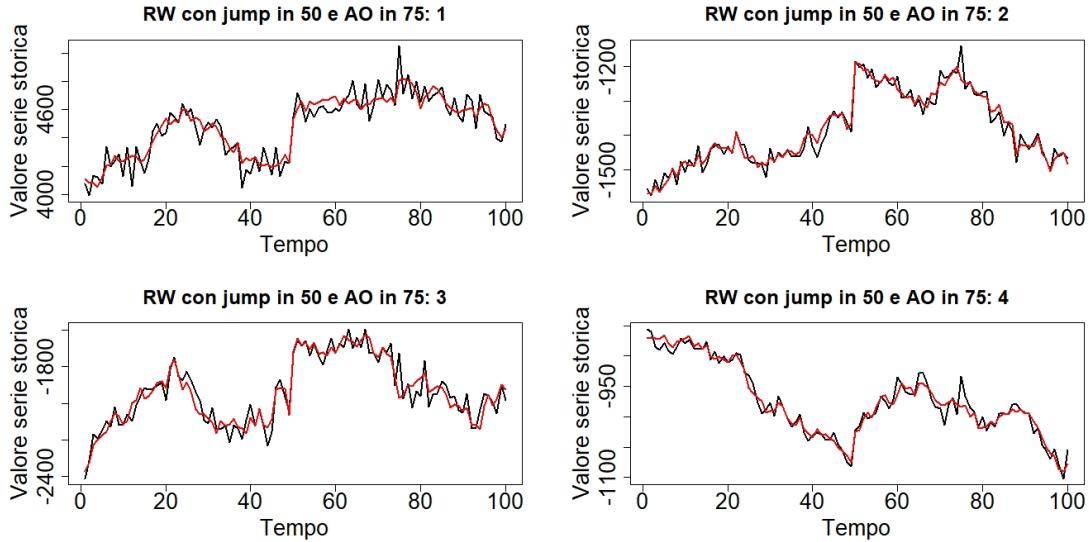


Figura 3.3: Esempi di serie storica con jump in 50 e AO in 75, usate per fare l’analisi. In nero la serie storica mentre in rosso il trend.

Applicando il nostro metodo alla serie storica ci rimane da decidere quando l’algoritmo “dice” che in quell’istante siamo in presenza di un outlier. Per fare ciò prendiamo tutti gli istanti in cui la varianza aggiuntiva è diverso da zero, ovvero è contata come grado di libertà, secondo le modalità descritte nella Sezione 1.4.6. Sottolineiamo come se il cambio di livello è nell’istante 50, allora la varianza aggiuntiva o il residuo ausiliare, saranno posti o individuati l’istante precedente, in 49. Questo è dovuto alla scelta del formalismo quando abbiamo definito il modello (3.1).

Sono diverse le metriche di valutazione che si possono considerare, noi abbiamo scelto le seguenti: *perfetti*, la proporzione di serie in cui l’identificazione di outlier è perfetta, ovvero viene individuato unicamente l’outlier nella posizione corretta; *recall*, rappresenta le serie in cui è stato identificato l’outlier correttamente, ma può anche essere che vengano identificati altri outlier non veritieri; *nessuno*, indica le serie in cui il metodo non ha individuato alcun outlier; *FP*, che sta per *falsi positivi*, quando vengono individuati unicamente outlier in posizioni non corrette, ovvero sono tutti sono falsi positivi. Tutte queste metriche sono valide e saranno riportate, tuttavia per avere un singolo valore di riferimento, utilizzeremo come metrica principale la F1-measure, ovvero

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}.$$

Per quanto riguarda l’analisi coi residui ausiliari bisogna prestare particolare attenzione, in quanto residui del trend e della parte additiva sono fortemente correlati, portando raramente ad una individuazione perfetta degli outlier, in quanto i residui del trend identifieranno anche l’outlier nella parte additiva.

Fatte queste premesse riportiamo i vari risultati ottenuti per queste simulazioni nella Tabella 3.2.

Cambi di livello			Outlier additivi		
	LODE	auxres		LODE	auxres
Perfetti	0.30	0.29	Perfetti	0.36	0.28
Recall	0.36	0.57	Recall	0.45	0.68
Nessuno	0.62	0.14	Nessuno	0.54	0.08
FP	0.02	0.29	FP	0.01	0.24
F1	0.47	0.49	F1	0.55	0.53

Tabella 3.2: Proporzione di serie che soddisfano le rispettive caratteristiche, nell’individuazione dei cambi di livello e degli outlier additivi, in serie storiche che presentavano entrambi i tipi di outlier. Legenda: *perfetti*, viene selezionato unicamente l’outlier nella posizione corretta; *recall* viene selezionato l’outlier nella posizione corretta; *nessuno*, non viene selezionato alcun outlier; *FP*, nessuno tra gli outlier selezionati è quello corretto; *F1*, misura F1.

Dai risultati si capisce come il nostro metodo sia meno propenso ad individuare outlier, ciò porta ad avere una recall più bassa. Però è anche interessante osservare il numero di errori dei due metodi. *Auxres* ha una maggiore probabilità di sbagliare totalmente la posizione degli outlier, mentre noi optiamo maggiormente per l’astensione. Le F1 sono molto simili per entrambi i casi, tuttavia bisogna ricordare quanto detto prima, riguardo alla correlazione dei residui ausiliari. Ciò porta *auxres* ad individuare spesso il jump tra gli outlier additivi e viceversa, facendo inevitabilmente peggiorare le performance.

Per risolvere questa problematica abbiamo generato altre serie ponendo unicamente il cambio di livello e cercando solo lui. Sono state simulate ben 500 serie, infatti ricordiamo che il metodo LODE è molto più veloce se si ricercano outlier su una sola componente. Le serie sono quasi identiche a quelle generate in precedenza, cambia solo che ora non è posto alcun outlier additivo.

Dalla Tabella 3.3 notiamo come ora il metodo *auxres* abbia un valore della F1 maggiore, poiché non individua più l’outlier additivo come jump. Per noi questo fattore è molto meno rilevante, capita alcune volte, ma sono rare, infatti i risultati sono quasi identici ai precedenti.

Abbiamo visto come *auxres* abbia in generale una miglior abilità nell’individuare gli outlier. LODE ha la tendenza ad astenersi molto, ponendo tutte le varianze aggiuntive pari a zero.

### 3.3.4 Stima dimensione cambio di livello

Il secondo fattore elencato in precedenza che influenza la stima del trend è la capacità di stimare la dimensione dell’outlier. Il nostro metodo crea un bias, dovuto allo shrinkage dei parametri, dunque in generale ci aspettiamo che sottostimi la grandezza del jump. Invece

Cambi di livello		
	LODE	auxres
Perfetti	0.32	0.42
Recall	0.41	0.70
Nessuno	0.59	0.15
FP	0.01	0.15
F1	0.52	0.63

Tabella 3.3: Proporzione di serie che soddisfano le rispettive caratteristiche, nell’individuazione dei cambi di livello, in serie storiche che presentavano solo questo tipo di outlier. Legenda: *perfetti*, viene selezionato unicamente l’outlier nella posizione corretta; *recall* viene selezionato l’outlier nella posizione corretta; *nessuno*, non viene selezionato alcun outlier; *FP*, nessuno tra gli outlier selezionati è quello corretto; *F1*, misura F1.

*Auxres* utilizza una variabile dummy a gradino per stimare la dimensione, e ci aspettiamo che questo approccio sia corretto. Per verificare quest’ultima teoria si possono simulare serie con uno shift in posizione nota e porre sempre la variabile dummy.

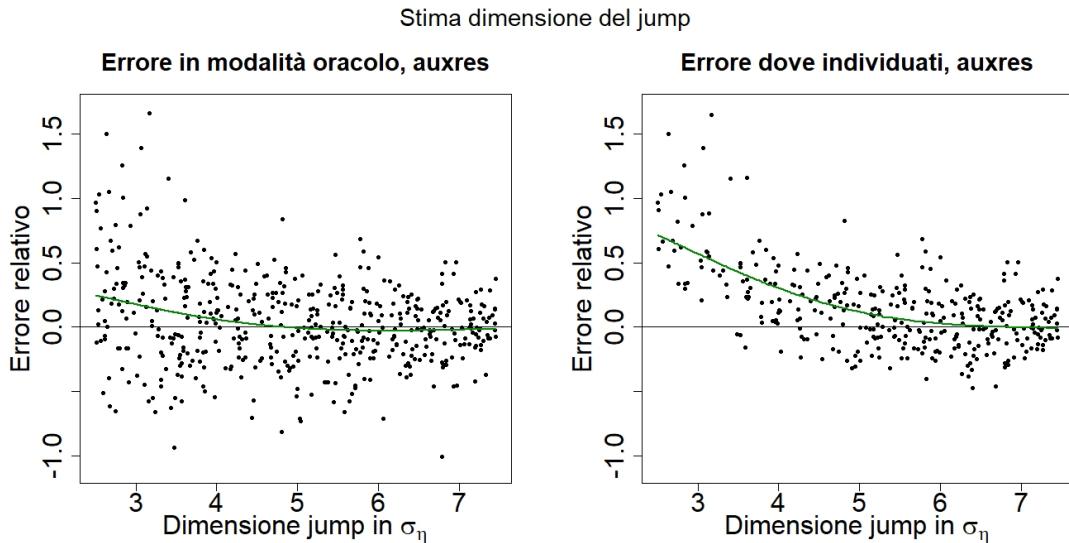


Figura 3.4: Dimensione reale del jump misurata in  $\sigma_\eta$  contro l’errore relativo ovvero dimensione stimata meno dimensione reale, diviso la dimensione reale; nei due casi: oracolo e solo dove lo shift viene identificato coi residui ausiliari. Nel primo caso lo stimatore basato sulle variabili dummy è corretto, nel secondo vi è un sovrastima generale. In verde un lisciamento con una spline con 3 gradi di libertà. Sono state simulate 500 serie in totale, dunque sono altrettanti i punti nella figura di sinistra, mentre in quella di destra sono 351.

Andiamo ad analizzare la Figura 3.4, dove è rappresentato l'errore relativo, ovvero dimensione stimata meno dimensione reale, diviso la dimensione reale. Osservando la modalità oracolo, a sinistra, tutto è come da aspettativa: lo stimatore a volte sovrastima, a volte sottostima, ma è corretto e l'errore decresce all'aumentare della grandezza, in quanto le fluttuazioni ordinarie di  $\sigma_\epsilon$  e  $\sigma_\eta$  diventano meno importanti.

Tuttavia questo è il caso dell'oracolo, ove sappiamo individuare sempre il jump. Andando ad osservare solo quando realmente saremmo in grado di individuarlo, per esempio basandoci sui residui ausiliari, le cose cambiano. Nella parte di destra della Figura 3.4, si osserva come lo stimatore con variabili dummy non sia più corretto, in quanto se un jump è "piccolo" e sarebbe anche sottostimato con la variabile dummy, non viene identificato e dunque nelle stime vengono a mancare le sottostime dei jump più piccoli. Di conseguenza si verifica un forte bias verso la sovrastima.

Tutto ciò è davvero molto interessante poiché noi effettuiamo uno shrinkage, che in generale porta ad uno uno stimatore con bias, tuttavia analizzando solo quando gli shift vengono individuati, lo shrinkage va a diminuire l'effetto di sovrastima che abbiamo appena visto. Ciò porta ad una stima migliore della dimensione che si traduce in un MSE minore nel calcolo del trend. Analizzando i jump più grandi, questo non accade, in quanto si è sempre, o quasi, capaci di identificare il jump, portandoci nella situazione dell'oracolo.

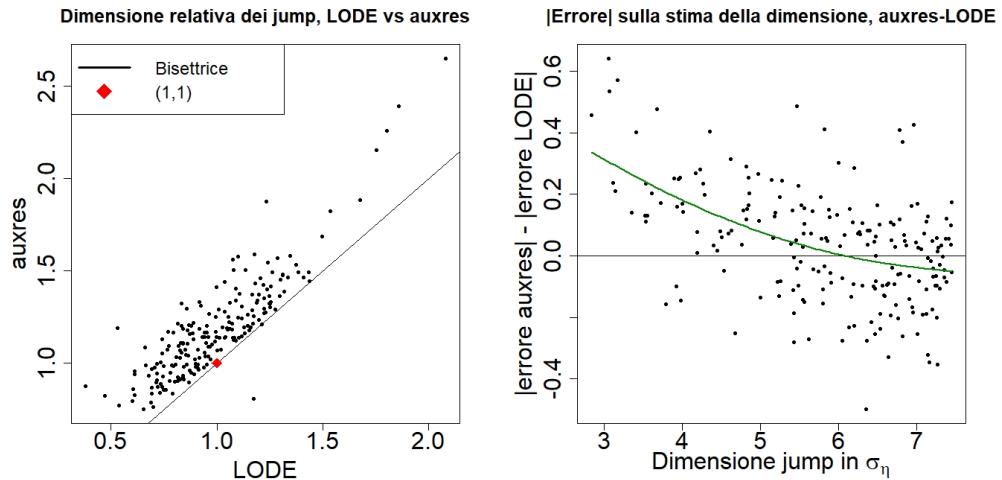


Figura 3.5: A sinistra il confronto tra le dimensioni relative dei jump individuati con i due approcci. Con dimensione relativa si intende la dimensione stimata diviso la dimensione reale del jump. La linea nera è la bisettrice, ovvero l'uguaglianza tra le due stime, mentre il punto rosso è (1,1), che indica una stima perfetta da parte di entrambi i metodi. Nella figura di destra abbiamo la differenza del valore assoluto dell'errore relativo nella stima della grandezza del jump, ottenuto con i due approcci, al variare della dimensione reale del jump misurata in  $\sigma_\eta$ . In verde una spline che estrae il trend dai dati. Ricordiamo che con errore relativo intendiamo la dimensione stimata meno dimensione reale, diviso la dimensione reale.

La Figura 3.5 illustra molto bene quanto detto. Nell'immagine di sinistra si nota come la stima tramite variabile dummy, fatta da *auxres*, sia quasi sempre maggiore della nostra,

infatti i punti sono sopra la bisettrice. Ciò, come anticipato, è causato dallo shrinkage che poniamo. Tuttavia questo fatto porta ad un minor errore nelle stima dei jump più piccoli, come si vede nella Figura di destra.

A questo punto sorge naturale la domanda: nella pratica è possibile distinguere i jump "piccoli" da quelli "grandi", e dunque individuare quando ci si aspetta che la stima con la variabile dummy sia migliore della nostra?

Per fare questo si può porre sull'asse delle ascisse la dimensione stimata del jump diviso  $\hat{\sigma}_\eta$ , il valore stimato da *auxres*, e non più  $\sigma_\eta$ . Come si può osservare dall'immagine 3.6, i

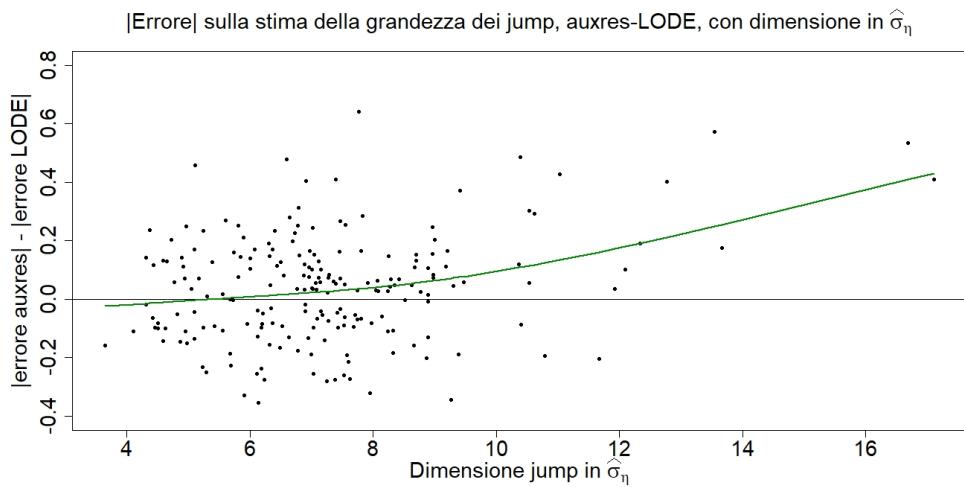


Figura 3.6: Dimensione stimata del jump misurata in  $\hat{\sigma}_\eta$ , ovvero quella stimata, contro la differenza tra i valori assoluti degli errori relativi. Ora non si riesce più ad identificare una zona di sovrastima ed una con stime corrette, come in precedenza. In verde un lisciamento con una spline con 3 gradi di libertà. La tendenza ad crescere della spline è dovuta soltanto alla presenza di pochi punti.

risultati purtroppo mettono in mostra un totale rimescolamento dei dati, che non rendono possibile la distinzione di zone corrette e di sovrastima, come accadeva precedentemente.

Sottolineo il fatto che le analisi sono state condotte su un certo range dei jump, da 2.5 a  $7.5 \sigma_\eta$ , ovvero quelli che si possono incontrare in applicazioni reali. Valori più piccoli non sarebbero considerati jump, mentre valori più grandi sono irrealistici, ma cambierebbero le analisi, a favore dello stimatore con variabili dummy, che risulterebbe "più corretto".

**Prime conclusioni** Questi primi risultati sono stati utili per capire quali sono le potenzialità del metodo LODE: è in grado di capire se non ci sono outlier, quindi funziona bene su serie poco "contaminante", inoltre abbiamo visto come sia in grado di stimare ottimamente la dimensione di jump "piccoli". Mentre pecca un poco nella capacità di individuare outlier, in quanto molto parsimonioso. Sottolineiamo come queste tre considerazioni sono fondamentali per capire i risultati che ora andremo a presentare.

## 3.4 Stima del trend della serie

Illustriamo ora i vari risultati riguardanti la stima del trend della serie, ottenuti con le diverse tipologie di serie storiche.

### 3.4.1 Random walk

La prima tipologia di serie analizzate sono i random walk, il modello più semplice esistente. Il modello utilizzato è quello già descritto nell'equazione (3.1).

La simulazione di queste serie è avvenuta tramite mistura di gaussiane, come descritto nella Sezione 3.1, ripetendo le operazioni fino ad ottenere 250 serie su cui è stato applicato sia il metodo LODE, sia *auxres*, per confrontare le prestazioni.

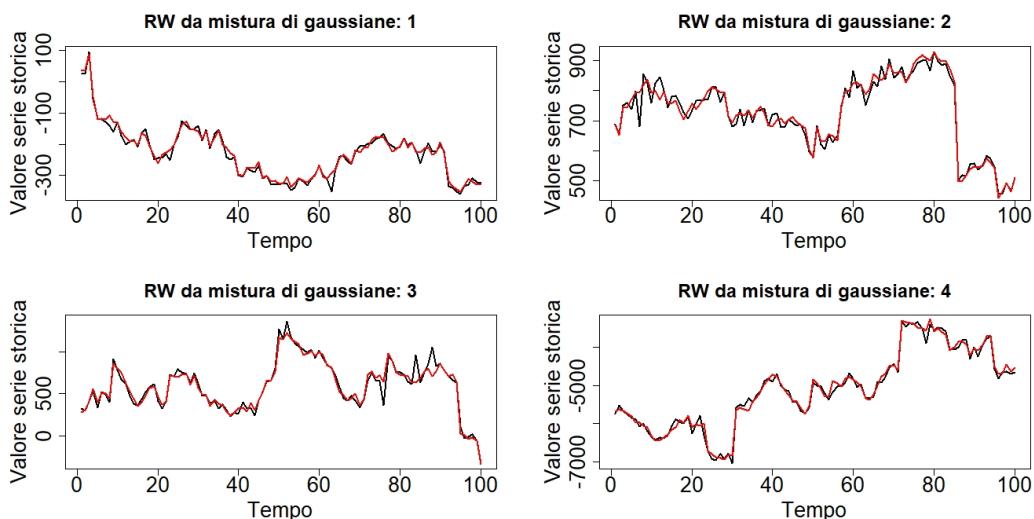


Figura 3.7: Esempio di alcuni random walk analizzati. In nero è raffigurato la serie storica, mentre in rosso il trend,  $\mu$ .

L'obiettivo di queste prime analisi è valutare l'errore quadratico medio nella stima del trend. Come spiegato nella Sezione 1.4.4, è difficile stabilire a priori un numero di iterazione per il nostro metodo, perché dipende da quanto tempo si ha a disposizione, per questo in diverse simulazioni abbiamo applicato il metodo LODE con 30 e 50 iterazioni, rispettivamente denominati LODE30 e LODE50.

I risultati sono presentati nella Tabella 3.4. Per prima cosa analizziamo le differenze tra LODE30 e LODE50. Come è giusto che sia, abbassando il numero di iterazioni, i risultati peggiorano, ma fortunatamente neanche di troppo. Non riportiamo la varianza associata a questi numeri, perché è poco sensata, in quanto è dovuta al diverso numero e grandezza degli outlier presenti nelle diverse serie storiche, dunque è significativa solo la varianza delle differenza di MSE tra due metodi. In ogni caso, provando sia un t-test, sia un test non parametrico, come quello di Wilcoxon, non abbiamo trovato alcuna significatività riguardante la differenza di MSE tra LODE30 e LODE50. Andiamo ora ad analizzare il confronto tra il BIC ottenuto coi due metodi. Su 250 serie solo per 65 aumentare le

MSE: random walk				
	Numerosità	LODE30	LODE50	<i>auxres</i>
<i>Tutte</i>	250	0.883	0.874	0.875
<i>MSE&lt;0.95</i>	121	/	0.735	0.724
<i>SW&lt;5%</i>	197	0.857	0.843	0.838

Tabella 3.4: MSE medio ottenuto su random walk con outlier provenienti da una mistura di gaussiane. Ricordiamo brevemente le varie categorie: *tutte* comprende tutte le serie simulate, *MSE<0.95* solo quelle ove LODE50 e *auxres* hanno un MSE<0.95, infine *SW<5%* contiene le serie in cui almeno un test di Shapiro-Wilk sui residui ausiliari abbia ottenuto un p-value minore del 5%.

iterazioni a 50 ha portato ad una stima differente. Consideriamo unicamente tali serie. La differenza di MSE tra LODE30 e LODE50 è di 0.036, mentre quella in BIC è di 1.069. Ciò mette in mostra due fattori: una unità di differenza nel BIC porta a differenze marcate nel valore del MSE, parlando si serie storiche con 100 istanti temporali. Seconda cosa è che per alcune serie è utile aumentare il numero massimo di iterazioni dell'ottimizzazione bayesiana. Tuttavia queste serie sono solo 65 su 250, ovvero il 26%, negli altri casi aumentare le iterazioni è solo una perdita di tempo. Concludendo il confronto tra LODE30 e LODE50, possiamo affermare che è sensato ridurre il numero di iterazioni per quasi dimezzare il tempo computazionale, infatti è ciò che faremo per alcune simulazioni particolarmente lunghe.

Concentriamoci ora sul confronto con *auxres*. Siamo in quasi perfetta parità se si guardando tutte le serie simulate, mentre un poco in svantaggio osservando le serie con i residui non gaussiani, *SW<5%*. Ciò è normale, e riaccadrà spesso, in quanto le serie che non rientrano in questa categoria sono quelle meno "contaminate", i cui outlier sono più piccoli o meno numerosi, proprio la tipologia di serie dove LODE ha ottime performance.

Per la categoria *MSE<0.95* non sono riportati i risultati per LODE30, in quanto per creare la categoria si sono considerati unicamente i valori del MSE di LODE50 e *auxres* minori di 0.95, e non anche di LODE30 per non ridurre di troppo la dimensione di questo sottogruppo. I risultati per questa categoria mostrano come *auxres* sia un poco meglio rispetto a LODE50. Questa è un fatto che si è ripetuto in numerose simulazioni e mette in luce la differenza nei modi di stimare la serie. *Auxres* rileva sempre molti jump o AO da porre nel modello, perciò le sue stime spesso portano a grandi errori, influenzando molto il MSE medio. Considerando solo le serie ove *MSE<0.95* si eliminano dal conto tali serie, mettendo in luce il fatto che *auxres* è in grado di portare a ottime stime, anche migliori di quelle del metodo LODE, ma purtroppo i numerosi errori che fa, gli alzano notevolmente il MSE medio. Per il nostro approccio ciò è meno evidente in quanto opta maggiormente per l'estensione piuttosto che porre outlier non strettamente necessari.

I risultati sono molto suscettibili al numero e alla dimensione degli outlier creati. Noi abbiamo generato i disturbi come descritto nella Sezione 3.1, ovvero

$$\tilde{\epsilon}_i \sim N(0, \sigma_\epsilon^2 \cdot c^2)$$

con  $c \sim Uniforme(2,6)$ . Tuttavia  $c$  è un parametro che può essere scelto come si vuole, e che influenza non poco i risultati. Per questo riportiamo anche i risultati per  $c$  diversi, ovvero tra le 250 serie simulate abbiamo selezionato quelle in cui  $c$  apparteneva all'intervallo 2-4 sia per  $\tilde{\epsilon}_i$  che per  $\tilde{\eta}_i$ , e anche quelle in cui  $c$  appartenevano a 4-6.

MSE con $c \in [2,4]$				
Numerosità	LODE30	LODE50	auxres	
Tutte	65	0.967	0.958	0.972
MSE con $c \in [4,6]$				
Numerosità	LODE30	LODE50	auxres	
Tutte	53	0.859	0.850	0.806

Tabella 3.5: MSE medio ottenuto su random walk con outlier provenienti da una mistura di gaussiane, qui differenziati per la loro grandezza. In alto le serie con gli outlier più piccoli dove  $c \in [2,4]$ , sotto con quelli più grandi dove  $c \in [4,6]$ . Nel primo caso LODE50 è meglio di *auxres*, nel secondo vale il viceversa.

Dalla Tabella 3.5 si nota proprio come  $c$  influenzi notevolmente i risultati, facendo cambiare anche quale sia l'approccio con MSE minore. Ovviamente non esiste un  $c$  corretto e uno sbagliato, dipende da come si vogliono generare le serie storiche, ed è per questo che abbiamo considerato un ampio range, da 2 a 6, per simulare tutti i possibili casi che possono accadere nella realtà.

**Con 200 istanti temporali** Fino ad ora ed anche prossimamente, ci focalizzeremo unicamente su serie con 100 dati, tuttavia è bene testare il metodo LODE anche su serie più lunghe. Aumentando il numero di istanti temporali, crescono anche il numero di varianze aggiuntive, ovvero il numero di parametri da minimizzare, ciò porta a due problemi principali. Più parametri implica più difficoltà a trovare l'ottimo globale della funzione, ciò potrebbe portare in alcuni casi a performance non eccellenti. Tuttavia la problematica principale si riscontra nel tempo computazionale, il principale difetto del metodo LODE, e che viene accentuata sempre di più allungando la serie storica.

Fatte queste premesse, riportiamo i risultati per simulazioni di serie con 200 istanti temporali e con outlier generati come in precedenza. Il metodo LODE è stato applicato con 30 iterazioni, per ridurre il tempo computazionale.

I risultati contenuti nella Tabella 3.6 mostrano un leggero decremento delle performance, dovuto ai motivi detti sopra. In ogni caso il metodo è applicabile con discrete prestazioni anche a serie storiche più lunghe di 100 dati, anche se ovviamente bisogna pagare in tempo di esecuzione.

**2 AO e 2 LS** Come già detto vi sono svariati modi per generare i disturbi, in questo paragrafo andiamo ad analizzare i risultati ottenuti simulando sempre RW, ma con gli outlier generati in un altro modo. La dimensione dei level shift e degli outlier additivi,

MSE: RW con 200 istanti temporali			
	Numerosità	LODE30	<i>auxres</i>
<i>Tutte</i>	100	0.892	0.871
<i>MSE&lt;0.95</i>	51	0.803	0.765
<i>SW&lt;5%</i>	93	0.885	0.864

Tabella 3.6: MSE medio ottenuto su random walk di 200 istanti temporali, con outlier provenienti da una mistura di gaussiane. Le varie categorie indicano: *tutte* comprende tutte le serie simulate, *MSE<0.95* solo quelle ove LODE30 e *auxres* hanno un MSE<0.95, infine *SW<5%* contiene le serie in cui almeno un test di Shapiro-Wilk sui residui ausiliari abbia ottenuto un p-value minore del 5%.

non ha più una distribuzione gaussiana, ma uniforme, precisamente da 2.5 a 7.5 volte la rispettiva varianza, ovvero

$$\begin{aligned} LS &\sim U(2.5, 7.5) \cdot \sigma_\eta \\ AO &\sim U(2.5, 7.5) \cdot \sigma_\epsilon. \end{aligned}$$

La scelta del range viene dal fatto che, considerando un pvalue del 99%, per dati provenienti da un  $N(0,1)$ , si ha che dati maggiori di 2.5 sono considerabili come outlier. Mentre il valore massimo di 7.5 è stato scelto ipotizzando che jump e AO più grandi siano poco realistici, ed in ogni caso molto facili da individuare, anche ad occhio nudo. La posizione dei jump e degli outlier invece è stata definita in modo casuale, lasciando i primi 5 istanti temporali senza outlier e poi posizionando un jump in maniera casuale ed un secondo altrettanto casualmente, ma non negli istanti adiacenti, in quanto due cambi di livello consecutivi sono irrealistici. Abbiamo posizionato anche due outlier, in modo causale tra le posizioni lasciate libere dai jump. Di seguito le righe di codice riproducibili per generare le serie storiche di questo tipo, riportate anche al seguente link [Sim\\_2ao2ls<sup>2</sup>](#).

---

```

1 n=100
2 jmp<-ao<-numeric(n)
3 sig_eps <- runif(1,1,100)
4 sig_eta <- runif(1,1,4)*sig_eps/2
5 jmp_size <- runif(1,2.5,7.5)
6 ao_size <-runif(1,2.5,7.5)
7 x_jmp<-sample(5:n,1)
8 xx_jmp<-sample((5:n)[-c(x_jmp-1,x_jmp,x_jmp+1)],1)
9 jmp[c(x_jmp,xx_jmp)]<-sample(c(-1,1),2,replace=T)*jmp_size*sig_eta
10 x_ao<-sample((5:n)[-c(x_jmp,xx_jmp)],2)
11 ao[x_ao] <- sample(c(-1,1),2,replace = T) * ao_size * sig_eps
12 mu<-cumsum(rnorm(n, sd=sig_eta)+jmp)+runif(1,-100*sig_eps,100*sig_eps)
13 y <- mu + rnorm(n, sd = sig_eps) + ao

```

---

<sup>2</sup>[https://github.com/simone-frigerio/LODE/blob/main/sim\\_2ao\\_2ls.R](https://github.com/simone-frigerio/LODE/blob/main/sim_2ao_2ls.R)

MSE: 2 AO e 2 LS				
	Numerosità	LODE30	LODE50	<i>auxres</i>
<i>Tutte</i>	100	0.910	0.908	0.877
<i>MSE&lt;0.95</i>	49	/	0.766	0.763
<i>SW&lt;5%</i>	73	0.886	0.885	0.840

Tabella 3.7: MSE medio ottenuto su random walk con due outlier additivi e due cambi di livello. Le varie categorie indicano: *tutte* comprende tutte le serie simulate, *MSE<0.95* solo quelle ove LODE50 e *auxres* hanno un  $MSE < 0.95$ , infine *SW<5%* contiene le serie in cui almeno un test di Shapiro-Wilk sui residui ausiliari abbia ottenuto un p-value minore del 5%.

I risultati presenti nella Tabella 3.7 sono simili a quanto visto prima, ed ancora una volta sono influenzati dalla scelta usate per le dimensioni degli outlier.

### 3.4.2 RW con $\sigma_\eta = 0$

Le serie generate in precedenza presentavano tutte un rapporto segnale rumore tra un mezzo e due, dunque era impossibile avere  $\sigma_\eta = 0$ . Questa eventualità però è plausibile nella realtà, come nella serie nel Nilo, perciò meritava un approfondimento.

Per generare dati di questa tipologia, ovviamente non è più possibile usare il codice descritto in precedenza, poiché in questo caso abbiamo  $\sigma_\eta = 0$ . Perciò abbiamo generato dei jump in maniera artigianale, selezionando tre istanti temporali, 25, 50 e 75, ove porre un cambio di livello, con dimensione  $D$  con distribuzione uniforme e moltiplicata rispettivamente per  $\text{seno}(D \cdot 2)$ ,  $\text{seno}(D)$ ,  $\text{seno}(D \cdot 4)$ , questo per rendere i jump diversi gli uni dagli altri, facendo in modo che a volte fossero positivi, altre negativi, più grandi o più piccoli. Gli outlier additivi sono generati da mistura di gaussiane, nel modo già descritto. Alcune serie storiche generate purtroppo erano quasi esclusivamente rumore, con un trend praticamente piatto, queste serie sono state escluse dai calcoli.

Una parte molto interessante di questa analisi riguarda la capacità dei metodi di individuare se la serie storica abbia  $\sigma_\eta = 0$  o meno. Il filtro di Kalman classico non restituirà mai  $\hat{\sigma}_{\eta KFS} = 0$ , ovvero un trend pari ad una retta orizzontale, a causa dei salti nel trend, poiché li liscerà mettendo  $\hat{\sigma}_{\eta KFS} \neq 0$ , come nella serie del Nilo. Invece *auxres* ed il nostro metodo hanno le potenzialità per indovinare. In precedenza abbiamo parlato delle diverse versioni del BIC utilizzabile, qua si capisce il perché. Utilizzare il BIC corretto, BICC, favorisce molto il porre  $\sigma_\eta = 0$  in quanto si avrebbe un grado di libertà in meno. Analizziamo ora i risultati, ove abbiamo considerato  $\hat{\sigma}_\eta \approx 0$  se minore di  $\hat{\sigma}_{\eta KFS}/10$ .

Osservando la Tabella 3.8, notiamo come utilizzando sia utilizzando il BIC corretto sia non corretto, il nostro metodo è maggiormente propenso a porre  $\sigma_\eta = 0$ , nelle serie analizzate, rispetto a *auxres*, portando quindi a risultati migliori. Tuttavia è da notare che questa propensione, qua è ottima, ma nelle altre analisi di RW, capitava purtroppo che il metodo ponesse comunque, erroneamente  $\sigma_\eta = 0$ . Si parla di circa il 1% delle serie, ma

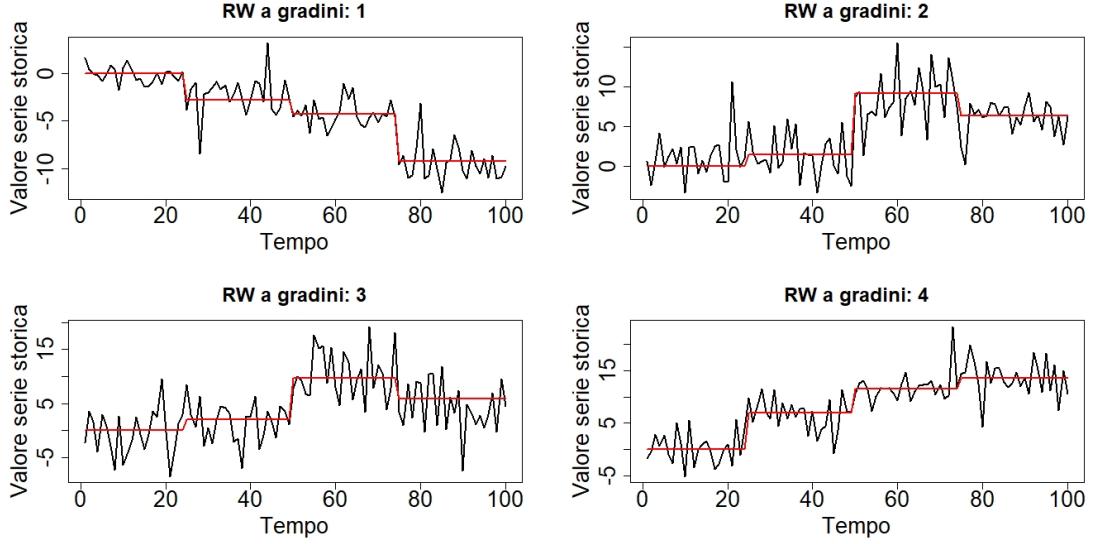


Figura 3.8: Esempio di alcuni random walk con  $\sigma_\eta = 0$  analizzati. In nero è raffigurato la serie storica, mentre in rosso il trend,  $\mu$ .

Proporzione di serie in cui $\hat{\sigma}_\eta \approx 0$				
LODE50 BIC	LODE50 BICC	auxres		
0.31	0.50	0.00		
MSE: RW con $\sigma_\eta = 0$				
Numerosità	LODE30 BIC	LODE50 BIC	LODE50 BICC	auxres
Tutte	83	0.593	0.591	0.596
SW<5%	80	0.591	0.590	0.595
				0.711
				0.714

Tabella 3.8: La tabella in alto mostra la percentuale di serie analizzate in cui è stata stimata un valore di  $\hat{\sigma}_\eta$  di circa zero, ovvero si è individuata correttamente questa caratteristica della serie storica. In basso vi è tabella contenente i MSE medi ottenuto su random walk a gradini, ove  $\sigma_\eta = 0$ .

comunque un'enormità in confronto allo 0% di *auxres*. Ovviamente, utilizzando il BICC si accentua molto di più questo comportamento.

La nostra maggior facilità di porre  $\sigma_\eta = 0$  può essere spiegata osservando che per valori di  $\lambda$  piccoli, quasi sempre si stima un trend a gradini, quindi per una qualsiasi serie storica viene sempre stimato un modello di questo tipo, che a volte può essere scelto come il migliore, altre volte no. *Auxres*, invece, è una semplice modifica del filtro classico.

Quanto detto lo si può notare nella Figura 3.9. Abbiamo riportato un esempio in cui il metodo LODE individua molto bene l'andamento a gradini della serie, ovvero pone  $\hat{\sigma}_\eta^{LODE} = 0$ , mentre non accade ciò per *auxres*, in quanto identifica solo l'ultimo jump.

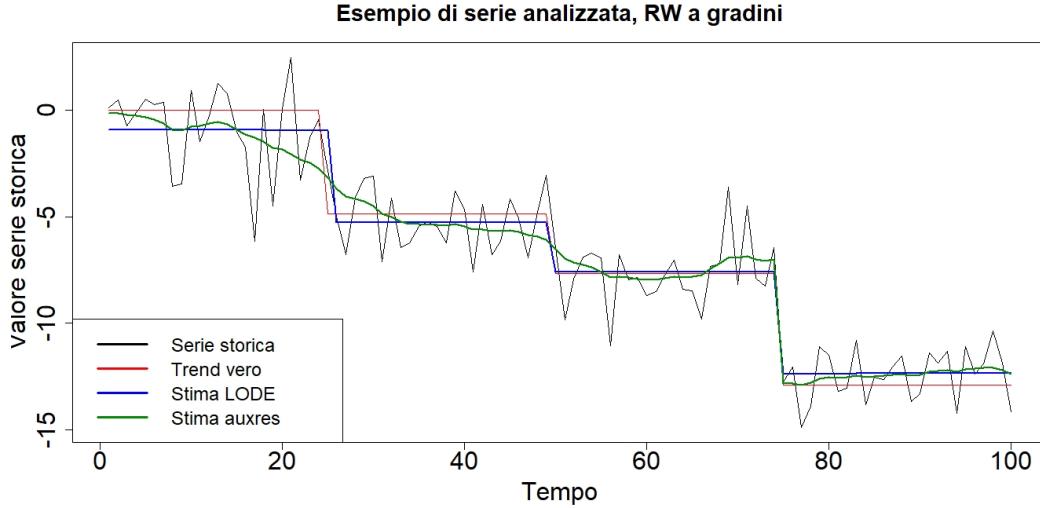


Figura 3.9: Esempio di trend stimato con serie in cui  $\sigma_\eta = 0$ . Possiamo osservare come il metodo LODE identifichi l’andamento a gradini, mentre *auxres* no.

Torniamo ora alla Tabella 3.8. I risultati riguardanti il MSE dei metodi LODE forse sono un poco sorprendenti. Sebbene utilizzando il BIC siamo in grado di individuare più volte che si tratta di serie a gradini, ciò non si ripercuote nel MSE. Sottolineiamo come i risultati siano soggetti ad una elevata variabilità poiché influenzati da valori estremi di MSE presenti in questo blocco di serie simulati. Infatti si è raggiunto un MSE relativo minimo di circa 0.04, estremamente basso, dovuto al fatto che una volta indovinato che la serie è a gradini, diventa facile fare un’ottima stima della serie.

I risultati sono comunque molto positivi per il metodo LODE, che testimoniano come, su questa tipologia di serie, il nuovo approccio che proponiamo porta indubbiamente ai migliori risultati. Infatti applicando il test dei ranghi con segno di Wilcoxon ai valori di MSE di LODE50 e *auxres*, si trova un p-value minore del 5%. Ricordiamo che l’ipotesi nulla del test è che i campioni provengano dalla medesima distribuzione.

### 3.4.3 LLT più stagionalità

Andiamo ora ad analizzare una serie storica completa, ovvero un local linear trend più stagionalità, modellata con dummy stocastiche. Le serie storiche sono state generate come descritto, nella Sezione 3.1, con gli outlier derivanti da una mistura di gaussiane. Il modello utilizzato è il seguente:

$$\begin{aligned}
 y_t &= \mu_t + S_t + \epsilon_t & \epsilon_t &\sim NID(0, \sigma_\epsilon^2 + \dot{\sigma}_{\epsilon_t}^2) \\
 \mu_{t+1} &= \mu_t + \beta_t + \eta_t & \eta_t &\sim NID(0, \sigma_\eta^2 + \dot{\sigma}_{\eta_t}^2) \\
 \beta_{t+1} &= \beta_t + \zeta_t & \zeta_t &\sim NID(0, \sigma_\zeta^2) \\
 S_{t+1} &= - \sum_{i=1}^{T-1} S_{t+1-i} + \omega_t & \omega_t &\sim NID(0, \sigma_\omega^2).
 \end{aligned}$$

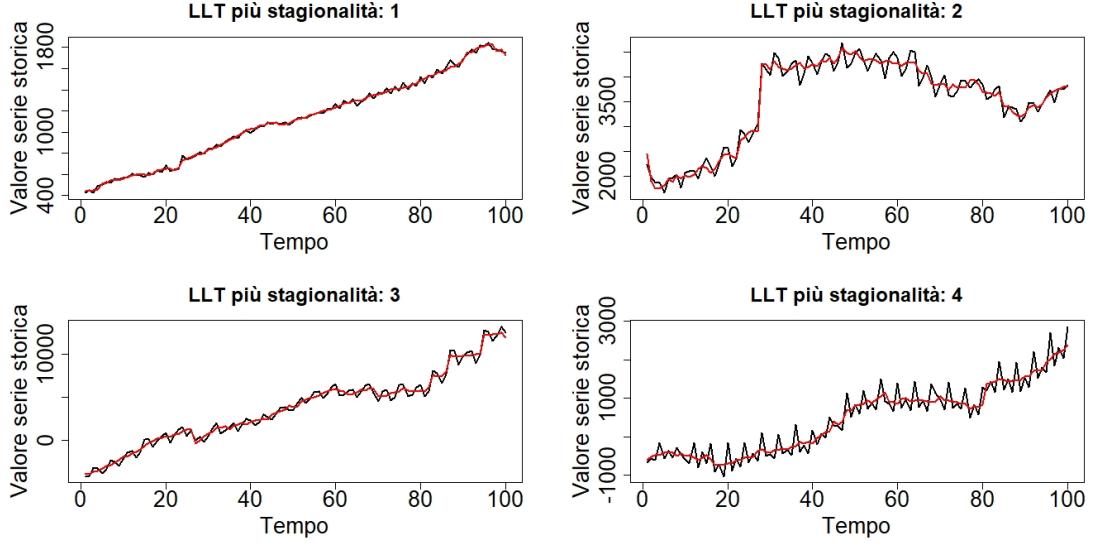


Figura 3.10: Esempio di alcune serie con LLT e stagionalità trimestrale analizzate. In nero è raffigurato la serie storica, mentre in rosso il trend,  $\mu$ .

Mentre la funzione da minimizzare è:

$$\mathcal{L}_{\lambda_\epsilon, \lambda_\eta}(\theta) = -\log(f(y, \theta)) + \lambda_\epsilon \sum_{t=1}^n |\dot{\sigma}_{\epsilon_t}| + \lambda_\eta \sum_{t=1}^n |\dot{\sigma}_{\eta_t}|$$

con  $\theta = (\sigma_\epsilon, \dot{\sigma}_{\epsilon_1}, \dots, \dot{\sigma}_{\epsilon_n}, \sigma_\eta, \dot{\sigma}_{\eta_1}, \dots, \dot{\sigma}_{\eta_n})$ .

MSE: LLT più stagionalità				
	Numerosità	LODE30	LODE50	auxres
Tutte	100	0.903	0.903	0.926
MSE<0.95	41	/	0.778	0.776
SW<5%	64	0.856	0.858	0.880

Tabella 3.9: MSE medio ottenuto su serie modellate da un local linear trend più stagionalità. Gli outlier sono stati generati da una mistura di gaussiane. Le varie categorie indicano: *tutte* comprende tutte le serie simulate, *MSE<0.95* solo quelle ove LODE50 e *auxres* hanno un MSE<0.95, infine *SW<5%* contiene le serie in cui almeno un test di Shapiro-Wilk sui residui ausiliari abbia ottenuto un p-value minore del 5%.

I risultati ottenuti, raccolti nella Tabella 3.9, sono molto simili a quelli presentati per i random walk. Questa volta il metodo LODE un MSE leggermente minore rispetto ad *auxres*, ma nessun test ritiene significativa tale differenza, in quanto è solo in media, a volte *auxres* è meglio di LODE e viceversa. Questa differenza di MSE potrebbe essere

spiegata dal fatto che ora abbiamo altre due componenti da calcolare e su cui possono essere distribuiti i residui, rendendo meno significativi i residui nell'istante dell'outlier per una singola componente.

Cambiando il tipo di serie storiche è cambiato anche il tempo di esecuzione dell'algoritmo. Essendo un modello più complesso, si impiega più tempo per trovare il minimo della funzione e dunque il metodo LODE diventa ancora più lento, infatti impiega circa 5 minuti per analizzare una singola serie storica con queste caratteristiche, mentre con un RW impiega indicativamente 3 minuti. Anche il filtro classico è più lento in questa situazione, ma per tale algoritmo il tempo di esecuzione non è un grosso problema, a differenza del nostro.

### 3.4.4 Cambio di pendenza

Fin'ora ci siamo concentrati ad analizzare outlier che erano additivi o cambi di livello, ma può capitare che essi influenzino la pendenza della serie, provocandone un forte cambio. Questo tipo di disturbi sono molto difficile da individuare, poiché generalmente lo smoother distribuisce l'aumento dei disturbi anche sugli istanti vicini, non creando un vero picco in un unico istante. Il metodo *auxres* per correggere il disturbo pone una variabile a forma di rampa, mentre noi andiamo come sempre a porre una varianza aggiuntiva. Sottolineiamo come in questo caso, se il cambio di pendenza è nell'istante 50, allora la varianza aggiuntiva o il residuo ausiliare, saranno posti o individuati due istanti precedenti, in 48. Andiamo dunque ad analizzare RW integrati che presentano un cambio di pendenza.

Il modello che andiamo ad utilizzare è il seguente:

$$\begin{aligned} y_t &= \mu_t + \epsilon_t & \epsilon_t &\sim NID(0, \sigma_\epsilon^2) \\ \mu_{t+1} &= \mu_t + \beta_t \\ \beta_{t+1} &= \beta_t + \zeta_t & \zeta_t &\sim NID(0, \sigma_\zeta^2 + \dot{\sigma}_\zeta^2). \end{aligned}$$

Mentre la funzione da minimizzare è:

$$\mathcal{L}_{\lambda_\zeta}(\theta) = -\log(f(y, \theta)) + \lambda_\zeta \sum_{t=1}^n |\dot{\sigma}_{\zeta_t}|.$$

Come outlier, per ogni serie ne è stato generato solo uno, sempre a metà serie in posizione 50. Essi sono stati generati compresi tra 5 e 20 volte  $\sigma_\zeta$ , sia positivi che negativi. Sono molto più grandi rispetto agli AO o agli shift generati in precedenza, ma come già detto, è veramente difficile individuare i cambi di pendenza, perciò è stato necessario porre outlier molto "grandi".

Andiamo ora a guardare i risultati. Considerando tutte le serie troviamo che il MSE medio di *auxres* è 0.926, mentre LODE si assesta su un 0.951. Per comprendere meglio questi numeri andiamo ad osservare la loro abilità di individuare il cambio di pendenza. Vista la difficoltà di individuare gli outlier, per questa volta si è deciso di considerarli individuati anche quando la posizione trovata era 49 o 51, ovvero le posizioni limitrofe a quella reale. Fortunatamente, non è mai capitato che i modelli individuassero sia un outlier in 49 e sia uno in 51, ciò avrebbe creato dei problemi nei calcoli.

I risultati presenti nella Tabella 3.10 sono abbastanza differenti con quanto visto per l'individuazione dei jump, i due approcci ora sono molto più simili, infatti *auxres* non ha

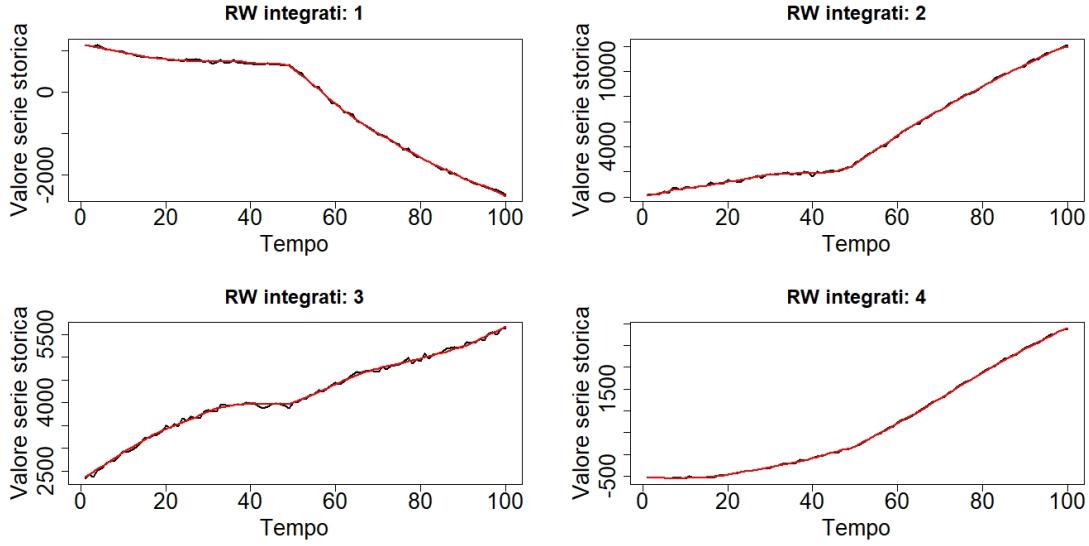


Figura 3.11: Esempio di alcuni random walk integrati analizzati. Per queste serie ricerciamo cambi di pendenza. In nero è raffigurato la serie storica, mentre in rosso il trend,  $\mu$ .

	Cambi di pendenza	
	LODE	auxres
Perfetti	0.50	0.68
Recall	0.70	0.74
Nessuno	0.24	0.21
FP	0.06	0.05
F1	0.67	0.79

Tabella 3.10: Proporzione di serie che soddisfano le rispettive caratteristiche, nell’individuazione dei cambi di pendenza, in serie modellate con random walk integrati. Legenda: *perfetti*, viene selezionato unicamente l’outlier nella posizione corretta; *recall* viene selezionato l’outlier nella posizione corretta; *nessuno*, non viene selezionato alcun outlier; *FP*, nessuno tra gli outlier selezionati è quello corretto; *F1*, misura F1.

più quella tendenza a sbagliare, ma anche lui si astiene. Ricordando che i cambi di pendenza generati sono molto grandi, lo shrinkage del metodo LODE influenza molto le stime, così si giustificano i risultati riguardanti il MSE di prima.

### 3.4.5 Cambio di stagionalità

Un'altra componente su cui si può andare alla ricerca di outlier è la stagionalità. Modellandola attraverso variabili dummy, un fenomeno esterno non influenza un unico istante temporale, ma cambia l'intero periodo, per maggiori informazioni consultare pag. 81 di (Pelagatti 2016). Di conseguenza è necessario cambiare il nostro metodo per far sì che un'unica varianza aggiuntiva influenzi l'intero periodo. Chiamando  $T$  il periodo della stagionalità, abbiamo reso ininfluenti i primi e gli ultimi  $T$  parametri, in quanto impossibile rilevare un cambio di stagionalità in questi istanti. Per i restanti parametri aggiuntivi, che chiameremo  $\pi_t$ , ognuno di essi deve influenzare un intero periodo perciò

$$\begin{aligned}\Pi_t &= \pi_t^2 + \pi_{t-1}^2 + \dots + \pi_{t-T+1}^2 \\ S_{t+1} &= -\sum_{i=1}^{T-1} S_{t+1-i} + \omega_t & \omega_t &\sim NID(0, \sigma_\omega^2 + \Pi_t).\end{aligned}$$

In tal modo la presenza di un cambio di stagionalità sarà modellata da un solo parametro aggiuntivo,  $\pi_t$ , diverso da zero.

Andiamo a considerare delle serie storiche formate da un RW più stagionalità, di conseguenza il nostro modello è il seguente:

$$\begin{aligned}y_t &= \mu_t + S_t + \epsilon_t & \epsilon_t &\sim NID(0, \sigma_\epsilon^2) \\ \mu_{t+1} &= \mu_t + \eta_t & \eta_t &\sim NID(0, \sigma_\eta^2) \\ S_{t+1} &= -\sum_{i=1}^{T-1} S_{t+1-i} + \omega_t & \omega_t &\sim NID(0, \sigma_\omega^2 + \Pi_t).\end{aligned}$$

Mentre la funzione da minimizzare è:

$$\mathcal{L}_{\lambda_\pi}(\theta) = -\log(f(y, \theta)) + \lambda_\pi \sum_{t=1}^n |\pi_t|.$$

con  $\theta = (\sigma_\epsilon, \sigma_\eta, \sigma_\omega, \pi_1, \dots, \pi_n)$ . Per il calcolo delle derivate si veda l'Appendice A.1.3.

Sono state considerate 100 serie storiche trimestrali, quindi con una stagionalità di periodo 4. Come già detto un singolo disturbo va ad influenzare un intero periodo, per cui per gli istanti 50, 51, 52 e 53 si sono considerati disturbi di grandezza da 5 a 10 volte  $\sigma_\omega$  sia positivi che negativi, diversi per ogni istante temporale. Segnaliamo anche un'accortezza che abbiamo dovuto applicare. In genere  $\lambda \in [0.1, 2]$ , ma si è notato che in questo caso un tale spazio parametrico non era ottimale, dunque abbiamo adottato  $\lambda_\pi \in [0.03, 1.3]$ .

Il metodo *auxres* in questo caso è un poco più complesso rispetto al solito. Non basta l'inserimento di una singola variabile dummy, ne servono tante quante è lungo il periodo. Inoltre bisogna tenere conto che l'effetto totale della variabile deve essere pari a zero. Ipotizzando che il cambio di stagionalità avvenga all'istante  $t_0$  e la serie abbia un periodo

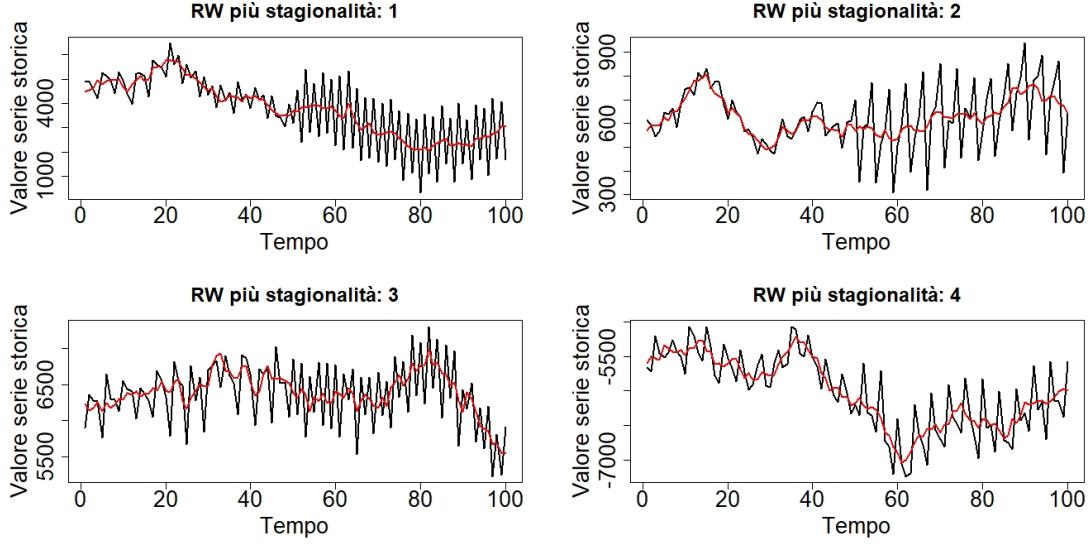


Figura 3.12: Esempio di alcuni random walk più stagionalità. Per queste serie ricerchiamo cambi di stagionalità. In nero è raffigurato la serie storica, mentre in rosso il trend,  $\mu$ .

$T = 4$ , allora si introducono nel modello le seguenti variabili:

$$\begin{aligned}
 s_{it} &= (0, 0, \dots) & t = 1, 2, \dots, t_0 - 1 & i = 1, \dots, 4 \\
 s_{1t} &= (1, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, 1, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, \dots) & t = t_0, \dots, n \\
 s_{2t} &= (-\frac{1}{3}, 1, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, 1, -\frac{1}{3}, -\frac{1}{3}, \dots) & t = t_0, \dots, n \\
 s_{3t} &= (-\frac{1}{3}, -\frac{1}{3}, 1, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, 1, -\frac{1}{3}, \dots) & t = t_0, \dots, n \\
 s_{4t} &= (-\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, 1, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, 1, \dots) & t = t_0, \dots, n
 \end{aligned}$$

ovvero variabili nulle prima del cambio di stagionalità e periodiche successivamente. La somma istante per istante di queste variabili permette di modellare il cambio di stagionalità. I momenti in cui porre queste variabili sono stati come al solito scelti con l'utilizzo dei residui ausiliari. Anche se in questo caso l'individuazione è difficile, in quanto un cambio di stagionalità influenza più istanti temporali, e dunque vi saranno più residui ausiliari con valori anomali.

Come al solito siamo andati a valutare il miglioramento nella stima del trend. Dalla Tabella 3.11 si può osservare come l'errore dei due metodi è pressoché identico a quello classico, tuttavia visto che ricerchiamo outlier sulla stagionalità, osserviamo come varia la sua stima. Calcolando il MSE relativo alla stagionalità, i risultati sono incoraggianti, in quanto si nota un miglioramento rispetto sia a KFS sia ad *auxres*. Probabilmente esso non si ripercuote sulla stima del trend in quanto la stagionalità è sola una delle varie componenti della serie, perciò le ripercussioni sulla stima del trend diventano minime.

Come già visto in precedenza un altro metodo per valutare le performance di un modello è osservare quante volte viene individuato l'outlier, in questo caso il cambio di stagionalità.

MSE: cambio di stagionalità		
	LODE	<i>auxres</i>
sul trend	0.989	1.006
sulla stagionalità	0.811	0.884

Tabella 3.11: MSE medio ottenuto su random walk più stagionalità con outlier su quest’ultima componente. Sono presentati i risultati sia per la stima del trend, sia per la stima della stagionalità.

Abbiamo notato che i diversi approcci difficilmente individuavano l’istante esatto, perciò abbiamo allargato ancora una volta la finestra in cui abbiamo considerato come individuato l’outlier. Abbiamo posto sempre un outlier in posizione 50 e lo abbiamo ritenuto individuato se i vari metodi ne identificavano uno nelle posizioni 49, 50, 51.

Cambio di stagionalità		
	LODE	<i>auxres</i>
Perfetti	0.37	0.23
Recall	0.59	0.48
Nessuno	0.16	0.10
FP	0.25	0.42
F1	0.51	0.43

Tabella 3.12: Proporzione di serie che soddisfano le rispettive caratteristiche, nell’individuazione dei cambi di stagionalità. Legenda: *perfetti*, viene selezionato unicamente l’outlier nella posizione corretta; *recall* viene selezionato l’outlier nella posizione corretta; *nessuno*, non viene selezionato alcun outlier; *FP*, nessuno tra gli outlier selezionati è quello corretto; *F1*, misura F1.

Dalla Tabella 3.12 si nota come i risultati siano ottimi, il metodo LODE ha migliori capacità di individuazione dei cambi di stagionalità di *auxres*. Quest’ultimo, spesso individua più cambi di stagionalità ravvicinati nella medesima serie storica, ciò fa peggiorare le sue performance.

### 3.4.6 Errata ricerca di outlier

Un’ulteriore analisi che abbiamo condotto riguarda la simulazione di una errata ricerca degli outlier. Abbiamo creato serie storiche ponendo outlier solo sulla parte additiva, ma cercandoli anche sul trend. Questo poiché non è sempre facile capire su quale componente sono presenti gli outlier. Infatti i residui delle varie componenti sono correlati, in quanto si tenta di spalmare il disturbo su più componenti.

Abbiamo simulato 100 RW con outlier additivi provenienti da mistura di gaussiane ed invece disturbi del trend perfettamente gaussiani.

MSE: errata ricerca outlier			
	Numerosità	LODE50	<i>auxres</i>
<i>Tutte</i>	100	0.953	1.001
<i>SW&lt;5%</i>	34	0.926	0.992

Tabella 3.13: MSE medio ottenuto su random walk con solo outlier additivi, ma cercando anche cambi di livello.

I risultati presenti nella Tabella 3.13 sono simili a quelli relativi alle serie puramente gaussiane, nella Sezione 3.3.2. Il nostro metodo ha un MSE medio minore, in quanto è più bravo a non porre alcun jump nelle varie serie. Analizzando gli errori di individuazione jump si nota come *auxres* abbia il solito problema di correlazione tra i residui, identificando un jump anche se si tratta di un AO. I risultati ci portano a confermare il nostro metodo come robusto sugli outlier e robusto su un'errata ricerca di essi.

### 3.4.7 Generazione disturbi da t-student

Nelle nostre analisi abbiamo generato gli outlier in diversi modi, tuttavia i metodi non fanno altro che alzare le code della distribuzione gaussiana da cui vengono generati i disturbi. La tipica distribuzione a code pesanti è la t-student, per questo abbiamo provato a generare disturbi da questa distribuzione. La cosa interessante di questa distribuzione è l'oggettiva "contaminazione" della serie, in quanto i gradi di libertà della t-student definiscono in modo preciso quanto i disturbi generati non siano gaussiani.

Abbiamo simulato delle serie con  $\epsilon_t$  e  $\eta_t$  provenienti da t-student con 3, 4, 5, 6 e 7 gradi di libertà, generando 50 serie per ognuno di essi. In tal modo abbiamo verificato l'andamento del MSE al diminuire del numero di outlier.

Osservando la Figura 3.13 notiamo risultati interessanti, perché mostrano proprio il diverso andamento dei due approcci, *auxres* e LODE. Per il primo il MSE continua a crescere, diventando maggiore di uno per le serie con i gradi di libertà più alti, ovvero dove gli outlier sono più piccoli. Per LODE il MSE tende a stabilizzarsi su uno, in quanto in grado di restituire automaticamente la stima del filtro classico quando non vi sono outlier importanti.

Le linee media(LODE, *auxres*) e media(LODE, KFS) verranno spiegate meglio nella Sezione 3.4.8, mentre la linea verde unione(LODE, *auxres*) la commenteremo nella omonima Sezione 3.5.

### 3.4.8 Media tra LODE, *auxres* e KFS

Nel mondo dell'analisi dati si dice sempre un singolo modello non è mai la scelta migliore, perciò abbiamo provato a unire il modello LODE a *auxres* e al filtro di kalman classico.

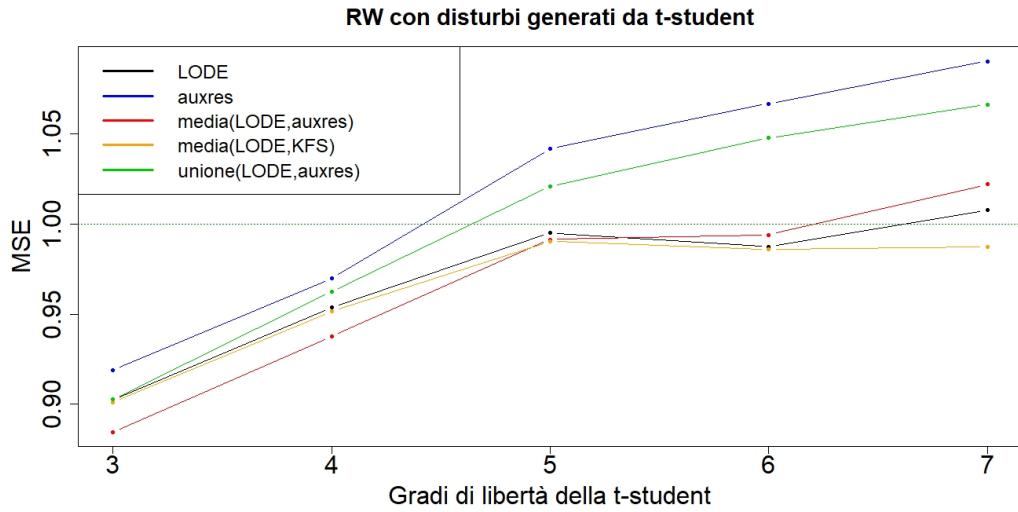


Figura 3.13: MSE all'aumentare dei gradi di libertà della t-student e dunque al diminuire del numero di outlier. Il metodo LODE in questo caso è stato applicato con 30 iterazioni. Nella Tabella 3.14 sono elencati i risultati presenti nella Figura.

MSE: disturbi da t-student		
Gradi di libertà	LODE30	auxres
3	0.903	0.919
4	0.954	0.970
5	0.995	1.042
6	0.987	1.067
7	1.007	1.090

MSE: disturbi da t-student			
Gradi di libertà	media(LODE,auxres)	media(LODE,KFS)	unione(LODE,auxres)
3	0.886	0.901	0.902
4	0.938	0.951	0.962
5	0.991	0.990	1.021
6	0.994	0.986	1.048
7	1.022	0.987	1.066

Tabella 3.14: MSE medio ottenuto con i diversi metodi per RW con disturbi generati da t-student. I risultati in tabella sono i medesimi rappresentati in Figura 3.13.

Il modo più semplice per unirli è la media istante per istante, serie per serie, delle stime del trend. La media aritmetica, ovvero con pesi pari a 0.5, potrebbe non essere la scelta migliore. L'ideale sarebbe assegnare peso maggiore al modello migliore. Per fare questo si possono utilizzare pesi,  $W_i$ , basati sul BIC dei vari modelli (Wagenmakers e Farrell 2004),

$$\Delta_i = BIC_i - \min BIC$$

$$W_i = \frac{\exp(-\frac{1}{2}\Delta_i)}{\sum_j \exp(-\frac{1}{2}\Delta_j)}$$

Siamo andati ad analizzare questo modo di procedere solo per disturbi generati da t-student, visti nella Sezione 3.4.7. Quindi ci riferiremo ai risultati presentati in tale sezione, in particolare alla Figura 3.13 e alla Tabella 3.14.

Concentriamoci sulla media tra LODE e *auxres*. Il nostro metodo in genere ottiene un BIC minore, in quanto è proprio il criterio di scelta del modello, ciò porta a far sì che nella media il metodo LODE sia preponderante. Tuttavia, in alcune serie, ove il metodo LODE performance male, probabilmente anche per problemi di natura numerica, la media può portare a notevoli miglioramenti.

Per questi motivi l'MSE si riduce per gradi di libertà bassi della t-student, come si può vedere dalla Figura 3.13. Il risultato è interessante e ci porta a sostenere che se si vuole stimare il trend di una serie con outlier, è conveniente fare la media tra i due metodi.

I risultati con la media tra LODE e KFS sono diversi. Ovviamente la stima di KFS per serie molto rumorose, non è ottimale, ma è la migliore per serie puramente gaussiane. Il BIC del metodo LODE è sempre minore o uguale di quello del filtro classico, proprio per come viene scelto il miglior  $\lambda$ . Ciò fa sì che ancora una volta il metodo LODE sia preponderante nella media. Questo si evince anche dai risultati, infatti i valori per la media(LODE,KFS) sono molto simili a LODE.

Visto che come detto, al metodo LODE in genere vengono assegnati pesi più alti, si può pensare di fare una semplice media aritmetica. Ciò è stato fatto, ma non riportiamo i risultati, in quanto in media non sono migliori di quelli presentati.

### 3.5 Unione(*LODE*, *auxres*)

Nelle analisi precedenti abbiamo visto come una ricerca degli outlier tramite residui ausiliari sia più efficiente, però il nostro metodo ha ottenuto risultati migliori nella stima della dimensione del jump. Di conseguenza è nata l'idea di individuare gli outlier con i residui ausiliari, per poi inserirli nel modello come varianze aggiuntive, come accade nel nostro algoritmo, e non porli come variabili dummy. In questo caso non si utilizza alcuna regolarizzazione, dunque le varianze aggiuntive sono parametri totalmente liberi.

Se si osserva ancora una volta la Figura 3.13, si notano risultati interessanti. Il MSE medio non è inferiore al metodo LODE, ma è comunque minore di *auxres*. Capiamo il perché di ciò analizzando le stime delle grandezze dei jump.

Dalla Figura 3.14 osserviamo come questo nuovo metodo abbia un errore minore, in quanto ha una stima sempre minore di quella fornita con la variabile dummy. Anche qui probabilmente effettuiamo un shrinkage implicito, ponendo una sorta di prior con media zero. Ciò porta per i motivi già elencati ad un miglior MSE.

Questo metodo è interessante perché migliora l'analisi che si può effettuare basandosi sui residui ausiliari, semplicemente cambiando il modo di modellare gli outlier.

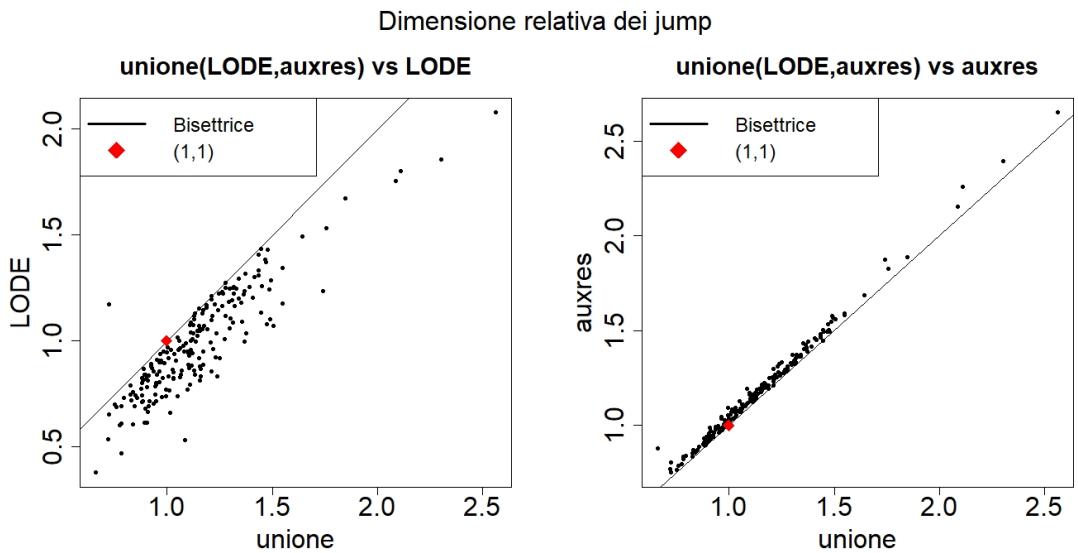


Figura 3.14: Dimensione relativa dei jump individuati. Con dimensione relativa si intende la dimensione stimata diviso la dimensione reale del jump. A sinistra il confronto tra unione e LODE, a destra quello tra unione e *auxres*. In quest’ultima si nota come i punti siano sopra la bisettrice, indicando che porre una varianza aggiuntiva, anche senza shrinkage porta ad una stima minore rispetto a quella della variabile dummy. Mentre nella figura di sinistra si vede l’effetto dello shrinkage del metodo LODE. Il punto rosso è (1,1) ovvero quando ambo i metodi restituiscono una stima perfetta.

### 3.6 Andamento delle varianze aggiuntive

Le varie funzioni di  $R$  che implementano la regressione LASSO in genere riportano come output un grafico del valore dei parametri al variare di  $\lambda$ . Abbiamo cercato di riprodurlo anche nel nostro metodo, per osservare come decadono le varianze aggiuntive. Per chiarezza del grafico si è deciso di usare un solo  $\lambda$ , ovvero ricerchiammo solo cambi di livello, e i valori sono stati fissati tramite griglia.

Osservando il valore delle varianze aggiuntive al variare di  $\lambda$ , si può capire cosa succede al suo crescere e dunque valutare quanto sono significativi gli outlier trovati. Oppure guardando valori minori del  $\lambda$  ottimo si può capire su quali altri istanti possa essere sensato porre un outlier. Queste a nostro avviso sono considerazioni utili per capire meglio le varie scelte che il nostro metodo effettua.

Riportiamo i grafici per due serie, la prima con un unico jump a metà serie, in posizione 50, mentre la seconda ha un trend generato da una t-student con tre gradi di libertà. Per entrambe ribadiamo che cerchiamo unicamente jump e non outlier additivi, per avere un singolo  $\lambda$ .

Analizzando l’immagine a sinistra della Figura 3.15 si nota come il metodo posizioni correttamente la varianza aggiuntiva in posizione 49, visto che il jump è in 50. Inoltre essa è molto significativa, in quanto è necessario un ampio valore di  $\lambda_\eta$  per azzerarla. Interessante è osservare l’effetto dello shrinkage. Sebbene la varianza citata sia sempre significativa,

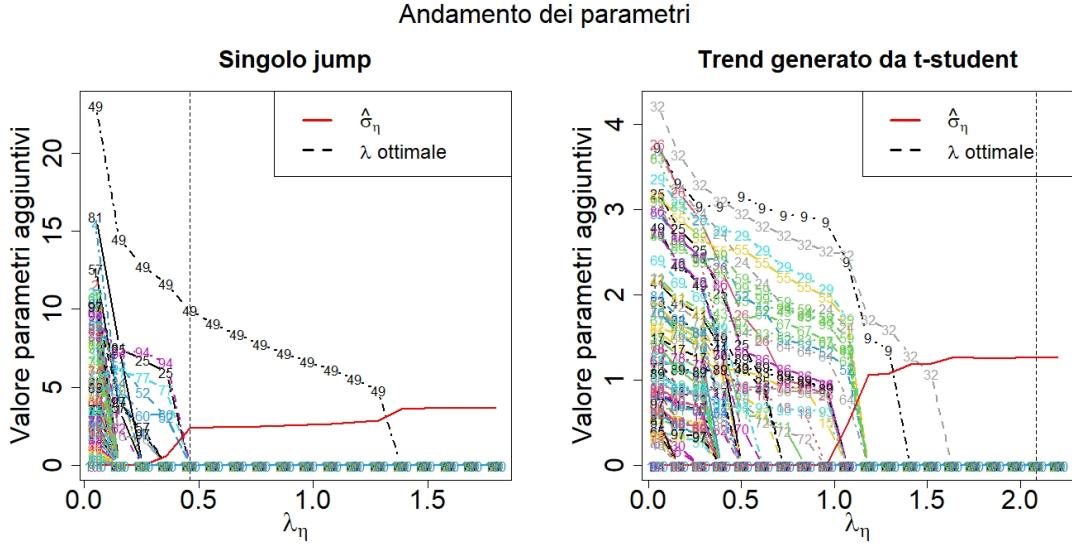


Figura 3.15: Valore dei parametri aggiuntivi all’incrementare di  $\lambda_\eta$ . In rosso è raffigurata il valore di  $\hat{\sigma}_\eta$ , la radice della varianza costante. La linea verticale tratteggiata rappresenta il valore di  $\lambda$  con il minore BIC, quindi il valore scelto. Ogni parametro aggiuntivo è rappresentato da una linea tratteggiata con sopra un numero che rappresenta il rispettivo istante temporale che va ad influenzare.

essa decresce, ciò porterà alle conseguenze descritte nella Sezione 3.3.4. La linea verticale rappresenta il modello con il BIC minore, che si ottiene con un unica varianza aggiuntiva, all’istante 49. Anche alcuni  $\lambda$  successivi a quello scelto hanno questa caratteristica, ma lo shrinkage abbassa la verosimiglianza del modello, per cui il BIC sceglie il valore di  $\lambda$  più basso a parità di numero di parametri aggiuntivi diversi da zero.

Osservando invece l’immagine di destra si nota come nessun jump venga individuato. Il  $\lambda$  scelto è uno dei tanti che pone a zero tutte le varianze aggiuntive, uno vale l’altro, in quanto la stima è la medesima, identica a quella del filtro classico. Il grafico evidenzia dei possibili shift negli istanti 10 e 33, poiché sono le ultime varianze aggiuntive ad essere poste pari a zero. Infine osserviamo la linea rossa continua, che rappresenta  $\hat{\sigma}_\eta$ . Essa è monotona non decrescente, infatti all’inizio tutti i disturbi sono spiegati tramite le varianze aggiuntive, poi man mano che vengono azzerate è necessario aumentare la stima di  $\sigma_\eta$  per spiegare i disturbi. Dal grafico si nota come i parametri abbiano una discesa lenta e graduale, tranne che all’ultimo step, quando vengono messi a zero. Ciò è spiegabile proprio comparando il loro valore con quello di  $\hat{\sigma}_\eta$ . Nell’istante prima di essere zero hanno un valore simile a  $\hat{\sigma}_\eta$ , per cui ridurli ancora di un poco non ha senso, in quanto non avrebbero più alcuna influenza nei calcoli dello smoother, visto che si sommano le varianze, ovvero i parametri del grafico al quadrato. Il metodo perciò preferisce porli direttamente a zero, per evitare la penalità nella funzione da minimizzare,  $\mathcal{L}$ .

Sottolineiamo che fin’ora abbiamo sempre parlato del valore assoluto dei parametri, non tenendo conto del segno, in quanto nei calcoli entra in gioco unicamente il loro quadrato.

### 3.7 Confronto con STAMP e SAS

In questa sezione andremo a confrontare il nostro metodo con quelli presenti nei principali software per l’analisi dei modelli a componenti non osservabili. Sia STAMP che SAS offrono la possibilità di ricercare automaticamente gli outlier, ma solo quelli additivi o i cambi di livello. STAMP lo abbiamo già presentato, ed ora valuteremo le sue performance. Per quanto riguarda SAS andremo ad usare la procedura UCM ([The UCM Procedure](#)) che permette di implementare i modelli a componenti non osservabili e di andare a ricercare outlier additivi e level shift attraverso l’algoritmo descritto in (de Jong e Penzer [1998](#)). Se in STAMP la modellizzazione degli outlier è automatica, in SAS non lo è, per cui, per ogni serie storica abbiamo salvato gli outlier individuati e poi modellizzati nel modo opportuno, attraverso variabili dummy.

Poiché per entrambi i software è necessaria una parte "manuale", per salvare i risultati e trasportarli in *R*, dove è stata condotta tutto il resto dell’analisi, non si sono potuto analizzare grandissimi quantità di serie storiche. Come in precedenza anche per questi metodi abbiamo analizzato le performance sotto due aspetti: la stima del trend e l’individuazione di outlier. Abbiamo generato 50 random walk con la mistura di gaussiane per valutare il MSE nella stima del trend e successivamente generato altre 50 serie come descritto nella Sezione 3.3.3 per valutare la loro abilità nell’identificare gli outlier. Entrambi i software non permettono l’identificazione di soltanto i level shift, ma è necessario individuare anche gli AO. Per SAS questo non è un problema in quanto le due identificazioni sono indipendenti e dunque prendiamo i risultati riguardi unicamente i cambi di livello. Per STAMP la questione è diversa, esso implementa un algoritmo ignoto per cui la selezione degli shift dipende anche dalle selezione degli outlier additivi, per cui per tale software saranno riportati unicamente i risultati relativi al MSE. Inoltre in SAS si è deciso di fissare un p-value dell’1% di significatività degli outlier, lo stesso usato in *auxres*. Il software spesso individuava level shift contigui, cosa alquanto improbabile, perciò in questi casi abbiamo considerato solo il più significativo.

MSE: confronto con STAMP e SAS					
	Numerosità	LODE50	<i>auxres</i>	STAMP	SAS
Tutte	50	0.846	0.854	0.836	0.876

Tabella 3.15: MSE medio ottenuto su random walk con outlier provenienti da una mistura di gaussiane. Sono presenti anche i risultati ottenuti usando i software STAMP e SAS.

La Tabella 3.15 confronta tutti i risultati ottenuti con i vari metodi. Il dato più interessante è il MSE di STAMP, che è il migliore, e soprattutto è meglio di *auxres*. La differenza non è eccessiva, ma comunque evidenzia il fatto che l’algoritmo implementato in STAMP probabilmente è un poco più efficace rispetto al nostro tentativo di copiarlo con *auxres*. Ci rendiamo conto che forse 50 serie sono un po’ poche, per un confronto così importante, però, come già detto, si impiega molto tempo ad analizzare le serie con STAMP.

Riportiamo ora i risultati relativi all’individuazione dei cambi di livello, con i vari metodi.

Cambi di livello			
	LODE	<i>auxres</i>	SAS
Perfetti	0.32	0.42	0.44
Recall	0.50	0.66	0.56
Nessuno	0.44	0.14	0.14
FP	0.06	0.20	0.30
F1	0.55	0.60	0.54

Tabella 3.16: Proporzione di serie che soddisfano le rispettive caratteristiche, nell’individuazione dei cambi di livello. Sono presenti anche i risultati ottenuti usando il software SAS. Legenda: *perfetti*, viene selezionato unicamente l’outlier nella posizione corretta; *recall* viene selezionato l’outlier nella posizione corretta; *nessuno*, non viene selezionato alcun outlier; *FP*, nessuno tra gli outlier selezionati è quello corretto; *F1*, misura F1.

Dalla Tabella 3.16 si nota come l’algoritmo di ricerca di outlier implementato in SAS sia un poco peggio rispetto ad *auxres*, anch’esso è propenso a porre numerosi outlier, ma purtroppo con scarsa efficacia. Il metodo LODE rimane sempre il più parsimonioso.

Dopo aver visto i vari risultati possiamo ritenerci soddisfatti, in quanto le prestazioni dei due metodi implementati da parte nostra, LODE e *auxres*, sono in linea con quelli dei principali software per le analisi di serie storiche.

**Errori numerici** Facciamo ora un piccolo approfondimento riguardante gli errori numerici che i software restituiscono. Abbiamo già ampiamente parlato del problema riscontrato con il pacchetto *KFAS* ([Package KFAS](#)), ma che in genere non si presentava quando analizzavamo random walk. Inaspettatamente STAMP presenta tali problematiche. Su ben due serie provate il software ha restituito errore, per la precisione “*collapse to Kalman filter has failed*”. Serie sulle quali i nostri metodi non avevano avuto alcun problema. Perciò abbiamo dovuto scartare tali serie e generarne delle altre. Tutto ciò per sottolineare che i problemi numerici di cui il metodo LODE soffre sono comuni anche agli altri metodi, anche a software molto importanti come STAMP, sviluppato da Koopman, un grandissimo esperto nei modelli a componenti non osservabili.

## 3.8 Comportamento su serie storiche reali

In questa ultima sezione andiamo ad analizzare il comportamento del metodo LODE su serie storiche reali. Ne prendiamo una per ciascun tipo di modello, scegliendo tra le più famose.

### 3.8.1 Nilo

Iniziamo con la serie storiche della portata annua d’acqua del Nilo, l’abbiamo già usata come esempio in precedenza ed è presente come serie di prova per numerosi metodi (Commandeur, Koopman e Ooms 2011). Essa è famosa perché a partire dall’anno 1899 è presente l’effetto della costruzione della diga di Assuan che ha fatto decrescere la portata del fiume. Questo è un esempio di serie storica modellizzata attraverso un random walk.

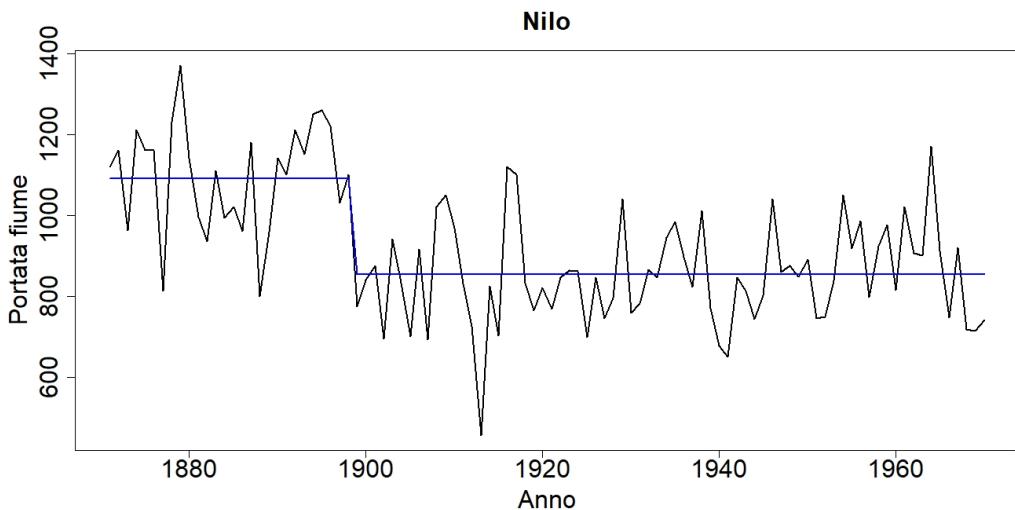


Figura 3.16: In nero la serie storica della portata del Nilo. In blu la stima del trend ottenuta col metodo LODE.

Abbiamo applicato il metodo LODE, con 50 iterazioni, a questa serie ed i risultati sono soddisfacenti, si veda la Figura 3.16. Viene individuato il cambio di livello dovuto alla diga di Assuan e conseguentemente il trend viene stimato a gradino, esattamente come diversi esperti dicono che dovrebbe essere. Tuttavia non viene identificato l’outlier additivo nell’anno 1913, dovuto all’eruzione del vulcano Katmai (Mendelssohn 2011), che ha causato un forte abbassamento del livello del fiume, e che dunque si potrebbe considerare come outlier additivo. Al seguente link [LODE Nilo<sup>3</sup>](#) si può trovare il codice totalmente riproducibile utilizzato per questa analisi.

### 3.8.2 Incidenti stradali UK

Come secondo esempio proponiamo una serie storica ben studiata in (Harvey e Koopman 1992), riguardante il numero automobilisti uccisi o gravemente feriti in Gran Bretagna. Nel febbraio del 1983 è stata introdotta la legge che ha reso obbligatorio l’utilizzo della cintura, ciò ha provocato una netta decrescita degli incidenti gravi. Sono disponibili diverse serie storiche a riguardo, a seconda della posizione del passeggero in auto. Noi andiamo ad analizzare sempre un problema univariato, considerando unicamente la serie storica

<sup>3</sup>[https://github.com/simone-frigerio/LODE/blob/main/LODE\\_Nilo.R](https://github.com/simone-frigerio/LODE/blob/main/LODE_Nilo.R)

dei guidatori. Inoltre prendiamo i dati fino al dicembre 1984, partendo dal luglio 1975, per evitare i dati influenzati dalle crisi del petrolio di qualche anno prima. Consideriamo anche il logaritmo della serie, sempre come suggerito in (Harvey e Koopman 1992). Essa rappresenta un esempio di dati modellati attraverso un local linear trend ed una stagionalità mensile.

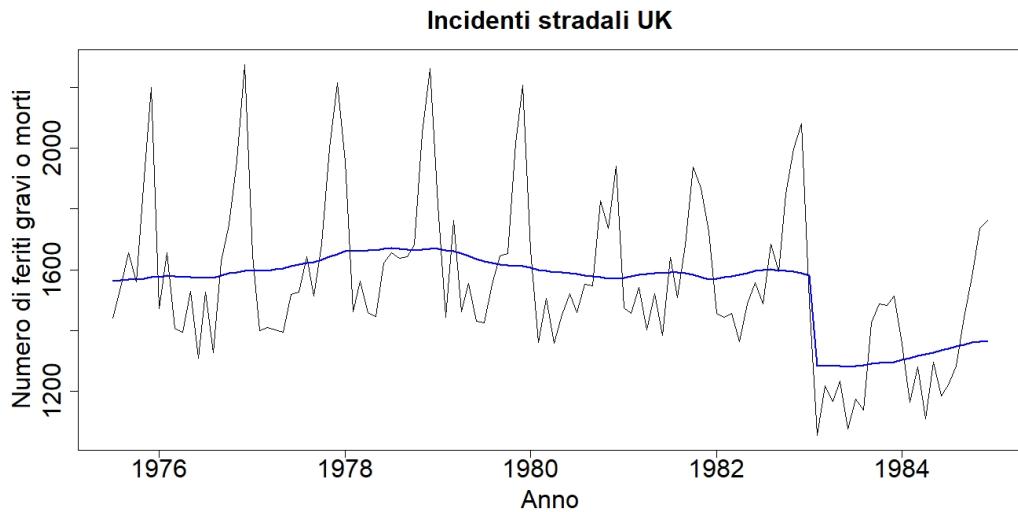


Figura 3.17: Serie storica del numero di automobilisti uccisi o gravemente feriti in UK. In blu la stima del trend ottenuta col metodo LODE.

Il metodo LODE è in grado di individuare correttamente il cambio di livello dovuto all'introduzione delle cinture di sicurezza (Figura 3.17), ponendo una varianza aggiuntiva a gennaio 1983, che va di conseguenza ad influenzare l'istante successivo, ovvero febbraio 1983, quando è stata introdotta la legge sulle cinture di sicurezza. Ancora una volta non viene identificato alcun outlier additivo, a differenza di come capita in (Harvey e Koopman 1992).

### 3.8.3 Inventario contabile USA

Andiamo ora alla ricerca di cambi di pendenza. Consideriamo la serie storica degli inventari contabili di tutte le aziende in USA. Noi trattiamo la serie già destagionalizzata, dal gennaio 1992 a dicembre 2016, con periodicità mensile. Essa presenta dei cambi di pendenza dovuti alla recessione del 2008. I dati sono quasi interamente segnale, è dunque molto facile individuare i cambi di pendenza, tuttavia il nostro è un semplice esempio esplicativo, per mostrare che il metodo LODE funziona su serie reali.

Applicando il nostro approccio alla serie si individuano due cambi di pendenza, nel settembre 2008 e ottobre 2009 (Figura 3.18).

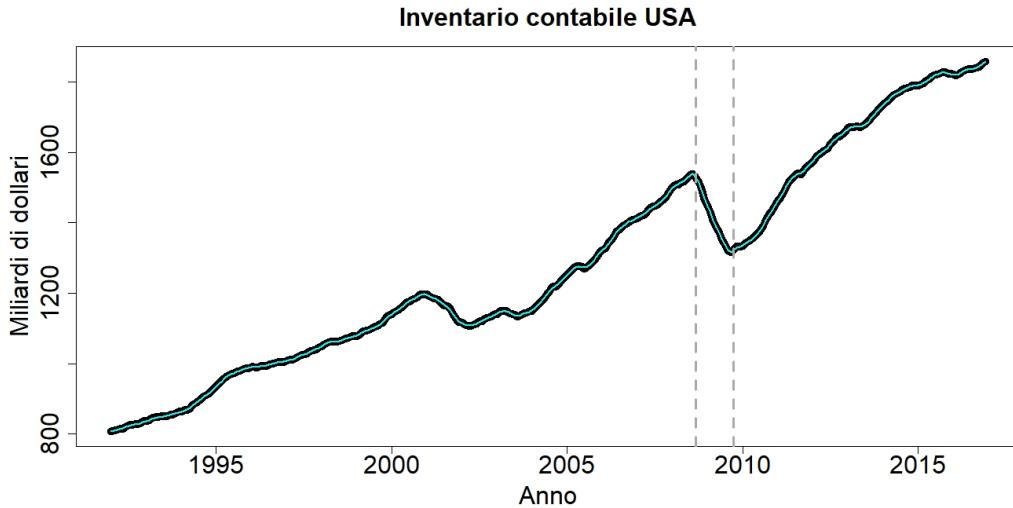


Figura 3.18: Serie storica dell’ammontare in dollari dell’inventario nelle aziende statunitensi. Le righe verticali indicano gli istanti in cui sono avvenuti cambi di pendenza. In nero, con un tratto molto largo, la serie storica, in azzurro la stima del trend, che copia l’andamento dei dati osservati, in quanto la serie è quasi interamente segnale.

### 3.8.4 Matrimoni in Inghilterra e Galles

Come ultimo esempio proponiamo la serie storica trimestrale dei matrimoni in Inghilterra (ENG) e Galles (WLS), proposta anche da Pelagatti nel suo libro, pag. 82 di (Pelagatti 2016). Essa è rilevante perché presenta l’influenza di una nuova legge che ha cambiato la tassazione dei matrimoni, non favorendo più i matrimoni in inverno. Ciò ha provocato un netto cambio nella stagionalità della serie storica, a partire dal 1969.

Abbiamo applicato il metodo LODE anche a questa serie storica, Figura 3.19. Esso è in grado di individuare il cambio di stagionalità del primo trimestre del 1969. Oltre ad esso identifica un altro cambio di stagionalità, all’inizio del 1950, quando è stata introdotta un’importante legge sui matrimoni.

## 3.9 Conclusioni

Abbiamo visto in tutte queste simulazioni come il nuovo metodo che proponiamo ottenga ottime performance. I risultati sono al pari dei metodi più consolidati basati sui residui ausiliari. Inoltre nelle serie a gradini, in cui  $\sigma_\eta$  è pari a zero, abbiamo ottenuto risultati significativamente migliori rispetto ad *auxres*.

Il metodo LODE non si comporta male anche nelle serie ove non vi sono outlier, ciò significa che prima di applicare questo approccio non è per forza richiesta un’analisi preliminare della serie, per stabilire se vi sia qualche valore anomalo o meno.

Il principale difetto di questo metodo è senza dubbio la lentezza, impiegare qualche minuto, o spesso anche più, per analizzare una serie storica è un fattore che influenza

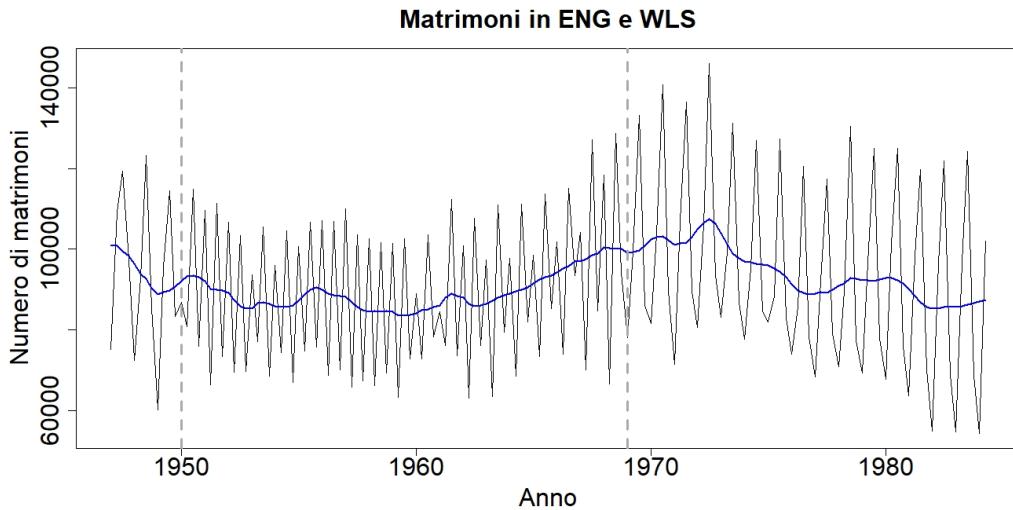


Figura 3.19: Serie storica del numero dei matrimoni in Galles e in Inghilterra. In blu la stima del trend ottenuta col metodo LODE ed in grigio gli istanti si cui si sono verificati i cambi di stagionalità.

molto le sue possibilità di utilizzo. Tuttavia, come spesso si dice in questi casi, il costo computazionale sarà sempre meno un limite col progredire della tecnologia.

Nel complesso possiamo ritenerci soddisfatti dei risultati ottenuti, per questo pensiamo in futuro di creare un pacchetto *R* cosicché chiunque possa utilizzare il metodo. Inoltre l'approccio può essere ulteriormente sviluppato, cercando di ridurre il tempo computazionale e allargando il metodo ad altri tipi di outlier e modelli.

# Appendice A

## Calcolo derivate e gradi di libertà per altre tipologie di modelli

### A.1 Calcolo delle derivate

#### A.1.1 LLT con stagionalità

Riportiamo i calcoli per trovare le derivate nel modello local linear trend più stagionalità a dummy stocastiche di periodo  $T$ , con ricerca degli outlier additivi level shift.

$$\begin{aligned} y_t &= \mu_t + S_t + \epsilon_t & \epsilon_t &\sim NID(0, \sigma_\epsilon^2 + \dot{\sigma}_{\epsilon_t}^2) \\ \mu_{t+1} &= \mu_t + \beta_t + \eta_t & \eta_t &\sim NID(0, \sigma_\eta^2 + \dot{\sigma}_{\eta_t}^2) \\ \beta_{t+1} &= \beta_t + \zeta_t & \zeta_t &\sim NID(0, \sigma_\zeta^2) \\ S_{t+1} &= - \sum_{i=1}^{T-1} S_{t+1-i} + \omega_t & \omega_t &\sim NID(0, \sigma_\omega^2). \end{aligned}$$

La funzione da minimizzare è:

$$\mathcal{L}_{\lambda_\epsilon, \lambda_\eta}(\theta) = -\log(f(y, \theta)) + \lambda_\epsilon \sum_{t=1}^n |\dot{\sigma}_{\epsilon_t}| + \lambda_\eta \sum_{t=1}^n |\dot{\sigma}_{\eta_t}|$$

con  $\theta = (\sigma_\epsilon, \dot{\sigma}_{\epsilon_1}, \dots, \dot{\sigma}_{\epsilon_n}, \sigma_\eta, \dot{\sigma}_{\eta_1}, \dots, \dot{\sigma}_{\eta_n})$ .

Per il calcolo delle derivate applichiamo la formula 3.2 dell'articolo (Koopman e Shepard 1992), ove  $\lambda_i$  sono i vari parametri:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \sigma_\epsilon} &= - \sum_{t=1}^n (e_t e'_t - D_t) \sigma_\epsilon \\ \frac{\partial \mathcal{L}}{\partial \sigma_\eta} &= - \sum_{t=1}^n \text{tr} \left( (r_{t-1} r'_{t-1} - N_{t-1}) \cdot \begin{pmatrix} \sigma_\eta & 0 & 0 \end{pmatrix} \right) \end{aligned}$$

ove  $r_t$  e  $N_t$  sono matrici.

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \sigma_\zeta} &= -\sum_{t=1}^n \text{tr} \left( (r_{t-1} r'_{t-1} - N_{t-1}) \cdot \begin{pmatrix} 0 & \sigma_\zeta & \\ & 0 & \end{pmatrix} \right) \\ \frac{\partial \mathcal{L}}{\partial \sigma_\omega} &= -\sum_{t=1}^n \text{tr} \left( (r_{t-1} r'_{t-1} - N_{t-1}) \cdot \begin{pmatrix} 0 & & \\ & 0 & \\ & & \sigma_\omega \end{pmatrix} \right) \\ \frac{\partial \mathcal{L}}{\partial \dot{\sigma}_{\epsilon_t}} &= -(e_t e'_t - D_t) \dot{\sigma}_{\epsilon_t} + \lambda_\epsilon \cdot \text{sign}(\dot{\sigma}_{\epsilon_t}) \\ \frac{\partial \mathcal{L}}{\partial \dot{\sigma}_{\eta_t}} &= -\text{tr} \left( (r_{t-1} r'_{t-1} - N_{t-1}) \cdot \begin{pmatrix} \dot{\sigma}_{\eta_t} & 0 & \\ & 0 & \\ & & 0 \end{pmatrix} \right) + \lambda_\eta \cdot \text{sign}(\dot{\sigma}_{\eta_t})\end{aligned}$$

### A.1.2 RW integrato con outlier sullo slope

Riportiamo i calcoli per trovare le derivate nel modello random walk integrato con outlier che provocano cambi di pendenza:

$$\begin{aligned}y_t &= \mu_t + \epsilon_t & \epsilon_t &\sim NID(0, \sigma_\epsilon^2) \\ \mu_{t+1} &= \mu_t + \beta_t \\ \beta_{t+1} &= \beta_t + \zeta_t & \zeta_t &\sim NID(0, \sigma_\zeta^2 + \dot{\sigma}_{\zeta_t}^2).\end{aligned}$$

La funzione da minimizzare è:

$$\mathcal{L}_{\lambda_\zeta}(\theta) = -\log(f(y, \theta)) + \lambda_\zeta \sum_{t=1}^n |\dot{\sigma}_{\zeta_t}|$$

con  $\theta = (\sigma_\epsilon, \sigma_\eta, \sigma_\zeta, \sigma_{\zeta_1}, \dots, \dot{\sigma}_{\zeta_n})$ .

Per il calcolo delle derivate applichiamo la formula 3.2 dell'articolo (Koopman e Shepard 1992), ove  $\lambda_i$  sono i vari parametri:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \sigma_\epsilon} &= -\sum_{t=1}^n (e_t e'_t - D_t) \sigma_\epsilon \\ \frac{\partial \mathcal{L}}{\partial \sigma_\eta} &= -\sum_{t=1}^n \text{tr} \left( (r_{t-1} r'_{t-1} - N_{t-1}) \cdot \begin{pmatrix} \sigma_\eta & \\ & 0 \end{pmatrix} \right) \\ \frac{\partial \mathcal{L}}{\partial \sigma_\zeta} &= -\sum_{t=1}^n \text{tr} \left( (r_{t-1} r'_{t-1} - N_{t-1}) \cdot \begin{pmatrix} 0 & \\ & \sigma_\zeta \end{pmatrix} \right) \\ \frac{\partial \mathcal{L}}{\partial \dot{\sigma}_{\zeta_t}} &= -\text{tr} \left( (r_{t-1} r'_{t-1} - N_{t-1}) \cdot \begin{pmatrix} 0 & \\ & \dot{\sigma}_{\zeta_t} \end{pmatrix} \right) + \lambda_\zeta \cdot \text{sign}(\dot{\sigma}_{\zeta_t})\end{aligned}$$

### A.1.3 RW con outlier su stagionalità

Riportiamo il calcolo delle derivate per il modello descritto nella sottosezione 3.4.5, che qui riportiamo:

$$\begin{aligned} y_t &= \mu_t + S_t + \epsilon_t & \epsilon_t &\sim NID(0, \sigma_\epsilon^2) \\ \mu_{t+1} &= \mu_t + \eta_t & \eta_t &\sim NID(0, \sigma_\eta^2) \\ S_{t+1} &= -\sum_{i=1}^{T-1} S_{t+1-i} + \omega_t & \omega_t &\sim NID(0, \sigma_\omega^2 + \Pi_t) \\ \Pi_t &= \pi_t^2 + \pi_{t-1}^2 + \dots + \pi_{t-T+1}^2 \end{aligned}$$

con  $\pi_i$  i parametri aggiuntivi. La funzione da minimizzare è:

$$\mathcal{L}_{\lambda_\eta}(\pi) = -\log(f(y, \theta)) + \lambda_\pi \sum_{t=1}^n |\pi_t|$$

con  $\theta = (\sigma_\epsilon, \sigma_\eta, \sigma_\omega, \pi_1, \dots, \pi_n)$ .

Per il calcolo delle derivate applichiamo la formula 3.2 dell'articolo (Koopman e Shepard 1992), ove  $\lambda_i$  sono i vari parametri:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \sigma_\epsilon} &= -\sum_{t=1}^n (e_t e'_t - D_t) \sigma_\epsilon \\ \frac{\partial \mathcal{L}}{\partial \sigma_\eta} &= -\sum_{t=1}^n \text{tr} \left( (r_{t-1} r'_{t-1} - N_{t-1}) \cdot \begin{pmatrix} \sigma_\eta & 0 \end{pmatrix} \right) \\ \frac{\partial \mathcal{L}}{\partial \sigma_\omega} &= -\sum_{t=1}^n \text{tr} \left( (r_{t-1} r'_{t-1} - N_{t-1}) \cdot \begin{pmatrix} 0 & \sigma_\omega \end{pmatrix} \right) \\ \frac{\partial \mathcal{L}}{\partial \pi_t} &= -\sum_{i=1}^{T-1} \text{tr} \left( (r_{i-1} r'_{i-1} - N_{i-1}) \cdot \begin{pmatrix} 0 & \pi_t \end{pmatrix} \right) + \lambda_\pi \cdot \text{sign}(\pi_t) \end{aligned}$$

## A.2 Calcolo dei gradi di libertà

Quando andiamo alla ricerca di cambi di pendenza o stagionalità le formule per calcolare i gradi di libertà sono le seguenti:

- nel caso di RW integrati su cui ricerchiamo cambi di pendenza definiamo

$$df = \sum_t \left( \dot{\sigma}_{\zeta_t} > \frac{\sigma_\zeta}{100} \right) + \left( \sigma_\zeta > \frac{\sigma_\zeta}{100} \right) + \left( \sigma_\epsilon > \frac{\sigma_\epsilon}{100} \right)$$

- nel caso di RW più stagionalità su cui ricerchiamo cambi di stagionalità definiamo

$$df = \sum_t \left( \pi_t > \frac{\sigma_\omega}{100} \right) + \left( \sigma_\omega > \frac{\sigma_\omega}{100} \right) + \left( \sigma_\eta > \frac{\sigma_\eta}{100} \right) + \left( \sigma_\epsilon > \frac{\sigma_\epsilon}{100} \right).$$

## Appendice B

# Ulteriori dettagli su *auxres*

Lo sviluppo del metodo che abbiamo denominato *auxres* ha richiesto diverse ore di lavoro, principalmente per la sua implementazione come codice *R*. In questa appendice andiamo a descrivere alcune scelte fatte ed alcuni problemi riscontrati.

**Soglia per outlier** Indubbiamente la scelta che influenza maggiormente i risultati riguarda la soglia oltre la quale considerare i residui ausiliari come outlier. Ricordiamo che noi abbiamo deciso di utilizzare un intervallo di confidenza(CL) del 99% e dunque una soglia di circa 2.5. In tal modo, considerando una serie senza outlier di 100 istanti temporali, in media un residuo ausiliare supera la soglia. Ciò potrebbe indurre a pensare di alzare la soglia, in modo da considerare meno outlier. Abbiamo provato a farlo, considerando un CL del 99.5%, purtroppo i risultati peggioravano significativamente sulle serie con gli outlier, in quanto non più in grado di identificarli, se non quelli "giganteschi". Al contrario, ridurre l'intervallo di confidenza, porta vantaggi quando ci sono molti outlier, ma si ottengono performance orribili su serie meno contaminate. Per questi motivi si è deciso di usare un CL del 99%, per avere ottime performance su tutti i tipi di serie.

**Cosa fa STAMP** Il metodo *auxres* dovrebbe essere una copia di quello implementato in STAMP, per questo si è cercato di fare un'operazione di *ingegneria inversa* per scoprire come funzioni. L'analisi non ha dato molti frutti, si è capito che anche STAMP utilizza più o meno una soglia di 2.5, tuttavia la selezione di outlier additivi e cambi di livello è sempre correlata. L'algoritmo implementato nel software non soffre della problematica di porre sia AO che LS nel medesimo istante temporale. Inoltre per qualche motivo quando vi sono numerosi residui superiori alla soglia di 2.5, non tutti vengono considerati outlier.

**Condizioni iniziali per le variabili** Una piccola accortezza di cui abbiamo dovuto tenere conto riguarda le condizioni iniziali per le variabili che modellano gli outlier. Lasciando condizioni diffuse alcune volte il codice restituiva un warning, per cui in questi casi si è deciso di fissare la varianza iniziale,  $P_1$ , pari ad un numero "grande", come 100 volte la varianza della serie.

**Esempio di codice** Tutti gli ulteriori dettagli del metodo possono essere visionati attraverso un esempio di codice che riportiamo al seguente link [auxres Nilo](#)<sup>1</sup>. Esso contiene il metodo *auxres* applicato alla serie storica della portata d'acqua del Nilo.

---

<sup>1</sup>[https://github.com/simone-frigerio/LODE/blob/main/auxres\\_Nilo.R](https://github.com/simone-frigerio/LODE/blob/main/auxres_Nilo.R)

# Bibliografia

- Candelieri, A. (2021). «Sequential model based optimization of partially defined functions under unknown constraints». In: *Journal of Global Optimization* 79.
- Carnell, R. (2020). *Basic Latin hypercube samples and designs with package lhs*. URL: [https://cran.r-project.org/web/packages/lhs/vignettes/lhs\\_basics.html](https://cran.r-project.org/web/packages/lhs/vignettes/lhs_basics.html).
- Commandeur, J. F., S. J. Koopman e M. Ooms (2011). «Statistical Software for State Space Methods». In: *Journal of Statistical Software, Articles* 41.1, pp. 1–18.
- De Jong, P. e J. Penzer (1998). «Diagnosing Shocks in Time Series». In: *Journal of the American Statistical Association* 93.442, pp. 796–806.
- Frazier, P. I. (2018). «A Tutorial on Bayesian Optimization». In: *arXiv e-prints*.
- Harvey, A. C. e S. J. Koopman (1992). «Diagnostic Checking of Unobserved-Components Time Series Models». In: *Journal of Business Economic Statistics* 10.4, pp. 377–389.
- Helske, J. *Package KFAS*. URL: <https://cran.r-project.org/web/packages/KFAS/KFAS.pdf>.
- Institute, SAS. *The UCM Procedure*. URL: [https://documentation.sas.com/doc/en/pgmsascdc/9.4\\_3.4/etsug/etsug\\_ucm\\_toc.htm](https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.4/etsug/etsug_ucm_toc.htm).
- Koopman, S. J. e N. Shepard (1992). «Exact score for time series models in state space form». In: *Biometrika* 79, pp. 823–826.
- Mebane, W. e J. Sekhon. *Package rgenoud*. URL: <https://cran.r-project.org/web/packages/rgenoud/rgenoud.pdf>.
- Mendelsohn, R. (2011). «The STAMP Software for State Space Models». In: *Journal of Statistical Software* 41.
- Pelagatti, M. M. (2016). *Time Series Modelling with Unobserved Components*. A cura di A Chapman Hall Book.
- Wagenmakers, E. J. e S. Farrell (2004). «AIC model selection using Akaike weights». In: *Psychonomic Bulletin Review* 11, pp. 192–196.
- Zou, H., T. Hastie e R. Tibshirani (2007). «On the degrees of freedom of the lasso». In: *The annals of statistics* 35, pp. 2173–2192.