

UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI SCIENZE STATISTICHE

**CORSO DI LAUREA SPECIALISTICA IN
STATISTICA ED INFORMATICA**

TESI DI LAUREA

**SULLA STIMA DEGLI
IPERPARAMETRI
DI UN MODELLO LINEARE
GENERALIZZATO DINAMICO**

RELATORE: CH.MO PROF. ADELCHI AZZALINI

LAUREANDO: TOMAS DAL CAPPELLO

ANNO ACCADEMICO 2005-06

*“Dubitare di tutto
e credere a tutto
sono due soluzioni
ugualmente comode
che ci dispensano,
l’una come l’altra,
dal riflettere.”*

JULES HENRI POINCARÉ

Indice

1	Introduzione	1
2	Filtro di Kalman e sua estensione a famiglie esponenziali	3
2.1	Modelli nello spazio degli stati di tipo lineare	4
2.2	Filtro di Kalman lineare	9
2.3	Modelli nello spazio degli stati di famiglia esponenziale	11
2.4	Estensione del filtro di Kalman a famiglie esponenziali	15
2.4.1	Stima della moda a posteriori e verosimiglianza penalizzata	15
2.4.2	Derivazione dell'algoritmo	17
3	Stima degli iperparametri	25
3.1	Algoritmo di tipo EM	25
3.2	Altri metodi di stima	31
3.3	Modifica all'algoritmo di tipo EM	32
3.4	Nota sulla verosimiglianza	36
4	Applicazioni illustrative	39
4.1	Aspetti computazionali	39
4.1.1	Mantenimento della simmetria delle matrici delle varianze	39
4.1.2	Criteri di convergenza	43
4.1.3	Modo di utilizzo delle funzioni di R	43
4.2	Esempio 1: precipitazioni a Tokyo	45

4.3	Esempio 2: chiamate telefoniche all'Università di Warwick . . .	50
4.3.1	Sola componente di trend	52
4.3.2	Trend più stagionalità	55
4.4	Vantaggi dell'algoritmo modificato	61
4.5	Ulteriori commenti alle tabelle dei risultati	63
4.6	Esempio 3: pubblicità di una tavoletta di cioccolato	66
5	Conclusioni	73
A	Le funzioni in R	75
	Bibliografia	91
	Ringraziamenti	95

Capitolo 1

Introduzione

Nel contesto delle serie storiche, l'approccio nello spazio degli stati si distingue perché offre una metodologia molto flessibile che riesce a trattare una gamma di problemi molto più ampia rispetto al sistema di analisi tradizionale, il sistema ARIMA di Box-Jenkins: nei modelli di serie storiche nello spazio degli stati, le osservazioni sono viste come somma di varie componenti, come trend, stagionalità, elementi di regressione e termini di errore, ognuna delle quali è modellata separatamente. I modelli delle componenti sono accorpati in un unico modello, chiamato *modello nello spazio degli stati*, che costituisce l'elemento base dell'analisi.

L'analisi delle serie storiche utilizzando l'approccio nello spazio degli stati ha avuto origine con il pionieristico lavoro di Kalman (1960 e 1963) nella teoria dei sistemi, con, tra le altre, famose applicazioni in astronautica: i modelli nello spazio degli stati lineari gaussiani sono stati quindi utilizzati in un numero notevole di casi pratici, adottando spesso delle trasformazioni per le osservazioni. Vi sono però molti casi in cui, anche adottando delle trasformazioni, questi modelli non risultano adeguati, ad esempio nel caso di variabili risposta categoriali, di conteggio, oppure non-negative: i modelli nello spazio degli stati di famiglia esponenziale ed i modelli lineari generalizzati dinamici hanno infatti avuto una nascita molto più recente, si veda West, Harrison e Migon (1985), e sono stati introdotti, appunto, per trattare

dati vistosamente non-gaussiani.

In questa tesi si prende in considerazione l'approccio ai modelli nello spazio degli stati di famiglia esponenziale presentato in Fahrmeir (1992) e, nello specifico, quello di Fahrmeir e Wagenpfeil (1997). Seguendo questo approccio, i modelli vengono trattati mediante iterative applicazioni del filtro di Kalman lineare ad un modello "costruito", allo stesso modo dei minimi quadrati pesati iterati che vengono applicati a variabili risposta "costruite" nei modelli lineari generalizzati statici: il capitolo 2 presenta una rassegna su questo lavoro.

Il capitolo 3 affronta invece il problema della stima degli iperparametri, il problema più ostico nei modelli nello spazio degli stati, perché può presentare notevoli difficoltà di carattere numerico oppure algoritmi caratterizzati dalla lenta convergenza: viene infatti introdotto un algoritmo di tipo EM, fornendo tutti i passaggi matematici necessari per la sua derivazione (completamente assenti nelle fonti bibliografiche), e ne viene proposta una modifica con l'obiettivo di rendere la stima degli iperparametri più veloce.

Nel capitolo 4, infine, vengono presentate delle applicazioni illustrative utilizzando gli stessi esempi del capitolo 8 della monografia di Fahrmeir e Tutz (2001) con un duplice obiettivo: il primo è quello di verificare la correttezza delle funzioni scritte mediante il programma statistico R per l'estensione del filtro di Kalman a famiglie esponenziali, confrontando i risultati ottenuti con quelli della monografia appena citata. Il secondo, invece, consiste nel quantificare il guadagno, in termini di tempo di esecuzione, ottenibile mediante l'utilizzo dell'algoritmo modificato proposto rispetto a quello originale.

Capitolo 2

Filtro di Kalman e sua estensione a famiglie esponenziali

La modellazione nello spazio degli stati fornisce una metodologia unificata per trattare un'ampia gamma di problemi nell'analisi delle serie storiche. Quando si utilizza questo approccio, si assume che lo sviluppo nel tempo del sistema che si sta studiando è determinato da una serie non osservata di vettori $\alpha_1, \dots, \alpha_T$, ai quali sono associati una serie di osservazioni y_1, \dots, y_T , dove la relazione tra gli α_t e le y_t è specificata dal modello nello spazio degli stati. Per i modelli lineari dinamici gaussiani, la relazione è data da $y_t = Z_t \alpha_t + \varepsilon_t$, assieme a un'equazione di transizione lineare $\alpha_t = F_t \alpha_{t-1} + \xi_t$ e le usuali assunzioni sui processi di rumore gaussiano $\{\varepsilon_t\}$, $\{\xi_t\}$: il filtro di Kalman (1960 e 1963) fornisce delle ottime stime per le medie condizionate.

Molti autori hanno considerato estensioni a sistemi non-gaussiani e non-lineari: tra questi, Sage e Melsa (1971), capitolo 9, ed Anderson e Moore (1979), capitolo 8, che hanno trattato dei modelli non-lineari con osservazioni condizionatamente gaussiane. A parte alcuni semplici modelli che ammettono delle soluzioni coniugate per la distribuzione a priori e la distribuzione a posteriori, le necessarie integrazioni devono però essere effet-

tuare numericamente, con enormi problemi computazionali, molte volte non risolvibili.

Dei filtri approssimati alternativi sono quindi d'interesse: uno dei primi è stato derivato da West, Harrison e Migon (1985), considerando il modello di osservazione solo come una “relazione guida” ed utilizzando il concetto di “sconto”. Anche se ben congegnato, il loro metodo presenta degli svantaggi, il principale dei quali è quello di non poterlo estendere a risposte multivariate, ad esempio modelli di serie storiche multicategoriali. Qualche dettaglio un po' più specifico su questo approccio verrà fornito alla fine della sezione 2.3, quando le nozioni riguardanti i modelli nello spazio degli stati saranno ormai consolidate.

In questa tesi si presenta uno di questi filtri approssimati, basato sull'idea del filtraggio della moda della distribuzione a posteriori: questo capitolo consiste infatti di una rassegna del lavoro svolto in Fahrmeir (1992), Fahrmeir e Tutz (2001) ed, in particolare, in Fahrmeir e Wagenpfeil (1997). Si fa presente, infine, che più recentemente hanno preso piede approcci di stima basati sulla simulazione, ad esempio metodi Markov Chain Monte Carlo (MCMC), in cui si estraggono ripetuti campioni dalle distribuzioni a posteriori: un lavoro di riferimento in cui i modelli nello spazio degli stati di famiglia esponenziale sono trattati mediante simulazione è quello di Durbin e Koopman (2001), che si basa su *importance sampling*.

2.1 Modelli nello spazio degli stati di tipo lineare

La caratteristica principale dei modelli nello spazio degli stati è quella di mettere in relazione una serie storica di osservazioni $\{y_t\}$ con una sequenza ignota di stati $\{\alpha_t\}$, i quali tipicamente includono una componente di trend e coefficienti di variabili esplicative che variano nel tempo. Date le osservazioni fino al tempo t , ossia y_1, \dots, y_t , il principale scopo inferenziale consiste nello stimare gli stati correnti, futuri e passati: tali procedure inferenziali vengono

denominate, rispettivamente, “filtraggio”, “previsione” ed “interpolazione” (quest’ultimo chiamata anche “lisciamento”).

Nel caso standard, ossia nei modelli nello spazio degli stati lineari, le osservazioni y_t (univariate o multivariate) sono messe in relazione coi vettori degli stati non osservati α_t per mezzo di una *equazione delle osservazioni* lineare

$$y_t = Z_t \alpha_t + \varepsilon_t, \quad t = 1, 2, \dots, \quad (2.1)$$

dove Z_t denota una matrice di disegno (detta anche matrice di osservazione) ed $\{\varepsilon_t\}$ è un processo di rumore bianco, ossia una sequenza di variabili d’errore mutualmente incorrelate con $E[\varepsilon_t] = 0$ e $\text{Cov}[\varepsilon_t] = R_t$. Nel caso di osservazioni scalari la matrice di disegno si riduce ad un vettore di disegno z'_t e la matrice delle covarianze alla varianza σ_t^2 .

La sequenza degli stati è definita da un’*equazione di transizione* lineare

$$\alpha_t = F_t \alpha_{t-1} + \xi_t, \quad t = 1, 2, \dots, \quad (2.2)$$

dove F_t è una matrice di transizione, $\{\xi_t\}$ è un processo di rumore bianco con $E[\xi_t] = 0$, $\text{Cov}[\xi_t] = Q_t$ e lo stato iniziale α_0 ha $E[\alpha_0] = a_0$ e $\text{Cov}[\alpha_0] = Q_0$. La media e la struttura di covarianza del modello sono pienamente specificate assumendo che $\{\varepsilon_t\}$ e $\{\xi_t\}$ siano mutualmente incorrelate ed incorrelate con lo stato iniziale α_0 . Nello specifico, nel caso di modelli nello spazio degli stati lineari gaussiani si ha che $\varepsilon_t \sim N(0, R_t)$, $\xi_t \sim N(0, Q_t)$, $\alpha_0 \sim N(a_0, Q_0)$ e che $\{\varepsilon_t\}$, $\{\xi_t\}$ ed α_0 sono mutualmente indipendenti. Si noti che le matrici di covarianza possono essere singolari: perciò, osservazioni parzialmente esatte e stati costanti nel tempo non sono esclusi dal modello.

Nella forma più semplice dei modelli nello spazio degli stati, le matrici di sistema Z_t , F_t , R_t , Q_t ed a_0 , Q_0 sono assunte note. In molte applicazioni, comunque, le matrici di covarianza R_t , Q_t ed i valori iniziali a_0 , Q_0 ed in qualche caso la matrice di transizione F_t sono completamente ignote o contengono degli ignoti iperparametri θ , così che

$$R_t = R_t(\theta), \quad Q_t = Q_t(\theta), \quad a_0 = a_0(\theta), \quad Q_0 = Q_0(\theta).$$

Inoltre, la matrice di disegno può dipendere da delle variabili esplicative o dalle osservazioni passate, ossia

$$Z_t = Z_t(x_t, y_{t-1}^*), \quad \text{dove } y_{t-1}^* = (y_{t-1}, \dots, y_1).$$

Nel capitolo 1 è stato detto che i modelli nello spazio degli stati hanno avuto, tra le altre, famose applicazioni in astronautica: in questo contesto, l'equazione delle osservazioni descrive le osservazioni y_t del radar, disturbate da del rumore, sullo stato (posizione, velocità, ...) di un'astronave, mentre l'equazione di transizione è un'approssimazione linearizzata e discretizzata delle leggi fisiche del moto nello spazio. Date le osservazioni y_1, \dots, y_t , l'interesse primario è rappresentato dalla stima in linea di α_t (filtraggio) e dalla previsione.

La principale ragione dello sviluppo e della diffusione dell'approccio nello spazio degli stati in statistica è che un numero notevole di modelli (ARMA, serie storiche strutturali, modelli di regressione dinamici, ...) possono essere trattati in una maniera univoca molto flessibile. Nel caso delle serie storiche strutturali scalari, ad esempio, l'idea base è quella di interpretare la scomposizione

$$y_t = \tau_t + \gamma_t + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma_\varepsilon^2), \quad (2.3)$$

di una serie storica in una componente di trend τ_t , in una componente stagionale γ_t e una componente d'errore ε_t come l'equazione delle osservazioni di un modello nello spazio degli stati e di definire le componenti stocastiche stagionali e di trend con equazioni di transizione ricorsive. Semplici modelli di trend non stazionari sono le passeggiate casuali di ordine uno o due, costituite rispettivamente da

$$\tau_t = \tau_{t-1} + u_t, \quad \tau_t = 2\tau_{t-1} - \tau_{t-2} + u_t, \quad u_t \sim N(0, \sigma_u^2). \quad (2.4)$$

Se nel modello non è presente nessuna componente stagionale γ_t , allora (2.3) e (2.4) possono essere scritte nella forma spazio degli stati definendo

$$\alpha_t = \tau_t = 1 \cdot \alpha_{t-1} + u_t, \quad y_t = 1 \cdot \alpha_t + \varepsilon_t,$$

per il modello con passeggiata casuale di ordine uno e

$$\alpha_t = \begin{pmatrix} \tau_t \\ \tau_{t-1} \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \tau_{t-1} \\ \tau_{t-2} \end{pmatrix} + \begin{pmatrix} u_t \\ 0 \end{pmatrix}, \quad y_t = (1, 0)\alpha_t + \varepsilon_t,$$

per il modello con passeggiata casuale di ordine due. Fahrmeir e Tutz (2001), capitolo 8, suggeriscono due diversi metodi per modellare una componente stagionale: nel primo, una componente stagionale stocastica γ_t di s periodi può essere specificata introducendo

$$\gamma_t = - \sum_{j=1}^{s-1} \gamma_{t-j} + \omega_t, \quad \omega_t \sim N(0, \sigma_\omega^2).$$

Nel caso di $s = 4$ periodi ed utilizzando una passeggiata casuale di ordine due per il trend, l'equazione di transizione e l'equazione delle osservazioni diventano

$$\alpha_t = \begin{pmatrix} \tau_t \\ \tau_{t-1} \\ \gamma_t \\ \gamma_{t-1} \\ \gamma_{t-2} \end{pmatrix} = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \tau_{t-1} \\ \tau_{t-2} \\ \gamma_{t-1} \\ \gamma_{t-2} \\ \gamma_{t-3} \end{pmatrix} + \begin{pmatrix} u_t \\ 0 \\ \omega_t \\ 0 \\ 0 \end{pmatrix}$$

e

$$y_t = (1, 0, 1, 0, 0)\alpha_t + \varepsilon_t. \quad (2.5)$$

Il secondo metodo suggerito da Fahrmeir e Tutz (2001), capitolo 8, consiste nel modellare la componente stagionale in forma trigonometrica, utilizzando le armoniche della serie di Fourier

$$\gamma_t = \sum_{j=1}^{[s/2]} \gamma_{jt},$$

con

$$\begin{aligned} \gamma_{jt} &= \gamma_{j,t-1} \cos \lambda_j + \tilde{\gamma}_{j,t-1} \sin \lambda_j + \omega_{jt} \\ \tilde{\gamma}_{jt} &= -\gamma_{j,t-1} \sin \lambda_j + \tilde{\gamma}_{j,t-1} \cos \lambda_j + \tilde{\omega}_{jt}, \quad j = 1, \dots, [s/2], \end{aligned}$$

con frequenze stagionali $\lambda_j = 2\pi j/s$ e processi di rumore bianco $\{\omega_{jt}\}$, $\{\tilde{\omega}_{jt}\}$ mutualmente indipendenti con varianza comune σ_ω^2 . Si noti che $\lambda_{[s/2]} = \pi$ per qualsiasi s e, quindi, si ha

$$\gamma_{[s/2],t} = -\gamma_{[s/2],t-1} + \omega_{[s/2],t}.$$

Nel caso visto in precedenza di $s = 4$ periodi e di trend modellato da una passeggiata casuale di ordine due, l'equazione di transizione e l'equazione delle osservazioni sono così definite:

$$\alpha_t = \begin{pmatrix} \tau_t \\ \tau_{t-1} \\ \gamma_{1t} \\ \tilde{\gamma}_{1t} \\ \gamma_{2t} \end{pmatrix} = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} = \begin{pmatrix} \tau_{t-1} \\ \tau_{t-2} \\ \gamma_{1,t-1} \\ \tilde{\gamma}_{1,t-1} \\ \gamma_{2,t-1} \end{pmatrix} + \begin{pmatrix} u_t \\ 0 \\ \omega_{1t} \\ \tilde{\omega}_{1t} \\ \omega_{2t} \end{pmatrix}$$

e

$$y_t = (1, 0, 1, 0, 1)\alpha_t + \varepsilon_t. \quad (2.6)$$

Tutti i modelli nello spazio degli stati visti finora sono invarianti nel tempo, ossia le matrici di sistema F_t e Z_t non dipendono dal tempo t . Assumendo che gli errori dei processi siano mutualmente indipendenti, le matrici di covarianza Q sono diagonali ma in molti casi singolari. Ad esempio, in (2.5) $Q = \text{diag}(\sigma_u^2, 0, \sigma_\omega^2, 0, 0)$ ed in (2.6) $Q = \text{diag}(\sigma_u^2, 0, \sigma_\omega^2, \sigma_\omega^2, \sigma_\omega^2)$. Le varianze σ_u^2 e σ_ω^2 sono iperparametri del modello, generalmente ignoti.

Si possono inoltre aggiungere delle variabili esplicative e le risposte precedenti: se il loro effetto è invariante nel tempo e se si omette la componente stagionale per semplicità, si ottiene

$$y_t = \tau_t + (x_t', y_{t-1}, \dots)\beta + \varepsilon_t.$$

L'invarianza nel tempo di β può essere descritta dalla relazione $\beta_t = \beta_{t-1}$ (= β). Assumendo una passeggiata casuale di ordine due per τ_t , si ha

$$\alpha_t = \begin{pmatrix} \tau_t \\ \tau_{t-1} \\ \beta_t \end{pmatrix} = \begin{pmatrix} 2 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & I \end{pmatrix} = \begin{pmatrix} \tau_{t-1} \\ \tau_{t-2} \\ \beta_{t-1} \end{pmatrix} + \begin{pmatrix} u_t \\ 0 \\ 0 \end{pmatrix},$$

$$y_t = (1, 0, x'_t, y_{t-1}, \dots)\alpha_t + \varepsilon_t.$$

Introducendo effetti che variano nel tempo per le variabili esplicative e stagionalità, si giunge a

$$y_t = \tau_t + \gamma_t + (x'_t, y_{t-1}, \dots)\beta_t + \varepsilon_t.$$

La scelta più semplice per $\{\beta_t\}$ è una passeggiata casuale di ordine uno con componenti d'errore mutualmente indipendenti. Il corrispondente blocco nell'equazione di transizione è quindi specificato da

$$\beta_t = \beta_{t-1} + \omega_t, \quad \omega_t \sim N(0, Q), \quad Q \text{ diagonale.}$$

2.2 Filtro di Kalman lineare

Una volta scritto il modello nella forma spazio degli stati, lo scopo principale è la stima di α_t date le osservazioni y_1, \dots, y_T . Questa operazione prende il nome di

- “filtraggio”, se $t = T$,
- “interpolazione” o “lisciamento”, se $t < T$,
- “previsione”, se $t > T$.

Sotto l'assunzione di normalità, la soluzione ottima al problema del filtraggio è data dalla *media condizionata* (o *media a posteriori*)

$$a_{t|t} := E[\alpha_t \mid y_t^*, x_t^*]$$

di α_t dati y_t^* e x_t^* , con $y_t^* = (y'_1, \dots, y'_t)'$ ed $x_t^* = (x'_1, \dots, x'_t)'$. Il simbolo “:=”, qui e nel seguito, sta per “è definito come”. Visto che il modello è lineare e gaussiano, la distribuzione a posteriori di α_t è anche gaussiana,

$$\alpha_t \mid y_t^*, x_t^* \sim N(a_{t|t}, V_{t|t})$$

con *matrice di covarianza a posteriori*

$$V_{t|t} := E[(\alpha_t - a_{t|t})(\alpha_t - a_{t|t})'].$$

Allo stesso modo, il miglior previsore ad un passo per α_t date le osservazioni fino al tempo $t - 1$, y_{t-1}^* , è

$$a_{t|t-1} := \mathbb{E}[\alpha_t \mid y_{t-1}^*, x_{t-1}^*],$$

e la densità di previsione ad un passo è gaussiana,

$$\alpha_t \mid y_{t-1}^*, x_{t-1}^* \sim N(a_{t|t-1}, V_{t|t-1}),$$

con matrice di covarianza a posteriori

$$V_{t|t-1} := \mathbb{E}[(\alpha_t - a_{t|t-1})(\alpha_t - a_{t|t-1})'].$$

Il filtro di Kalman (1960 e 1963) lineare calcola le medie e le matrici di covarianza a posteriori in modo ricorsivo, traendo vantaggio dal fatto che le distribuzioni a posteriori sono normali. Viene qui di seguito riportato l'algoritmo del filtro.

Filtro di Kalman lineare

Inizializzazione: $a_{0|0} = a_0, \quad V_{0|0} = Q_0.$

Per $t = 1, \dots, T$:

Passo di previsione: $a_{t|t-1} = F_t a_{t-1|t-1}$

$$V_{t|t-1} = F_t V_{t-1|t-1} F_t' + Q_t$$

Passo di correzione: $a_{t|t} = a_{t|t-1} + K_t (y_t - Z_t a_{t|t-1})$

$$V_{t|t} = V_{t|t-1} - K_t Z_t V_{t|t-1},$$

con guadagno di Kalman $K_t = V_{t|t-1} Z_t' [Z_t V_{t|t-1} Z_t' + R_t]^{-1}.$

Si noti che si sta considerando il caso in cui gli iperparametri (matrici di covarianza e di sistema, valori iniziali, ...) siano noti o dati. Il problema della loro stima verrà affrontato nel capitolo seguente.

Date le osservazioni y_1, \dots, y_{t-1} , il passo di previsione aggiorna l'ultimo valore del filtro $a_{t-1|t-1}$ alla previsione ad un passo $a_{t|t-1}$, in accordo con

l'equazione di transizione lineare (2.2). Osservato y_t , $a_{t|t-1}$ è corretto aggiungendo l'errore di previsione ad un passo pesato con il guadagno di Kalman K_t per ottenere la nuova stima del filtraggio $a_{t|t}$.

L'interpolatore di α_t date le osservazioni $y_T^* = (y_1, \dots, y_T)$ ed x_T^* è

$$a_{t|T} := E[\alpha_t | y_T^*, x_T^*].$$

Di nuovo la distribuzione a posteriori è normale

$$\alpha_t | y_T^*, x_T^* \sim N(a_{t|T}, V_{t|T}),$$

con

$$V_{t|T} := E[(\alpha_t - a_{t|T})(\alpha_t - a_{t|T})'].$$

La tradizionale forma di interpolazione è data dall'interpolatore "ad intervallo fisso", si veda Anderson e Moore (1979), sezione 7.4. Le stime $a_{t|T}$ sono ottenute con dei passi all'indietro, ossia per $t = T, \dots, 1$ si ha:

$$\begin{aligned} a_{t-1|T} &= a_{t-1|t-1} + B_t(a_{t|T} - a_{t|t-1}), \\ V_{t-1|T} &= V_{t-1|t-1} + B_t(V_{t|T} - V_{t|t-1})B_t', \end{aligned}$$

dove

$$B_t = V_{t-1|t-1}F_t'V_{t|t-1}^{-1}.$$

Ad ogni passo, la stima di lisciamiento $a_{t-1|T}$ è ottenuta dalla stima di filtraggio $a_{t-1|t-1}$, aggiungendo la differenza, opportunamente pesata, tra la stima di lisciamiento $a_{t|T}$ del passo precedente e la stima di previsione $a_{t|t-1}$.

2.3 Modelli nello spazio degli stati di famiglia esponenziale

È ovvio che, quando si è in presenza di serie storiche di conteggi, di dati categoriali oppure di dati non-negativi, l'assunzione di normalità dell'equazione delle osservazioni (2.1) viene a cadere: è quindi necessario sostituire il modello con un appropriato modello non-gaussiano. Per comprendere meglio

i passaggi che permettono di scrivere un'equazione delle osservazioni per le variabili risposta di famiglia esponenziale, Fahrmeir e Wagenpfeil (1997) suggeriscono di riscrivere l'equazione (2.1) delle osservazioni lineare gaussiana nella forma

$$y_t | \alpha_t \sim N(\eta_t, R_t), \quad y_t \in \mathbb{R}^q, \alpha_t \in \mathbb{R}^p, \quad (2.7)$$

dove $R_t = \text{Var}[y_t | \alpha_t]$ è la matrice delle covarianze di y_t dato α_t ed $\eta_t = Z_t \alpha_t$. L'ovvia modifica per le osservazioni da famiglia esponenziale non-gaussiana consiste nello specificare il modello delle osservazioni per y_t dato α_t con una distribuzione q -dimensionale del tipo famiglia esponenziale naturale:

$$y_t | \alpha_t \sim p(y_t | \alpha_t) = c_t(y_t) \exp\{\nu_t' y_t - b_t(\nu_t)\}, \quad (2.8)$$

dove ν_t , il parametro naturale, è funzione del predittore lineare $\eta_t = Z_t \alpha_t$ e $c_t(\cdot)$ e $b_t(\cdot)$ sono funzione note. Dalle proprietà delle famiglie esponenziali, la media e la varianza sono date da

$$E[y_t | \alpha_t] = \mu_t(\alpha_t) = \frac{\partial b_t(\nu_t)}{\partial \nu_t}, \quad (2.9)$$

$$\text{Var}[y_t | \alpha_t] = \Sigma_t(\alpha_t) = \frac{\partial^2 b_t(\nu_t)}{\partial \nu_t \partial \nu_t'}. \quad (2.10)$$

Come nei modelli lineari generalizzati statici, la media $\mu_t(\alpha_t)$ è legata al predittore lineare $\eta_t = Z_t \alpha_t$ da

$$\mu_t(\alpha_t) = h(Z_t \alpha_t), \quad (2.11)$$

dove $h : \mathbb{R}^q \rightarrow \mathbb{R}^p$ è una funzione continua e differenziabile due volte e Z_t è una matrice $q \times p$ che, come già detto nella sezione 2.1, può dipendere da delle variabili esplicative x_t o dalle risposte passate y_{t-1}^* .

L'assunzione di famiglia esponenziale (2.8), assieme alla specificazione della media (2.11), sostituisce quindi l'equazione delle osservazioni (2.7) dei modelli lineari gaussiani. Per quanto riguarda il modello (2.2) di transizione di stato, Fahrmeir e Wagenpfeil (1997) mantengono l'assunzione di un'equazione di transizione lineare gaussiana

$$\alpha_t = F_t \alpha_{t-1} + \xi_t, \quad t = 1, \dots, T, \quad (2.12)$$

con matrice di transizione F_t , rumore bianco gaussiano $\{\xi_t\}$ con $\xi_t \sim N(0, Q_t)$ e stato iniziale $\alpha_0 \sim N(a_0, Q_0)$. Per specificare completamente i modelli in termini di densità, Fahrmeir e Wagenpfeil (1997) aggiungono le seguenti assunzioni di indipendenza condizionata.

(A1) Condizionatamente ad α_t , le risposte attuali y_t sono indipendenti dagli stati passati $\alpha_{t-1}, \dots, \alpha_0$, ossia

$$p(y_t \mid \alpha_t, \alpha_{t-1}, \dots, \alpha_0) = p(y_t \mid \alpha_t), \quad t = 1, \dots, T.$$

Questa assunzione è implicita nei modelli nello spazio degli stati lineari gaussiani, data la mutua indipendenza delle sequenze d'errore $\{\varepsilon_t\}$ e $\{\xi_t\}$.

(A2) La sequenza $\{\alpha_t\}$ è markoviana, ossia

$$p(\alpha_t \mid \alpha_{t-1}, \dots, \alpha_0) = p(\alpha_t \mid \alpha_{t-1}).$$

In accordo con la (2.12), si ottiene $p(\alpha_t \mid \alpha_{t-1}) \sim N(F_t \alpha_{t-1}, Q_t)$

Nel caso di risposte scalari ($q = 1$), si possono ottenere dei modelli lineari generalizzati dinamici scalari. Ad esempio, nel caso di conteggi, la scelta più comune sono i modelli log-lineari con distribuzione di Poisson:

$$y_t \mid \alpha_t \sim \mathcal{P}(\lambda_t), \quad \lambda_t = \exp\{\eta_t\}, \quad \eta_t = Z_t \alpha_t.$$

Il predittore lineare può essere scelto come nei modelli di serie storiche strutturali per osservazioni gaussiane:

$$\eta_t = \tau_t + \gamma_t + x_t' \beta_t, \quad t = 1, \dots, T,$$

dove gli stati sono componenti stocastiche non osservate di trend τ_t e di stagionalità γ_t ed effetti β_t , che possono variare nel tempo, di variabili esplicative x_t .

Naturalmente, un modello log-lineare con distribuzione di Poisson non sarà sempre adatto e altre scelte, come la binomiale negativa, possono essere considerate. Se il numero dei conteggi è limitato, ad esempio, da n_t , i

modelli di regressione binomiali, come i modelli logit o probit, sono spesso appropriati:

$$y_t \mid \alpha_t \sim \mathcal{B}i(n_t, \pi_t), \quad \pi_t = h(\eta_t),$$

dove $h(\cdot)$ è la funzione legame: per $h(\cdot) = \exp(\cdot)/[1 + \exp(\cdot)]$ si ottiene il modello logit, per $h(\cdot) = \Phi(\cdot)$ il modello probit. Con $n_t = 1$ si possono modellare serie storiche binarie. Per quanto riguarda le serie storiche con risposte multicategoriali, si procede in maniera simile: per approfondimenti si veda Fahrmeir e Tutz (2001), pagg. 347–349 e, per il caso particolare di categorie ordinate, Fahrmeir e Wagenpfeil (1997).

Si noti che il tipo di modellazione proposto da West, Harrison e Migon (1985) è diverso: invece di un'esplicita equazione di transizione, essi assumono delle distribuzioni a priori e a posteriori coniugate per il parametro naturale ν_t ed impongono un'equazione di previsione lineare per α_t , utilizzando il concetto di “sconto” per evitare la stima delle matrici di covarianza degli errori ignoti. Inoltre, essi utilizzano la “relazione guida” $g(\nu_t) \simeq \eta_t = Z_t \alpha_t$ solo per determinare i parametri delle distribuzioni a priori e a posteriori di ν_t con un metodo dei momenti: non c'è comunque un modo ovvio di estendere questo metodo a risposte multidimensionali. L'approccio di West, Harrison e Migon (1985) è comunque caratterizzato da un aggiornamento sequenziale ricorsivo per le stime e le previsioni e, quando è stato introdotto, questo rappresentava un vantaggio non indifferente poiché l'onere computazionale era paragonabile a quello dell'algoritmo per i modelli nello spazio degli stati lineari e, quindi, non erano necessari calcolatori dalle elevate prestazioni.

2.4 Estensione del filtro di Kalman a famiglie esponenziali

2.4.1 Stima della moda a posteriori e verosimiglianza penalizzata

A differenza dei modelli nello spazio degli stati lineari in cui viene utilizzato il filtro di Kalman lineare, che stima le medie a posteriori, per i modelli nello spazio degli stati di famiglia esponenziale è stato adottato l'approccio di Fahrmeir e Wagenpfeil (1997), che si basa sulla stima della moda a posteriori. In questa sottosezione, gli algoritmi relativi sono derivati assumendo che gli iperparametri siano noti: la loro stima verrà affrontata nel prossimo capitolo. Le sequenze delle risposte e degli stati passati vengono denotate come

$$y_t^* = (y'_1, \dots, y'_t)', \quad \alpha_t^* = (\alpha'_1, \dots, \alpha'_t)'$$

e viene posto $\alpha = \alpha_T^*$. L'interpolatore della moda della distribuzione a posteriori, $a \equiv (a'_{0|T}, a'_{1|T}, \dots, a'_{T|T})' \in \mathbb{R}^m$, con $m = (T + 1)p$, è quindi definito come

$$a \equiv (a'_{0|T}, a'_{1|T}, \dots, a'_{T|T})' := \arg \max_{\alpha} \{p(\alpha | y_T^*)\},$$

ossia la moda della distribuzione a posteriori dell'intera sequenza. L'obiettivo di Fahrmeir e Wagenpfeil (1997) è quindi quello di massimizzare $p(\alpha | y_T^*)$. Applicando il teorema di Bayes si ottiene

$$p(\alpha | y_T^*) = \frac{1}{p(y_T^*)} \left\{ \prod_{t=1}^T p(y_t | \alpha_t^*, y_{t-1}^*) \prod_{t=1}^T p(\alpha_t | \alpha_{t-1}^*, y_{t-1}^*) p(\alpha_0) \right\},$$

e tenendo in considerazione le assunzioni (A1), (A2), poiché $p(y_T^*)$ non dipende da α , si giunge a

$$p(\alpha | y_T^*) \propto \prod_{t=1}^T p(y_t | \alpha_t, y_{t-1}^*) \prod_{t=1}^T p(\alpha_t | \alpha_{t-1}) p(\alpha_0).$$

Se poi si utilizzano i logaritmi e si inseriscono le densità normali del modello di transizione (2.12), si ottiene quella che Fahrmeir e Wagenpfeil (1997) chiamano funzione di log-verosimiglianza penalizzata, data da

$$\begin{aligned} \text{PL}(\alpha) &:= \sum_{t=1}^T \log p(y_t | \alpha_t, y_{t-1}^*) - \frac{1}{2}(\alpha_0 - a_0)' Q_0^{-1}(\alpha_0 - a_0) + \\ &\quad - \frac{1}{2} \sum_{t=1}^T (\alpha_t - F_t \alpha_{t-1})' Q^{-1}(\alpha_t - F_t \alpha_{t-1}) \end{aligned} \quad (2.13)$$

dove le densità $p(y_t | \alpha_t, y_{t-1}^*)$ sono definite dal modello di osservazione di famiglia esponenziale. La funzione obiettivo può essere quindi riscritta in questo modo:

$$a \equiv (a'_{0|T}, a'_{1|T}, \dots, a'_{T|T})' := \arg \max_{\alpha} \{\text{PL}(\alpha)\}. \quad (2.14)$$

Si noti che, quando Fahrmeir e Wagenpfeil (1997) utilizzano la denominazione “log-verosimiglianza penalizzata”, essi prendono come riferimento la verosimiglianza del solo modello di osservazione, la quale viene penalizzata dalle densità degli stati α_t , mentre le varianze Q_0 e Q svolgono il ruolo di parametri di lisciamiento. È infatti possibile dare a (2.13) e (2.14) un’interpretazione non parametrica: considerando $\{\alpha_t\}$ come una sequenza di parametri fissati ma ignoti, il primo termine nella (2.13) misura la bontà di adattamento ottenuta dal predittore lineare $Z_t \alpha_t$, mentre il secondo termine penalizza il “sovra-adattamento” ai dati. Ad esempio, nel caso di un modello logit dinamico binario $\log[\pi_t/(1 - \pi_t)] = \tau_t + \beta_t x_t$, se il trend τ_t e l’effetto variante nel tempo β_t sono modellati da una passeggiata casuale di ordine uno, allora $\alpha_t = (\tau_t, \beta_t)'$ e la (2.13) diventa

$$\begin{aligned} \text{PL}(\alpha) &= \sum_{t=1}^T \{y_t \log \pi_t + (1 - y_t) \log(1 - \pi_t)\} + \\ &\quad - \frac{1}{2\sigma_\tau^2} \sum_{t=1}^T (\tau_t - \tau_{t-1})^2 - \frac{1}{2\sigma_\beta^2} \sum_{t=1}^T (\beta_t - \beta_{t-1})^2, \end{aligned}$$

tralasciando le distribuzioni a priori di τ_0 e β_0 per semplicità. Mentre il primo termine misura la bontà di adattamento in termini di devianza, gli

altri termini penalizzano il “sovrà-adattamento” dei trend $\{\tau_t\}$ e degli effetti varianti nel tempo $\{\beta_t\}$. Facendo un confronto con le *spline* di lisciamiento, in questo caso si sta lisciando trend, componenti stagionali ed effetti di variabili esplicative invece di funzioni di variabili esplicative, mentre le varianze σ_τ^2 , σ_β^2 (in generale, le componenti di Q_t), come già detto, svolgono il ruolo di parametri di lisciamiento.

2.4.2 Derivazione dell’algoritmo

È facile notare che, nel caso di un modello di osservazione lineare gaussiano $y_t = Z_t\alpha_t + \varepsilon_t$, il termine di log-verosimiglianza nella (2.13) diventa

$$\sum_{t=1}^T \log p(y_t | \alpha_t, y_{t-1}^*) = -\frac{1}{2} \sum_{t=1}^T (y_t - Z_t\alpha_t)' R_t^{-1} (y_t - Z_t\alpha_t),$$

e la (2.13) un criterio di minimi quadrati penalizzati. Visto che le medie e le mode nei modelli nello spazio degli stati lineari coincidono, il problema di ottimizzazione è risolto dall’usuale filtro di Kalman lineare, risultando in un problema di complessità $O(T)$.

Per trovare una soluzione alla (2.14) nel caso generale, ossia per il modello di osservazione di famiglia esponenziale, potrebbe essere usato qualsiasi algoritmo di programmazione non-lineare. Per scopi statistici, si trae vantaggio dall’algoritmo del punteggio di Fisher, come nei modelli lineari generalizzati statici. Comunque, come nel caso dei modelli lineari gaussiani, gli algoritmi devono tener conto della speciale struttura dinamica del criterio di log-verosimiglianza penalizzata: seguendo l’impostazione di Fahrmeir e Wagenpfeil (1997), verrà ora derivato un singolo passo dell’algoritmo del punteggio di Fisher, in analogia coi modelli lineari generalizzati statici, e verrà dimostrato che esso può essere calcolato applicando il filtro di Kalman lineare a delle variabili risposta “costruite”, ottenendo nuovamente un algoritmo di complessità $O(T)$.

Notazione matriciale

Per prima cosa, Fahrmeir e Wagenpfeil (1997) riscrivono la (2.13) in forma matriciale come

$$\text{PL}(\alpha) = l(\alpha) - \frac{1}{2}\alpha'K\alpha,$$

dove

$$l(\alpha) = \sum_{t=0}^T l_t(\alpha_t), \quad l_t(\alpha_t) := \log p(y_t | \alpha_t, y_{t-1}^*), \quad t = 1, \dots, T,$$

$l_0(\alpha_0) := -\frac{1}{2}(\alpha_0 - a_0)'Q_0^{-1}(\alpha_0 - a_0)$ e la matrice di penalità K è simmetrica e tri-diagonale a blocchi:

$$K = \begin{pmatrix} K_{00} & K_{01} & & & 0 \\ K_{10} & K_{11} & K_{12} & & \\ & K_{21} & \ddots & \ddots & \\ & & \ddots & \ddots & K_{T-1,T} \\ 0 & & & K_{T,T-1} & K_{TT} \end{pmatrix}$$

con

$$\begin{aligned} K_{t-1,t} &= K'_{t,t-1} = -F'_t Q_t^{-1}, & t = 1, \dots, T, \\ K_{00} &= F'_1 Q_1^{-1} F_1, \\ K_{tt} &= Q_t^{-1} + F'_{t+1} Q_{t+1}^{-1} F_{t+1}, & t = 1, \dots, T, \\ F_{T+1} &= 0. \end{aligned}$$

Per poter descrivere un passo dell'algorithmo del punteggio di Fisher in notazione matriciale, è conveniente introdurre il vettore delle osservazioni

$$y' = (a'_0, y'_1, \dots, y'_T)$$

aumentato di a_0 . Corrispondentemente, vengono definiti il vettore delle medie aumentato di α_0 ,

$$\mu(\alpha)' = \{\alpha'_0, \mu'_1(\alpha_1), \dots, \mu'_T(\alpha_T)\}, \quad \text{dove } \mu_t(\alpha_t) = h(Z_t \alpha_t),$$

la matrice di covarianza diagonale a blocchi

$$\Sigma(\alpha) = \text{diag}\{Q_0, \Sigma_1(\alpha_1), \dots, \Sigma_T(\alpha_T)\},$$

la matrice di disegno diagonale a blocchi

$$Z = \text{diag}(I, Z_1, \dots, Z_T),$$

con $I \in \mathbb{R}^{p \times p}$ matrice identità, e la matrice diagonale a blocchi

$$D(\alpha) = \text{diag}\{I, D_1(\alpha_1), \dots, D_T(\alpha_T)\},$$

dove $D_t(\alpha_t) = \partial h(\eta_t) / \partial \eta_t$ è la derivata prima della funzione legame inversa $h(\eta_t)$ calcolata in $\eta_t = Z_t \alpha_t$. Utilizzando le proprietà (2.10) e (2.11), la funzione di punteggio di $l(\alpha)$ è quindi

$$s(\alpha) = \{s'_0(\alpha_0), s'_1(\alpha_1), \dots, s'_T(\alpha_T)\}' := Z' D(\alpha) \Sigma^{-1}(\alpha) \{y - \mu(\alpha)\},$$

con componenti $s_0(\alpha_0) = Q_0^{-1}(a_0 - \alpha_0)$, $s_t(\alpha_t) = Z' D_t(\alpha_t) \Sigma_t^{-1} \{y_t - \mu_t(\alpha_t)\}$, $t = 1, \dots, T$. La matrice dei pesi è

$$W(\alpha) = \text{diag}\{W_0, W_1(\alpha_1), \dots, W_T(\alpha_T)\} := D(\alpha) \Sigma^{-1}(\alpha) D'(\alpha)$$

con blocchi $W_0 = Q_0^{-1}$, $W_t(\alpha_t) = D_t(\alpha_t) \Sigma_t^{-1}(\alpha) D'_t(\alpha)$, $t = 1, \dots, T$ e la matrice d'informazione attesa di $l(\alpha)$ è

$$S(\alpha) = \text{diag}\{S_0, S_1(\alpha_1), \dots, S_T(\alpha_T)\} := Z' W(\alpha) Z$$

con blocchi diagonali $S_0 = Q_0^{-1}$, $S_t(\alpha_t) = Z'_t W_t(\alpha) Z_t$, $t = 1, \dots, T$. Entrambe le matrici $W(\alpha)$ ed $S(\alpha)$ sono diagonali a blocchi.

Singolo passo dell'algoritmo

Utilizzando la notazione matriciale, Fahrmeir e Wagenpfeil (1997) derivano un singolo passo dell'algoritmo del punteggio di Fisher: la derivata prima di $\text{PL}(\alpha)$ è data da

$$u(\alpha) = \frac{\partial \text{PL}(\alpha)}{\partial \alpha} = s(\alpha) - K\alpha,$$

mentre la matrice d'informazione attesa tri-diagonale a blocchi da

$$U(\alpha) = -\mathbb{E} \left[\frac{\partial^2 \text{PL}(\alpha)}{\partial \alpha \partial \alpha'} \right] = S(\alpha) + K = Z'W(\alpha)Z + K.$$

Un singolo passo dell'algoritmo del punteggio di Fisher dall'iterazione corrente $\alpha^k \in \mathbb{R}^m$ alla successiva $\alpha^{k+1} \in \mathbb{R}^m$ è quindi rappresentato da

$$U(\alpha^k)\{\alpha^{k+1} - \alpha^k\} = u(\alpha^k).$$

Poiché $U(\alpha^k)\alpha^{k+1} - U(\alpha^k)\alpha^k = u(\alpha^k)$, segue che $\alpha^{k+1} = \alpha^k + U(\alpha^k)^{-1}u(\alpha^k)$ e questa può essere riscritta come

$$\alpha^{k+1} = \{Z'W(\alpha^k)Z + K\}^{-1}Z'W(\alpha^k)\tilde{y}(\alpha^k), \quad (2.15)$$

con variabile risposta “costruita”

$$\tilde{y}(\alpha^k) = \{\tilde{y}'_0, \tilde{y}'_1(\alpha^k_1), \dots, \tilde{y}'_T(\alpha^k_T)\}' := \{D^{-1}(\alpha^k)\}'\{y - \mu(\alpha^k)\} + Z\alpha^k,$$

le cui componenti sono $\tilde{y}_0 = a_0$, $\tilde{y}_t(\alpha^k_t) = \{D_t^{-1}(\alpha^k_t)\}'\{y_t - \mu_t(\alpha^k_t)\} + Z_t\alpha^k_t$, $t = 1, \dots, T$. Una formula simile, ma senza la matrice di penalità K (che contiene le informazioni del modello di transizione), è ottenuta per la stima dei minimi quadrati pesati iterati nei modelli lineari generalizzati statici: per un confronto, si veda Azzalini (2001), pagg. 253–259.

Si assuma ora il particolare caso di un modello nello spazio degli stati lineare gaussiano, definito dalla (2.7) e dalla (2.12). Allora $\mu(\alpha) = Z\alpha$, $D(\alpha)$ si riduce alla matrice identità e la funzione di punteggio di $l(\alpha)$ diventa

$$s(\alpha) = Z'R^{-1}(y - Z\alpha),$$

con $R = \text{diag}(Q_0, R_1, \dots, R_T)$, $R_t = \text{Cov}[y_t | \alpha_t]$. La matrice dei pesi $W(\alpha)$ si riduce a R^{-1} e la variabile risposta “costruita” alla reale osservazione y , visto che $D(\alpha^k) = I$ e $\mu(\alpha^k) = Z\alpha^k$. La (2.15) diventa

$$a = (Z'R^{-1}Z + K)^{-1}Z'R^{-1}y, \quad (2.16)$$

dove $a = (a'_{0|T}, a'_{1|T}, \dots, a'_{T|T})'$ è il vettore delle stime interpolate. Come già detto in precedenza, il filtro di Kalman risolve in maniera efficiente la

(2.16), senza il bisogno di invertire esplicitamente la matrice tri-diagonale a blocchi $Z'R^{-1}Z + K$. Confrontando la (2.16) con la (2.15), si può concludere quanto segue: per risolvere la (2.15), ossia per effettuare un singolo passo dell'algoritmo del punteggio di Fisher nel caso di famiglia esponenziale, si può applicare qualsiasi versione efficiente del filtro di Kalman lineare, sostituendo R_t con $W_t^{-1}(\alpha^k)$ ed y_t con $\tilde{y}_t(\alpha^k)$. Questo verrà chiamato filtro di Kalman "costruito": per un confronto, si veda l'algoritmo del filtro di Kalman lineare riportato nella sezione 2.2. Nel seguente algoritmo, $a_{t|t}$, $V_{t|t}$, $a_{t|t-1}$, $V_{t|t-1}$, $a_{t|T}$, $V_{t|T}$ sono approssimazioni numeriche dei valori filtrati, previsti ed interpolati di α_t e delle corrispondenti matrici di covarianza degli errori.

Filtro ed interpolatore di Kalman costruito (WKFS)

Inizializzazione: $a_{0|0} = a_0$, $V_{0|0} = Q_0$.

Per $t = 1, \dots, T$:

Passo di previsione: $a_{t|t-1} = F_t a_{t-1|t-1}$,

$$V_{t|t-1} = F_t V_{t-1|t-1} F_t' + Q_t.$$

Passo di correzione: $a_{t|t} = a_{t|t-1} + K_t \{\tilde{y}_t(\alpha^k) - Z_t a_{t|t-1}\}$,

$$V_{t|t} = V_{t|t-1} - K_t Z_t V_{t|t-1},$$

con guadagno di Kalman $K_t = V_{t|t-1} Z_t' \{Z_t V_{t|t-1} Z_t' + W_t^{-1}(\alpha^k)\}^{-1}$.

Per $t = T, \dots, 1$ (interpolazione):

$$a_{t-1|T} = a_{t-1|t-1} + B_t (a_{t|T} - a_{t|t-1}),$$

$$V_{t-1|T} = V_{t-1|t-1} + B_t (V_{t|T} - V_{t|t-1}) B_t',$$

$$\text{dove } B_t = V_{t-1|t-1} F_t' V_{t|t-1}^{-1}.$$

Si ottiene così $\alpha^{k+1} = (a'_{0|T}, a'_{1|T}, \dots, a'_{T|T})'$.

Applicando il lemma d'inversione di matrici, ad esempio Anderson e Moore (1979), sezione 6.3, il passo di correzione può essere riscritto in forma di punteggio come

Passo di correzione*: $V_{t|t} = \{V_{t|t-1}^{-1} + S_t(\alpha_t^k)\},$

$$a_{t|t} = a_{t|t-1} + V_{t|t} \tilde{s}_t(\alpha_t^k),$$

con funzione di punteggio “costruita” $\tilde{s}_t(\alpha_t^k) := s_t(\alpha_t^k) - S_t(\alpha_t^k)\{a_{t|t-1} - \alpha_t^k\}$. I passi di correzione in forma di punteggio sono più generali rispetto a quelli scritti in forma di guadagno di Kalman, in quanto si applicano anche a modelli di osservazione non-esponenziali.

Algoritmo completo

Verrà ora mostrato l’algoritmo del punteggio di Fisher completo derivato da Fahrmeir e Wagenpfeil (1997): poiché l’obiettivo è quello di risolvere la (2.14), si deve iterare WKFS, dove la soluzione α^k della precedente iterazione costituisce la matrice di partenza per il ciclo successivo. È però necessaria una matrice α^0 per inizializzare l’algoritmo. Essa è fornita dal filtro ed interpolatore di Kalman esteso generalizzato (GKFS) presentato in Fahrmeir (1992), il quale differisce da WKFS solamente per il passo di correzione:

Passo di correzione GKFS: $a_{t|t} = a_{t|t-1} + K_t \{y_t - h(Z_t a_{t|t-1})\},$

$$V_{t|t} = V_{t|t-1} - K_t D'_t(a_{t|t-1}) Z_t V_{t|t-1},$$

con $K_t = V_{t|t-1} Z'_t D'_t(a_{t|t-1}) [D'_t(a_{t|t-1}) Z_t V_{t|t-1} Z'_t D'_t(a_{t|t-1}) + \Sigma_t(a_{t|t-1})]^{-1}$. L’algoritmo del punteggio di Fisher completo è quindi dato dal

Filtro ed interpolatore di Kalman pesato iterato (IWKFS)

Inizializzazione: calcolare $\alpha^0 = (a_{0|T}^0, a_{1|T}^0, \dots, a_{T|T}^0)'$ con GKFS; porre l’indice di iterazione $k = 0$.

Passo 1: partendo da α^k , calcolare α^{k+1} mediante l’applicazione di WKFS.

Passo 2: se qualche criterio di convergenza è soddisfatto, STOP; altrimenti, $k = k + 1$ ed andare al Passo 1.

Alla convergenza si ottiene l'interpolatore della moda a posteriori $a = \alpha^{\bar{k}}$. Inoltre, le matrici di covarianza degli errori $V_{t|T}$, calcolati con WKFS, alla convergenza sono i blocchi diagonali di $U(a)^{-1}$, cfr. Fahrmeir e Kaufmann (1991), e non è quindi necessario nessun ulteriore carico computazionale per ricavarli. Questo è un risultato molto conveniente per la stima degli iperparametri, come si vedrà nel prossimo capitolo.

Capitolo 3

Stima degli iperparametri

In questo capitolo verrà affrontato il problema della stima degli iperparametri che, fino ad ora, sono stati trattati come noti: verrà introdotto l'algoritmo di tipo EM proposto da Fahrmeir (1992) e Fahrmeir e Wagenpfeil (1997), completandolo con i passaggi matematici necessari per la sua derivazione (del tutto assenti nelle fonti bibliografiche appena citate), e verrà proposta una modifica per renderne la convergenza più veloce.

3.1 Algoritmo di tipo EM

Considerando il caso di un modello nello spazio degli stati invariante nel tempo con vettore degli iperparametri $\theta = (a_0, Q_0, Q)$, un metodo per la stima degli iperparametri è rappresentato dall'algoritmo EM di Dempster, Laird e Rubin (1977). Tralasciando le costanti, la log-verosimiglianza congiunta dei dati completi è data da

$$l(y_1, \dots, y_T, \alpha_0, \dots, \alpha_T; \theta) = \sum_{t=1}^T l_t(\alpha_t) + l(\alpha_0 | a_0, Q_0) + l(\alpha | Q), \quad (3.1)$$

con

$$\begin{aligned} l_t(\alpha_t) &= \log p(y_t | \alpha_t, y_{t-1}^*), \\ l(\alpha_0 | a_0, Q_0) &= -\frac{1}{2} \log |Q_0| - \frac{1}{2} (\alpha_0 - a_0)' Q_0^{-1} (\alpha_0 - a_0), \end{aligned}$$

$$l(\alpha | Q) = -\frac{T}{2} \log |Q| - \frac{1}{2} \sum_{t=1}^T (\alpha_t - F_t \alpha_{t-1})' Q^{-1} (\alpha_t - F_t \alpha_{t-1}),$$

in accordo con la terminologia EM, dove $(y_1, \dots, y_T)'$ sono i dati osservati, $(\alpha_0, \dots, \alpha_T)'$ i dati non osservati e, quindi, $(y_1, \dots, y_T, \alpha_0, \dots, \alpha_T)'$ i dati completi.

Verranno ora mostrati i passaggi matematici per derivare l'algoritmo di tipo EM per la stima degli iperparametri: questi passaggi sono del tutto assenti in Fahrmeir (1992), Fahrmeir e Wagenpfeil (1997) e Fahrmeir e Tutz (2001), che riportano solamente i risultati finali. Il passo E della p -esima iterazione dell'algoritmo EM consiste quindi nel calcolare

$$\mathbb{E}[l(y_1, \dots, y_T, \alpha_0, \dots, \alpha_T; \theta) | y_1, \dots, y_T; \theta^{(p)}],$$

il valore atteso condizionato della log-verosimiglianza date le osservazioni ed il vettore degli iperparametri corrente $\theta^{(p)}$. Data l'indipendenza di $l_t(\alpha_t)$ da θ , verrà considerata solamente

$$S(\theta | \theta^{(p)}) = S_1(a_0, Q_0 | \theta^{(p)}) + S_2(Q | \theta^{(p)}),$$

dove

$$\begin{aligned} S_1(a_0, Q_0 | \theta^{(p)}) &= \mathbb{E}[l(\alpha_0 | a_0, Q_0) | \theta^{(p)}] = \\ &= -\frac{1}{2} \log |Q_0| - \frac{1}{2} \mathbb{E}[(\alpha_0 - a_0)' Q_0^{-1} (\alpha_0 - a_0) | \theta^{(p)}] = \\ &= -\frac{1}{2} \log |Q_0| - \frac{1}{2} \text{tr}\{Q_0^{-1} \mathbb{E}[(\alpha_0 - a_0)(\alpha_0 - a_0)' | \theta^{(p)}]\}, \end{aligned}$$

e

$$\begin{aligned} S_2(Q | \theta^{(p)}) &= \mathbb{E}[l(\alpha | Q) | \theta^{(p)}] = \\ &= -\frac{T}{2} \log |Q| - \frac{1}{2} \sum_{t=1}^T \mathbb{E}[(\alpha_t - F_t \alpha_{t-1})' Q^{-1} (\alpha_t - F_t \alpha_{t-1}) | \theta^{(p)}] = \\ &= -\frac{T}{2} \log |Q| - \frac{1}{2} \sum_{t=1}^T \text{tr}\{Q^{-1} \mathbb{E}[(\alpha_t - F_t \alpha_{t-1})(\alpha_t - F_t \alpha_{t-1})' | \theta^{(p)}]\}. \end{aligned}$$

Si possono scrivere le due equazioni per $S_1(a_0, Q_0 | \theta^{(p)})$ e $S_2(Q | \theta^{(p)})$ in questo modo perché, tenendo in considerazione che $\mathbb{E}[(\alpha_0 - a_0) | \theta^{(p)}] =$

$E[(\alpha_t - F_t \alpha_{t-1}) \mid \theta^{(p)}] = 0$, è possibile applicare il seguente risultato a $E[(\alpha_0 - a_0)' Q_0^{-1} (\alpha_0 - a_0) \mid \theta^{(p)}]$ ed a $E[(\alpha_t - F_t \alpha_{t-1})' Q^{-1} (\alpha_t - F_t \alpha_{t-1}) \mid \theta^{(p)}]$.

Lemma 1 *Sia $X = (x_1, x_2, \dots, x_k)'$ una variabile casuale multipla con $E[X] = \mu$ e $\text{Var}[X] = V$ ed $A = (a_{ij})$ una matrice quadrata di ordine k . Allora*

$$E[X'AX] = \mu' A \mu + \text{tr}(AV).$$

Per la dimostrazione si veda, ad esempio, Azzalini (2001), pagg. 303–304.

Dal passo M della p -esima iterazione si ottiene il vettore degli iperparametri per la $p + 1$ -esima iterazione e consiste nel massimizzare $S(\theta \mid \theta^{(p)})$ rispetto a θ , ossia

$$\theta^{(p+1)} = \arg \max_{\theta} S(\theta \mid \theta^{(p)}).$$

Si tratta quindi di calcolare

$$\begin{cases} \frac{\partial S(\theta \mid \theta^{(p)})}{\partial a_0} = \frac{\partial S_1(a_0, Q_0 \mid \theta^{(p)})}{\partial a_0} = 0, \\ \frac{\partial S(\theta \mid \theta^{(p)})}{\partial Q_0} = \frac{\partial S_1(a_0, Q_0 \mid \theta^{(p)})}{\partial Q_0} = 0, \\ \frac{\partial S(\theta \mid \theta^{(p)})}{\partial Q} = \frac{\partial S_2(Q \mid \theta^{(p)})}{\partial Q} = 0. \end{cases}$$

Tralasciando il condizionamento a $\theta^{(p)}$ per semplificare la notazione, si può vedere che

$$\begin{aligned} S_1(a_0, Q_0) &= -\frac{1}{2} \log |Q_0| - \frac{1}{2} \text{tr} \{ Q_0^{-1} (\text{Cov}[\alpha_0] + E[\alpha_0] E[\alpha_0'] + \\ &\quad - a_0 E[\alpha_0'] - E[\alpha_0] a_0' + a_0 a_0') \} = \\ &= -\frac{1}{2} \log |Q_0| - \frac{1}{2} \text{tr} \{ Q_0^{-1} \text{Cov}[\alpha_0] \} + \\ &\quad - \frac{1}{2} \text{tr} \{ Q_0^{-1} (E[\alpha_0] - a_0) (E[\alpha_0] - a_0)' \}, \end{aligned}$$

e, ricordando che $\text{tr}(AB) = \text{tr}(BA)$ e che $\text{tr}(b) = b$ se b è uno scalare, si ottiene

$$\begin{aligned} \frac{\partial S_1(a_0, Q_0)}{\partial a_0} &= -\frac{1}{2} \frac{\partial}{\partial a_0} \text{tr} \{ (E[\alpha_0] - a_0)' Q_0^{-1} (E[\alpha_0] - a_0) \} = \\ &= -Q_0^{-1} (E[\alpha_0] - a_0), \end{aligned}$$

per cui

$$\frac{\partial S_1(a_0, Q_0)}{\partial a_0} = 0 \Leftrightarrow a_0 = E[\alpha_0]. \quad (3.2)$$

Per risolvere la seconda equazione conviene procedere ponendo $N = Q_0^{-1}$; quindi

$$\begin{aligned} \frac{\partial S_1(a_0, N)}{\partial N} &= \frac{\partial}{\partial N} \left\{ \frac{1}{2} \log |N| - \frac{1}{2} \text{tr}\{N \text{Cov}[\alpha_0]\} + \right. \\ &\quad \left. - \frac{1}{2} \text{tr}\{N(E[\alpha_0] - a_0)(E[\alpha_0] - a_0)'\} \right\} = \\ &= \frac{1}{2} \frac{\partial}{\partial N} \left\{ \log |N| - \text{tr}\{N \text{Cov}[\alpha_0]\} + \right. \\ &\quad \left. - \text{tr}\{N(E[\alpha_0] - a_0)(E[\alpha_0] - a_0)'\} \right\} \end{aligned}$$

e, tenendo in considerazione che

$$\frac{\partial}{\partial V} \log |V| = 2V^{-1} - \text{diag}(V^{-1}),$$

se V simmetrica, e che

$$\frac{\partial}{\partial X} \text{tr}(XY) = Y + Y' - \text{diag}(Y),$$

se X simmetrica, si ottiene

$$\begin{aligned} \frac{\partial S_1(a_0, N)}{\partial N} &= \frac{1}{2} \left\{ 2N^{-1} - \text{diag}(N^{-1}) - 2\text{Cov}[\alpha_0] + \text{diag}(\text{Cov}[\alpha_0]) + \right. \\ &\quad \left. - 2(E[\alpha_0] - a_0)(E[\alpha_0] - a_0)' + \text{diag}[(E[\alpha_0] - a_0)(E[\alpha_0] - a_0)'] \right\} = \\ &= \frac{1}{2} \left\{ 2[N^{-1} - \text{Cov}[\alpha_0] - (E[\alpha_0] - a_0)(E[\alpha_0] - a_0)'] + \right. \\ &\quad \left. - \text{diag}[N^{-1} - \text{Cov}[\alpha_0] - (E[\alpha_0] - a_0)(E[\alpha_0] - a_0)'] \right\} = \\ &= \frac{1}{2} \{2M - \text{diag}(M)\}, \end{aligned}$$

con $M = Q_0 - \text{Cov}[\alpha_0] - (E[\alpha_0] - a_0)(E[\alpha_0] - a_0)'$. Uguagliando l'equazione a zero e tenendo presente il risultato della prima equazione (3.2), si giunge a

$$\begin{aligned} \frac{\partial S_1(a_0, N)}{\partial N} = 0 &\Leftrightarrow 2M - \text{diag}(M) = 0 \\ &\Leftrightarrow M = 0 \\ &\Leftrightarrow Q_0 = \text{Cov}[\alpha_0]. \end{aligned}$$

Similmente, per risolvere la terza equazione, conviene effettuare la sostituzione $P = Q^{-1}$; allora

$$\begin{aligned}
\frac{\partial S_2(P)}{\partial P} &= \frac{\partial}{\partial P} \left\{ \frac{T}{2} \log |P| - \frac{1}{2} \sum_{t=1}^T \text{tr} \{ P E [(\alpha_t - F_t \alpha_{t-1}) (\alpha_t - F_t \alpha_{t-1})'] \} \right\} = \\
&= \frac{T}{2} [2P^{-1} - \text{diag}(P^{-1})] - \frac{1}{2} \sum_{t=1}^T \left[2E [(\alpha_t - F_t \alpha_{t-1}) (\alpha_t - F_t \alpha_{t-1})'] + \right. \\
&\quad \left. - \text{diag} (E [(\alpha_t - F_t \alpha_{t-1}) (\alpha_t - F_t \alpha_{t-1})']) \right] = \\
&= \frac{1}{2} \left\{ 2TQ - T \text{diag}(Q) - 2 \sum_{t=1}^T E [(\alpha_t - F_t \alpha_{t-1}) (\alpha_t - F_t \alpha_{t-1})'] + \right. \\
&\quad \left. + \sum_{t=1}^T \text{diag} (E [(\alpha_t - F_t \alpha_{t-1}) (\alpha_t - F_t \alpha_{t-1})']) \right\} = \\
&= \frac{1}{2} \left\{ 2 \left(TQ - \sum_{t=1}^T E [(\alpha_t - F_t \alpha_{t-1}) (\alpha_t - F_t \alpha_{t-1})'] \right) + \right. \\
&\quad \left. - \text{diag} \left(TQ - \sum_{t=1}^T E [(\alpha_t - F_t \alpha_{t-1}) (\alpha_t - F_t \alpha_{t-1})'] \right) \right\} = \\
&= \frac{1}{2} \{ 2L - \text{diag}(L) \},
\end{aligned}$$

dove

$$L = TQ - \sum_{t=1}^T E [(\alpha_t - F_t \alpha_{t-1}) (\alpha_t - F_t \alpha_{t-1})'],$$

per cui

$$\begin{aligned}
\frac{\partial S_2(P)}{\partial P} = 0 &\Leftrightarrow 2L - \text{diag}(L) = 0 \\
&\Leftrightarrow L = 0 \\
&\Leftrightarrow Q = \frac{1}{T} \sum_{t=1}^T E [(\alpha_t - F_t \alpha_{t-1}) (\alpha_t - F_t \alpha_{t-1})'].
\end{aligned}$$

Riassumendo, quindi, abbiamo che

$$\begin{aligned}
a_0^{(p+1)} &= E[\alpha_0 \mid \theta^{(p)}], \\
Q_0^{(p+1)} &= \text{Cov}[\alpha_0 \mid \theta^{(p)}], \\
Q^{(p+1)} &= \frac{1}{T} \sum_{t=1}^T E[\alpha_t \alpha_t' - \alpha_t \alpha_{t-1}' F_t' - F_t \alpha_{t-1} \alpha_t' + F_t \alpha_{t-1} \alpha_{t-1}' F_t' \mid \theta^{(p)}].
\end{aligned}$$

Considerando

$$\begin{aligned} E[\alpha_t \alpha'_t | \theta^{(p)}] &= \text{Cov}[\alpha_t | \theta^{(p)}] + E[\alpha_t | \theta^{(p)}] E[\alpha'_t | \theta^{(p)}], \\ E[\alpha_{t-1} \alpha'_t | \theta^{(p)}] &= \text{Cov}[\alpha_{t-1}, \alpha_t | \theta^{(p)}] + E[\alpha_{t-1} | \theta^{(p)}] E[\alpha'_t | \theta^{(p)}], \end{aligned}$$

si può notare che, per ottenere le stime degli iperparametri, sono necessarie le medie e le matrici di covarianza di α_t . Per evitare integrazioni numeriche di elevate dimensioni, si sostituiscono le medie con le stime della moda a posteriori $a_{t|T}$ e le matrici di covarianza con le matrici di covarianza della stima della moda $V_{t|T}$ risultanti da IWKFS della sottosezione 2.4.2: da qui la denominazione di algoritmo “di tipo” EM.

Tenendo in considerazione Schneider (1986), citato in Biller (1997), $\text{Cov}[\alpha_{t-1}, \alpha_t | \theta^{(p)}]$ può essere sostituito da $B_t V_{t|T}$, con B_t definito in WKFS della sottosezione 2.4.2; quindi,

$$\begin{aligned} Q^{(p+1)} &= \frac{1}{T} \sum_{t=1}^T \{ E[\alpha_t \alpha'_t | \theta^{(p)}] - E[\alpha_t \alpha'_{t-1} | \theta^{(p)}] F'_t - F_t E[\alpha_{t-1} \alpha'_t | \theta^{(p)}] + \\ &\quad + F_t E[\alpha_{t-1} \alpha'_{t-1} | \theta^{(p)}] F'_t \} = \\ &= \frac{1}{T} \sum_{t=1}^T \{ V_{t|T} + a_{t|T} a'_{t|T} - V_{t|T} B'_t F'_t - a_{t|T} a'_{t-1|T} F'_t - F_t B_t V_{t|T} + \\ &\quad - F_t a_{t-1|T} a_{t|T} + F_t V_{t-1|T} F'_t + F_t a_{t-1|T} a'_{t-1|T} F'_t \}. \end{aligned}$$

La procedura per la stima congiunta di α , a_0 , Q_0 e Q può essere allora riassunta dal seguente

Algoritmo di tipo “EM”

1. Scegliere i valori iniziali $Q^{(0)}$, $Q_0^{(0)}$, $a_0^{(0)}$ e porre l'indice di iterazione $p = 0$.
2. Interpolazione (lisciamento): calcolare $a_{t|T}^{(p)}$, $V_{t|T}^{(p)}$ con IWKFS, sostituendo i parametri ignoti con $Q^{(p)}$, $Q_0^{(p)}$, $a_0^{(p)}$.
3. Passo EM: calcolare $Q^{(p+1)}$, $Q_0^{(p+1)}$, $a_0^{(p+1)}$ con

$$\begin{aligned}
a_0^{(p+1)} &= a_{0|T}^{(p)}, \\
Q_0^{(p+1)} &= V_{0|T}^{(p)}, \\
Q^{(p+1)} &= \frac{1}{T} \sum_{t=1}^T \{ (a_{t|T}^{(p)} - F_t a_{t-1|T}^{(p)}) (a_{t|T}^{(p)} - F_t a_{t-1|T}^{(p)})' + V_{t|T}^{(p)} - F_t B_t^{(p)} V_{t|T}^{(p)} + \\
&\quad - V_{t|T}^{(p)} B_t^{(p)} F_t' + F_t V_{t-1|T}^{(p)} F_t' \}.
\end{aligned}$$

4. Se qualche criterio di convergenza è soddisfatto, STOP; altrimenti, $p = p + 1$ ed andare al punto 2.

Si fa notare che né Fahrmeir e Wagenpfeil (1997) né Fahrmeir e Tutz (2001) affrontano l'argomento degli errori standard delle stime degli iperparametri θ : lo sviluppo e l'esplorazione di questo problema costituisce quindi un interessante argomento da approfondire.

3.2 Altri metodi di stima

Esistono anche altri metodi per stimare gli iperparametri. Uno di questi, ad esempio, può essere ottenuto utilizzando un criterio di convalida incrociata generalizzata: si veda Fahrmeir e Wagenpfeil (1997). Gli stessi autori appena citati fanno però notare che questo metodo ha il difetto di poter presentare parecchi minimi locali: i risultati possono così dipendere molto dai valori di partenza scelti per l'algoritmo di ottimizzazione e sono quindi necessari numerosi tentativi per trovare il valore minimo del criterio di convalida incrociata, come riportato in Chiogna e Gaetan (2002). Wagenpfeil (1995) effettua un confronto tra l'algoritmo di tipo EM, il criterio di convalida incrociata generalizzata ed un terzo metodo basato su una verosimiglianza approssimata, riportando le seguenti conclusioni: l'algoritmo di tipo EM si conferma un metodo robusto per la stima degli iperparametri ma anche un algoritmo dalla convergenza molto lenta; i metodi basati sul criterio di convalida incrociata generalizzata e sulla verosimiglianza approssimata costituiscono una possibile alternativa visto che sono molto più veloci, anche se è molto più probabile incorrere in problemi numerici, specialmente nel caso di iperparametri θ multidimensionali.

3.3 Modifica all'algoritmo di tipo EM

Uno svantaggio dell'algoritmo di tipo EM ed, in generale, di tutti gli algoritmi EM, è la loro lenta convergenza: infatti, Dempster, Laird e Rubin (1977) dimostrano che essi convergono solo linearmente alla stima di massima verosimiglianza.

L'algoritmo di tipo EM descritto nella sezione 3.1 è rappresentato dalla Figura 3.1: nello schema, cc_α e cc_θ denotano, rispettivamente, il criterio di convergenza sugli stati α e sugli iperparametri θ . Come si può notare, nel caso del filtro di Kalman esteso a famiglie esponenziali la lentezza in convergenza dell'algoritmo di tipo EM è accentuata per la presenza dell'algoritmo interno IWKFS che, a sua volta, ha un proprio criterio di convergenza da raggiungere iterando più volte WKFS.

Si propone qui una modifica all'algoritmo, con l'obiettivo di ridurre il numero di iterazioni di WKFS. Lo svantaggio principale di WKFS è che esso deve essere inizializzato, oltre che dagli iperparametri, anche da una matrice iniziale di stati α^0 che serve per calcolare le variabili risposta "costruite" nel primo ciclo di IWKFS. Per questo motivo, ogni volta che vengono ristimati gli iperparametri, è necessario utilizzare GKFS per calcolare un nuovo α^0 . La modifica proposta è la seguente: utilizzare GKFS per calcolare la matrice iniziale degli stati solamente per la prima iterazione ${}^{(1)}\alpha^0$ mentre, per le successive iterazioni, utilizzare come matrice iniziale ${}^{(p)}\alpha^0$, $p > 1$, dove p indica la p -esima iterazione dell'algoritmo esterno, la matrice degli stati per cui è stata raggiunta la convergenza alla $(p-1)$ -esima iterazione, ossia ${}^{(p-1)}\alpha^{\bar{k}}$. Quest'ultimo dovrebbe costituire un migliore valore iniziale rispetto a quello ottenuto da GKFS e, quindi, il numero di iterazioni di WKFS necessario per raggiungere la convergenza dovrebbe risultare minore.

La procedura per la stima è quindi riassunta dal seguente

Algoritmo di tipo "EM" modificato

1. Scegliere i valori iniziali $Q^{(0)}$, $Q_0^{(0)}$, $a_0^{(0)}$.

2. Calcolare ${}^{(0)}\alpha^{\bar{k}} = \{a_{t|T}^{(0)}\}$, $V_{t|T}^{(0)}$ con GKFS, sostituendo i parametri ignoti con $Q^{(0)}$, $Q_0^{(0)}$, $a_0^{(0)}$, e porre l'indice di iterazione $p = 0$.
3. Passo EM: calcolare $Q^{(p+1)}$, $Q_0^{(p+1)}$, $a_0^{(p+1)}$ con

$$a_0^{(p+1)} = a_{0|T}^{(p)},$$

$$Q_0^{(p+1)} = V_{0|T}^{(p)},$$

$$Q^{(p+1)} = \frac{1}{T} \sum_{t=1}^T \{(a_{t|T}^{(p)} - F_t a_{t-1|T}^{(p)})(a_{t|T}^{(p)} - F_t a_{t-1|T}^{(p)})' + V_{t|T}^{(p)} - F_t B_t^{(p)} V_{t|T}^{(p)} +$$

$$- V_{t|T}^{(p)} B_t^{(p)} F_t' + F_t V_{t-1|T}^{(p)} F_t'\}.$$
4. Calcolare ${}^{(p+1)}\alpha^{\bar{k}} = \{a_{t|T}^{(p+1)}\}$, $V_{t|T}^{(p+1)}$ iterando WKFS fino alla convergenza, sostituendo i parametri ignoti con ${}^{(p)}\alpha^{\bar{k}}$, $Q^{(p+1)}$, $Q_0^{(p+1)}$, $a_0^{(p+1)}$.
5. Se qualche criterio di convergenza sugli iperparametri è soddisfatto, STOP; altrimenti, $p = p + 1$ ed andare al punto 3.

Un confronto fra l'algorithmo originale e quello modificato, schematizzato dalla Figura 3.2, verrà effettuato, per mezzo di esempi illustrativi, nel prossimo capitolo.

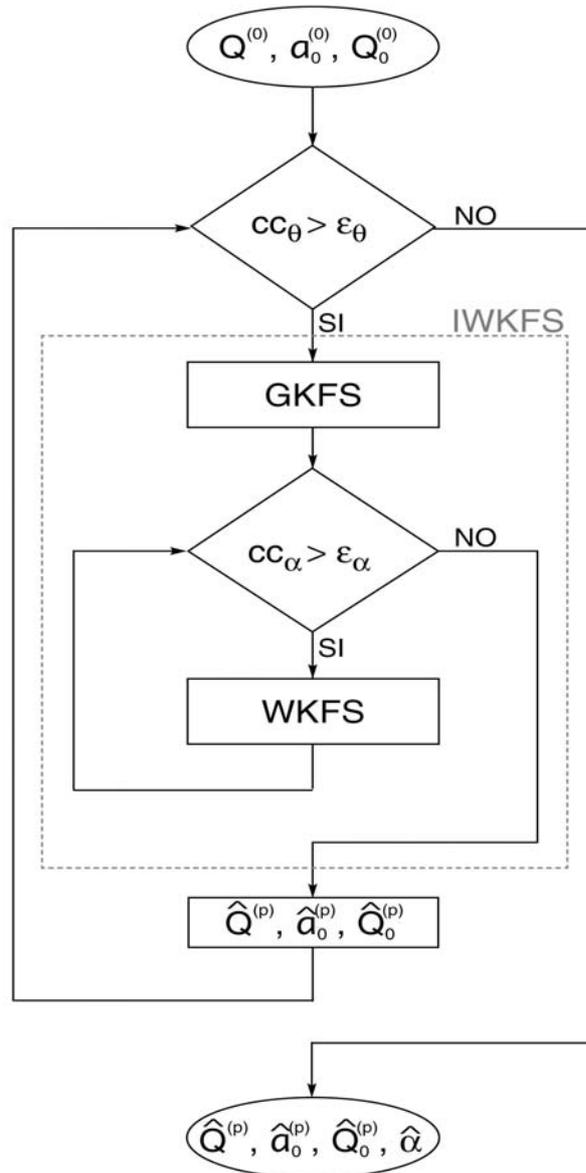


Figura 3.1: Schema dell'algoritmo di tipo EM per il filtro di Kalman esteso a famiglie esponenziali.

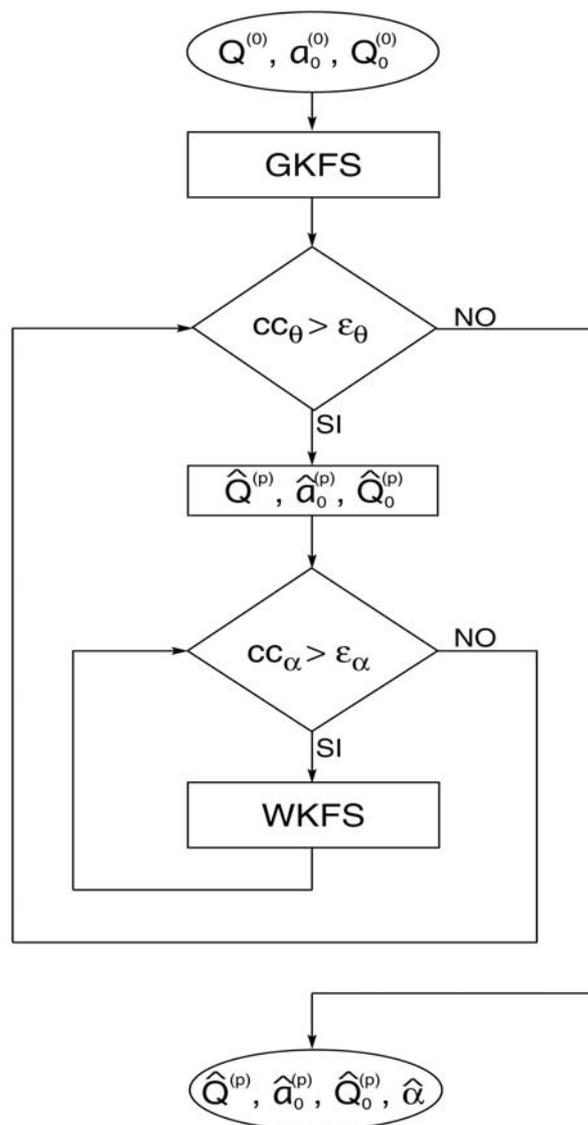


Figura 3.2: Schema dell'algorithmo modificato per il filtro di Kalman esteso a famiglie esponenziali.

[186,]	74388.40	75417.80	75709.98	75784.63	75799.59	75801.48	75801.54	75801.54	NA
[187,]	74388.59	75417.93	75710.10	75784.74	75799.70	75801.59	75801.66	75801.66	NA
[188,]	74388.78	75418.06	75710.21	75784.86	75799.82	75801.70	75801.77	75801.77	NA
[189,]	74388.97	75418.19	75710.33	75784.97	75799.93	75801.81	75801.88	75801.88	NA
[190,]	74389.15	75418.32	75710.44	75785.08	75800.04	75801.92	75801.99	75801.99	NA

dove la prima matrice di R riporta i valori di $PL(\alpha)$ e la seconda quelli di $l(y_1, \dots, y_T, \alpha_0, \dots, \alpha_T; \theta)$. I valori di riga, ad eccezione di quelli nella prima colonna, che riporta i valori della verosimiglianza calcolati con GKFS, indicano i valori di verosimiglianza per ogni iterazione dell'algoritmo WKFS fino a quando il criterio di convergenza imposto sugli α non è soddisfatto: i valori NA della matrice indicano che la verosimiglianza non è stata calcolata perché, appunto, l'algoritmo aveva già raggiunto la convergenza. Da una riga alla successiva cambia il valore degli iperparametri, che vengono ristimati secondo le equazioni riportate nell'algoritmo di tipo EM della sezione 3.1. In pratica, si possono identificare le righe con l'indice p dell'algoritmo di tipo EM e le colonne con l'indice k dell'algoritmo IWKFS della sottosezione 2.4.2.

Come si può notare, i valori per ogni riga aumentano andando da sinistra verso destra, mentre, passando da una riga alla successiva, il valore per cui viene raggiunta la convergenza (ossia l'ultimo valore della riga prima di NA) nel caso di $PL(\alpha)$ diminuisce, in apparente contrasto con la definizione di massima verosimiglianza: infatti, il valore corretto è quello di $l(y_1, \dots, y_T, \alpha_0, \dots, \alpha_T; \theta)$, riportato nella seconda matrice. Le funzioni costruite in R per gli algoritmi (originali e modificati) per il filtro di Kalman esteso a famiglie esponenziali calcolano quindi quest'ultimo valore.

Capitolo 4

Applicazioni illustrative

Sulla base delle parti teoriche presentate nei due capitoli precedenti, sono state scritte, mediante l'utilizzo del programma statistico R, le funzioni inserite nell'appendice A. In questo capitolo verrà prima illustrato il principale problema computazionale che si è incontrato durante la loro costruzione, la non simmetria delle matrici $V_{t|t-1}$, $V_{t|t}$, $V_{t|T}$ e, di conseguenza, Q_0 e Q . Le funzioni saranno poi utilizzate con i dati di tre esempi presentati nel capitolo 8 di Fahrmeir e Tutz (2001), in modo da verificarne la correttezza di funzionamento mediante un confronto dei risultati. Verranno inoltre presentati i vantaggi ottenuti dalla modifica all'algoritmo originale.

4.1 Aspetti computazionali

4.1.1 Mantenimento della simmetria delle matrici delle varianze

Durante la costruzione delle funzioni in R, si è incorsi in un problema molto grave, che avrebbe potuto precludere il funzionamento dell'algoritmo se non fosse stato risolto: la mancanza di simmetria delle matrici di varianza e covarianza $V_{t|t-1}$, $V_{t|t}$ e $V_{t|T}$. Si ricorda che queste matrici sono risultati di calcoli ricorsivi e, quindi, il valore di una influenza l'altra e viceversa.

L'origine del problema è di natura computazionale: infatti, se si osserva come sono costruite algebricamente le matrici di covarianza $V_{t|t-1}$, $V_{t|t}$ e $V_{t|T}$ nell'algoritmo WKFS della sezione 2.4.2, è evidente che esse debbano essere definite positive e, quindi, simmetriche.

Ad esempio, utilizzando i dati dell'esempio delle chiamate telefoniche che verrà presentato nella sezione 4.3, si nota che già alla prima iterazione dell'algoritmo di tipo EM (ossia per $p = 1$, mantenendo la notazione del capitolo precedente) e per $t = 10$ la matrice $V_{t|t-1}$ non è simmetrica. La seguente matrice di R rappresenta ${}^{(1)}V_{10|9} - {}^{(1)}V'_{10|9}$:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	0.000000e+00	0.000000e+00	0.000000e+00	5.551115e-17	2.220446e-16	-1.665335e-16
[2,]	0.000000e+00	0.000000e+00	-3.885781e-16	-2.775558e-16	-2.220446e-16	8.326673e-17
[3,]	0.000000e+00	3.885781e-16	0.000000e+00	-2.775558e-17	1.110223e-16	-1.665335e-16
[4,]	-5.551115e-17	2.775558e-16	2.775558e-17	0.000000e+00	1.172673e-15	2.220446e-16
[5,]	-2.220446e-16	2.220446e-16	-1.110223e-16	-1.172673e-15	0.000000e+00	-1.110223e-16
[6,]	1.665335e-16	-8.326673e-17	1.665335e-16	-2.220446e-16	1.110223e-16	0.000000e+00
[7,]	1.665335e-16	0.000000e+00	8.326673e-17	-1.110223e-16	-5.551115e-17	4.024558e-16
[8,]	0.000000e+00	-2.220446e-16	2.220446e-16	1.387779e-16	0.000000e+00	4.440892e-16
[9,]	2.775558e-17	-1.665335e-16	-1.110223e-16	1.110223e-16	1.110223e-16	0.000000e+00
[10,]	2.220446e-16	-7.632783e-17	4.163336e-17	-2.775558e-17	2.775558e-17	-8.326673e-17
[11,]	3.330669e-16	2.220446e-16	-3.330669e-16	-5.551115e-17	2.220446e-16	-2.220446e-16
	[,7]	[,8]	[,9]	[,10]	[,11]	
[1,]	-1.665335e-16	0.000000e+00	-2.775558e-17	-2.220446e-16	-3.330669e-16	
[2,]	0.000000e+00	2.220446e-16	1.665335e-16	7.632783e-17	-2.220446e-16	
[3,]	-8.326673e-17	-2.220446e-16	1.110223e-16	-4.163336e-17	3.330669e-16	
[4,]	1.110223e-16	-1.387779e-16	-1.110223e-16	2.775558e-17	5.551115e-17	
[5,]	5.551115e-17	0.000000e+00	-1.110223e-16	-2.775558e-17	-2.220446e-16	
[6,]	-4.024558e-16	-4.440892e-16	0.000000e+00	8.326673e-17	2.220446e-16	
[7,]	0.000000e+00	1.665335e-16	1.110223e-16	-1.665335e-16	-8.326673e-17	
[8,]	-1.665335e-16	0.000000e+00	3.885781e-16	0.000000e+00	-2.220446e-16	
[9,]	-1.110223e-16	-3.885781e-16	0.000000e+00	1.110223e-16	2.220446e-16	
[10,]	1.665335e-16	0.000000e+00	-1.110223e-16	0.000000e+00	-1.276756e-15	
[11,]	8.326673e-17	2.220446e-16	-2.220446e-16	1.276756e-15	0.000000e+00	

Si nota che la mancanza di simmetria è piccolissima, al massimo dell'ordine di 10^{-15} , ma questa si ripercuote sulle successive e già per $t = 300$ (e sempre per $p = 1$) è diventata dell'ordine di 10^{-14} . Infatti, ${}^{(1)}V_{300|299} - {}^{(1)}V'_{300|299}$ dà

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	0.000000e+00	0.000000e+00	-2.486900e-14	1.687539e-14	-1.776357e-15	-3.552714e-15
[2,]	0.000000e+00	0.000000e+00	1.421085e-14	1.421085e-14	-1.776357e-14	3.552714e-15

```

[3,] 2.486900e-14 -1.421085e-14 0.000000e+00 -2.131628e-14 2.842171e-14 1.243450e-14
[4,] -1.687539e-14 -1.421085e-14 2.131628e-14 0.000000e+00 5.773160e-15 4.973799e-14
[5,] 1.776357e-15 1.776357e-14 -2.842171e-14 -5.773160e-15 0.000000e+00 4.618528e-14
[6,] 3.552714e-15 -3.552714e-15 -1.243450e-14 -4.973799e-14 -4.618528e-14 0.000000e+00
[7,] -3.552714e-15 -8.881784e-15 -3.552714e-15 2.486900e-14 0.000000e+00 3.019807e-14
[8,] 1.065814e-14 1.776357e-15 1.332268e-15 -1.865175e-14 1.776357e-15 -2.486900e-14
[9,] -1.554312e-15 9.769963e-15 1.953993e-14 0.000000e+00 5.828671e-16 1.421085e-14
[10,] 3.330669e-15 1.243450e-14 -1.776357e-15 5.773160e-15 -4.440892e-16 -5.662137e-15
[11,] -4.440892e-15 -2.664535e-15 7.105427e-15 -2.664535e-15 2.220446e-15 1.776357e-15
      [,7]      [,8]      [,9]      [,10]     [,11]
[1,] 3.552714e-15 -1.065814e-14 1.554312e-15 -3.330669e-15 4.440892e-16
[2,] 8.881784e-15 -1.776357e-15 -9.769963e-15 -1.243450e-14 2.664535e-15
[3,] 3.552714e-15 -1.332268e-15 -1.953993e-14 1.776357e-15 -7.105427e-15
[4,] -2.486900e-14 1.865175e-14 0.000000e+00 -5.773160e-15 2.664535e-15
[5,] 0.000000e+00 -1.776357e-15 -5.828671e-16 4.440892e-16 -2.220446e-15
[6,] -3.019807e-14 2.486900e-14 -1.421085e-14 5.662137e-15 -1.776357e-15
[7,] 0.000000e+00 -7.105427e-15 1.065814e-14 -1.065814e-14 -1.010303e-14
[8,] 7.105427e-15 0.000000e+00 2.091383e-14 3.552714e-15 7.105427e-15
[9,] -1.065814e-14 -2.091383e-14 0.000000e+00 0.000000e+00 -1.065814e-14
[10,] 1.065814e-14 -3.552714e-15 0.000000e+00 0.000000e+00 -1.199041e-14
[11,] 1.010303e-14 -7.105427e-15 1.065814e-14 1.199041e-14 0.000000e+00

```

Alla decima iterazione dell' algoritmo esterno, il difetto è molto più evidente. Infatti, non è più necessario presentare la differenza tra la matrice e la sua trasposta, ma direttamente $^{(10)}V_{10|9}$:

```

      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,] 7.784336445 -1.01452963 0.675535630 -0.24460352 0.90748102 0.23856018
[2,] -1.071284230 8.74782644 0.139831129 -0.37111383 0.63245576 -0.02756026
[3,] 0.630696923 0.81681519 9.010357150 -0.05788867 -0.61359235 -0.45210805
[4,] -0.273461253 -0.47005797 0.001809085 8.80254349 -0.39058933 -0.60640857
[5,] 0.887654520 0.57003964 -0.709424083 0.51174941 8.63422080 -0.36983214
[6,] 0.229623820 -0.05578247 -0.434968995 -0.70684942 -0.30125501 8.46088873
[7,] 0.538341906 0.54059559 -0.232304692 0.32856535 -0.57713001 0.46581380
[8,] 0.181526761 0.07628416 -0.298175123 -0.26810772 -0.44339638 -0.75579756
[9,] 0.085466894 0.30769154 0.215522250 0.64967087 -0.03522620 0.54000024
[10,] -0.138138938 -0.01353323 0.070812957 0.13370757 -0.18162374 -0.14265860
[11,] -0.001349530 0.14649052 0.235299748 0.48382375 0.21574090 0.74028090
      [,7]      [,8]      [,9]      [,10]     [,11]
[1,] 0.5436534 0.1816908 0.08802076 -0.13852733 0.0006714456
[2,] 0.5608500 0.0858573 0.31302977 -0.01046813 0.1448905225
[3,] -0.2020980 -0.3017645 0.22504575 0.06837085 0.2376140762
[4,] 0.3978215 -0.2409867 0.66663744 0.14227117 0.4840857467
[5,] -0.4767264 -0.4646319 -0.00849877 -0.19104810 0.2289714746
[6,] -0.2597159 -0.6885540 0.58964581 -0.12958454 0.7532040706
[7,] 8.6829682 -0.3015559 -0.61167805 -0.57989949 -0.1560574440

```

```

[8,] -0.2479367  8.3197084 -0.10854887 -0.84602692  0.6054874535
[9,] -0.6812130  0.2825895  8.28335870 -0.52693054 -0.8161349672
[10,] -0.5623659 -0.8756372 -0.49510552  7.52291123 -0.0017854093
[11,] -0.1765602  0.5738368 -0.84897042  0.13614833  7.6012584040

```

Proseguendo con le iterazioni, i valori tendono ad “esplodere”: la seguente matrice rappresenta ${}^{(13)}V_{10|9}$:

```

          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,]  212.56017 -710.5263  3607.0195  510.2905  53.23610 -28.4300
[2,]  747.46409 21714.8646  6654.2253  143.5151 5903.42458 -5026.0272
[3,] -1817.80562 16532.8547 10136.2578 -605.9518 7633.61382 -2830.7223
[4,] -382.83299 -10149.3978 -3886.9830 1587.8531 -6278.23461 4360.4991
[5,] -603.30149 1151.2029 -5546.6552 4650.2168 1224.45583 -2175.2795
[6,] -495.17434 -5978.4461 -4706.1304 -1898.6640 1023.91074 1908.8699
[7,] -354.48440 2780.2265 -453.8592 -2867.8856 -2379.34497 3660.4398
[8,] -416.97762 -966.7098 -1984.6967 -274.1525 11.09288 -1272.5287
[9,] 147.41233 3338.2623 2174.7848 -1160.9357 79.65886 -2543.1330
[10,] -97.30704 1149.0399 169.8269 -408.0762 191.60016 -536.3972
[11,] 257.58753 1417.9305 1717.4145 -551.6237 108.01373 -689.0488

          [,7]      [,8]      [,9]      [,10]      [,11]
[1,] -324.7301 -91.34374 -38.71068 21.09197 -20.63609
[2,] 3008.7229 -3101.37786 3829.28789 -734.85110 3024.79709
[3,] 1126.0999 -2149.75848 1887.15702 -542.21803 1566.79978
[4,] 1297.8581 1131.62219 -515.31316 284.27021 -454.46208
[5,] 3824.7838 -347.79257 1006.90405 -179.39276 524.05407
[6,] -4546.6758 2676.54955 510.50240 414.54927 -626.27687
[7,] 1033.7136 -1531.97249 2287.08999 -102.05457 600.43594
[8,] 1672.7798 543.98851 -3137.68425 829.49949 306.86402
[9,] -1429.3081 2425.54624 730.47727 -611.44285 1167.15527
[10,] 293.1531 -701.46469 766.70655 121.08079 -1390.47848
[11,] -297.8193 -978.22173 -413.96259 1365.80520 320.22569

```

e la stima corrente della matrice delle varianze degli errori dell’equazione di transizione, ${}^{(13)}Q$, presenta degli elementi diagonali, e quindi dei valori per le varianze, addirittura negativi.

Per ovviare a questo grave problema, è stato scelto la soluzione di rappresentare le matrici di covarianza dell’algoritmo come una media tra la matrice stessa e la sua trasposta: in questo modo, i valori delle matrici sono risultati più stabili e gli algoritmi hanno potuto funzionare correttamente.

Si fa presente, infine, che in Fahrmeir e Tutz (2001) il problema del non mantenimento della simmetria da parte delle matrici di covarianza non è nemmeno segnalato e, quindi, non è noto come sia stato risolto.

4.1.2 Criteri di convergenza

Gli algoritmi presentati nei capitoli 2 e 3 sono caratterizzati da due criteri di convergenza da raggiungere, uno per gli stati α ed un altro per gli iperparametri $\theta = (a_0, Q_0, Q)$. È stato scelto di basare i due criteri sulla differenza dei valori fra l'iterazione corrente e quella precedente e nello specifico, poiché α , a_0 , Q_0 e Q sono tutte matrici, sulla differenza media dei valori all'interno della matrice degli stati e degli iperparametri. Quindi, per gli iperparametri $a_0 = \{a_{0i}\}$, $Q_0 = \{q_{0ij}\}$ e $Q = \{q_{ij}\}$, con a_0 vettore di lunghezza I e Q_0 e Q matrici $I \times I$, il criterio di convergenza è costituito da

$$cc_\theta = \frac{1}{3} \left[\frac{\frac{1}{I} \sum_{i=1}^I |a_{0i}^{(p+1)} - a_{0i}^{(p)}|}{1 + \frac{1}{I} \sum_{i=1}^I |a_{0i}^{(p+1)} - a_{0i}^{(p)}|} + \frac{\frac{1}{I^2} \sum_{i=1}^I \sum_{j=1}^I |q_{0ij}^{(p+1)} - q_{0ij}^{(p)}|}{1 + \frac{1}{I^2} \sum_{i=1}^I \sum_{j=1}^I |q_{0ij}^{(p+1)} - q_{0ij}^{(p)}|} + \frac{\frac{1}{I^2} \sum_{i=1}^I \sum_{j=1}^I |q_{ij}^{(p+1)} - q_{ij}^{(p)}|}{1 + \frac{1}{I^2} \sum_{i=1}^I \sum_{j=1}^I |q_{ij}^{(p+1)} - q_{ij}^{(p)}|} \right].$$

Similmente, il criterio di convergenza per gli stati α è dato da

$$cc_\alpha = \frac{\frac{1}{IT} \sum_{i=1}^I \sum_{t=0}^T |\alpha_{it}^{k+1} - \alpha_{it}^k|}{1 + \frac{1}{IT} \sum_{i=1}^I \sum_{t=0}^T |\alpha_{it}^{k+1} - \alpha_{it}^k|}.$$

4.1.3 Modo di utilizzo delle funzioni di R

Come già detto, le funzioni costruite in R per essere utilizzate sui dati degli esempi delle sezioni che seguono, sono riportate nell'appendice A e sono: `poisDGLMun`, `poisDGLMun2`, `binomDGLMun` e `binomDGLMun2`, dove le prime lettere indicano il tipo di distribuzione di cui la variabile risposta y_t è realizzazione (Poisson o Binomiale) e “un” sta per univariato. Infatti, anche se gli algoritmi fin qui presentati funzionano identicamente in entrambi i casi univariato e multivariato, le funzioni sono state scritte appositamente per il caso univariato perché così si è ottenuto un guadagno nelle velocità di esecuzione delle stesse: ad esempio, la funzione `solve` di R, qui utilizzata per invertire le matrici, compare solo una volta in tutto l'algoritmo WKFS. La presenza di un “2” alla fine del nome indica che la funzione usa l'algoritmo modificato presentato nella sezione 3.3.

I parametri obbligatori da passare alle funzioni sono:

- `data`, il vettore delle osservazioni y_t , $t = 1, \dots, T$,
- `Z`, la matrice di disegno $Z = \{z'_t\}$, $t = 1, \dots, T$,
- `F`, la matrice di transizione $F_t = F$, costante nel tempo,
- `a_0`, `Q_0`, `Q`, i valori iniziali degli iperparametri.

Si noti la differenza tra il parametro `Z` ed il parametro `F`: le funzioni sono state costruite considerando solo il caso in cui la matrice di transizione sia costante nel tempo (il caso più comune), mentre `Z` è una matrice molto più grande, per tenere conto di variabili esplicative che variano nel tempo, come nell'esempio 3 di questo capitolo. Se, come negli esempi 1 e 2, z'_t è un vettore di disegno costante nel tempo, la matrice Z avrà invece le righe tutte uguali.

Le funzioni sono inoltre caratterizzate dai seguenti parametri:

- `epsTHETA`, il valore che regola la severità del criterio di convergenza sugli iperparametri $\theta = (a_0, Q_0, Q)$; il valore di default è 10^{-5} ,
- `epsALPHA`, il valore che regola la severità del criterio di convergenza sugli stati α ; il valore di default è 10^{-3} ,
- `Qdiag`, variabile di tipo vero / falso, necessaria per indicare se le matrici Q e Q_0 sono diagonali, ossia se gli errori ξ_t dell'equazione di transizione sono incorrelati o meno; il valore di default è **FALSE**.

Infine, si fa notare che è possibile che, eseguendo le funzioni, si verifichi il seguente errore

```
Error in while (convcrit >epsALPHA) { : missing value where TRUE/FALSE needed
```

dovuto al fatto che sono stati utilizzati dei valori troppo elevati per gli iperparametri, che hanno generato un valore `NaN` per il criterio di convergenza sugli stati α . Nel caso si verifichi questo errore, il rimedio è quello di rieseguire la funzione utilizzando dei valori degli iperparametri più piccoli.

4.2 Esempio 1: precipitazioni a Tokyo

Il primo esempio in cui sono state utilizzate le funzioni costruite in R riguarda le precipitazioni cadute a Tokyo negli anni 1983–1984: come già detto, è un esempio riportato dal capitolo 8 di Fahrmeir e Tutz (2001) che era già stato usato in Kitagawa (1987). I dati sono stati scaricati da http://www.stat.uni-muenchen.de/service/datenarchiv/welcome_e.html del gruppo di ricerca SFB 386 presente presso l'Institut für Statistik, Ludwig-Maximilians-Universität, München. Per ogni giorno t dell'anno, $t = 1, \dots, 366$, è stato registrato se fosse caduto almeno 1 mm di pioggia. La variabile risposta y_t può quindi assumere tre valori:

- 0, se non si sono verificate precipitazioni (ossia non è caduto almeno 1 mm di pioggia),
- 1, se le precipitazioni si sono verificate in uno solo dei due anni,
- 2, se le precipitazioni si sono verificate sia nel 1983 sia nel 1984.

I dati sono rappresentati nella Figura 4.1.

Per ottenere una stima interpolata della probabilità π_t che si verifichino delle precipitazioni il giorno t , ed ipotizzando un modello con solo trend, che si distribuisce come una passeggiata casuale di ordine uno, si può usare il seguente modello logit binomiale dinamico

$$y_t \sim \mathcal{Bi}(n_t, \pi_t), \quad \pi_t = h(\alpha_t) = \frac{\exp\{\alpha_t\}}{1 + \exp\{\alpha_t\}},$$

$$\alpha_t = \alpha_{t-1} + \xi_t, \quad \xi_t \sim N(0, q), \quad \alpha_0 \sim N(a_0, q_0),$$

con $n_t = 1$ per $t = 60$ ed $n_t = 2$ per ogni $t \neq 60$; bisogna infatti tener conto del fatto che il 1984 è stato bisestile (il sessantesimo giorno dell'anno è il 29 febbraio). Per quanto riguarda la matrice di transizione e la matrice di disegno, utilizzando il modello appena descritto, si ha

$$F_t = Z_t = 1,$$

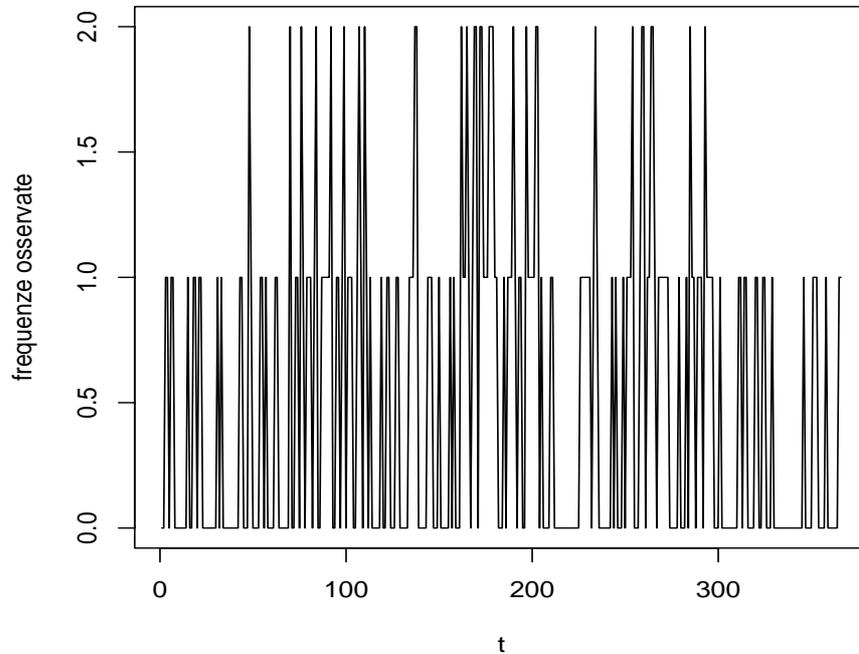


Figura 4.1: Numero di volte in cui si sono verificate precipitazioni nell'area di Tokyo per ogni giorno del biennio 1983–1984.

e la funzione di log-verosimiglianza, a meno delle costanti, è data da

$$l(y_1, \dots, y_{366}, \alpha; a_0, q_0, q) = \sum_{t=1}^{366} \left\{ y_t \alpha_t - n_t \log(1 + e^{\alpha_t}) \right\} - \frac{1}{2} \log q + \\ - \frac{1}{2q} \sum_{t=1}^{366} (\alpha_t - \alpha_{t-1})^2 - \frac{1}{2} \log q_0 - \frac{1}{2q_0} (\alpha_0 - a_0)^2,$$

ottenuta scrivendo la verosimiglianza per il modello di osservazione in forma di famiglia esponenziale naturale.

La Tabella 4.1 mostra le stime ottenute per gli iperparametri a seconda dei diversi valori degli iperparametri iniziali, della severità del criterio di convergenza e dell'algoritmo usato (originale o modificato).

Tabella 4.1: Valori di stima per gli iperparametri e misura del tempo di esecuzione delle funzioni per l'esempio delle precipitazioni a Tokyo per un modello con sola componente di trend ($\varepsilon_\alpha = 10^{-3}$).

ε_θ	$a_0^{(0)}$	$q_0^{(0)}$	$q^{(0)}$	Algoritmo	$l(\hat{a}_0, \hat{Q}_0, \hat{Q})$	\hat{a}_0	$10^4 \times \hat{q}_0$	$10^2 \times \hat{q}$	Tot.it. EM	Media it. WKFS	Tempo
10^{-5}	1	1	1	originale	226931.8	-1.526	8.592	3.358	481	3.131	9 ^m
				modificato	226904.6	-1.526	8.596	3.359	480	1.210	4 ^m
	0	0.1	0.1	originale	227167.3	-1.520	6.140	3.346	610	3.023	10 ^m
				modificato	227143.0	-1.520	6.109	3.347	612	1.088	4 ^m
	0	0.01	0.01	originale	227113.5	-1.496	2.594	3.349	1262	3.000	21 ^m
				modificato	227088.1	-1.496	2.586	3.350	1265	1.044	8 ^m
	-1	1	1	originale	226802.5	-1.527	9.302	3.364	450	3.140	8 ^m
				modificato	226784.5	-1.527	9.258	3.365	451	1.224	3 ^m
	-2	1	1	originale	226725.0	-1.527	9.663	3.368	436	3.144	8 ^m
				modificato	226708.3	-1.527	9.616	3.369	437	1.231	4 ^m
-2	0.1	0.01	originale	227519.3	-1.553	6.302	3.328	502	3.000	9 ^m	
			modificato	227495.2	-1.553	6.303	3.330	501	1.068	3 ^m	
10^{-6}	1	1	1	originale	227260.5	-1.533	2.751	3.341	1365	3.046	24 ^m
				modificato	227235.2	-1.533	2.758	3.343	1361	1.074	9 ^m
	0	0.1	0.1	originale	227257.7	-1.531	1.943	3.341	1867	3.007	33 ^m
				modificato	227232.3	-1.531	1.935	3.343	1874	1.029	12 ^m
	0	0.01	0.01	originale	227237.0	-1.524	0.818	3.343	4244	3.000	1 ^h 12 ^m
				modificato	227211.6	-1.524	0.817	3.344	4253	1.013	28 ^m

ε_θ	$a_0^{(0)}$	$q_0^{(0)}$	$q^{(0)}$	Algoritmo	$l(\hat{a}_0, \hat{Q}_0, \hat{Q})$	\hat{a}_0	$10^4 \times \hat{q}_0$	$10^2 \times \hat{q}$	Tot. it. EM	Media it. WKFS	Tempo
10^{-6}	-1	1	1	originale	227260.8	-1.533	2.997	3.341	1259	3.050	23 ^m
				modificato	227235.5	-1.533	2.990	3.343	1261	1.080	9 ^m
	-2	1	1	originale	227260.9	-1.534	3.127	3.341	1210	3.052	21 ^m
				modificato	227235.6	-1.534	3.114	3.343	1214	1.083	8 ^m
	-2	0.1	0.01	originale	227286.9	-1.542	2.025	3.340	1703	3.000	30 ^m
				modificato	227261.5	-1.542	2.028	3.341	1700	1.020	11 ^m
10^{-7}	1	1	1	originale	227273.0	-1.536	0.868	3.341	4186	3.015	1 ^h 13 ^m
				modificato	227247.7	-1.536	0.871	3.342	4172	1.024	28 ^m
	0	0.1	0.1	originale	227272.3	-1.535	0.613	3.341	5860	3.002	1 ^h 41 ^m
				modificato	227247.0	-1.535	0.611	3.342	5881	1.009	38 ^m
	0	0.01	0.01	originale	227266.6	-1.533	0.258	3.341	13712	3.000	3 ^h 52 ^m
				modificato	227241.3	-1.533	0.258	3.342	13739	1.004	1 ^h 27 ^m
	-1	1	1	originale	227273.1	-1.536	0.946	3.341	3849	3.016	1 ^h 7 ^m
				modificato	227247.8	-1.536	0.944	3.342	3856	1.026	25 ^m
	-2	1	1	originale	227273.1	-1.536	0.987	3.341	3692	3.017	1 ^h 3 ^m
				modificato	227247.8	-1.536	0.983	3.342	3704	1.027	25 ^m
	-2	0.1	0.01	originale	227281.2	-1.538	0.640	3.340	5531	3.000	1 ^h 34 ^m
				modificato	227255.9	-1.538	0.641	3.342	5521	1.006	35 ^m

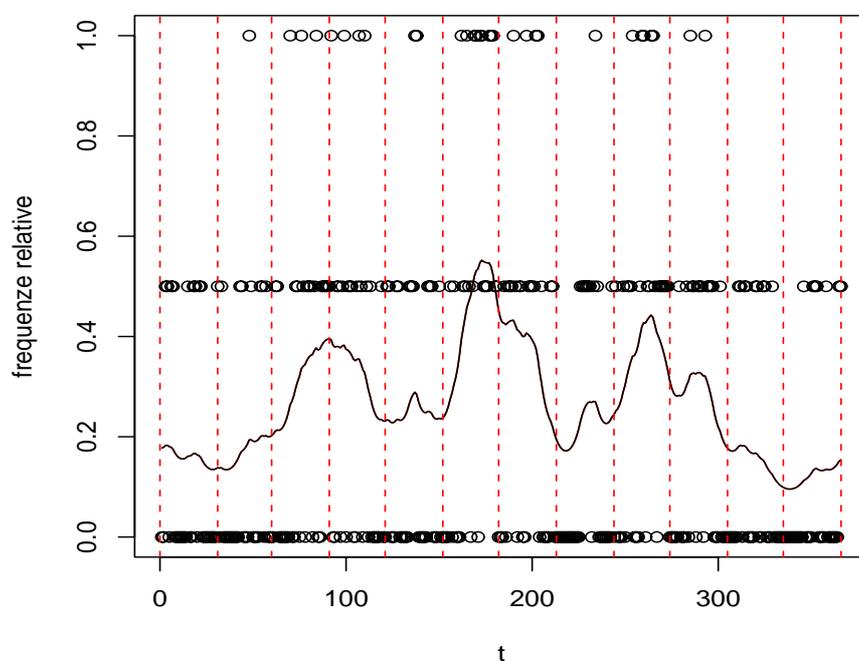


Figura 4.2: Frequenze relative osservate y_t/n_t e curva delle probabilità giornaliere di precipitazione stimate $\hat{\pi}_t$ per l'esempio 1 (le linee verticali tratteggiate delimitano i mesi dell'anno).

I risultati ottenuti sono stati $\hat{a}_0 = -1.54$, $\hat{q}_0 = 0.0001$ e $\hat{q} = 0.0334$ e le corrispondenti stime interpolate $\hat{\pi}_t = h(a_t|_{366})$ sono mostrate nella Figura 4.2 assieme ai dati originali: si nota che i mesi più piovosi sono marzo, aprile, giugno e settembre e che gli inverni sono secchi. Fahrmeir e Tutz (2001) riportano invece i seguenti risultati: $\hat{a}_0 = -1.51$, $\hat{q}_0 = 0.0019$ e $\hat{q} = 0.032$.

In riferimento a quanto detto nella sottosezione 2.4.1 sul ruolo di q come parametro di lisciamiento, viene mostrata la Figura 4.3, in cui la curva dei valori stimati è stata disegnata con il valore della varianza fissato a $q = 0.005$: si noti come, utilizzando un valore che è almeno sei volte più piccolo di quello stimato, la curva sia molto più liscia rispetto alla Figura 4.2.

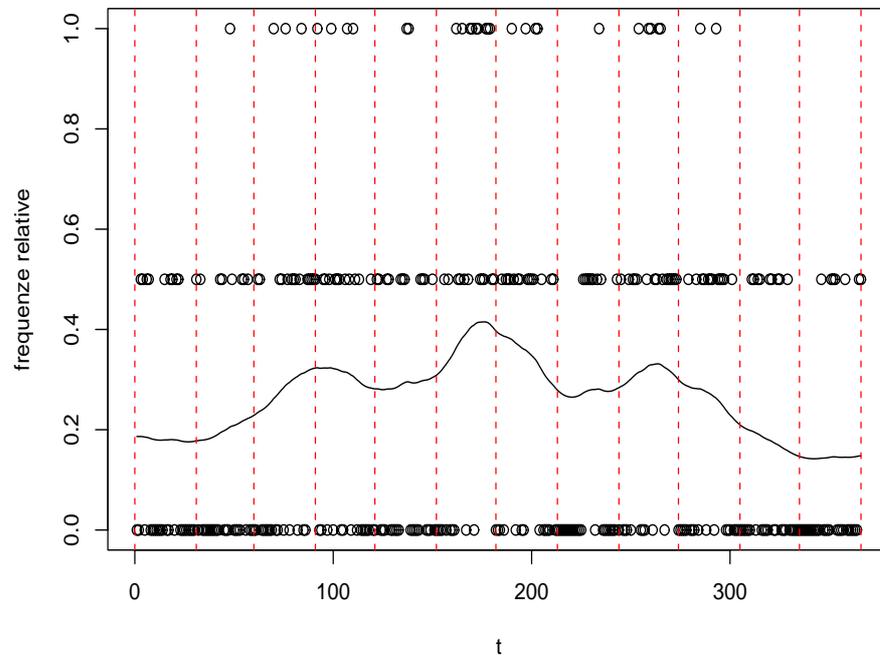


Figura 4.3: Frequenze relative osservate y_t/n_t e curva delle probabilità giornaliere di precipitazione stimate $\hat{\pi}_t$ ottenute con $q = 0.005$ (le linee verticali tratteggiate delimitano i mesi dell'anno).

4.3 Esempio 2: chiamate telefoniche all'Università di Warwick

Il secondo esempio preso in considerazione dal capitolo 8 di Fahrmeir e Tutz (2001) è, a sua volta, riportato da West, Harrison e Migon (1985) e riguarda il conteggio del numero di chiamate telefoniche arrivate durante successivi periodi di trenta minuti all'Università di Warwick dalle ore 00:00 di lunedì 6 settembre 1982 alle ore 24:00 di domenica 12 settembre, ossia esattamente per un'intera settimana. I dati sono sempre disponibili allo stesso sito citato nella sezione 4.2.

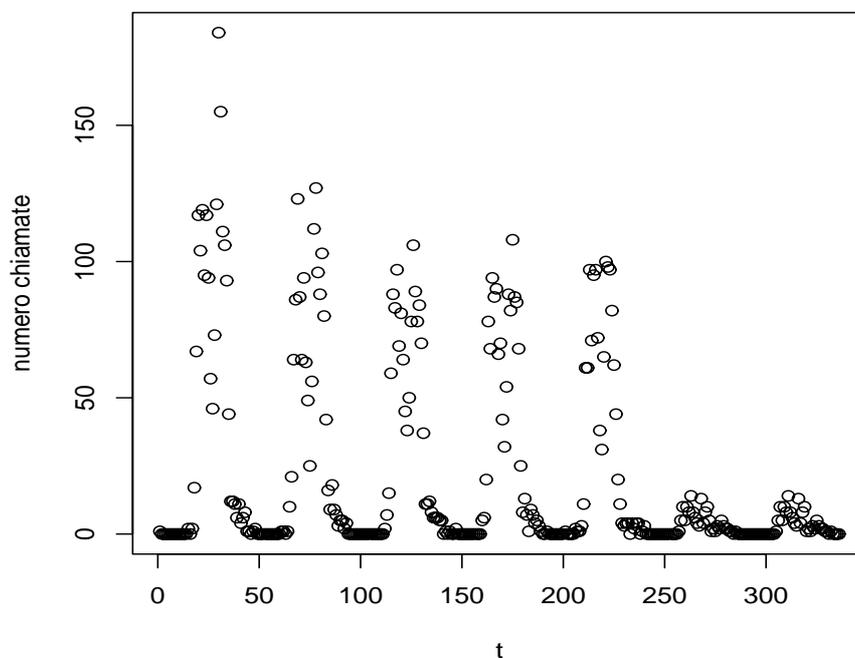


Figura 4.4: Numero di chiamate telefoniche ad intervalli di mezz'ora all'Università di Warwick dal 6 al 12 settembre 1982.

Dalla Figura 4.4 si evince una grande variabilità, visto che sono registrate anche le chiamate nelle ore notturne, le quali sono caratterizzate da lunghi periodi con zero chiamate.

I dati sono stati analizzati utilizzando un modello di Poisson log-lineare dinamico:

$$y_t \sim \mathcal{P}(\exp\{z_t' \alpha_t\}), \quad t = 1, \dots, 336,$$

e la funzione di verosimiglianza per una osservazione y_t , scritta in forma di famiglia esponenziale naturale, è data da

$$p(y_t | \alpha_t) = \frac{1}{y_t!} \exp\{\nu_t y_t - \exp(\nu_t)\}, \quad \nu_t = z_t' \alpha_t;$$

inoltre, nel caso del modello di Poisson, si ha

$$E[y_t | \alpha_t] = \mu_t(\alpha_t) = \exp\{z_t' \alpha_t\} = \Sigma_t(\alpha_t) = \text{Var}[y_t | \alpha_t].$$

Per il modello di transizione, invece, sono stati utilizzati due modelli diversi. Prima, un modello caratterizzato dalla sola componente di trend, utilizzato come esempio in Wagenpfeil (1995). Poi, un modello caratterizzato da una componente di trend e da una componente stagionale, visto che i picchi ed i momenti di stallo sembrano verificarsi sempre negli stessi periodi della giornata che, come già detto, è preso dal capitolo 8 di Fahrmeir e Tutz (2001). I due modelli, assieme ai risultati ottenuti, sono riportati nelle due sottosezioni che seguono.

4.3.1 Sola componente di trend

Come nell'esempio della sezione precedente, si ha $z_t' = F_t = 1$ per ogni t e, quindi

$$\alpha_t = \alpha_{t-1} + \xi_t, \quad \xi_t \sim N(0, q) \quad \alpha_0 \sim N(a_0, q_0);$$

rappresenta il modello di transizione.

I risultati ottenuti utilizzando sia l'algoritmo originale sia quello modificato e diversi valori iniziali sono riportati nella Tabella 4.2: si nota un valore molto piccolo per \hat{q}_0 , mentre $\hat{a}_0 = -1.34$ e $\hat{q} = 0.4$. La Figura 4.5 mostra un buon adattamento dei valori stimati ai valori osservati, evidenziando così la capacità del modello di cambiare di giorno in giorno ma di riuscire anche a cogliere le caratteristiche stagionali della serie storica come i picchi al mattino e al pomeriggio. Wagenpfeil (1995), invece, riporta i seguenti risultati: $\hat{a}_0 = -0.864$, $\hat{q}_0 = 0.015$ e $\hat{q} = 0.44$. Si noti che, nonostante una stima della varianza q sensibilmente più grande (di circa il 10%), il grafico della curva dei valori stimati ottenuto da Wagenpfeil (1995) e quello della Figura 4.5 sono praticamente identici.

Tabella 4.2: Valori di stima per gli iperparametri e misura del tempo di esecuzione delle funzioni per l'esempio delle chiamate telefoniche per un modello con sola componente di trend ($\varepsilon_\alpha = 10^{-3}$).

ε_θ	$a_0^{(0)}$	$q_0^{(0)}$	$q^{(0)}$	Algoritmo	$l(\hat{a}_0, \hat{q}_0, \hat{q})$	\hat{a}_0	$10^2 \times \hat{q}_0$	\hat{q}	Tot. it. EM	Media it. WKFS	Tempo
10^{-4}	1	0.1	0.5	originale	75811.46	-1.2261	0.3792	0.3965	372	8.0054	15^m
			0.1	modificato	75809.63	-1.2219	0.3660	0.3965	383	1.0392	2^m
	0	0.001	0.1	originale	75685.98	-0.5537	0.0543	0.3974	1020	7.9922	40^m
			0.1	modificato	75684.27	-0.5536	0.0543	0.3974	1020	1.0157	6^m
	1	1	1	originale	75815.33	-1.3113	1.0332	0.3965	151	8.1258	6^m
			1	modificato	75813.67	-1.3040	0.8991	0.3965	171	1.1579	1^m
	-2	1	1	originale	75815.47	-1.3526	1.9722	0.3965	83	8.2289	3^m
			1	modificato	75813.43	-1.3482	2.1499	0.3965	75	1.3467	1^m
	-1	1	1	originale	75815.30	-1.3348	1.6538	0.3965	97	8.1959	4^m
			1	modificato	75813.85	-1.3315	1.5086	0.3965	105	1.2476	1^m
-2	0.1	0.5	originale	75818.85	-1.3973	0.8389	0.3964	185	8.0108	7^m	
		0.5	modificato	75817.23	-1.3967	0.8431	0.3965	183	1.0765	1^m	
10^{-5}	1	0.1	0.5	originale	75817.58	-1.3087	0.1200	0.3965	1275	8.0016	51^m
			0.1	modificato	75815.96	-1.3073	0.1159	0.3965	1317	1.0114	7^m
	0	0.001	0.1	originale	75798.78	-1.0765	0.0172	0.3966	6694	7.9988	$4^b 37^m$
			0.1	modificato	75797.18	-1.0764	0.0172	0.3966	6697	1.0024	37^m
	1	1	1	originale	75818.04	-1.3361	0.3269	0.3965	487	8.0390	19^m
			1	modificato	75816.49	-1.3337	0.2863	0.3965	553	1.0488	3^m

ε_θ	$a_0^{(0)}$	$q_0^{(0)}$	$q^{(0)}$	Algoritmo	$l(\hat{a}_0, \hat{q}_0, \hat{q})$	\hat{a}_0	$10^2 \times \hat{q}_0$	\hat{q}	Tot. it. EM	Media it. WKFS	Tempo
10^{-5}	-2	1	1	originale	75817.82	-1.3492	0.6220	0.3964	261	8.0728	10^m
				modificato	75816.06	-1.3478	0.6843	0.3965	236	1.1102	1^m
	-1	1	1	originale	75817.83	-1.3436	0.5211	0.3965	309	8.0615	12^m
				modificato	75816.31	-1.3425	0.4814	0.3965	333	1.0781	2^m
10^{-6}	1	0.1	0.5	originale	75819.24	-1.3633	0.2656	0.3964	604	8.0033	23^m
				modificato	75817.65	-1.3631	0.2673	0.3965	599	1.0234	3^m
10^{-6}	0	0.001	0.1	originale	75819.48	-1.3353	0.0380	0.3964	4169	8.0005	$2^h 46^m$
				modificato	75817.90	-1.3348	0.0367	0.3965	4309	1.0035	24^m
	1	1	1	originale	75816.48	-1.2596	0.0054	0.3965	26203	7.9997	$16^h 42^m$
				modificato	75814.90	-1.2595	0.0054	0.3965	26216	1.0006	$2^h 24^m$
10^{-6}	-2	1	1	originale	75819.24	-1.3440	0.1033	0.3964	1555	8.0122	$1^h 1^m$
				modificato	75817.72	-1.3432	0.0906	0.3965	1769	1.0153	10^m
	-1	1	1	originale	75818.95	-1.3481	0.1967	0.3964	823	8.0231	32^m
				modificato	75817.28	-1.3477	0.2172	0.3965	744	1.0349	4^m
10^{-6}	-2	0.1	0.5	originale	75819.01	-1.3464	0.1650	0.3964	978	8.0194	38^m
				modificato	75817.48	-1.3460	0.1527	0.3965	1055	1.0246	6^m
10^{-6}	-2	0.1	0.5	originale	75819.70	-1.3526	0.0840	0.3964	1923	8.0010	$1^h 18^m$
				modificato	75818.11	-1.3525	0.0846	0.3965	1908	1.0073	10^m

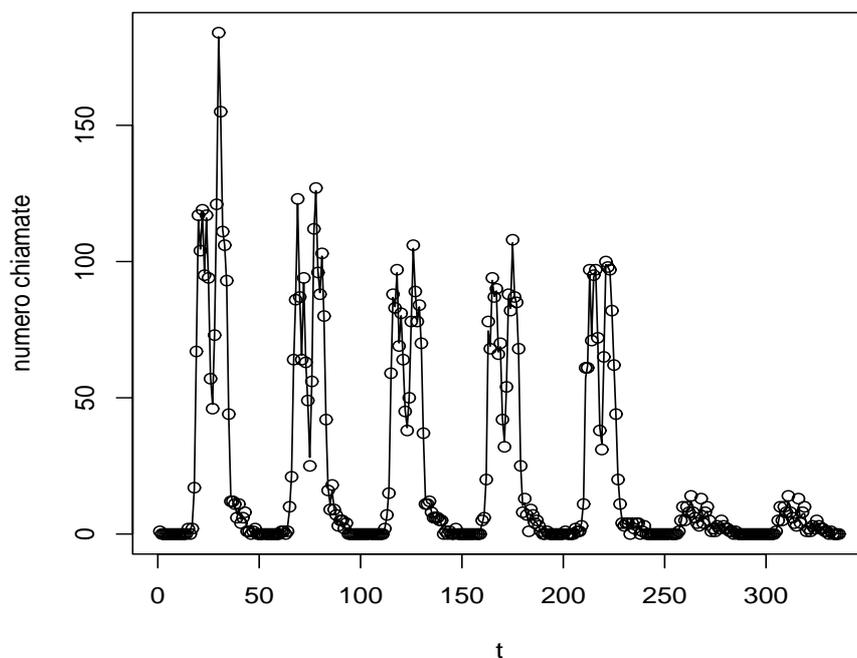


Figura 4.5: Frequenze osservate y_t e curva dei valori attesi stimati $\hat{\mu}_t(\alpha_t)$ per il numero di telefonate ad intervalli di mezz'ora dell'esempio 2 (modello con sola componente di trend).

4.3.2 Trend più stagionalità

Anche se la Figura 4.5 mostra che il modello con sola componente di trend sembra essere un buon modello, visto che i dati vengono interpolati quasi perfettamente, solamente per esercizio numerico viene ora introdotto un modello caratterizzato da una componente di trend e da una componente stagionale: servirà infatti a mostrare che le funzioni costruite in R funzionano correttamente anche nel caso in cui gli stati α_t siano dei vettori e le varianze Q_0 e Q delle matrici.

Il modello di transizione nel caso di stati caratterizzati da trend e stagionalità risulta molto più complesso: la componente stagionale viene infatti

modellata usando l'approccio trigonometrico presentato nella sezione 2.1, utilizzando però solo le prime cinque armoniche della serie di Fourier (se fossero state utilizzate tutte, visto che i periodi giornalieri sono $s = 48$, la dimensione della matrice F sarebbe risultata troppo elevata). La matrice di transizione è quindi rappresentata da

$$F = \begin{pmatrix} 1 & & & & 0 \\ & F_1 & & & \\ & & \ddots & & \\ & & & \ddots & \\ 0 & & & & F_5 \end{pmatrix},$$

con

$$F_i = \begin{pmatrix} \cos i \frac{\pi}{24} & \sin i \frac{\pi}{24} \\ -\sin i \frac{\pi}{24} & \cos i \frac{\pi}{24} \end{pmatrix}, \quad i = 1, \dots, 5.$$

Il vettore di disegno, invece, è dato da

$$z'_t = (1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0),$$

mentre gli stati da

$$\alpha'_t = (\tau_t, \gamma_{1t}, \dots, \gamma_{10t}).$$

I risultati ottenuti utilizzando le funzioni dell'appendice A sono riportati nella Tabella 4.3: anche questa volta, essi sono molto simili a quelli ottenuti da Fahrmeir e Tutz (2001), che riporta $\hat{Q} = \text{diag}(0.05, 0.003, 0.003, 0.0001, 0.0001, 0.0001, 0.0001, 0.0001, 0.0001, 0.00009, 0.00009)$, mentre \hat{a}_0 e \hat{Q}_0 non sono riportati. Infine, la Figura 4.6 mostra di nuovo i valori osservati e la curva dei valori stimati: anche in questo caso il modello riesce a cogliere i picchi mattutini e pomeridiani ed il basso numero di chiamate durante il pranzo e le ore notturne. Si nota anche che lunedì ha il più elevato numero di chiamate e che il comportamento giornaliero rimane simile per tutta la settimana, anche se il livello cambia: martedì ha un livello più elevato di mercoledì, giovedì e venerdì, mentre ci sono poche chiamate il sabato e la domenica.

Tabella 4.3: Valori di stima per gli iperparametri e misura del tempo di esecuzione delle funzioni per l'esempio delle chiamate telefoniche per un modello con trend e stagionalità ($\varepsilon_\alpha = 10^{-3}$); $a_0^* = (1, -3, 1, -1, 0, 0, 0, 0, 0)'$.

ε_θ	$a_0^{(0)}$	$Q_0^{(0)}$	$Q^{(0)}$	Algoritmo	$l(\hat{a}_0, \hat{Q}_0, \hat{Q})$	\hat{a}_0
10^{-5}	1	0.1	0.1	originale	(1.173, -2.919, -2.786, 0.933, -0.860, 0.401, 0.445, -0.324, -0.350, 0.010, 0.119)	(1.173, -2.919, -2.786, 0.933, -0.860, 0.401, 0.445, -0.324, -0.350, 0.010, 0.119)
				modificato	(1.173, -2.919, -2.785, 0.932, -0.861, 0.401, 0.444, -0.323, -0.351, 0.010, 0.119)	(1.173, -2.919, -2.785, 0.932, -0.861, 0.401, 0.444, -0.323, -0.351, 0.010, 0.119)
	0	0.1	0.1	originale	(1.194, -2.924, -2.799, 0.936, -0.867, 0.404, 0.443, -0.323, -0.352, 0.010, 0.118)	(1.194, -2.924, -2.799, 0.936, -0.867, 0.404, 0.443, -0.323, -0.352, 0.010, 0.118)
				modificato	(1.194, -2.924, -2.798, 0.936, -0.866, 0.404, 0.443, -0.323, -0.352, 0.010, 0.118)	(1.194, -2.924, -2.798, 0.936, -0.866, 0.404, 0.443, -0.323, -0.352, 0.010, 0.118)
	0	0.05	0.05	originale	(1.195, -2.898, -2.751, 0.915, -0.826, 0.371, 0.446, -0.335, -0.368, 0.013, 0.112)	(1.195, -2.898, -2.751, 0.915, -0.826, 0.371, 0.446, -0.335, -0.368, 0.013, 0.112)
				modificato	(1.195, -2.899, -2.752, 0.915, -0.827, 0.372, 0.445, -0.334, -0.368, 0.013, 0.112)	(1.195, -2.899, -2.752, 0.915, -0.827, 0.372, 0.445, -0.334, -0.368, 0.013, 0.112)
	-1	0.1	0.1	originale	(1.210, -2.932, -2.817, 0.942, -0.877, 0.412, 0.442, -0.321, -0.349, 0.009, 0.119)	(1.210, -2.932, -2.817, 0.942, -0.877, 0.412, 0.442, -0.321, -0.349, 0.009, 0.119)
				modificato	(1.233, -2.944, -2.843, 0.945, -0.897, 0.423, 0.434, -0.313, -0.348, 0.011, 0.121)	(1.233, -2.944, -2.843, 0.945, -0.897, 0.423, 0.434, -0.313, -0.348, 0.011, 0.121)
	a_0^*	0.05	0.05	originale	(1.202, -2.956, -2.852, 0.960, -0.906, 0.438, 0.445, -0.313, -0.330, 0.006, 0.128)	(1.202, -2.956, -2.852, 0.960, -0.906, 0.438, 0.445, -0.313, -0.330, 0.006, 0.128)
				modificato	(1.202, -2.956, -2.852, 0.960, -0.906, 0.438, 0.445, -0.313, -0.330, 0.006, 0.128)	(1.202, -2.956, -2.852, 0.960, -0.906, 0.438, 0.445, -0.313, -0.330, 0.006, 0.128)
10^{-6}	1	0.1	0.1	originale	(1.188, -2.831, -2.714, 0.934, -0.887, 0.378, 0.450, -0.348, -0.327, -0.009, 0.097)	(1.188, -2.831, -2.714, 0.934, -0.887, 0.378, 0.450, -0.348, -0.327, -0.009, 0.097)
				modificato	(1.189, -2.832, -2.713, 0.934, -0.887, 0.378, 0.449, -0.347, -0.327, -0.008, 0.097)	(1.189, -2.832, -2.713, 0.934, -0.887, 0.378, 0.449, -0.347, -0.327, -0.008, 0.097)
	0	0.1	0.1	originale	(1.195, -2.834, -2.719, 0.936, -0.891, 0.381, 0.449, -0.347, -0.326, -0.009, 0.096)	(1.195, -2.834, -2.719, 0.936, -0.891, 0.381, 0.449, -0.347, -0.326, -0.009, 0.096)
				modificato	(1.195, -2.833, -2.718, 0.936, -0.891, 0.380, 0.449, -0.347, -0.326, -0.009, 0.096)	(1.195, -2.833, -2.718, 0.936, -0.891, 0.380, 0.449, -0.347, -0.326, -0.009, 0.096)
	0	0.05	0.05	originale	(1.203, -2.815, -2.683, 0.918, -0.860, 0.354, 0.450, -0.358, -0.340, -0.007, 0.090)	(1.203, -2.815, -2.683, 0.918, -0.860, 0.354, 0.450, -0.358, -0.340, -0.007, 0.090)
				modificato	(1.203, -2.815, -2.683, 0.918, -0.861, 0.355, 0.450, -0.357, -0.340, -0.007, 0.090)	(1.203, -2.815, -2.683, 0.918, -0.861, 0.355, 0.450, -0.357, -0.340, -0.007, 0.090)
	-1	0.1	0.1	originale	(1.199, -2.838, -2.728, 0.941, -0.899, 0.387, 0.449, -0.345, -0.323, -0.009, 0.097)	(1.199, -2.838, -2.728, 0.941, -0.899, 0.387, 0.449, -0.345, -0.323, -0.009, 0.097)
				modificato	(1.205, -2.846, -2.741, 0.944, -0.913, 0.395, 0.444, -0.340, -0.321, -0.008, 0.099)	(1.205, -2.846, -2.741, 0.944, -0.913, 0.395, 0.444, -0.340, -0.321, -0.008, 0.099)
	a_0^*	0.05	0.05	originale	(1.187, -2.853, -2.758, 0.958, -0.923, 0.409, 0.451, -0.339, -0.309, -0.012, 0.103)	(1.187, -2.853, -2.758, 0.958, -0.923, 0.409, 0.451, -0.339, -0.309, -0.012, 0.103)
				modificato	(1.187, -2.854, -2.759, 0.958, -0.923, 0.409, 0.451, -0.338, -0.309, -0.012, 0.103)	(1.187, -2.854, -2.759, 0.958, -0.923, 0.409, 0.451, -0.338, -0.309, -0.012, 0.103)

ε_θ	$a_0^{(0)}$	$Q_0^{(0)}$	$Q^{(0)}$	Algoritmo	$l(\hat{a}_0, \hat{Q}_0, \hat{Q})$	$10^4 \times \text{diag}(\hat{Q}_0)$
10^{-5}	1	0.1	0.1	originale	5233098	(4.495, 0.937, 1.015, 0.442, 0.463, 0.264, 0.276, 0.214, 0.212, 0.121, 0.121)
				modificato	5234296	(4.483, 0.934, 1.011, 0.440, 0.461, 0.263, 0.275, 0.213, 0.212, 0.121, 0.121)
	0	0.1	0.1	originale	5216230	(4.505, 0.972, 1.053, 0.459, 0.481, 0.274, 0.287, 0.221, 0.219, 0.124, 0.124)
				modificato	5216687	(4.501, 0.971, 1.053, 0.458, 0.481, 0.274, 0.287, 0.221, 0.219, 0.124, 0.124)
	0	0.05	0.05	originale	5175496	(4.766, 1.060, 1.146, 0.504, 0.525, 0.303, 0.317, 0.245, 0.243, 0.140, 0.139)
				modificato	5178119	(4.748, 1.053, 1.139, 0.501, 0.521, 0.301, 0.315, 0.243, 0.241, 0.139, 0.139)
	-1	0.1	0.1	originale	5209190	(4.496, 0.984, 1.068, 0.466, 0.489, 0.279, 0.291, 0.223, 0.221, 0.125, 0.125)
				modificato	5204780	(4.376, 1.009, 1.097, 0.465, 0.492, 0.279, 0.292, 0.222, 0.221, 0.123, 0.123)
	a_0^*	0.05	0.05	originale	5230519	(4.429, 0.922, 1.001, 0.436, 0.460, 0.264, 0.276, 0.212, 0.210, 0.121, 0.121)
				modificato	5232227	(4.420, 0.919, 0.998, 0.435, 0.458, 0.263, 0.275, 0.211, 0.209, 0.120, 0.120)
10^{-6}	1	0.1	0.1	originale	6289655	(1.355, 0.107, 0.115, 0.041, 0.043, 0.025, 0.026, 0.026, 0.026, 0.012, 0.012)
				modificato	6289592	(1.354, 0.107, 0.115, 0.041, 0.043, 0.025, 0.026, 0.026, 0.026, 0.012, 0.012)
	0	0.1	0.1	originale	6311206	(1.307, 0.103, 0.110, 0.039, 0.041, 0.023, 0.025, 0.025, 0.011, 0.011)
				modificato	6310216	(1.308, 0.103, 0.110, 0.039, 0.041, 0.023, 0.025, 0.025, 0.011, 0.011)
	0	0.05	0.05	originale	6287652	(1.364, 0.108, 0.116, 0.041, 0.043, 0.025, 0.026, 0.027, 0.027, 0.012, 0.012)
				modificato	6287413	(1.363, 0.108, 0.116, 0.041, 0.043, 0.025, 0.026, 0.027, 0.027, 0.012, 0.012)
	-1	0.1	0.1	originale	6330406	(1.267, 0.099, 0.106, 0.037, 0.039, 0.022, 0.023, 0.024, 0.024, 0.010, 0.011)
				modificato	6358951	(1.202, 0.094, 0.100, 0.035, 0.037, 0.021, 0.022, 0.023, 0.023, 0.010, 0.010)
	a_0^*	0.05	0.05	originale	6340803	(1.243, 0.096, 0.103, 0.036, 0.038, 0.022, 0.023, 0.023, 0.023, 0.010, 0.010)
				modificato	6340111	(1.243, 0.097, 0.103, 0.036, 0.038, 0.021, 0.023, 0.024, 0.023, 0.010, 0.010)

ε_θ	$a_0^{(0)}$	$Q_0^{(0)}$	$Q^{(0)}$	Algoritmo	$l(\hat{a}_0, \hat{Q}_0, \hat{Q})$	$10^4 \times \text{diag}(\hat{Q})$
10^{-5}	1	0.1	0.1	originale	5233098	(563.976, 4.123, 3.245, 1.679, 1.442, 0.938, 0.857, 1.367, 1.230, 0.597, 0.564)
				modificato	5234296	(564.113, 4.111, 3.236, 1.672, 1.437, 0.935, 0.855, 1.367, 1.230, 0.596, 0.563)
	0	0.1	0.1	originale	5216230	(563.633, 4.318, 3.367, 1.744, 1.493, 0.971, 0.886, 1.340, 1.259, 0.606, 0.572)
				modificato	5216687	(563.651, 4.315, 3.364, 1.742, 1.491, 0.970, 0.885, 1.399, 1.258, 0.605, 0.572)
	0	0.05	0.05	originale	5175496	(562.380, 4.493, 3.525, 1.904, 1.625, 1.057, 0.963, 1.479, 1.329, 0.667, 0.628)
				modificato	5178119	(562.485, 4.469, 3.506, 1.893, 1.616, 1.051, 0.957, 1.475, 1.326, 0.664, 0.625)
	-1	0.1	0.1	originale	5209190	(563.511, 4.443, 3.436, 1.773, 1.515, 0.985, 0.898, 1.411, 1.268, 0.606, 0.572)
				modificato	5204780	(564.075, 4.685, 3.572, 1.767, 1.511, 0.989, 0.902, 1.422, 1.277, 0.597, 0.564)
	a_0^*	0.05	0.05	originale	5230519	(564.257, 4.344, 3.342, 1.660, 1.421, 0.946, 0.862, 1.346, 1.209, 0.600, 0.566)
				modificato	5232227	(564.311, 4.332, 3.333, 1.653, 1.416, 0.942, 0.859, 1.344, 1.207, 0.598, 0.564)
10^{-6}	1	0.1	0.1	originale	6289655	(568.409, 0.569, 0.491, 0.153, 0.144, 0.093, 0.089, 0.863, 0.706, 0.059, 0.059)
				modificato	6289592	(569.172, 0.568, 0.491, 0.153, 0.144, 0.093, 0.089, 0.865, 0.707, 0.059, 0.059)
	0	0.1	0.1	originale	6311206	(568.223, 0.551, 0.475, 0.145, 0.137, 0.088, 0.085, 0.875, 0.710, 0.056, 0.055)
				modificato	6310216	(568.924, 0.551, 0.475, 0.146, 0.137, 0.088, 0.085, 0.876, 0.711, 0.056, 0.055)
	0	0.05	0.05	originale	6287652	(567.718, 0.559, 0.485, 0.156, 0.147, 0.093, 0.089, 0.879, 0.719, 0.059, 0.058)
				modificato	6287413	(568.448, 0.559, 0.485, 0.156, 0.147, 0.093, 0.089, 0.880, 0.719, 0.059, 0.059)
	-1	0.1	0.1	originale	6330406	(568.151, 0.537, 0.462, 0.139, 0.131, 0.084, 0.081, 0.882, 0.710, 0.053, 0.052)
				modificato	6358951	(569.159, 0.517, 0.443, 0.128, 0.121, 0.078, 0.075, 0.899, 0.715, 0.049, 0.048)
	a_0^*	0.05	0.05	originale	6340803	(568.709, 0.539, 0.461, 0.133, 0.126, 0.083, 0.079, 0.852, 0.688, 0.053, 0.052)
				modificato	6340111	(569.430, 0.540, 0.462, 0.133, 0.126, 0.083, 0.080, 0.853, 0.688, 0.053, 0.052)

ε_θ	$a_0^{(0)}$	$Q_0^{(0)}$	$Q^{(0)}$	Algoritmo	$l(\hat{a}_0, \hat{Q}_0, \hat{Q})$	Tot. it. EM	Media it. WKFS	Tempo
10^{-5}	1	0.1	0.1	originale	5233098	1227	4.785	46^m
				modificato	5234296	1230	1.083	12^m
	0	0.1	0.1	originale	5216230	1208	4.788	46^m
				modificato	5216687	1208	1.081	12^m
	0	0.05	0.05	originale	5175496	1119	4.606	41^m
				modificato	5178119	1124	1.062	11^m
	-1	0.1	0.1	originale	5209190	1204	5.051	46^m
				modificato	5204780	1249	1.349	15^m
	a_0^*	0.05	0.05	originale	5230519	1217	4.767	50^m
				modificato	5232227	1219	1.049	12^m
10^{-6}	1	0.1	0.1	originale	6289655	4519	3.634	$2^h 6^m$
				modificato	6289592	4525	1.023	41^m
	0	0.1	0.1	originale	6311206	4668	3.616	$2^h 11^m$
				modificato	6310216	4666	1.021	42^m
	0	0.05	0.05	originale	6287652	4446	3.556	$2^h 2^m$
				modificato	6287413	4452	1.016	40^m
	-1	0.1	0.1	originale	6330406	4814	3.671	$2^h 29^m$
				modificato	6358951	5090	1.086	49^m
	a_0^*	0.05	0.05	originale	6340803	4905	3.604	$2^h 39^m$
				modificato	6340111	4906	1.012	47^m

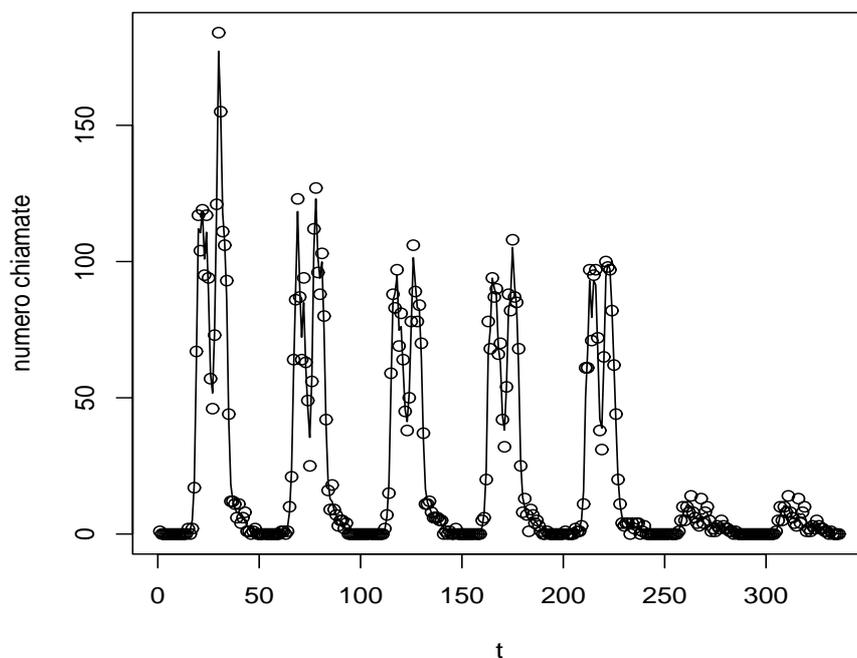


Figura 4.6: Frequenze osservate y_t e curva dei valori attesi stimati $\hat{\mu}_t(\alpha_t)$ per il numero di telefonate ad intervalli di mezz'ora dell'esempio 2 (modello con componente di trend e stagionalità).

4.4 Vantaggi dell'algorithmo modificato

Osservando le Tabelle 4.1, 4.2 e 4.3, dal confronto fra l'algorithmo originale e quello modificato, a parità di valori iniziali, risulta che il secondo è nettamente più veloce: infatti, le stime ottenute per gli iperparametri sono pressoché identiche, ma i tempi di esecuzione sono molto minori, con un risparmio di tempo che va dal 60% dell'esempio delle precipitazioni a Tokyo (vedi Tabella 4.1), fino all'85% dell'esempio delle chiamate telefoniche con modello che include solo la componente di trend (vedi Tabella 4.2). Si fa presente che i calcolatori utilizzavano un processore Pentium IV da 2.8 GHz ed una RAM

Si vede come il numero di iterazioni di IWKFS salga vertiginosamente da dieci prima a ventitre e poi a quarantotto, mantenendosi attorno a questa cifra per altre tre iterazioni dell'algoritmo di tipo EM prima di scendere nuovamente a dieci. Tutto ciò è dovuto al fatto che GKFS fornisce dei valori iniziali a WKFS non proprio buoni e, quindi, l'algoritmo interno ha bisogno di un numero notevole di iterazioni per raggiungere la convergenza.

L'algoritmo modificato, invece, è caratterizzato da un numero di iterazioni di WKFS sempre minore o uguale a quello precedente, ma mai superiore.

4.5 Ulteriori commenti alle tabelle dei risultati

Dopo aver illustrato i vantaggi dell'algoritmo modificato, osservando attentamente le Tabelle 4.1, 4.2 e 4.3, si possono inoltre evincere le seguenti considerazioni.

Influenza dei valori iniziali sui tempi. La prima cosa che si nota è quanto i valori iniziali $a_0^{(0)}$, $Q_0^{(0)}$ e $Q^{(0)}$ influenzino in maniera diversa la durata dell'algoritmo. In particolare, è risultato molto svantaggioso, in termini di tempo di esecuzione, utilizzare dei valori iniziali per le varianze Q_0 e Q che si sono poi rivelati più piccoli di quelli delle stime finali. Infatti, se si osserva il caso con sola componente di trend dell'esempio delle chiamate telefoniche rappresentato nella Tabella 4.2, quando sono stati usati i valori iniziali $a_0^{(0)} = 0$, $q_0^{(0)} = 0.0001$ e $q^{(0)} = 0.1$ già con $\varepsilon_\theta = 10^{-4}$ i tempi di esecuzione dell'algoritmo erano più lenti anche di oltre dieci volte rispetto a quelli ottenuti utilizzando altri valori iniziali, fino ad arrivare ad una convergenza più lenta di più di trenta volte nel caso di $\varepsilon_\theta = 10^{-6}$: utilizzando l'algoritmo originale, sono state necessarie più di sedici ore, mentre cambiando i valori iniziali è bastata poco più di mezz'ora. La lentezza in convergenza utilizzando valori molto piccoli di $a_0^{(0)}$, $Q_0^{(0)}$ e $Q^{(0)}$ è stata confermata anche nel caso del mo-

dello caratterizzato dalla componente stagionale, oltre alla componente di trend: in questo caso la procedura si è rivelata talmente lenta che è stata fermata prima che giungesse al termine e, quindi, non è stato possibile riportare i risultati finali.

Un altro valore che sembra influenzare molto la durata dell'algoritmo è $a_0^{(0)}$. Infatti, quando, negli esempi delle precipitazioni a Tokyo e del modello con sola componente di trend per le chiamate telefoniche, è stato usato un valore iniziale negativo (visto che la stima finale è negativa), i tempi sono minori: ossia, come è naturale aspettarsi, quando la stima iniziale è meno distorta rispetto alla stima finale, i tempi di esecuzione sono molto più veloci. Questo non sembra però confermato nel caso in cui gli stati α_t siano dei vettori caratterizzati da più di una componente: infatti, nell'esempio del modello composto da trend e stagionalità per le chiamate telefoniche, dopo aver osservato i primi risultati, è stato posto $a_0^{(0)} = a_0^* = (1, -3, -3, 1, -1, 0, 0, 0, 0, 0, 0)'$, in modo da avere dei valori iniziali molto prossimi alle stime definitive, ma non se ne è tratto nessun vantaggio in termini di tempo di esecuzione ed, anzi, l'algoritmo è risultato sensibilmente più lento.

Visto che dalle tabelle si nota che i valori iniziali che determinano le stime più veloci quando il criterio di convergenza ε_θ è più largo sono gli stessi che determinano le stime più veloci anche quando il criterio di convergenza è più severo (e similmente per quei valori iniziali che determinano le stime più lente), nella pratica si può consigliare la seguente procedura di stima: utilizzare prima un criterio di convergenza meno severo con diversi valori iniziali per avere una prima indicazione sui tempi di esecuzione; restringere poi il criterio di convergenza e ripetere la stima utilizzando quei valori iniziali per cui l'algoritmo è risultato prima più veloce, ottenendo così delle stime più accurate.

Varianza Q_0 . Dalle tabelle si notano valori molto piccoli per la varianza Q_0 , dell'ordine di 10^{-3} , 10^{-4} e 10^{-5} per i tre esempi finora riportati: questo giustifica una certa difficoltà nella stima di un valore identico

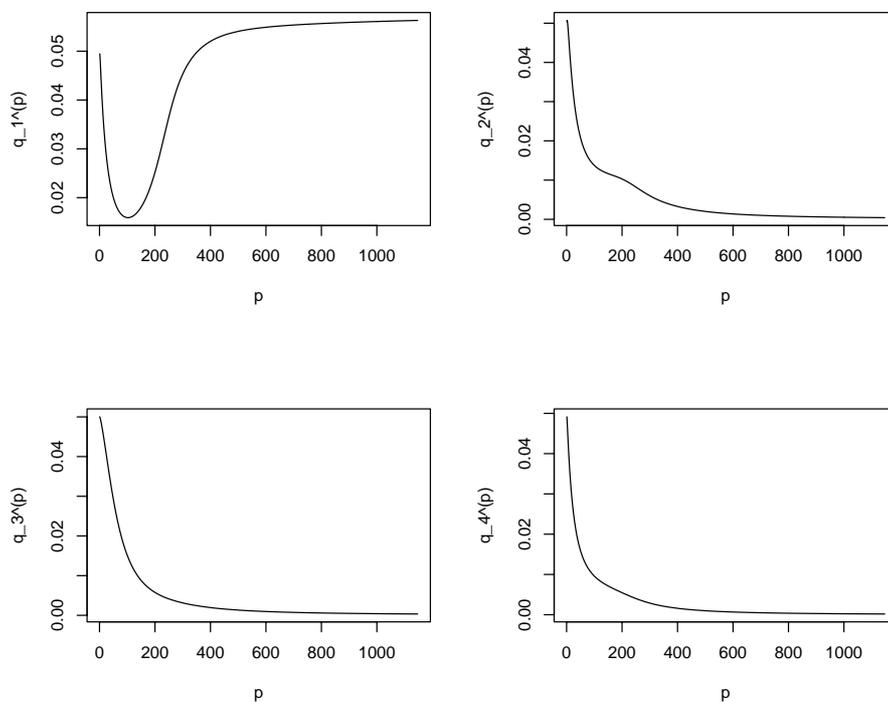


Figura 4.7: Convergenza delle prime quattro componenti della matrice $Q = \text{diag}(q_1, q_2, \dots, q_{11})$ dell'esempio delle chiamate telefoniche nel caso del modello caratterizzato da trend e stagionalità.

per tutte le righe delle tabelle.

Algoritmo di tipo EM. L'algoritmo di tipo EM, come tutti gli algoritmi EM, si dimostra molto lento. Per avere una visione grafica di come le stime degli iperparametri cambino all'aumentare delle iterazioni, viene presentata la Figura 4.7, che rappresenta i primi quattro elementi della matrice diagonale Q dell'esempio delle chiamate telefoniche nel caso di modello caratterizzato da trend e stagionalità: si noti come la prima componente q_1 parta molto vicino a quella che sarà la stima definitiva, scenda verso valori più piccoli per poi risalire di nuovo.

Differenze con le stime di Fahrmeir e Tutz (2001). Come già fatto notare, Fahrmeir e Tutz (2001) riportano delle stime per gli iperparametri leggermente diverse da quelle ottenute in questa tesi, ma non tali da influenzare le curve dei valori attesi stimati, che risultano praticamente identiche. Queste piccole differenze potrebbero essere dovute all'utilizzo di diversi criteri di convergenza, a dei diversi valori per i parametri che regolano la severità di questi criteri, oppure, anche ad un diverso trattamento del problema computazionale illustrato nella sottosezione 4.1.1 del non mantenimento della simmetria delle matrici di covarianza: nella fonte bibliografica appena citata non è infatti riportato come questi tre problemi vengano affrontati e risolti.

4.6 Esempio 3: pubblicità di una tavoletta di cioccolato

Visti i notevoli vantaggi, in termini di tempi di esecuzione, che si conseguono utilizzando la versione modificata dell'algoritmo di tipo EM, verrà ora mostrato un terzo esempio, utilizzando per la stima solamente l'algoritmo modificato: esso è caratterizzato, a differenza dei due precedenti, dalla presenza di una variabile esplicativa che varia nel tempo. Si tratta di un esempio che Fahrmeir e Tutz (2001) riportano da West, Harrison e Migon (1985) ed i dati consistono di conteggi settimanali del numero di persone, su un totale di $n_t = n = 66$, che hanno dato una risposta positiva alla pubblicità di una tavoletta di cioccolato; la variabile esplicativa x_t , invece, fornisce una misura dell'influenza pubblicitaria. È quindi d'interesse sapere se esiste un trend nel tempo per la risposta positiva, quanto misura l'effetto della pubblicità e se questo è costante oppure cambia nel tempo. Si noti che, per i dati di quest'esempio (scaricati dallo stesso sito citato nella sezione 4.2), sono presenti dei dati mancanti per la variabile risposta y_t che conta il numero di risposte positive per ogni settimana, per un totale di dodici dati mancanti.

Un modello adeguato è rappresentato dal modello logit binomiale dina-

mico

$$\begin{aligned} y_t &\sim \mathcal{B}i(66, \pi_t), & \pi_t &= h(\tau_t + x_t\beta_t) = \frac{\exp\{\tau_t + x_t\beta_t\}}{1 + \exp\{\tau_t + x_t\beta_t\}}, \\ \alpha_t &= \alpha_{t-1} + \xi_t, & \xi_t &\sim N(0, Q), \quad \alpha_0 \sim N(a_0, Q_0), \end{aligned}$$

con $\alpha_t = (\tau_t, \beta_t)'$ e $Q = \text{diag}(\sigma_0^2, \sigma_1^2)$. Per quanto riguarda la matrice di transizione, utilizzando il modello appena descritto, si ha

$$F_t = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I_2,$$

mentre il vettore di disegno è dato da

$$z_t' = (1, x_t), \quad t = 1, \dots, 171.$$

La Tabella 4.4 mostra le stime ottenute per gli iperparametri a seconda dei diversi valori degli iperparametri iniziali e della severità del criterio di convergenza: le stime ottenute per la matrice di varianza Q sono state $\sigma_0^2 = 0.006$ e $\sigma_1^2 = 0.00003$. Fahrmeir e Tutz (2001), invece, riportano $\sigma_0^2 = 0.0025$ e $\sigma_1^2 = 0.0002$. In effetti, è stato notato che la figura in basso di pag. 359 di Fahrmeir e Tutz (2001) riporta delle osservazioni dove dovrebbero essere assenti: i dati mancanti sono stati quindi sostituiti.

Si è pertanto cercato di riportare gli stessi valori della figura di Fahrmeir e Tutz (2001) per sostituire le osservazioni mancanti: i valori degli iperparametri ristimati sostituendo i dati mancanti si trovano nella Tabella 4.5. In questo caso le stime ottenute per la matrice di varianza Q sono state $\sigma_0^2 = 0.0036$ e $\sigma_1^2 = 0.00014$, molto più vicine quindi a quelle ottenute da Fahrmeir e Tutz (2001). Si noti che, questa volta, le piccole differenze nelle stime con la fonte bibliografica appena citata, potrebbero essere dovute, oltre a dei diversi criteri di convergenza, a dei diversi gradi di severità di questi o ad un diverso modo di affrontare il problema computazionale esposto nella sottosezione 4.1.1, anche al fatto che sia difficile estrapolare dei valori proiettando dei punti sull'asse delle ordinate e, quindi, i dati mancanti sostituiti potrebbero essere leggermente diversi.

Le Figure 4.8 e 4.9 riportano, rispettivamente, le curve dei valori attesi stimati e le curve per il trend e per l'effetto della pubblicità nel tempo: in ogni figura, il grafico in alto è stato disegnato utilizzando i dati originali, mentre quello in basso è stato disegnato sostituendo i dati mancanti. I due grafici di Figura 4.8 sono praticamente identici: questo denota la versatilità del filtro di Kalman a funzionare correttamente anche nel caso in cui alcuni dati non siano disponibili. Dalla Figura 4.9, invece, si evince che vi è un effetto positivo, anche se di poco, della pubblicità e che questo è pressoché costante nel tempo, mentre il trend è negativo e diminuisce nel tempo.

Tabella 4.4: Valori di stima per gli iperparametri e misura del tempo di esecuzione delle funzioni per l'esempio della pubblicità della tavoletta di cioccolato ($\varepsilon_\alpha = 10^{-3}$, algoritmo modificato, dati originali).

ε_θ	$\alpha_0^{(0)}$	$Q_0^{(0)}$	$Q^{(0)}$	$l(\hat{\alpha}_0, \hat{Q}_0, \hat{Q})$	$\hat{\alpha}_0$	$10^6 \times \text{diag}(\hat{Q}_0)$	$10^3 \times \text{diag}(\hat{Q})$	Tot. it. EM	Media it. WKFS	Tempo
10^{-6}	-1	1	1	219937.8	(-1.537, 0.195)	(8.794, 0.584)	(6.035, 0.033)	2914	1.016	11 ^m
	0	0.01	0.01	218110.5	(-1.463, 0.178)	(5.581, 0.346)	(5.759, 0.039)	4518	1.009	16 ^m
10^{-7}	-1	1	1	220448.9	(-1.522, 0.190)	(1.842, 0.100)	(6.033, 0.032)	14238	1.003	52 ^m
	0	0.1	0.1	219802.0	(-1.507, 0.187)	(8.048, 0.479)	(5.952, 0.034)	3206	1.014	12 ^m
10^{-8}	0	0.01	0.01	219044.1	(-1.487, 0.183)	(1.229, 0.070)	(5.865, 0.036)	20879	1.002	1 ^h 16 ^m
	0	0.1	0.1	219858.1	(-1.508, 0.187)	(3.253, 0.180)	(5.961, 0.034)	7992	1.006	29 ^m
	0	0.01	0.01	219475.2	(-1.498, 0.185)	(0.266, 0.015)	(5.916, 0.035)	97224	1.000	5 ^h 47 ^m

Tabella 4.5: Valori di stima per gli iperparametri e misura del tempo di esecuzione delle funzioni per l'esempio della pubblicità della tavoletta di cioccolato ($\varepsilon_\alpha = 10^{-3}$, algoritmo modificato, dati mancanti sostituiti); $\alpha_0^\# = (-2, 0)'$.

ε_θ	$\alpha_0^{(0)}$	$Q_0^{(0)}$	$Q^{(0)}$	$l(\hat{\alpha}_0, \hat{Q}_0, \hat{Q})$	$\hat{\alpha}_0$	$10^6 \times \text{diag}(\hat{Q}_0)$	$10^3 \times \text{diag}(\hat{Q})$	Tot. it. EM	Media it. WKFS	Tempo
10^{-7}	0	0.1	0.1	205486.4	(-1.531, 0.210)	(11.349, 1.256)	(3.646, 0.143)	1917	1.023	7 ^m
	0	1	1	204536.0	(-1.558, 0.221)	(1.321, 0.156)	(3.416, 0.163)	15980	1.003	58 ^m
	0	0.01	0.01	206746.0	(-1.501, 0.197)	(0.928, 0.090)	(3.918, 0.122)	23739	1.002	1 ^h 31 ^m
	$\alpha_0^\#$	0.01	0.01	204235.1	(-1.567, 0.225)	(0.957, 0.116)	(3.338, 0.170)	21889	1.001	1 ^h 26 ^m
10^{-8}	$\alpha_0^\#$	0.1	0.1	204265.1	(-1.566, 0.225)	(0.985, 0.119)	(3.345, 0.170)	21287	1.002	1 ^h 17 ^m
	0	0.1	0.1	205479.3	(-1.532, 0.210)	(2.734, 0.297)	(3.649, 0.143)	7894	1.006	29 ^m
	0	1	1	204872.4	(-1.548, 0.217)	(0.216, 0.025)	(3.501, 0.156)	98468	1.000	6 ^h 3 ^m

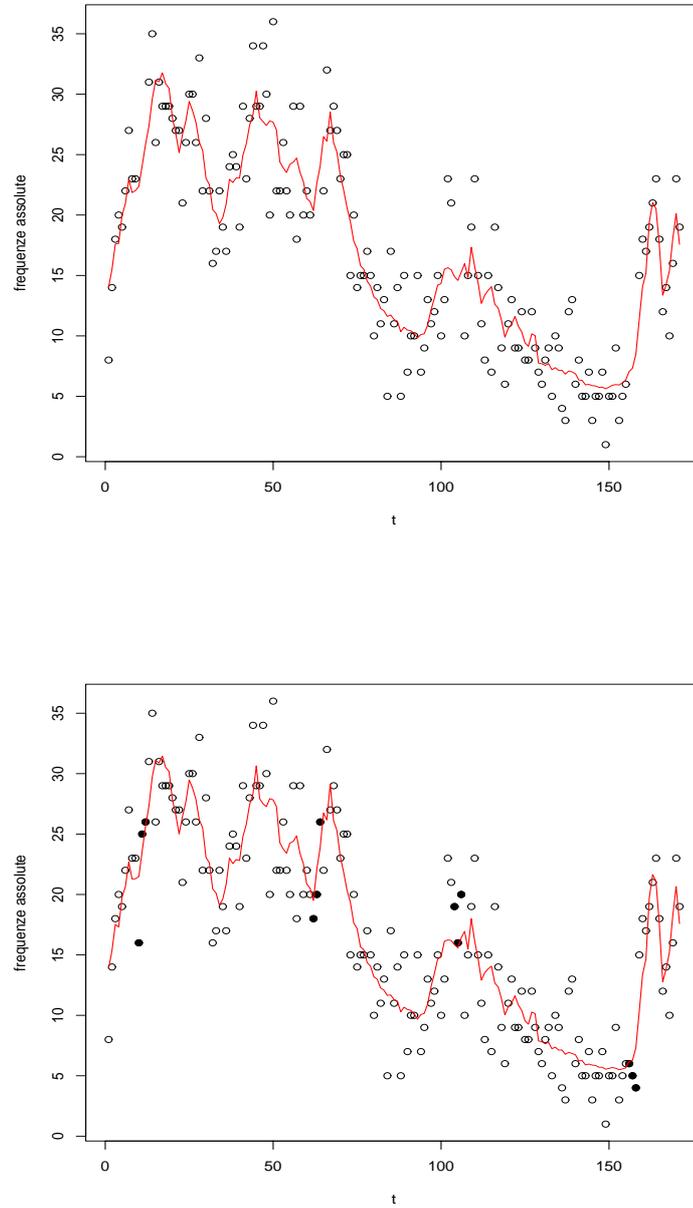


Figura 4.8: Frequenze osservate e curva dei valori attesi stimati per il numero di persone che hanno dato una risposta positiva alla pubblicità della tavoletta di cioccolato: nel grafico in alto sono stati utilizzati i dati originali, in quello in basso i dati mancanti sono stati sostituiti (pallini neri).

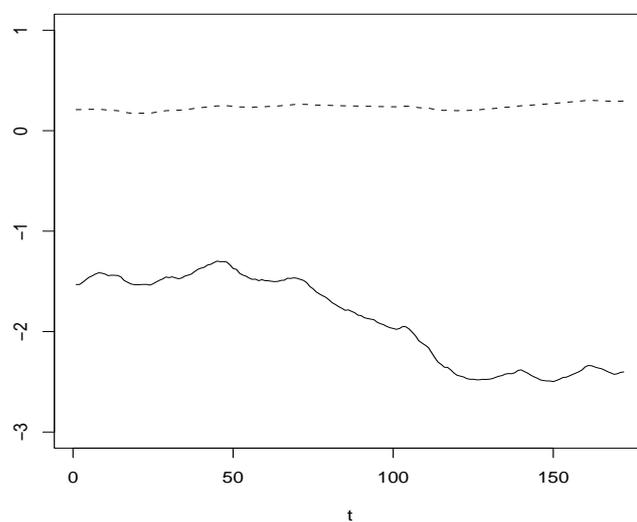
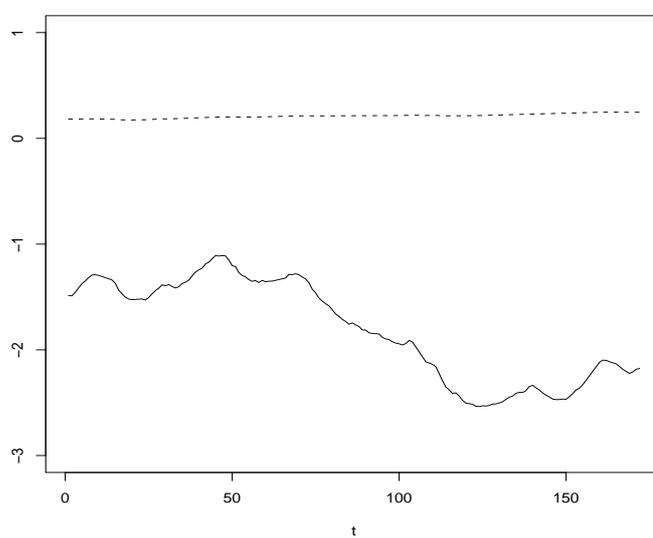


Figura 4.9: Trend interpolato (linea continua) ed effetto della pubblicità (linea tratteggiata): nel grafico in alto sono stati utilizzati i dati originali, in quello in basso i dati mancanti sono stati sostituiti.

Capitolo 5

Conclusioni

Gli obiettivi che ci si era fissati con questa tesi erano essenzialmente due. Il primo, quello di implementare delle funzioni in R da utilizzare con i modelli nello spazio degli stati di famiglia esponenziale, seguendo l'approccio di Fahrmeir e Wagenpfeil (1997) che itera il filtro di Kalman lineare su un modello "costruito". Il secondo, quello di collaudare una versione modificata dell'algoritmo di tipo EM tradizionale per la stima degli iperparametri, verificando se apportasse dei vantaggi in termini di tempi di esecuzione. Inoltre, si riporta che, per quanto riguarda l'algoritmo di tipo EM, sono stati forniti i passaggi matematici necessari per la derivazione dell'algoritmo, del tutto assenti in Fahrmeir e Wagenpfeil (1997).

Il primo obiettivo sembra essere stato raggiunto: applicando le funzioni di R a tre esempi riportati dal capitolo 8 della monografia di Fahrmeir e Tutz (2001), i risultati ottenuti confrontando le curve dei valori attesi stimati sono praticamente identici. Vi sono delle piccole differenze nelle stime degli iperparametri, molto probabilmente dovute all'applicazione di diversi criteri di convergenza oppure al differente grado di severità del parametro che regola la convergenza di questi criteri oppure, ancora, ad un diverso modo di risolvere il problema computazionale esposto nella sottosezione 4.1.1 del non mantenimento della simmetria da parte delle matrici delle covarianze del modello: Fahrmeir e Tutz (2001), infatti, non illustrano come affrontino e risolvano

questi tre problemi.

Per quanto riguarda il secondo obiettivo, la modifica dei valori iniziali per l'algoritmo interno ha apportato un notevole vantaggio: negli esempi su cui è stato provato, i tempi di esecuzione si sono ridotti notevolmente rispetto all'algoritmo originale, del 60% nel caso dell'esempio 1 presentato nel capitolo 4, ed addirittura dell'85% nel caso dell'esempio 2, quando è stato utilizzato un modello con sola componente di trend. Questo vantaggio è essenzialmente dovuto alla bontà dei nuovi valori iniziali per l'algoritmo interno, che permettono un numero di iterazioni notevolmente minore: utilizzando l'algoritmo di tipo EM modificato, infatti, il numero medio di iterazioni dell'algoritmo interno per ogni iterazione di quello esterno è molto prossimo ad uno.

Nonostante la modifica apportata permetta di ridurre il numero di iterazioni dell'algoritmo interno praticamente al minimo, almeno in media, l'algoritmo di tipo EM resta un algoritmo alquanto lento: sono infatti necessarie numerose iterazioni dell'algoritmo esterno per raggiungere la convergenza delle stime degli iperparametri. Comunque, con la velocità raggiunta dai processori oggi giorno e, soprattutto, con l'introduzione della modifica, i tempi di stima si sono ridotti notevolmente rispetto a quando l'approccio di Fahrmeir e Wagenpfeil (1997) ai modelli nello spazio degli stati di famiglia esponenziale venne introdotto. Negli esempi del capitolo 4 su cui è stato utilizzato, inoltre, l'algoritmo si è rivelato molto sensibile ai valori iniziali: infatti, a parità di grado di severità dei criteri di convergenza, a seconda dei valori iniziali l'algoritmo ha riportato dei tempi di esecuzione più lenti anche di trenta volte. Nella sezione 4.5 sono stati illustrati degli accorgimenti che permettono di evitare che la procedura di stima si riveli troppo lunga.

L'algoritmo di tipo EM si conferma tuttavia un algoritmo robusto e, quindi, da preferire ad altri metodi di stima per gli iperparametri come, ad esempio, il metodo basato sul criterio di convalida incrociata generalizzata, che presenta il difetto di poter essere caratterizzato da numerosi minimi locali.

Appendice A

Le funzioni in R

Le seguenti sono le funzioni che sono state costruite in R per l'algoritmo di tipo EM per il filtro di Kalman esteso a famiglie esponenziali. In totale le funzioni riportate sono quattro: sono stati infatti implementati l'algoritmo originale e quello modificato, descritti nel capitolo 3, sia per un modello log-lineare di Poisson dinamico sia per un modello logit binomiale dinamico.

```
poisDGLMun <- function(data, Z, F, a_0, Q_0, Q, epsTHETA=0.00001,
                        epsALPHA=0.001, Qdiag=FALSE) {

  Ti<-length(data)
  y<-matrix(data,1,Ti)
  dimF<-dim(F) [1]
  a_tt1<-array(NA,c(dimF,Ti))
  V_tt1<-array(NA,c(dimF,dimF,Ti))
  a_tt<-array(NA,c(dimF,Ti+1))
  V_tt<-array(NA,c(dimF,dimF,Ti+1))
  a_tT<-array(NA,c(dimF,Ti+1))
  V_tT<-array(NA,c(dimF,dimF,Ti+1))
  B_t<-array(NA,c(dimF,dimF,Ti))
  itv<-NULL i<-1
  convcrit2<-10*epsTHETA
  while (convcrit2 >epsTHETA) {

  a_tt[,1] <- a_0
```

```

V_tt[, ,1] <- Q_0
for(t in 1:Ti) {

a_tt1[,t] <- F %>% a_tt[,t]
V_tt1[, ,t] <- F %>% V_tt[, ,t] %>% t(F) + Q
V_tt1[, ,t] <- (V_tt1[, ,t]+t(V_tt1[, ,t]))/2

Z_t <- t(as.matrix(Z[t,]))
D_t <- exp(Z_t %>% a_tt1[,t])
sigma_t <- D_t
#Kalman gain:
K_t <- V_tt1[, ,t] %>% t(Z_t) %>% D_t %>% (D_t %>% Z_t %>% V_tt1[, ,t] %>%
      t(Z_t) %>% D_t + sigma_t)^-1
a_tt[,t+1] <- a_tt1[,t]
if(!is.na(y[,t])) a_tt[,t+1] <- a_tt1[,t] + K_t %>% (y[,t]-exp(Z_t %>%
      a_tt1[,t]))

V_tt[, ,t+1] <- V_tt1[, ,t]
if(!is.na(y[,t])) V_tt[, ,t+1] <- V_tt1[, ,t] - K_t %>% D_t %>% Z_t %>%
      V_tt1[, ,t]
V_tt[, ,t+1] <- (V_tt[, ,t+1]+t(V_tt[, ,t+1]))/2
}

#smoothing:
a_tT[,Ti+1] <- a_tt[,Ti+1]
V_tT[, ,Ti+1] <- V_tt[, ,Ti+1]
for (t in Ti:1) {

B_t[, ,t] <- V_tt[, ,t] %>% t(F) %>% solve(V_tt1[, ,t])
a_tT[,t] <- a_tt[,t] + B_t[, ,t] %>% (a_tT[,t+1]-a_tt1[,t])
V_tT[, ,t] <- V_tt[, ,t] + B_t[, ,t] %>% (V_tT[, ,t+1]-V_tt1[, ,t]) %>%
      t(B_t[, ,t])
V_tT[, ,t] <- (V_tT[, ,t]+t(V_tT[, ,t]))/2
}

it<-0
convcrit<-10*epsALPHA
while (convcrit >epsALPHA) {

```

```

alpha0<-a_tT
for(t in 1:Ti) {

a_tt1[,t] <- F %% a_tt[,t]
V_tt1[, ,t] <- F %% V_tt[, ,t] %% t(F) + Q
V_tt1[, ,t] <- (V_tt1[, ,t]+t(V_tt1[, ,t]))/2

Z_t <- t(as.matrix(Z[t,]))
D_t <- exp(Z_t %% alpha0[,t+1])
#sigma_t <- D_t
if(!is.na(y[,t])) y_work <- ((D_t)^-1) %% (y[,t]-exp(Z_t %%
                                alpha0[,t+1])) + Z_t %% alpha0[,t+1]

W_t <- D_t
#Kalman gain:
K_t <- V_tt1[, ,t] %% t(Z_t) %% (Z_t %% V_tt1[, ,t] %% t(Z_t)+ W_t^-1)^-1
a_tt[,t+1] <- a_tt1[,t]
if(!is.na(y[,t])) a_tt[,t+1] <- a_tt1[,t] + K_t %% (y_work - Z_t %%
                                a_tt1[,t])

V_tt[, ,t+1] <- V_tt1[, ,t]
if(!is.na(y[,t])) V_tt[, ,t+1] <- (diag(1,dimF) - K_t %% Z_t) %%
                                V_tt1[, ,t]

V_tt[, ,t+1] <- (V_tt[, ,t+1]+t(V_tt[, ,t+1]))/2
}

#smoothing:
a_tT[,Ti+1] <- a_tt[,Ti+1]
V_tT[, ,Ti+1] <- V_tt[, ,Ti+1]
for (t in Ti:1) {

B_t[, ,t] <- V_tt[, ,t] %% t(F) %% solve(V_tt1[, ,t])
a_tT[,t] <- a_tt[,t] + B_t[, ,t] %% (a_tT[,t+1]-a_tt1[,t])
V_tT[, ,t] <- V_tt[, ,t] + B_t[, ,t] %% (V_tT[, ,t+1]-V_tt1[, ,t]) %%
                                t(B_t[, ,t])
V_tT[, ,t] <- (V_tT[, ,t]+t(V_tT[, ,t]))/2
}

convcrit <- mean(abs(alpha0-a_tT))/(1+mean(abs(alpha0-a_tT)))
it<-it+1

```

```

}

a_0prev<-a_0
Q_0prev<-Q_0
Qprev<-Q

a_0 <- a_tT[,1]
Q_0 <- V_tT[, ,1]
if(Qdiag==TRUE) Q_0 <-diag(diag(Q_0))
Q <- matrix(0,dimF,dimF)
for (t in 1:Ti) {

S <- (a_tT[,t+1] - F %%% a_tT[,t]) %%% t(a_tT[,t+1] - F %%% a_tT[,t]) +
      V_tT[, ,t+1] - F %%% B_t[, ,t] %%% V_tT[, ,t+1] - t(F %%% B_t[, ,t] %%%
      V_tT[, ,t+1]) + F %%% V_tT[, ,t] %%% t(F)
Q <- Q+S
}

Q <- Q/Ti
Q <- (Q+t(Q))/2
if(Qdiag==TRUE) Q <- diag(diag(Q))
convcrit2 <- (mean(abs(a_0prev-a_0))/(1+mean(abs(a_0prev-a_0))) +
             mean(abs(Q_0prev-Q_0))/(1+mean(abs(Q_0prev-Q_0))) +
             mean(abs(Qprev-Q))/(1+mean(abs(Qprev-Q))))/3
itv[i]<-it+1 i<-i+1
}

loglik<-0
for (t in 1:Ti) if(!is.na(y[,t])) loglik <- loglik + t(as.matrix(Z[t,]))
             %%% a_tT[,t+1] %%% y[1,t] - exp(t(as.matrix(Z[t,])) %%% a_tT[,t+1])

l_0 <- t(a_tT[,1] - a_0prev) %%% solve(Q_0prev) %%% (a_tT[,1] - a_0prev)/2
      + log(det(matrix(Q_0prev,dimF,dimF)))/2

pen<-0
for (t in 1:Ti) pen <- pen + (t(a_tT[,t+1] - F %%% a_tT[,t]) %%%
      solve(Qprev) %%% (a_tT[,t+1] - F %%% a_tT[,t]))/2 + log(det(Qprev))*Ti/2

```

```

penloglik <- loglik - l_0 - pen

list(penloglik=penloglik, alpha=a_tT, V_tT=V_tT, alphas=a_tt, V_tt=V_tt,
      alphas1=a_tt1, V_tt1=V_tt1, a_0=a_0prev, Q_0=Q_0prev, Q=Qprev,
      itvm=mean(itv), itn=length(itv))

}

poisDGLMun2 <- function(data, Z, F, a_0, Q_0, Q, epsTHETA=0.00001,
                        epsALPHA=0.001, Qdiag=FALSE) {
  Ti<-length(data)
  y<-matrix(data,1,Ti)
  dimF<-dim(F) [1]
  a_tt1<-array(NA,c(dimF,Ti))
  V_tt1<-array(NA,c(dimF,dimF,Ti))
  a_tt<-array(NA,c(dimF,Ti+1))
  V_tt<-array(NA,c(dimF,dimF,Ti+1))
  a_tT<-array(NA,c(dimF,Ti+1))
  V_tT<-array(NA,c(dimF,dimF,Ti+1))
  B_t<-array(NA,c(dimF,dimF,Ti))
  itv<-NULL i<-1

  a_tt[,1] <- a_0
  V_tt[, ,1] <- Q_0
  for(t in 1:Ti) {

    a_tt1[,t] <- F %*% a_tt[,t]
    V_tt1[, ,t] <- F %*% V_tt[, ,t] %*% t(F) + Q
    V_tt1[, ,t] <- (V_tt1[, ,t]+t(V_tt1[, ,t]))/2

    Z_t <- t(as.matrix(Z[t,]))
    D_t <- exp(Z_t %*% a_tt1[,t])
    sigma_t <- D_t
  }
}

```

```

#Kalman gain:
K_t <- V_tt1[, ,t] %*% t(Z_t) %*% D_t %*% (D_t %*% Z_t %*% V_tt1[, ,t] %*%
      t(Z_t) %*% D_t + sigma_t)^-1
a_tt[,t+1] <- a_tt1[,t]
if(!is.na(y[,t])) a_tt[,t+1] <- a_tt1[,t] + K_t %*% (y[,t]-exp(Z_t %*%
      a_tt1[,t]))

V_tt[, ,t+1] <- V_tt1[, ,t]
if(!is.na(y[,t])) V_tt[, ,t+1] <- V_tt1[, ,t] - K_t %*% D_t %*% Z_t %*%
      V_tt1[, ,t]
V_tt[, ,t+1] <- (V_tt[, ,t+1]+t(V_tt[, ,t+1]))/2
}

#smoothing:
a_tT[,Ti+1] <- a_tt[,Ti+1]
V_tT[, ,Ti+1] <- V_tt[, ,Ti+1]
for (t in Ti:1) {

B_t[, ,t] <- V_tt[, ,t] %*% t(F) %*% solve(V_tt1[, ,t])
a_tT[,t] <- a_tt[,t] + B_t[, ,t] %*% (a_tT[,t+1]-a_tt1[,t])
V_tT[, ,t] <- V_tt[, ,t] + B_t[, ,t] %*% (V_tT[, ,t+1]-V_tt1[, ,t]) %*%
      t(B_t[, ,t])
V_tT[, ,t] <- (V_tT[, ,t]+t(V_tT[, ,t]))/2
}

convcrit2<-10*epsTHETA
while (convcrit2 >epsTHETA) {

a_0prev<-a_0
Q_0prev<-Q_0
Qprev<-Q

a_0 <- a_tT[,1]
Q_0 <- V_tT[, ,1]
if(Qdiag==TRUE) Q_0 <-diag(diag(Q_0))
Q <- matrix(0,dimF,dimF)
for (t in 1:Ti) {

S <- (a_tT[,t+1] - F %*% a_tT[,t]) %*% t(a_tT[,t+1] - F %*% a_tT[,t]) +

```

```

        V_tT[, ,t+1] - F %*% B_t[, ,t] %*% V_tT[, ,t+1] - t(F %*% B_t[, ,t] %*%
        V_tT[, ,t+1]) + F %*% V_tT[, ,t] %*% t(F)
    Q <- Q+S
}

Q <- Q/Ti
Q <- (Q+t(Q))/2
if(Qdiag==TRUE) Q <- diag(diag(Q))

a_tt[,1] <- a_0
V_tt[, ,1] <- Q_0
it<-0
convcrit<-10*epsALPHA
while (convcrit >epsALPHA) {

alpha0<-a_tT
for(t in 1:Ti) {

a_tt1[,t] <- F %*% a_tt[,t]
V_tt1[, ,t] <- F %*% V_tt[, ,t] %*% t(F) + Q
V_tt1[, ,t] <- (V_tt1[, ,t]+t(V_tt1[, ,t]))/2

Z_t <- t(as.matrix(Z[t,]))
D_t <- exp(Z_t %*% alpha0[,t+1])
#sigma_t <- D_t
if(!is.na(y[,t])) y_work <- ((D_t)^-1) %*% (y[,t]-exp(Z_t %*% alpha0[,t+1]))
+ Z_t %*% alpha0[,t+1]

W_t <- D_t
#Kalman gain:
K_t <- V_tt1[, ,t] %*% t(Z_t) %*% (Z_t %*% V_tt1[, ,t] %*% t(Z_t)+ W_t^-1)^-1

a_tt[,t+1] <- a_tt1[,t]
if(!is.na(y[,t])) a_tt[,t+1] <- a_tt1[,t] + K_t %*% (y_work - Z_t %*%
a_tt1[,t])

V_tt[, ,t+1] <- V_tt1[, ,t]
if(!is.na(y[,t])) V_tt[, ,t+1] <- (diag(1,dimF) - K_t %*% Z_t) %*% V_tt1[, ,t]
V_tt[, ,t+1] <- (V_tt[, ,t+1]+t(V_tt[, ,t+1]))/2
}

```

```

#smoothing:
a_tT[,Ti+1] <- a_tt[,Ti+1]
V_tT[, ,Ti+1] <- V_tt[, ,Ti+1]
for (t in Ti:1) {

B_t[, ,t] <- V_tt[, ,t] %*% t(F) %*% solve(V_tt1[, ,t])
a_tT[,t] <- a_tt[,t] + B_t[, ,t] %*% (a_tT[,t+1]-a_tt1[,t])
V_tT[, ,t] <- V_tt[, ,t] + B_t[, ,t] %*% (V_tT[, ,t+1]-V_tt1[, ,t]) %*%
      t(B_t[, ,t])
V_tT[, ,t] <- (V_tT[, ,t]+t(V_tT[, ,t]))/2
}

convcrit <- mean(abs(alpha0-a_tT))/(1+mean(abs(alpha0-a_tT)))
it<-it+1
}

convcrit2 <- (mean(abs(a_0prev-a_0))/(1+mean(abs(a_0prev-a_0))) +
      mean(abs(Q_0prev-Q_0))/(1+mean(abs(Q_0prev-Q_0))) +
      mean(abs(Qprev-Q))/(1+mean(abs(Qprev-Q))))/3
itv[i]<-it i<-i+1
}

loglik<-0
for (t in 1:Ti) if(!is.na(y[,t])) loglik <- loglik + t(as.matrix(Z[t,]))
      %*% a_tT[,t+1] %*% y[1,t] - exp(t(as.matrix(Z[t,])) %*% a_tT[,t+1])

l_0 <- t(a_tT[,1] - a_0) %*% solve(Q_0) %*% (a_tT[,1] - a_0)/2 +
      log(det(matrix(Q_0,dimF,dimF)))/2

pen<-0
for (t in 1:Ti) pen <- pen + (t(a_tT[,t+1] - F %*% a_tT[,t]) %*% solve(Q)
      %*% (a_tT[,t+1] - F %*% a_tT[,t]))/2 + log(det(Q))*Ti/2

penloglik <- loglik - l_0 - pen

list(penloglik=penloglik, alpha=a_tT, V_tT=V_tT, alphas=a_tt, V_tt=V_tt,
      alphas1=a_tt1, V_tt1=V_tt1, a_0=a_0, Q_0=Q_0, Q=Q, itvm=mean(itv),

```

```

        itn=length(itv))

}

binomDGLMun <- function(data, n, Z, F, a_0, Q_0, Q, epsTHETA=0.00001,
                        epsALPHA=0.001, Qdiag=FALSE) {

Ti<-length(data)
y<-matrix(data,1,Ti)
dimF<-dim(F) [1]
a_tt1<-array(NA,c(dimF,Ti))
V_tt1<-array(NA,c(dimF,dimF,Ti))
a_tt<-array(NA,c(dimF,Ti+1))
V_tt<-array(NA,c(dimF,dimF,Ti+1))
a_tT<-array(NA,c(dimF,Ti+1))
V_tT<-array(NA,c(dimF,dimF,Ti+1))
B_t<-array(NA,c(dimF,dimF,Ti))
itv<-NULL i<-1
convcrit2<-10
while (convcrit2 >epsTHETA) {

a_tt[,1] <- a_0
V_tt[, ,1] <- Q_0
for(t in 1:Ti) {

a_tt1[,t] <- F %% a_tt[,t]
V_tt1[, ,t] <- F %% V_tt[, ,t] %% t(F) + Q
V_tt1[, ,t] <- (V_tt1[, ,t]+t(V_tt1[, ,t]))/2

Z_t <- t(as.matrix(Z[t,]))
D_t <- n[t]*exp(Z_t %% a_tt1[,t])/(1+exp(Z_t %% a_tt1[,t]))^2
sigma_t <- D_t
#Kalman gain:
K_t <- V_tt1[, ,t] %% t(Z_t) %% D_t %% (D_t %% Z_t %% V_tt1[, ,t] %%

```

```

      t(Z_t) %*% D_t + sigma_t)^-1
a_tt[,t+1] <- a_tt1[,t]
if(!is.na(y[,t])) a_tt[,t+1] <- a_tt1[,t] + K_t %*% (y[,t]-n[t]*exp(Z_t
      %*% a_tt1[,t]))/(1+exp(Z_t %*% a_tt1[,t]))
V_tt[, ,t+1] <- V_tt1[, ,t]
if(!is.na(y[,t])) V_tt[, ,t+1] <- V_tt1[, ,t] - K_t %*% D_t %*% Z_t %*%
      V_tt1[, ,t]
V_tt[, ,t+1] <- (V_tt[, ,t+1]+t(V_tt[, ,t+1]))/2 }

#smoothing:
a_tT[,Ti+1] <- a_tt[,Ti+1]
V_tT[, ,Ti+1] <- V_tt[, ,Ti+1]
for (t in Ti:1) {

B_t[, ,t] <- V_tt[, ,t] %*% t(F) %*% solve(V_tt1[, ,t])
a_tT[,t] <- a_tt[,t] + B_t[, ,t] %*% (a_tT[,t+1]-a_tt1[,t])
V_tT[, ,t] <- V_tt[, ,t] + B_t[, ,t] %*% (V_tT[, ,t+1]-V_tt1[, ,t]) %*%
      t(B_t[, ,t])
V_tT[, ,t] <- (V_tT[, ,t]+t(V_tT[, ,t]))/2
}

it<-0
convcrit<-10*epsALPHA
while (convcrit >epsALPHA) {

alpha0<-a_tT
for(t in 1:Ti) {

a_tt1[,t] <- F %*% a_tt[,t]
V_tt1[, ,t] <- F %*% V_tt[, ,t] %*% t(F) + Q
V_tt1[, ,t] <- (V_tt1[, ,t]+t(V_tt1[, ,t]))/2

Z_t <- t(as.matrix(Z[t,]))
D_t <- n[t]*exp(Z_t %*% alpha0[,t+1])/(1+exp(Z_t %*% alpha0[,t+1]))^2
#sigma_t <- D_t
if(!is.na(y[,t])) y_work <- ((D_t)^-1) %*% (y[,t]-n[t]*exp(Z_t %*%
      alpha0[,t+1]))/(1+exp(Z_t %*% alpha0[,t+1])) + Z_t %*% alpha0[,t+1]
W_t <- D_t

```

```

#Kalman gain:
K_t <- V_tt1[, ,t] %*% t(Z_t) %*% (Z_t %*% V_tt1[, ,t] %*% t(Z_t)+ W_t^-1)^-1
a_tt[,t+1] <- a_tt1[,t]
if(!is.na(y[,t])) a_tt[,t+1] <- a_tt1[,t] + K_t %*% (y_work - Z_t %*%
                                a_tt1[,t])

V_tt[, ,t+1] <- V_tt1[, ,t]
if(!is.na(y[,t])) V_tt[, ,t+1] <- (diag(1,dimF) - K_t %*% Z_t) %*%
                                V_tt1[, ,t]
V_tt[, ,t+1] <- (V_tt[, ,t+1]+t(V_tt[, ,t+1]))/2
}

#smoothing:
a_tT[,Ti+1] <- a_tt[,Ti+1]
V_tT[, ,Ti+1] <- V_tt[, ,Ti+1]
for (t in Ti:1) {

B_t[, ,t] <- V_tt[, ,t] %*% t(F) %*% solve(V_tt1[, ,t])
a_tT[,t] <- a_tt[,t] + B_t[, ,t] %*% (a_tT[,t+1]-a_tt1[,t])
V_tT[, ,t] <- V_tt[, ,t] + B_t[, ,t] %*% (V_tT[, ,t+1]-V_tt1[, ,t]) %*%
                                t(B_t[, ,t])
V_tT[, ,t] <- (V_tT[, ,t]+t(V_tT[, ,t]))/2
}

convcrit <- mean(abs(alpha0-a_tT))/(1+mean(abs(alpha0-a_tT)))
it<-it+1
}

a_0prev<-a_0
Q_0prev<-Q_0
Qprev<-Q

a_0 <- a_tT[,1]
Q_0 <- V_tT[, ,1]
if(Qdiag==TRUE) Q_0 <-diag(diag(Q_0))
Q <- matrix(0,dimF,dimF)
for (t in 1:Ti) {

S <- (a_tT[,t+1] - F %*% a_tT[,t]) %*% t(a_tT[,t+1] - F %*% a_tT[,t]) +

```

```

      V_tT[, ,t+1] - F %>% B_t[, ,t] %>% V_tT[, ,t+1] - t(F %>% B_t[, ,t]
      %>% V_tT[, ,t+1]) + F %>% V_tT[, ,t] %>% t(F)
Q <- Q+S
}

Q <- Q/Ti
Q <- (Q+t(Q))/2
if(Qdiag==TRUE) Q <- diag(diag(Q))
convcrit2 <- (mean(abs(a_0prev-a_0))/(1+mean(abs(a_0prev-a_0))) +
              mean(abs(Q_0prev-Q_0))/(1+mean(abs(Q_0prev-Q_0))) +
              mean(abs(Qprev-Q))/(1+mean(abs(Qprev-Q))))/3
itv[i]<-it+1
i<-i+1
}

loglik<-0
for (t in 1:Ti) if(!is.na(y[,t])) loglik <- loglik + t(as.matrix(Z[t,]))
              %>% a_tT[,t+1] %>% y[1,t] - n[t]*log(1+exp(t(as.matrix(Z[t,]))
              %>% a_tT[,t+1]))

l_0 <- t(a_tT[,1] - a_0prev) %>% solve(Q_0prev) %>% (a_tT[,1] - a_0prev)/2
      + log(det(matrix(Q_0prev,dimF,dimF)))/2

pen<-0
for (t in 1:Ti) pen <- pen + (t(a_tT[,t+1] - F %>% a_tT[,t]) %>%
      solve(Qprev) %>% (a_tT[,t+1] - F %>% a_tT[,t]))/2
      + log(det(Qprev))*Ti/2

penloglik <- loglik - l_0 - pen

list(penloglik=penloglik, alpha=a_tT, V_tT=V_tT, alphas=a_tt, V_tt=V_tt,
      alphas1=a_tt1, V_tt1=V_tt1, a_0=a_0prev, Q_0=Q_0prev, Q=Qprev,
      itvm=mean(itv), itn=length(itv))
}

```

```

binomDGLMun2 <- function(data, n, Z, F, a_0, Q_0, Q, epsTHETA=0.00001,
                        epsALPHA=0.001, Qdiag=FALSE) {

  Ti<-length(data)
  y<-matrix(data,1,Ti)
  dimF<-dim(F) [1]
  a_tt1<-array(NA,c(dimF,Ti))
  V_tt1<-array(NA,c(dimF,dimF,Ti))
  a_tt<-array(NA,c(dimF,Ti+1))
  V_tt<-array(NA,c(dimF,dimF,Ti+1))
  a_tT<-array(NA,c(dimF,Ti+1))
  V_tT<-array(NA,c(dimF,dimF,Ti+1))
  B_t<-array(NA,c(dimF,dimF,Ti))
  itv<-NULL i<-1

  a_tt[,1] <- a_0
  V_tt[, ,1] <- Q_0
  for(t in 1:Ti) {

    a_tt1[,t] <- F %% a_tt[,t]
    V_tt1[, ,t] <- F %% V_tt[, ,t] %% t(F) + Q
    V_tt1[, ,t] <- (V_tt1[, ,t]+t(V_tt1[, ,t]))/2

    Z_t <- t(as.matrix(Z[t,]))
    D_t <- n[t]*exp(Z_t %% a_tt1[,t])/(1+exp(Z_t %% a_tt1[,t]))^2
    sigma_t <- D_t
    #Kalman gain:
    K_t <- V_tt1[, ,t] %% t(Z_t) %% D_t %% (D_t %% Z_t %% V_tt1[, ,t] %%
      t(Z_t) %% D_t + sigma_t)^-1
    a_tt[,t+1] <- a_tt1[,t]
    if(!is.na(y[,t])) a_tt[,t+1] <- a_tt1[,t] + K_t %% (y[,t]-n[t]*exp(Z_t
      %% a_tt1[,t])/(1+exp(Z_t %% a_tt1[,t])))
    V_tt[, ,t+1] <- V_tt1[, ,t]
    if(!is.na(y[,t])) V_tt[, ,t+1] <- V_tt1[, ,t] - K_t %% D_t %% Z_t %%
      V_tt1[, ,t]
    V_tt[, ,t+1] <- (V_tt[, ,t+1]+t(V_tt[, ,t+1]))/2 }

```

```

#smoothing:
a_tT[,Ti+1] <- a_tt[,Ti+1]
V_tT[, ,Ti+1] <- V_tt[, ,Ti+1]
for (t in Ti:1) {

B_t[, ,t] <- V_tt[, ,t] %*% t(F) %*% solve(V_tt1[, ,t])
a_tT[,t] <- a_tt[,t] + B_t[, ,t] %*% (a_tT[,t+1]-a_tt1[,t])
V_tT[, ,t] <- V_tt[, ,t] + B_t[, ,t] %*% (V_tT[, ,t+1]-V_tt1[, ,t]) %*%
      t(B_t[, ,t])
V_tT[, ,t] <- (V_tT[, ,t]+t(V_tT[, ,t]))/2 }

convcrit2<-10*epsTHETA
while (convcrit2 >epsTHETA) {

a_0prev<-a_0
Q_0prev<-Q_0
Qprev<-Q

a_0 <- a_tT[,1]
Q_0 <- V_tT[, ,1]
if(Qdiag==TRUE) Q_0 <-diag(diag(Q_0))
Q <- matrix(0,dimF,dimF)
for (t in 1:Ti) {

S <- (a_tT[,t+1] - F %*% a_tT[,t]) %*% t(a_tT[,t+1] - F %*% a_tT[,t])
      + V_tT[, ,t+1] - F %*% B_t[, ,t] %*% V_tT[, ,t+1] - t(F %*% B_t[, ,t]
      %*% V_tT[, ,t+1]) + F %*% V_tT[, ,t] %*% t(F)
Q <- Q+S
}

Q <- Q/Ti
Q <- (Q+t(Q))/2
if(Qdiag==TRUE) Q <- diag(diag(Q))
a_tt[,1] <- a_0
V_tt[, ,1] <- Q_0
it<-0
convcrit<-10*epsALPHA

```

```

while (convcrit >epsALPHA) {

alpha0<-a_tT
for(t in 1:Ti) {

a_tt1[,t] <- F %*% a_tt[,t]
V_tt1[, ,t] <- F %*% V_tt[, ,t] %*% t(F) + Q
V_tt1[, ,t] <- (V_tt1[, ,t]+t(V_tt1[, ,t]))/2

Z_t <- t(as.matrix(Z[t,]))
D_t <- n[t]*exp(Z_t %*% alpha0[,t+1])/(1+exp(Z_t %*% alpha0[,t+1]))^2
#sigma_t <- D_t
if(!is.na(y[,t])) y_work <- ((D_t)^-1) %*% (y[,t]-n[t]*exp(Z_t %*%
  alpha0[,t+1])/(1+exp(Z_t %*% alpha0[,t+1]))) + Z_t %*% alpha0[,t+1]
W_t <- D_t
#Kalman gain:
K_t <- V_tt1[, ,t] %*% t(Z_t) %*% (Z_t %*% V_tt1[, ,t] %*% t(Z_t)+ W_t^-1)^-1
a_tt[,t+1] <- a_tt1[,t]
if(!is.na(y[,t])) a_tt[,t+1] <- a_tt1[,t] + K_t %*% (y_work - Z_t %*%
  a_tt1[,t])

V_tt[, ,t+1] <- V_tt1[, ,t]
if(!is.na(y[,t])) V_tt[, ,t+1] <- (diag(1,dimF) - K_t %*% Z_t) %*%
  V_tt1[, ,t]

V_tt[, ,t+1] <- (V_tt[, ,t+1]+t(V_tt[, ,t+1]))/2
}

#smoothing:
a_tT[,Ti+1] <- a_tt[,Ti+1]
V_tT[, ,Ti+1] <- V_tt[, ,Ti+1]
for (t in Ti:1) {

B_t[, ,t] <- V_tt[, ,t] %*% t(F) %*% solve(V_tt1[, ,t])
a_tT[,t] <- a_tt[,t] + B_t[, ,t] %*% (a_tT[,t+1]-a_tt1[,t])
V_tT[, ,t] <- V_tt[, ,t] + B_t[, ,t] %*% (V_tT[, ,t+1]-V_tt1[, ,t]) %*%
  t(B_t[, ,t])
V_tT[, ,t] <- (V_tT[, ,t]+t(V_tT[, ,t]))/2
}

```

```

convcrit <- mean(abs(alpha0-a_tT))/(1+mean(abs(alpha0-a_tT)))
it<-it+1
}

convcrit2 <- (mean(abs(a_0prev-a_0))/(1+mean(abs(a_0prev-a_0))) +
             mean(abs(Q_0prev-Q_0))/(1+mean(abs(Q_0prev-Q_0))) +
             mean(abs(Qprev-Q))/(1+mean(abs(Qprev-Q))))/3
itv[i]<-it i<-i+1
}

loglik<-0
for (t in 1:Ti) if(!is.na(y[,t])) loglik <-loglik + t(as.matrix(Z[t,]))
             %*% a_tT[,t+1] %*% y[1,t] - n[t]*log(1+exp(t(as.matrix(Z[t,]))
             %*% a_tT[,t+1]))

l_0 <- t(a_tT[,1] - a_0) %*% solve(Q_0) %*% (a_tT[,1] - a_0)/2 +
      log(det(matrix(Q_0,dimF,dimF)))/2

pen<-0
for (t in 1:Ti) pen <- pen + (t(a_tT[,t+1] - F %*% a_tT[,t]) %*%
                             solve(Q) %*% (a_tT[,t+1] - F %*% a_tT[,t]))/2 + log(det(Q))*Ti/2

penloglik <- loglik - l_0 - pen

list(penloglik=penloglik, alpha=a_tT, V_tT=V_tT, alphas=a_tt, V_tt=V_tt,
     alphas1=a_tt1, V_tt1=V_tt1, a_0=a_0, Q_0=Q_0, Q=Q, itvm=mean(itv),
     itn=length(itv))
}

```

Bibliografia

Anderson, B. D. O. e Moore, J. B. (1979). *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ.

Azzalini, A. (2001). *Inferenza statistica. Una presentazione basata sul concetto di verosimiglianza*, 2^a edizione. Springer-Verlag Italia, Milano.

Biller, C. (1997). Posterior mode estimation in dynamic generalized linear mixed models (REVISED, June 2000). *Discussion Paper 70*, Sonderforschungsbereich 386, Ludwig-Maximilians-Universität, München.

Chiogna, M. e Gaetan, C. (2002). Dynamic generalized linear models with application to environmental epidemiology. *Applied Statistics* **51**, 453–468.

Datasets at the Department of Statistics, University of Munich, and the SFB386. http://www.stat.uni-muenchen.de/service/datenarchiv/welcome_e.html.

Dempster, A. P., Laird, N. M. e Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc., B* **39**, 1–38.

Durbin, J. e Koopman, S. J. (2001). *Time series analysis by state space methods*. Oxford University Press, Oxford.

Fahrmeir, L. (1992). Posterior mode estimation by extended Kalman filte-

- ring for multivariate dynamic generalized linear models. *J. Am. Statist. Assoc.* **87**, 501–509.
- Fahrmeir, L. e Kaufmann, H. (1991). On Kalman filtering, posterior mode estimation and Fisher scoring in dynamic exponential family regression. *Metrika* **38**, 37–60.
- Fahrmeir, L. e Tutz, G. (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models*, 2nd edition. Springer-Verlag, New York.
- Fahrmeir, L. e Wagenpfeil, S. (1997). Penalized likelihood estimation and iterative Kalman smoothing for non-Gaussian dynamic regression models. *Comput. Statist. Data Anal.* **24**, 295–320.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *J. Basic Engineering, Transactions ASME, Series D* **82**, 35–45.
- Kalman, R. E. (1963). New methods in Wiener filtering theory. In *Proceedings of the first symposium on engineering applications of random function theory and probability*. J. L. Bogdanoff and F. Kozin, eds. Wiley, New York.
- Kitagawa, G. (1987). Non-Gaussian state-space modeling of nonstationary time series. *J. Am. Statist. Assoc.* **82**, 1032–1063.
- Sage, A. P. e Melsa, J. L. (1971). *Estimation theory with applications to communications and control*. McGraw-Hill, New York.
- Schneider, W. (1986). *Der Kalmanfilter als Instrument zur Diagnose und Schätzung variabler Parameter in ökonomischen Modellen*. Physica, Heidelberg.
- Wagenpfeil, S. (1995). Hyperparameter estimation in exponential family state space models. *Discussion Paper 6*, Sonderforschungsbereich 386, Ludwig-Maximilians-Universität, München.

-
- West, M., Harrison, P. J. e Migon, H. S. (1985). Dynamic generalized linear models and Bayesian forecasting. *J. Am. Statist. Assoc.* **80**, 73–83.

Ringraziamenti

Sulla via del ritorno da un lungo viaggio, quando i luoghi, i paesaggi, la lingua, gli odori si fanno man mano sempre più familiari e noti, di solito quello è il momento in cui improvvisamente mi rendo conto che il viaggio sta effettivamente per terminare, ed allora i ricordi riaffiorano e mi vengono in mente tutte le persone che ho incontrato lungo il percorso, le sensazioni provate in certe situazioni ed i momenti che hanno reso quell'itinerario unico ed irripetibile.

Allo stesso modo, solamente il fatto di essere oltre la bibliografia mi fa capire che anche questo viaggio fatto di levatacce il lunedì mattina, corse verso la stazione, pranzi in mensa, spritz ma, soprattutto, di appunti, di lezioni e di esami sta per giungere alla sua conclusione naturale e sento quindi il bisogno di utilizzare queste ultime righe per spendere alcune (poche, come mio solito) parole per ricordare e ringraziare chi mi è stato vicino.

Innanzitutto, per il supporto morale (e finanziario), per avermi sempre incoraggiato, per avermi fatto diventare la persona che sono, i miei genitori Dario e Gertraud e mio fratello Davide.

Per la parte tecnica, il Prof. Adelchi Azzalini, per la disponibilità dimostratami ed i preziosi consigli statistici, informatici, di \LaTeX , che mi hanno permesso di concludere questa tesi di laurea.

Poi, gli amici di sempre, quelli che conosco da una vita e che ultimamente per lo studio ho dovuto un po' trascurare: Mac, Mice, Gian, Bagaj, Zotti, Kezzy, Radu, Lorenz, Andrea, Tati, Calia, Ano, Fede, Nico, Mario, Ale, Valdo, Bolo, Nani, Freddy, Teo.

Coloro con cui ho condiviso quest'avventura, con cui ho alternato studio e divertimento: in particolare, Andrea P., Gianluca, Angela, Paola, Erika, Daniela, Emma, Alessandra, Mauro, Alberto, Andrea B., Gessica, Veronica, Marghe, Fedè.

Tutti quelli che hanno suonato o cantato per me: fra i tanti, Robert Johnson, Miles Davis, Johnny Cash, Janis Joplin, i Clash.

Infine, un pensiero particolare va a quella che è stata "la mia casa" in questi ultimi quattro anni: Matteo, Cristian, Beppe, Izio, per l'amicizia, la disponibilità sempre pronta, per avermi introdotto all'arte culinaria e con cui, dopo lo studio, ho condiviso piacevoli serate a chiacchierare, "a stratraccannare" e "a stramaledire le donne, il tempo ed il governo" (F. De André, *La città vecchia*, 1968).

Grazie a tutti.

*"...io sono un principe libero
ed ho altrettanta autorità
di fare guerra al mondo intero
quanto colui che ha cento navi in mare."*

SAMUEL BELLAMY

(Pirata alle Antille nel XVIII secolo)