

Università degli Studi di Padova  
Dipartimento di Scienze Statistiche  
Corso di Laurea Triennale in  
Statistica, Economia e Finanza



RELAZIONE FINALE  
**MIXED DATA SAMPLING REGRESSION: ASPETTI  
METODOLOGICI E APPLICATIVI**

Relatore Prof. Tommaso Di Fonzo

Laureando: Chiara Magrin  
Matricola N 1075729

Anno Accademico 2015/2016



# Indice

<b>Introduzione .....</b>	<b>III</b>
<b>Capitolo 1 Aspetti metodologici: i modelli MIDAS .....</b>	<b>1</b>
1.1 Introduzione .....	1
1.2 La specificazione del modello .....	2
1.3 L'allineamento delle frequenze .....	4
1.4 Modelli MIDAS basati sugli aggregati ( <i>aggregates-based</i> ) e funzione vincolo .....	5
1.5 Stima del modello .....	8
1.6 Scelta della specificazione e test sul modello .....	9
1.7 Previsioni .....	10
1.8 Altre rappresentazioni di modelli MIDAS .....	12
1.9 Modelli MIDAS in R .....	13
<b>Capitolo 2 Aspetti applicativi: il pacchetto <code>midasr</code> .....</b>	<b>15</b>
2.1 Introduzione .....	15
2.2 Allineamento di frequenze tramite manipolazione dei dati .....	16
2.3 Simulazione di un modello MIDAS .....	18
2.4 Esempi di specificazioni di regressioni MIDAS .....	20
2.5 Test di adeguatezza sui vincoli .....	26
2.6 Scelta del modello .....	27
2.7 Previsioni .....	30
2.8 Applicazione e confronto .....	34

<b>Capitolo 3 Real GDP: stima con modelli MIDAS .....</b>	<b>37</b>
3.1 Il problema.....	37
3.2 La stima dei modelli MIDAS in R.....	40
3.3 Confronto tra modelli e test di adeguatezza.....	43
3.4 Previsioni.....	45
3.5 Stima di un modello ARIMA e confronto con i modelli MIDAS.....	46
<b>Conclusioni .....</b>	<b>49</b>
<b>Appendice .....</b>	<b>51</b>
<b>Bibliografia .....</b>	<b>61</b>

# Introduzione

Uno degli aspetti più importanti nell'ambito dell'analisi delle serie temporali è quello predittivo. Da sempre analisti di serie macroeconomiche e finanziarie si sono interessati all'adattamento delle serie temporali tramite modelli, ma anche alla capacità di questi modelli di produrre previsioni di buona qualità.

Solitamente i modelli a disposizione raccolgono e adattano dati tutti campionati alla stessa frequenza, con la conseguenza, però, di escludere quei dati che vengono raccolti a frequenze differenti. Le serie macroeconomiche, ad esempio, in alcuni casi hanno frequenza mensile ed in altri trimestrale, mentre le variabili finanziarie possiedono frequenza giornaliera o addirittura una frequenza maggiore.

Si pone, dunque, il problema di come utilizzare al meglio i dati a disposizione, per ottenere previsioni sempre più precise ed attendibili.

In questo lavoro viene presentata una delle possibili soluzioni a questo problema: i modelli MIDAS, che hanno la particolare caratteristica di utilizzare dati ad alta frequenza come variabili esplicative e una variabile a bassa frequenza come variabile dipendente.

In particolare nel Capitolo 1 verrà studiata la formalizzazione matematica del modello, a partire dalla specificazione dello stesso fino alla questione delle previsioni e dei test di adeguatezza.

Nel secondo capitolo, invece, si analizzeranno le funzioni del pacchetto `midasr`, appositamente creato per l'implementazione di questi modelli in ambiente R.

Nell'ultimo capitolo si proporrà un esempio pratico di stima di modelli MIDAS per la serie del *Real Gross Domestic Product Growth rate* degli Stati Uniti. Lo scopo non è solo quello di proporre una prova pratica di quanto spiegato nei capitoli precedenti, ma anche quello di mostrare le potenzialità predittive di un modello MIDAS rispetto ad altri modelli per serie storiche. Questo avverrà tramite il confronto tra i modelli MIDAS e un classico modello ARIMA.

# Capitolo 1

## Aspetti metodologici: i modelli MIDAS

### 1.1 Introduzione

I modelli MIDAS sono uno dei più recenti frutti della ricerca di Eric Ghysels. Questi rappresentano un'importante innovazione per quanto riguarda l'utilizzo di dati a frequenza mista in un modello. Uno dei primi articoli scritti a questo riguardo fu di *Ghysels et al.* nel 2002. Successivamente Ghysels continuò ad arricchire la letteratura di nuovi articoli su questi modelli.

In questo capitolo ci si sofferma sugli aspetti che si ritengono fondamentali nella specificazione di un modello MIDAS, a partire dal tipo di dati utilizzati per la stima, le possibili restrizioni sui parametri e i metodi di stima che si possono adottare.

Altri aspetti su cui ci si vuole focalizzare sono gli strumenti utilizzabili per la scelta del modello migliore e la possibilità di produrre previsioni a partire dal modello stimato.

## 1.2 La specificazione del modello

Per la specificazione di un modello MIDAS si suppone di possedere una serie storica  $\{y_t, t \in \mathbb{Z}\}$  di un processo univariato osservato a bassa frequenza. Si definiscono i ritardi della serie con l'operatore ritardo  $B$ , tale che  $By_t = y_{t-1}$ . Si possiede anche una variabile esplicativa osservata ad alta frequenza:  $\{x_\tau^{(i)}, \tau \in \mathbb{Z}\}, i = 1, \dots, k$ , in modo che, per ogni periodo  $t$  a bassa frequenza si osservi il processo  $x_\tau^{(i)}$  per  $m_i \in \mathbb{N}$  volte. La frequenza  $m_i$  del processo  $x_\tau^{(i)}$  deve essere  $m_i \geq 1$ , così che il processo  $y_t$  sia quello a frequenza inferiore (o uguale, nel caso  $m_i = 1$ ). Si definisce, inoltre, l'operatore ritardo  $L$  anche per il processo  $x_\tau^{(i)}$  in modo analogo al precedente operatore:  $Lx_\tau^{(i)} = x_{\tau-1}^{(i)}$ . Per chiarezza espositiva si definisce l' $i$ -esimo periodo  $\tau$  ad alta frequenza in termini del periodo a bassa frequenza:  $\tau = (t-1)m_i + j, j = 1, \dots, m_i$ .

L'equazione del modello è:

$$y_t - \alpha_1 y_{t-1} - \dots - \alpha_p y_{t-p} = \sum_{i=0}^k \sum_{j=0}^{l_i} \beta_j^{(i)} x_{tm_i-j}^{(i)} + \varepsilon_t, \quad (1)$$

per cui è richiesto che:

$$E(\varepsilon_t | y_{t-1}, \dots, y_{t-p}, x_{tm_0}^{(0)}, \dots, x_{tm_0-l_0}^{(0)}, \dots, x_{tm_k}^{(k)}, \dots, x_{tm_k-l_k}^{(k)}) = 0.$$

È possibile riscrivere il modello precedente in forma matriciale, per poterlo scrivere in forma più compatta:

$$\alpha(B)y_t = \beta(L)'x_{t,0} + \varepsilon_t, \quad (2)$$

dove

$$\alpha(z) = 1 - \sum_{j=0}^p \alpha_j z^j ;$$

$$\mathbf{x}_{t,0} := \left( x_{tm_0}^{(0)}, \dots, x_{tm_1}^{(1)}, \dots, x_{tm_l}^{(l)} \right)' ;$$

$$\boldsymbol{\beta}(z) = \sum_{j=0}^l \boldsymbol{\beta}_j z^j, \boldsymbol{\beta}_j = \left( \beta_j^{(0)}, \dots, \beta_j^{(i)}, \dots, \beta_j^{(l)} \right)' ;$$

$$L^j \mathbf{x}_{t,0} := \mathbf{x}_{t,j} = \left( L^j x_{tm_0}^{(0)}, \dots, L^j x_{tm_i}^{(i)}, \dots, L^j x_{tm_l}^{(l)} \right)' .$$

Consideriamo i parametri di un modello MIDAS. Generalmente i parametri totali da stimare sono:  $d = p + \sum_{i=0}^k l_i$ : si tratta di un numero piuttosto considerevole che può provocare difficoltà nella stima del modello, fino ad impedirne la stima del caso in cui il numero di osservazioni sia minore o uguale al numero di parametri. Per questo motivo viene solitamente imposta una funzione vincolo sui parametri originali, che permette di ridurre il numero di parametri del modello. Il parametro vincolato sarà così identificato:

$$\beta_j^{(i)} = f_i(\gamma_i, j), \quad j = 0, \dots, l_i, \quad \gamma_i = \left( \gamma_1^{(i)}, \dots, \gamma_{q_i}^{(i)} \right), \quad q_i \in \mathbb{N}.$$

Dunque, grazie al vincolo introdotto, il numero di parametri si riduce notevolmente, diventando:  $q = \sum_{i=0}^{h_i} q_i$ , notevolmente inferiore anche al numero di osservazioni.

L'introduzione di un vincolo funzionale richiede, però, che questo soddisfi la condizione di esistenza e continuità della derivata seconda  $\left( \frac{\partial^2 f_i}{\partial \gamma_i \partial \gamma_i'} \right)$ , dove  $\gamma$  è il vettore di tutti i parametri di un modello ristretto.

### 1.3 L'allineamento delle frequenze

In un modello MIDAS (1) in cui, per definizione, vengono mescolati campioni di dati a frequenze diverse, sorge il problema di trattarli in maniera da ottenere una stima dei parametri del modello.

Questa difficoltà viene risolta grazie alla riscrittura del modello (1) nella forma matriciale (2), che oltre ad essere più compatta permette il cosiddetto *allineamento delle frequenze*, poiché la variabile  $x_t$  viene riscritta come un vettore a bassa frequenza  $(x_{tm_i}^{(i)}, \dots, x_{tm_{i-1}}^{(i)}, \dots, x_{tm_{i-l}}^{(i)})'$ . Quindi si esprime di seguito il modello (1) in forma matriciale, esplicitando l'allineamento delle frequenze:

$$\begin{bmatrix} y_l \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} y_{l-1} & \cdots & y_{l-p} \\ \vdots & \ddots & \vdots \\ y_{n-1} & \cdots & y_{n-p} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_p \end{bmatrix} + \sum_{i=0}^k X^{(i)} \begin{bmatrix} \beta_0^{(i)} \\ \vdots \\ \beta_l^{(i)} \end{bmatrix} + \begin{bmatrix} \varepsilon_l \\ \vdots \\ \varepsilon_n \end{bmatrix},$$

dove

$$X^{(i)} := \begin{bmatrix} x_{um_i}^{(i)} & x_{um_i-1}^{(i)} & \cdots & x_{um_i-l}^{(i)} \\ x_{(u+1)m_i}^{(i)} & x_{(u+1)m_i-1}^{(i)} & \cdots & x_{(u+1)m_i-l}^{(i)} \\ \vdots & \vdots & \cdots & \vdots \\ x_{tm_i}^{(i)} & x_{tm_i-1}^{(i)} & \cdots & x_{tm_i-l}^{(i)} \\ \vdots & \vdots & \cdots & \vdots \\ x_{(n-1)m_i}^{(i)} & x_{(n-1)m_i-1}^{(i)} & \cdots & x_{(n-1)m_i-l}^{(i)} \\ x_{nm_i}^{(i)} & x_{nm_i-1}^{(i)} & \cdots & x_{nm_i-l}^{(i)} \end{bmatrix}, \quad (3)$$

$u$  è il più piccolo intero tale che  $um_i - l > 0$  e tale che  $u > p$ .

In seguito alla riscrittura del modello (1) nella notazione matriciale, che mostra come dati a differenti frequenze possono essere confrontati tramite vettori a bassa frequenza, è possibile stimare il modello utilizzando una classica regressione per serie temporali.

Di seguito si vuole proporre un semplice esempio per dare un'intuizione sul funzionamento dell'allineamento di frequenze.

Sia  $y_t$  una serie d'interesse osservata a bassa frequenza, per esempio trimestralmente, con  $t = 1, \dots, 12$ . Inoltre sia  $x_\tau$  una serie ad alta frequenza, osservata mensilmente, con  $\tau = 1, \dots, 3 \times 12$ , con  $m = 3$ . Volendo scrivere il modello in forma matriciale, come nell'equazione (3), si ottiene:

$$\begin{bmatrix} y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \end{bmatrix} \alpha_1 + \begin{bmatrix} x_6 & x_5 & x_4 & x_3 & x_2 & x_1 \\ x_9 & x_8 & x_7 & x_6 & x_5 & x_4 \\ x_{12} & x_{11} & x_{10} & x_9 & x_8 & x_7 \\ x_{15} & x_{14} & x_{13} & x_{12} & x_{11} & x_{10} \\ x_{18} & x_{17} & x_{16} & x_{15} & x_{14} & x_{13} \\ x_{21} & x_{20} & x_{19} & x_{18} & x_{17} & x_{16} \\ x_{24} & x_{23} & x_{22} & x_{21} & x_{20} & x_{19} \\ x_{27} & x_{26} & x_{25} & x_{24} & x_{23} & x_{22} \\ x_{30} & x_{29} & x_{28} & x_{27} & x_{26} & x_{25} \\ x_{33} & x_{32} & x_{31} & x_{30} & x_{29} & x_{28} \\ x_{36} & x_{35} & x_{34} & x_{33} & x_{32} & x_{31} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{bmatrix} + \begin{bmatrix} \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_7 \\ \varepsilon_8 \\ \varepsilon_9 \\ \varepsilon_{10} \\ \varepsilon_{11} \\ \varepsilon_{12} \end{bmatrix}$$

Si nota che la matrice  $X$  è costituita da una combinazione lineare delle osservazioni di  $x_\tau$  necessarie a modellare  $y_t$  e  $y_{t-1}$ , ovvero da:  $x_{mt}, x_{mt-1}, x_{mt-2}, x_{m(t-1)}, x_{m(t-1)-1}, x_{m(t-1)-2}$ .

## 1.4 Modelli MIDAS basati sugli aggregati (*aggregates-based*) e funzione vincolo

Talvolta, in un modello MIDAS, può essere utile per questioni di parsimonia scrivere la parte ad alta frequenza in termini di aggregati:

$$\beta(L)'x_{t,0} = \sum_{i=0}^k \sum_{j=0}^l \beta_j^{(i)} x_{tm_i-j}^{(i)} = \sum_{i=0}^k \sum_{r=0}^q \lambda_r^{(i)} \tilde{x}_{t-r}^{(i)}, \quad (4)$$

dove  $q \in \mathbb{N}$  sono i lag a bassa frequenza.

Questa particolare rappresentazione non è altro che un caso particolare di rappresentazione del modello specificato dall'equazione (2) con una specifica funzione vincolo sui parametri. Non è altrettanto vero, però, che ogni vincolo su  $\beta(L)$  può essere rappresentato usando aggregati periodici.

Si consideri la riscrittura della componente ad alta frequenza. Per approfondire l'argomento, sia:

$$\tilde{x}_{t-r}^{(i)} := x_{t-r}^{(i)}(\delta_{i,r}) = \sum_{s=1}^{m_i} w_r^{(i)}(\delta_{i,r}; s) x_{(t-1-r)m_i+s}^{(i)},$$

funzione di  $w_r^{(i)}(\delta_{i,r}; s)$ , ovvero una funzione di ponderazione con vettore di parametri  $\delta_{i,r}$ , che varia a seconda della variabile e della sua frequenza e/o dipende dall'ordine dei ritardi  $r \in \mathbb{N}$  della variabile a bassa frequenza.

Perché i pesi soddisfino alcune condizioni richieste, come la non negatività di questi ultimi e la soddisfazione del vincolo di normalizzazione ( $\sum_{s=0}^{m_i-1} w_r^{(i)}(\delta_{i,r}; s) = 1$ ), necessario all'identificazione dei parametri, si definisce la funzione di ponderazione come:

$$\forall i, r \quad w_r^{(i)}(\delta_{i,r}; s) = \frac{\psi_r^{(i)}(\delta_{i,r}; s)}{\sum_{j=1}^{m_i} \psi_r^{(i)}(\delta_{i,r}; j)}, \quad s = 1, \dots, m_i, \quad (5)$$

data la funzione sottostante  $\psi_r^{(i)}(\cdot)$  non negativa.

Esistono molte possibili funzioni sottostanti che mantengono la caratteristica di non negatività dei pesi e permettono l'identificazione dei parametri.

Si portano alcuni esempi di funzioni di ponderazione:

- *Exponential Almon lag polynomial*:  $\psi(\delta; s) = \exp(\sum_{j=1}^p \delta_j s^j)$ ,  $p \in \mathbb{N}$ , dove  $\delta = (\delta_1, \dots, \delta_j, \dots, \delta_p)' \in \mathbb{R}^p$ ;

- *Beta*, analoga alla funzione di densità:  $\psi(\boldsymbol{\delta}; s) = x_s^{\delta_1-1} (1 - x_s)^{\delta_2-1}$ , dove  $x_s := \xi + (1 - \xi)h(s)$ ,  $h(s) := \frac{s-1}{m-1}$ , con  $\xi > 0$  una quantità molto piccola e  $\boldsymbol{\delta} = (\delta_1, \delta_2)' \in \mathbb{R}_+^2$ ;
- *Gompertz*, analoga alla funzione di densità di probabilità:  $\psi(\boldsymbol{\delta}; s) = z(s)e^{-\delta_1 z(s)}$ , dove  $z(s) = \exp(\delta_2 s)$ , e  $\boldsymbol{\delta} = (\delta_1, \delta_2)' \in \mathbb{R}_+^2$ ;
- *Log-Cauchy*, analoga alla funzione di densità di probabilità:  $\psi(\boldsymbol{\delta}; s) = s^{-1}(\delta_2^2 + (\ln s - \delta_1)^2)^{-1}$ , dove  $\boldsymbol{\delta} = (\delta_1, \delta_2)' \in \mathbb{R} \times \mathbb{R}_+$ ;
- *Nakagami*:  $\psi(\boldsymbol{\delta}; s) = s^{2\delta_1-1} \exp\left(-\frac{\delta_1}{\delta_2 s^2}\right)$ , dove  $\boldsymbol{\delta} = (\delta_1, \delta_2)'$ ,  $\delta_1 \geq 0,5$ ,  $\delta_2 \in \mathbb{R}_+$ .

È importante sottolineare che la scelta di una funzione di ponderazione è uno dei possibili vincoli che si possono applicare a  $\beta(L)$ , altri esempi sono i sotto riportati ed impongono tutti l'uguaglianza di:

- lo schema di ponderazione/funzione applicata tra le variabili e/o le frequenze ( $\forall i, w_r^{(i)}(\cdot) = w_r(\cdot)$ );
- lo schema di ponderazione/funzione applicata tra tutti i ritardi a bassa frequenza  $r = 0, 1, \dots, q$  degli aggregati ( $\forall r, w_r(\cdot) = w(\cdot)$ );
- i parametri della funzione di ponderazione in ogni lag ( $\forall r, \delta_{i,r} = \delta_i$ );
- l'impatto contemporaneo e ritardato di tutti i lag ( $\forall r, \lambda_r^{(i)} = \lambda^{(i)}$ ).

A questo punto si vuole approfondire la motivazione dell'affermazione fatta in precedenza, per cui un modello MIDAS aggregates-based è un caso particolare del modello specificato dall'equazione (2) con una specifica funzione vincolo.

Sia  $s_i$  l'enumeratore dell' $i$ -esimo periodo ad alta frequenza, entro un periodo a bassa frequenza. Data, poi, la frequenza  $m_i$  (del periodo ad alta

frequenza) si nota che esiste una relazione uno a uno tra l'indice ad alta frequenza  $j \in \mathbb{N}$  e la coppia  $(r, s_i) \in \mathbb{N} \times \{1, 2, \dots, m_i\}$ :

$$j = r m_i + s_i,$$

da cui segue:

$$f_i(\gamma_i; r m_i + s_i) = \lambda_r^{(i)} w_r^{(i)}(\delta_{i,r}; s). \quad (6)$$

Dunque, come affermato in precedenza, siamo in presenza di un caso particolare della rappresentazione dell'equazione (2), con una specifica funzione vincolo.

## 1.5 Stima del modello

La stima di un modello MIDAS si differenzia in due casi: la stima di un modello senza restrizioni sui parametri e la stima di un modello con restrizioni.

Nel caso della stima di un modello senza restrizioni, anche chiamato modello U-MIDAS, nell'articolo di *Foroni et al.*, si può procedere utilizzando le stime dei minimi quadrati (OLS).

È diverso il caso di un modello in cui siano presenti delle restrizioni non lineari sui parametri. In questo caso, infatti, non è più possibile utilizzare lo stimatore OLS, poiché non è più soddisfatta l'ipotesi di linearità dei parametri.

Dunque lo stimatore più adatto a stimare i parametri di un modello MIDAS generale risulta essere lo stimatore dei minimi quadrati non lineari (NLS).

## 1.6 Scelta della specificazione e test sul modello

Per poter operare una scelta per la specificazione di un modello MIDAS sono importanti due fattori: la scelta del vincolo funzionale (funzione vincolo) e la scelta dell'ordine massimo dei ritardi.

Il modo più immediato per prendere queste decisioni, dunque selezionare il modello migliore in base ai due fattori elencati in precedenza, è utilizzare i criteri d'informazione, secondo quanto affermato nell'articolo di Ghysels *et al.* (2016). Come nei classici modelli per serie storiche e per i modelli OLS questi criteri sono l'AIC e il BIC.

Esistono, però, anche altri modi per testare l'adeguatezza dei vincoli funzionali imposti. Un elemento di particolare interesse risulta essere la verifica della presenza o meno di un impatto stagionale nel modello, questo perché non si ha una grande conoscenza a priori della componente stagionale.

Secondo Ghysels *et al.* (2006), In un modello che presenta un termine autoregressivo vale che:  $\phi(L) = \beta(L)/\alpha(B)$  avrà andamento stagionale a cui corrisponderà un impatto stagionale delle variabili esplicative sulla variabile dipendente. Per evitare questo effetto, nel caso non fosse rilevante, Clements *et al.* proposero di introdurre una restrizione a fattore comune che può essere formulata come una restrizione polinomiale con un vincolo sul polinomio  $\beta(L)$  per soddisfare la fattorizzazione  $\beta(L) = \alpha(B)\phi(L)$ . Così, invertendo l'equazione (2) in termini del polinomio  $\alpha(B)$  lascia  $\phi(L)$  inalterato, senza creare o distruggere qualsiasi andamento stagionale.

Si vuole introdurre ora il test sopradescritto, utile a confrontare un modello con restrizioni e un modello sprovvisto di vincoli.

Sia  $\beta(z)$  il vettore di parametri di un modello MIDAS privo di restrizioni. Sia  $f_\gamma(z)$  il corrispondente vettore di coefficienti ristretti da una funzione vincolo. Quindi, siano  $\hat{\beta}$  e  $\hat{f}_\gamma$  le corrispondenti stime dei vettori di coefficienti del modello non vincolato (ottenute con lo stimatore OLS) e del modello con funzione vincolo (ottenute a partire dalle stime NLS di  $\gamma$ ). Inoltre siano  $\hat{\alpha}$  e  $\hat{\alpha}_\gamma$  i vettori di coefficienti corrispondenti al polinomio  $\alpha(z)$ , il primo stimato con i minimi quadrati dal modello non ristretto e il secondo stimato con lo stimatore NLS dal modello con la restrizione. Si definiscono:  $\theta := (\alpha', \beta')$ ,  $\hat{\theta} := (\hat{\alpha}', \hat{\beta}')$  e  $\tilde{\theta} := (\hat{\alpha}'_\gamma, \hat{f}_\gamma(z)')$  i vettori composti dagli elementi sopra definiti.

Il test, proposto da Kvedaras e Zemlys, vuole verificare l'ipotesi nulla  $H_0: f_\gamma = \beta$ , per  $\exists \gamma \in \mathbb{R}^q$ , con statistica test:

$$(\hat{\theta} - \tilde{\theta})' \mathbf{A} (\hat{\theta} - \tilde{\theta}) \sim \chi^2(d - q),$$

dove  $\mathbf{A}$  è una matrice appropriata di normalizzazione,  $q = \dim(\gamma)$  e  $d = \dim(\theta)$ .

## 1.7 Previsioni

Uno degli aspetti che forse si ritengono più importanti in un modello MIDAS, come per tutti i modelli di serie storiche, è quello delle previsioni. Però, nei modelli MIDAS esiste una componente in più da tenere in considerazione: la variabile esplicativa  $x$ .

Intanto, prima di fare ulteriori considerazioni sulle esplicative, si vuole scrivere la previsione di  $y$  ad un passo avanti:

$$y_{t+1} = \alpha' y_{t,0} + \beta(L)' x_{t+1,0} + \varepsilon_{t+1},$$

dove  $\mathbf{y}_{t,0} = (y_t, \dots, y_{t-p+1})'$  e  $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_p)'$  è il vettore di parametri del termine autoregressivo.

Come anticipato in precedenza, la variabile esplicativa può influire sulla previsione (in questo caso  $y_{t+1}$ ), o meglio, se si posseggono le informazioni relative a  $\mathbf{x}_{t+1,0}$ , non ci saranno problemi con la previsioni, mentre, nel caso in cui non si posseggono queste informazioni sarà necessaria la previsione di  $\mathbf{x}_{t+1,0}$  a partire dal processo  $(y_t, \mathbf{x}_{t,0})$ . Quest'ultimo, però, è difficile da specificare e da stimare.

Per produrre una previsione di  $y_t$  a  $h$  passi in avanti si definisce  $I_{t,0} = \{\mathbf{y}_{t,j}, \mathbf{x}_{t,j}\}_{j=0}^{\infty}$ , ovvero l'insieme delle informazioni disponibili al momento  $t$ , dove  $\mathbf{y}_{t,j} = (y_{t-j}, \dots, y_{t-j-p+1})'$  e  $\mathbf{x}_{t,j} = (x_{tm_0}^{(0)}, \dots, x_{tm_i}^{(i)}, \dots, x_{tm_k}^{(k)})'$ .

La previsione diretta sarà, dunque, il valore atteso condizionato al set informativo disponibile al tempo  $t$ :

$$\tilde{y}_{t+h} = E(y_{t+h} | I_{t,0}) = \boldsymbol{\alpha}_h' \mathbf{y}_{t,0} + \boldsymbol{\beta}_h(L)' \mathbf{x}_{t,0}, \quad h \in \mathbb{N},$$

che si può anche scrivere come:

$$y_{t+h} = \boldsymbol{\alpha}_h' \mathbf{y}_{t,0} + \boldsymbol{\beta}_h(L)' \mathbf{x}_{t,0} + \varepsilon_{t,h}, \quad E(\varepsilon_{t,h} | I_{t,0}), \quad (7)$$

dove  $\boldsymbol{\alpha}_h$  e  $\boldsymbol{\beta}_h(L)$  sono i vettori di parametri all'orizzonte  $h$ .

È importante sottolineare che l'equazione (4) è una scrittura generale dell'equazione (2), ma, a differenza dell'equazione (2), in cui i vincoli imposti sui parametri erano certi/fissi, nell'equazione (4), a seconda dei passi in avanti della previsione sarà necessario adattare i vincoli.

## 1.8 Altre rappresentazioni di modelli MIDAS

Il modello MIDAS finora presentato è una delle rappresentazioni possibili e risulta essere la più generale. Esistono, tuttavia, come affermato nell'articolo di Andreou *et al.*, molte altre possibili specificazioni del modello MIDAS che sono casi particolari del modello (1) e che, al contrario di quest'ultimo, si possono specificare con un'unica variabile ad alta frequenza ( $m$ ).

Di seguito si propongono alcune particolari specificazioni:

1. D(istributed) L(ag)-MIDAS ( $p_X$ ):

$$y_{t+1} = \mu + \sum_{r=0}^{p_X} \sum_{j=0}^{m-1} \beta_{rm+j} x_{(t-r)m-j} + \varepsilon_{t+1}$$

2. A(utoregressive) D(istributed) L(ag)-MIDAS ( $p_X, p_Y$ ):

$$y_{t+1} = \mu + \sum_{j=0}^{p_Y} \mu_j y_{t-j} + \sum_{r=0}^{p_X} \sum_{j=0}^{m-1} \beta_{rm+j} x_{(t-r)m-j} + \varepsilon_{t+1}$$

3. F(actor) A(utoregressive) D(istributed) L(ag)-MIDAS ( $p_F, p_X, p_Y$ ):

$$y_{t+1} = \mu + \sum_{i=0}^{p_F} \alpha_i F_{t-1} \sum_{j=0}^{p_Y} \mu_j y_{t-j} + \sum_{r=0}^{p_X} \sum_{j=0}^{m-1} \beta_{rm+j} x_{(t-r)m-j} + \varepsilon_{t+1}$$

Dove  $F_t$  è un fattore prodotto tramite una regressione svolta utilizzando dati aggiuntivi.

4. A(utoregressive) D(istributed) L(ag)-MIDAS-M (multiplicative)

$(p_X, p_Y)$ :

$$y_{t+1} = \mu + \sum_{j=0}^{p_Y} \mu_j y_{t-j} + \sum_{r=0}^{p_X} \alpha_r X_{t-r} + \varepsilon_{t+1}$$

Dove  $X_{t-r} = \sum_{j=0}^{m_i-1} \beta_j x_{(t-r)m-j}$ .

5. MIDAS con  $J < m$  osservazioni disponibili per la variabile  $x_\tau$  al tempo  $t + 1$ :

$$y_{t+1} = \mu + \sum_{j=0}^{p_Y} \mu_j Y_{t-j} + \sum_{j=1}^J \beta_{-j} x_{tm+j} + \sum_{r=0}^{p_X} \sum_{j=0}^{m-1} \beta_{rm+j} x_{(t-r)m-j} + \varepsilon_{t+1}$$

Come per il modello generale finora visto le restrizioni vengono imposte sui coefficienti  $\beta_j$  in modo che  $\beta_j = f(\gamma, j)$ , ma a differenza di quest'ultimo la variabile dipendente  $y$  compare al ritardo  $t+1$ .

## 1.9 Modelli MIDAS in R

Dopo aver chiarito gli aspetti teorici riguardanti i modelli MIDAS, si vuole approfondire l'aspetto applicativo riguardante questi ultimi.

Tra i software in grado di effettuare l'analisi delle serie temporali, R implementa il modello MIDAS, grazie al pacchetto `midasR`, che offre all'utente notevoli possibilità per sviluppare questo tipo di modelli.



## Capitolo 2

### Aspetti applicativi: il pacchetto `midasr`

#### 2.1 Introduzione

In questo capitolo viene presentata l'implementazione dei modelli MIDAS tramite il software R. Ci si soffermerà sulle principali funzioni necessarie alla stima di questo tipo di modelli tramite l'utilizzo di serie simulate. Si approfondiranno, poi, gli argomenti trattati nel capitolo precedente: dal problema della scelta della specificazione, all'adeguatezza della scelta dei vincoli, alla produzione di previsioni. Tutto ciò verrà fatto partendo dalla riproposizione dei risultati ottenuti ed esposti nell'articolo di Ghysels *et al.*<sup>1</sup>, con l'intenzione di mostrare la correttezza del prodotto ottenuto in questo lavoro, tramite il confronto con i suddetti risultati.

---

<sup>1</sup> Ghysels *et al.* (2016).

## 2.2 Allineamento di frequenze tramite manipolazione dei dati

L'aspetto più banale, ma anche uno dei fondamentali, nella costruzione di un modello riguarda il trattamento dei dati di interesse nel software di riferimento, in questo caso R.

Per la natura particolare dei dati necessari alla stima di un modello MIDAS, che hanno per definizione frequenze differenti, non è possibile prendere tutti i dati da un unico `data.frame`. Dunque le variabili potranno essere sia vettori, sia passate come elementi di una `list`.

È inoltre necessario che il numero di osservazioni a frequenze differenti corrisponda perfettamente, ovvero  $n_i = nm_i$ , che le frequenze di ogni serie siano perfettamente allineate, se questo non risulta possibile, a causa di dati mancanti, sarà necessario inserire degli NA.

Dopo aver importato ed eventualmente sistemato le variabili necessarie alla stima del modello, è necessario occuparsi di *allineare le frequenze* delle variabili a bassa frequenza e quelle ad alta frequenza. In R, in particolare nel pacchetto `midasr`, sono state implementate delle funzioni che permettono di compiere quest'allineamento in modo immediato. Queste funzioni sono:

- `m1s(x, k, m)`, sistema il vettore di dati ad alta frequenza in una matrice  $\frac{\dim x}{m} \times \dim k$  di osservazioni a bassa frequenza, dove  $k$  è un vettore che definisce i ritardi ad alta frequenza;
- `fmls(x, k, m)`, opera allo stesso modo della funzione precedente, a differenza del fatto che  $k$  è uno scalare e sono prodotti  $k + 1$  ritardi a partire da 0 fino a  $k$ ;

- `dmls (x, k, m)`, uguale a `fmls` tranne per il fatto che la matrice risultante contiene  $k + 1$  differenze di primo ordine della variabile  $x$  ad alta frequenza.

La funzione `fmls (x, k, m)` produce l'esatto equivalente di quanto descritto dall'equazione (3): costruisce una matrice di osservazioni di una serie a bassa frequenza (con  $k$  ritardi) utilizzando un vettore contenente la variabile  $x$  ad alta frequenza. Per esempio in R:

```
x <- 1:24
fmls(x, 2, 3)
      X.0/m X.1/m X.2/m
[1,]      3      2      1
[2,]      6      5      4
[3,]      9      8      7
[4,]     12     11     10
[5,]     15     14     13
[6,]     18     17     16
[7,]     21     20     19
[8,]     24     23     22
```

si ottiene la stessa matrice descritta nell'equazione (3).

La funzione `mls` è più flessibile rispetto alla `fmls` poiché i ritardi possono iniziare sia da un ordine deliberatamente scelto, sia da zero (ad esempio per costruire una matrice analoga alla precedente si sarebbe dovuto scrivere: `mls (x, 0:2, 3)`), mentre la funzione `dmls`, rispetto alla `fmls` esegue anche una differenza prima sui dati, quest'ultima risulta conveniente per le serie integrate.

## 2.3 Simulazione di un modello MIDAS

Dopo aver chiarito come ottenere l'allineamento delle frequenze tramite specifici comandi, si vuole procedere con la simulazione di una serie il cui processo generatore di dati è un processo MIDAS. Dunque, date due variabili,  $x$  e  $z$ , ad alta frequenza, i cui parametri soddisfano l'*Exponential Almon lag polynomial*, si vuole generare la variabile risposta a bassa frequenza  $y$ <sup>2</sup>:

$$y_t = 2 + 0.1t + \sum_{j=0}^7 \beta_j^{(1)} x_{4t-j} + \sum_{j=0}^{16} \beta_j^{(2)} z_{12t-j} + \varepsilon_t$$

$$x_{\tau_1} \sim n.i.d.(0,1), \quad z_{\tau_2} \sim n.i.d.(0,1), \quad \varepsilon_t \sim n.i.d.(0,1), \quad (8)$$

dove  $(x_{\tau_1}, z_{\tau_2}, \varepsilon_t)$  sono indipendenti per ogni  $(\tau_1, \tau_2, t) \in \mathbb{Z}^3$ , e

$$\beta_j^{(i)} = \gamma_0^{(i)} \frac{\exp(\sum_{s=1}^{q_i-1} \gamma_s^{(i)} j^s)}{\sum_{j=0}^{d_i-1} \exp(\sum_{s=1}^{q_i-1} \gamma_s^{(i)} j^s)}, \quad i = 1, 2,$$

dove  $d_1 = k_1 + 1 = 8$  è un multiplo della frequenza  $m_i = 4$ , dove  $d_2 = k_2 + 1 = 17$  non è un multiplo di  $m_2 = 12$ . Inoltre  $q_1 = 2$  e  $q_2 = 3$  con le seguenti parametrizzazioni:

$$\gamma_1 = (1, -0,5)',$$

$$\gamma_2 = (2, 0,5, -0,1)'$$

Di seguito si riportano i comandi R necessari a riprodurre quanto descritto sopra, ovvero creare una serie  $y$  a bassa frequenza a partire da un processo generatore di dati caratterizzato come precedentemente esposto:

```
set.seed(1001)
```

---

<sup>2</sup> Ghysels *et al.* (2016).

```

n <- 250
trend <- c(1:n)
x <- rnorm(4*n)
z <- rnorm(12*n)
fnx <- nealmon(p = c(1, -0.5), d = 8)
fnz <- nealmon(p = c(2, 0.5, -0.1), d = 17)
y <- 2 + 0.1 * trend + fmls(x, 7, 4)%*%fnx +
+fmls(z, 16, 12)%*%fnz + rnorm(n)

```

La funzione `nealmon` è il comando R che riproduce la funzione vincolo *Exponential Almon lag polynomial*. Il pacchetto `midasr` è provvisto di molte altre funzioni vincolo che, se necessarie all'analisi di un modello, verranno introdotte in seguito.

Si riporta, inoltre il grafico che descrive i vincoli funzionali sopra imposti:

```

d <- c(1:17)
v <- c(1:8)
plot(v, fnx, type = "l", xlim = c(1,17), ylim = c(0, 0.45),
xlab = "Ritardi ad alta frequenza", ylab = "Pesi" , col =
1)
lines(d, fnz, col= 2)

```

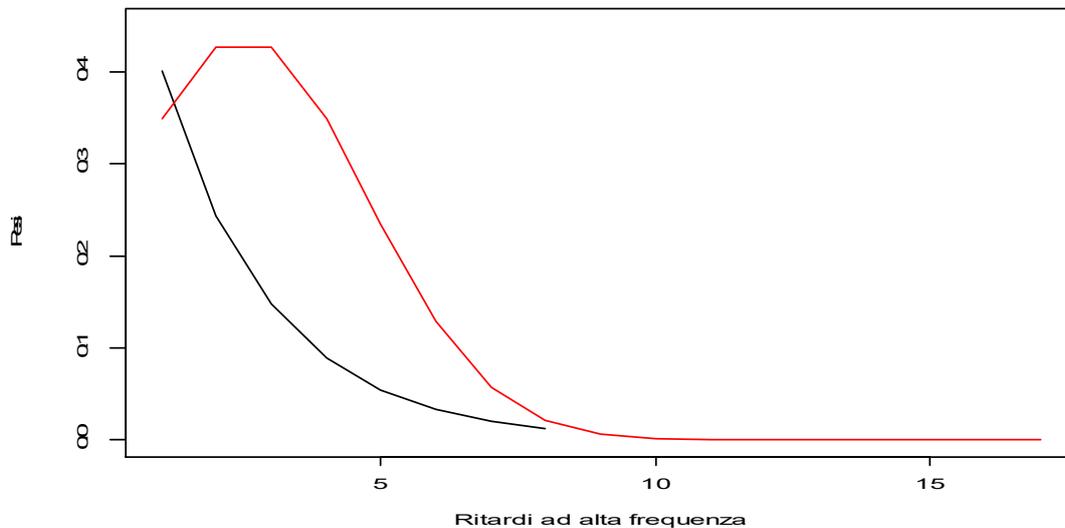


FIGURA 1 FUNZIONI VINCOLO:  $x_\tau$  IN NERO,  $y_\tau$  IN ROSSO.

## 2.4 Esempi di specificazioni di regressioni MIDAS

Dopo aver simulato la variabile  $y$  a partire dal processo generatore di dati sopra descritto si vuole stimare una regressione MIDAS a partire dai dati  $x$ ,  $y$  e  $z$  precedentemente individuati e presenti nell'ambiente R.

Si vogliono stimare due modelli:

- i. un U-MIDAS privo di restrizioni e stimabile con il metodo OLS;
- ii. un MIDAS vincolato con *Exponential Almon lag polynomial* stimabile con il metodo NLS.

Il primo modello può essere stimato in questo modo<sup>3</sup>:

```
mols <- lm(y ~ trend + fmls(x, 7, 4) + fmls(z, 16, 12))
```

o, equivalentemente:

---

<sup>3</sup> Ghysels *et al.* (2016).

```
mols <- midas_r(y ~ trend + fmls(x, 7, 4) +
+ fmls(z, 16, 12), start = NULL)
```

Il summary di queste equazioni stimate mostrerà la stima di 27 parametri ( $d_2 + d_1 + trend + intercetta$ ).

La stima di un modello MIDAS con funzione vincolo *Exponential Almon lag polynomial*, invece, produrrà un numero notevolmente inferiore di stime di parametri<sup>4</sup>:

```
eq_r <- midas_r(y ~ trend + fmls(x, 7, 4, nealmon) +
+ fmls(z, 16, 12, nealmon), start = list(x = c(1, -0.5), z =
+ c(2, 0.5, -0.1)))
```

```
summary(eq_r)
```

```
Formula y ~ trend + mls(x, 0:7, 4, nealmon) + mls(z, 0:16,
12, nealmon)
```

```
Parameters:
```

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	1.9881956	0.1152990	17.244	< 2e-16	***
trend	0.0998828	0.0007769	128.571	< 2e-16	***
x1	1.3533434	0.1512202	8.949	< 2e-16	***
x2	-0.5075657	0.0966696	-5.251	3.32e-07	***
z1	2.2634729	0.1728150	13.098	< 2e-16	***
z2	0.4096532	0.1556852	2.631	0.009052	**
z3	-0.0729794	0.0203915	-3.579	0.000417	***

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.9316 on 242 degrees of freedom
```

---

<sup>4</sup> Ghysels *et al.* (2016).

Il numero di parametri si è ridotto fino a sette grazie all'introduzione della funzione vincolo. Questo rende sicuramente migliore il modello dal punto di vista della parsimonia.

La funzione `midas_r` permette di scegliere tra un'ampia scelta di algoritmi di ottimizzazione numerica. La scelta è controllata tramite la funzione `Ofunction`. È possibile scegliere tra le funzioni `nls`, `optim` o `optimix` (dal pacchetto **optimx**).

È importante notare che vengono forniti degli *starting value* per ogni variabile provvista di vincoli, poiché questi ultimi definiscono implicitamente il numero di parametri della funzione vincolo per ogni serie ad alta frequenza.

È possibile ristimare un modello utilizzando il metodo NLS, che è molto sensibile alla scelta degli *starting value*, aggiornandoli, sfruttando una regressione fatta con: `Ofunction = optim` e `method = "Nelder - Mead"`:

- si stima il modello utilizzando le funzioni sopra citate<sup>5</sup>:

```
eq_r1 <- midas_r(y ~ trend + mls(x, 0:7, 4, nealmon) +  
+ mls(z, 0:16, 12, nealmon), start =  
+ list(x = c(1, -0.5), z = c(2, 0.5, -0.1)),  
+ Ofunction = "optim", method = "Nelder-Mead");
```

- si aggiorna il modello utilizzando `Ofunction = nls`:

```
eq_r2 <- update(eq_r1, Ofunction = "nls").
```

È possibile controllare il risultato ottenuto dalla funzione di ottimizzazione (`optim`) utilizzando il comando `opt`:

```
eq_r2$opt  
$par  
(Intercept)      trend          x1          x2          z1  
1.98564830  0.09990172  1.35251875 -0.50816124  2.26317742
```

---

<sup>5</sup> Ghysels *et al.* (2016).

```

z2          z3
0.40886037 -0.07283773
$value
[1] 210.0091
$count
function gradient
      502      NA
$convergence
[1] 1
$message
NULL

```

L'output ottenuto permette di conoscere lo status di convergenza dell'algoritmo di ottimizzazione: lo status 0 indica una convergenza soddisfacente, mentre lo status 1 indica una convergenza insoddisfacente e che il limite di iterazione della funzione di ottimizzazione è stato raggiunto.

Fino ad ora gli esempi svolti si sono focalizzati su tipologie di regressioni in cui il vincolo funzionale veniva imposto direttamente sul termine  $\beta(L)$  corrispondente ad ogni serie. Però, alla luce di quanto esposto precedentemente nel paragrafo 1.4, in particolare dalla relazione (5), è noto che è sempre possibile ricondursi alla scrittura con un vincolo generale da una scrittura *aggregates-based*.

Per ottenere tale risultato in ambiente R viene utilizzata la funzione `amweights`, che è in grado di produrre parecchi vincoli funzionali periodici espressi dall'equazione (5). Di seguito si forniscono alcuni esempi<sup>6</sup>:

```

amweights(p = c(1, -0.5), d = 8, m = 4, weight =
+ nealmon, type = "C")
[1] 0.4550542 0.2760043 0.1674051 0.1015363 0.4550542
0.2760043 0.1674051 0.1015363

```

---

<sup>6</sup> Ghysels *et al.* (2016).

dove `type = "C"` corrisponde ad un'espressione *aggregates-based* completamente vincolata tranne che per il vincolo di uguaglianza di schema di ponderazione tra variabili/frequenze diverse (si veda l'elenco di vincoli di uguaglianza a p. 6). Si nota che il codice produce lo stesso risultato, ma ripetuto, della funzione<sup>7</sup>:

```
nealmon(p = c(1, -0.5), d = 4)
[1] 0.4550542 0.2760043 0.1674051 0.1015363
```

L'output viene ripetuto due volte poiché  $\frac{d}{m} = 2$  con  $d = 8$  e  $m = 4$ .)

Dunque la funzione `amweights` permette di definire esplicitamente una nuova funzione vincolo dipendente dalla relazione (6).

È inoltre possibile indicare direttamente nella funzione `midas_r` la restrizione *aggregates-based* nel seguente modo:

```
eq_3 <- midas_r(y ~ trend + mls(x, 0:7, 4, amweights,
nealmon, "C") + mls(z, 0:16, 12, nealmon), start = list(x =
c(1, -0.5), z = c(2, 0.5, -0.1)))
```

La selezione alternativa di `type = "A"` o `type = "B"` dipende dal numero di parametri della specificazione, in particolare questi `type` vengono utilizzati nel caso di specificazioni con un grande numero di parametri. In questa situazione è necessario adattare gli *starting values* per tenere conto dell'aumento del numero di parametri.

Inoltre è importante tenere presente che, ogni qualvolta si volessero utilizzare restrizioni collegate agli aggregati, il numero di periodo deve essere un multiplo della frequenza  $m$ . Un esempio di quanto detto è l'`eq_3` sopra riportata: per la variabile  $x$  è possibile utilizzare gli aggregati, mentre per la

---

<sup>7</sup> Ghysels *et al.* (2016).

variabile  $z$  questo non è possibile, poiché  $d_2 = 17$  non è multiplo della frequenza ( $m = 4$ ).

Si propongono nella Tabella 1 alcuni esempi di specificazioni analitiche di modelli MIDAS corredati con il corrispondente codice R<sup>8</sup>.

Descrizione	Espressione analitica	Codice R
Funzioni vincolo differenti	$y_t = c + \sum_{j=0}^7 \beta_j^{(1)} x_{4t-j} + \sum_{j=0}^{16} \beta_j^{(2)} z_{12t-j} + \varepsilon_t$	<pre>midas_r(y~mls(x,0:7,4, nealmon) + mls(z, 0:16, 12, gompertzp), start = list(x = c(1, -0.5), z = c(1, 0.5, 0.1)))</pre>
Vincolo parziale (solo su $z$ )	$y_t = c + \sum_{j=0}^7 \beta_j^{(1)} x_{4t-j} + \sum_{j=0}^{16} \beta_j^{(2)} z_{12t-j} + \varepsilon_t$	<pre>midas_r(y~mls(x,0:7,4) + mls(z, 0:16, 12, nealmon), start = list(z = c(1, 0.5, 0.1)))</pre>
Con termine autoregressivo non vincolato	$y_t = c + \sum_{j=1}^2 \alpha_j y_{t-j} + \sum_{j=0}^{16} \beta_j x_{4t-j} + \varepsilon_t$	<pre>midas_r(y~mls(y,1:2,1) + mls(x,0:7,4, nealmon), start = list(x = c(1, -0.5)))</pre>
<i>Aggregates-based</i> (caso C)	$y_t = c + \lambda \sum_{r=0}^1 \sum_{s=1}^4 w(\delta; s) x_{4(t-1-r)+s} + \varepsilon_t$	<pre>midas_r(y~mls(x,0:7,4, amweights, nealmon, "C") start = list(z = c(1, -0.5)))</pre>

**TABELLA 1:** ALCUNI ESEMPI DI SPECIFICAZIONI DI MODELLI MIDAS CON CODICE R

<sup>8</sup> Ghysels *et al.* (2016).

## 2.5 Test di adeguatezza sui vincoli

Dopo aver specificato un modello, è necessario testare l'adeguatezza dei vincoli funzionali imposti ( $H_0$ ). Questo può essere fatto sotto alcune assunzioni standard, che non vengono approfondite in questo lavoro. Nel pacchetto `midasr` è possibile condurre un test di questo genere utilizzando le funzioni `hAh_test` e `hAhr_test`.

Nel caso di una serie  $\{y_t\}$  stazionaria il test potrà essere applicato direttamente sulla variabile, invece, nel caso di  $\{y_t\}$  non stazionaria sarà necessario applicare una trasformazione particolare. Quest'ultimo caso non verrà approfondito.

Sotto le ipotesi di indipendenza e identità della distribuzione del termine dell'errore  $\varepsilon_t$  del processo è possibile utilizzare la funzione `hAh_test`. Nel caso in cui, invece, quest'ipotesi non sia soddisfatta è possibile utilizzare la funzione `hAhr_test`, basata sulla versione HAC robusta del test.

È sempre preferibile utilizzare la funzione `hAh_test` nel caso in cui non si riscontrassero HAC significativi tra i residui, poiché questa produce un risultato per il test più preciso in piccoli campioni.

Di seguito si propone un esempio di test sul modello `eq_r`:

	<code>hAh_test(eq_r)</code>	<code>hAhr_test(eq_r)</code>
<b>Test</b>	16.552	14.854
<b>df</b>	20	20
<b>p-value</b>	0.6818	0.7847

**TABELLA 2:** RISULTATI DEI TEST DI ADEGUATEZZA

Secondo i test sopra riportati, sia per `hAh_test`, sia per `hAhr_test` non è possibile rifiutare l'ipotesi nulla di adeguatezza dei vincoli. Questo non è vero, invece, se, modificando il numero di parametri della funzione vincolo da tre a due, si ristima il modello e si riconduce il test<sup>9</sup>:

```
eq_rb <- midas_r(y ~ trend + mls(x, 0:7, 4, nealmon) +
mls(z, 0:16, 12, nealmon), start = list(x = c(1, -0.5), z =
c(2, -0.1)))
```

	<code>hAh_test(eq_rb)</code>	<code>hAhr_test(eq_rb)</code>
<b>Test</b>	37.994	33.364
<b>df</b>	21	21
<b>p-value</b>	0.01291	0.04233

**TABELLA 3:** RISULTATI DEI TEST DI ADEGUATEZZA

Nel caso in cui, per evidenza empirica, non si possa rifiutare l'ipotesi di adeguatezza dei vincoli per due o più modelli, per la scelta del modello migliore è possibile fare affidamento sui criteri d'informazione.

## 2.6 Scelta del modello

Selezionare un modello, tra tutte le possibili combinazioni prodotte dalla scelta di differenti funzioni vincolo e il relativo ordine del polinomio (che può essere diverso per le variabili ad alta frequenza e  $x$  e  $z$ , come nell'esempio sopra riportato) non è semplice. Nel pacchetto `midasr`, però, esiste la funzione `expand_weights_lags` che produce l'insieme di tutti i

<sup>9</sup> Ghysels *et al.* (2016).

potenziali modelli corrispondenti ad ogni variabile esplicativa. Di seguito si propone un esempio<sup>10</sup>:

```
set_z <- expand_weights_lags(weights = c("nealmon",  
"nealmon"), from = 0, to = c(10, 20), m = 1, start =  
list(nealmon = c(1, -1), nealmon = c(1, -1, 0)))
```

```
set_x <- expand_weights_lags(weights = c("nealmon",  
"almonp"), from = 0, to = c(5, 10), m = 1, start =  
list(nealmon = c(1, -1), almonp = c(1, 0, 0)))
```

La funzione produce un output in cui vengono mostrati, per ogni variabile, i possibili vincoli (sotto la colonna *weights*) e gli appropriati *starting value* (sotto la colonna *starts*), che definiscono implicitamente il numero di parametri per ogni funzione vincolo.

La struttura potenziale dei ritardi è ottenuta a partire dal seguente intervallo di ritardi ad alta frequenza: da  $[from; m \cdot \min(to)]$  a  $[from; m \cdot \max(to)]$ . Nel caso si stia modellando un *aggregates-based* la funzione *amweights* sarà presente in *midas\_r* e sarà necessario impostare *m* uguale alla frequenza. In questo modo si assicura che la struttura dei ritardi sia multiplo della frequenza, condizione fondamentale per questo tipo di modello MIDAS. In tutti gli altri casi sarà sufficiente mantenere il valore di *m* impostato di default, ovvero  $m = 1$ .

Dunque l'insieme dei potenziali modelli viene definito come tutte le possibili combinazioni di funzioni e di strutture dei ritardi con i corrispettivi *starting value*.

Si fornisce un esempio di seguito di quanto detto in precedenza:

---

<sup>10</sup> Ghysels *et al.* (2016).

```

expand_weights_lags(weights = c("nealmon", "nbeta"),
from = 1, to = c(2, 3), m = 1, start = list(nealmon = c(1,
-1), nbeta = rep(0.5, 3)))
weights lags          starts
1 nealmon 1:2          c(1, -1)
2 nealmon 1:3          c(1, -1)
3  nbeta  1:2 c(0.5, 0.5, 0.5)
4  nbeta  1:3 c(0.5, 0.5, 0.5)

```

Il prossimo comando, invece, permette la stima di tutti i modelli le cui potenziali specificazioni erano state definite in precedenza e produce una tabella riassuntiva di tutti i modelli, in cui sono presenti anche i criteri d'informazione e i test di adeguatezza sui vincoli funzionali imposti sui parametri<sup>11</sup>:

```

eqs_ic <- midas_r_ic_table(y ~ trend + mls(x, 0, m = 4) +
fmls(z, 0, m =12), table = list(z = set_z, x = set_x))

```

La tabella riassuntiva è un `data.frame` in cui ogni riga corrisponde ad un potenziale modello. È inoltre possibile aggiornare la tabella precedentemente prodotta tramite l'utilizzo della funzione `update`:

```

eqs_ic <- update(eqs_ic)

```

Si può selezionare il miglior modello in base ai criteri d'informazione utilizzando il seguente comando:

```

modsel(eqs_ic, IC = "AIC", type = "restricted")

```

```

Selected model with AIC = 674.5565

```

```

Based on restricted MIDAS regression model

```

```

The p-value for the null hypothesis of the test hAh_test
is 0.7413531

```

---

<sup>11</sup> Ghysels *et al.* (2016.).

Il comando produce anche la stampa a video del summary del modello selezionato:

```
Formula y ~ trend + mls(x, 0:7, m = 4, almonp) + mls(z,
0:10, m = 12,          nealmon)
```

Parameters:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	2.0090716	0.1192771	16.844	< 2e-16	***
trend	0.0997984	0.0008025	124.354	< 2e-16	***
x1	0.7726653	0.0788237	9.802	< 2e-16	***
x2	-0.2634821	0.0449879	-5.857	1.55e-08	***
x3	0.0231120	0.0051073	4.525	9.49e-06	***
z1	2.2396888	0.1828210	12.251	< 2e-16	***
z2	0.3844866	0.1533595	2.507	0.012835	*
z3	-0.0703943	0.0203889	-3.453	0.000656	***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9242 on 240 degrees of freedom

Viene indicato anche il p-value del test `hAh_test`, su cui, però, è necessario fare una precisazione. Il test è il prodotto di una complessa procedura di selezione del modello, dunque il risultato ottenuto in questo modo potrebbe non coincidere con il test svolto singolarmente sul modello.

## 2.7 Previsioni

Il pacchetto `midasr` offre anche la possibilità di produrre delle previsioni di un modello specifico.

Nel caso di un modello U-MIDAS, dunque privo di restrizioni, è possibile produrre delle previsioni condizionate utilizzando la funzione `predict.lm`. Invece, nel caso di un modello specifico, è possibile ottenere delle previsioni puntuali utilizzando la funzione `predict`.

La funzione `predict` permette di ottenere una previsione  $\hat{y}_{T+1|T}$  per un dato modello. Si supponga, di essere interessati alla previsione  $\hat{y}_{T+1|T}$  del modello in equazione (x). Per produrre questa previsione, però, sono necessari anche i nuovi dati  $x_{4(T+1)}, \dots, x_{4T-3}$  e  $z_{12(T+1)}, \dots, z_{12T-4}$ . Il pacchetto `midasr` fornisce già la funzione `forecast` che permette il calcolo dei dati in interesse. Quest'ultima funzione assume che il modello venga stimato con i dati fino all'indice a bassa frequenza  $T$ . Inoltre si assume che i nuovi dati siano quelli dopo l'indice  $T$  a bassa frequenza. In questo modo viene calcolata la previsione appropriata. Si propone un esempio in R<sup>12</sup>:

```
newx <- rnorm(4)
newz <- rnorm(12)

forecast(eq_rb, newdata = list(x = newx, z = newz, trend =
251))
  Point Forecast
1          26.95917
```

Se, invece, non sono richiesti nuovi dati per la previsione del modello si procede diversamente.

Prendendo per riferimento il modello:

$$y_{t+l} = 2 + 0.1t + \sum_{j=0}^7 \beta_j^{(1)} x_{4t-j} + \sum_{j=0}^{16} \beta_j^{(2)} z_{12t-j} + \varepsilon_{t+l}$$

dove  $l$  è l'orizzonte di previsione. Il modello può essere scritto come:

---

<sup>12</sup> Ghysels *et al.* (2016).

$$y_t = 2 + 0.1(t + l) + \sum_{j=4l}^{7+4l} \beta_j^{(1)} x_{4t-j} + \sum_{j=12l}^{16+12} \beta_j^{(2)} z_{12t-j} + \varepsilon_t$$

e può essere stimato con la funzione `midas_r` e possono essere ricavate le previsioni  $\hat{y}_{T+l|T}, \dots, \hat{y}_{T+1|T}$  utilizzando le variabili esplicative ad alta frequenza fino a  $T$ , l'indice a bassa frequenza. Per ovviare al problema della mancanza dei valori delle variabili esplicative a tempi successivi rispetto  $T$ , è necessario inserire degli NA nelle variabili esplicative. Per  $l = 1$  la previsione sarà la seguente<sup>13</sup>:

```
eq_f <- midas_r(y ~ trend + mls(x, 4 + 0:7, 4, nealmon) +
mls(z, 12 + 0:16, 12, nealmon), start = list(x = c(1, -0.5),
z = c(2, + 0.5, -0.1)))

forecast(eq_f, newdata = list(x = rep(NA, 4), z =
rep(NA, 12), trend = 251))

Point Forecast
1          27.20452
```

Il pacchetto `midasr` è, inoltre, provvisto di un'ulteriore funzione di previsione: `select_and_forecast`, che permette di ottenere previsioni out-sample e permette di confrontare e valutare diverse specificazioni di modelli MIDAS anche in base alle performance delle previsioni. Dunque, più precisamente, la funzione sopra indicata permette la selezione del modello migliore per ogni orizzonte di previsione, per alcune potenziali specificazioni, definite, come nel caso visto nel paragrafo precedente, dalle combinazioni delle funzioni vincolo e dall'ordine dei ritardi considerato. Inoltre produce conduce combinazioni di previsioni secondo uno specifico schema di ponderazione.

---

<sup>13</sup> Ghysels *et al.* (2016).

La funzione `select_and_forecast` possiede la particolarità, quindi, di segnalare il la miglior specificazione in base all'orizzonte di previsione  $l \in \{1, 2, 3\}$  a bassa frequenza.

Quindi sarà importante definire l'insieme di potenziali modelli, ovvero le funzioni che impongono restrizioni sui parametri e l'ordine dei ritardi, per ogni orizzonte di previsione.

Si vuole proporre un esempio dell'utilizzo di questa funzione. Avendo due variabili  $x$  e  $z$  ad alta frequenza, rispettivamente con frequenza  $m_1 = 4$  e  $m_2 = 12$ . Supponendo di essere interessanti alla previsione della variabile  $y$  a bassa frequenza fino a tre periodi  $l \in \{1, 2, 3\}$  in avanti, in termini di variabili ad alta frequenza l'orizzonte corrispondente, alla bassa frequenza, per le variabili ad alta frequenza sarà:  $lm_1 \in \{4, 8, 12\}$  per la variabile  $x$  e  $lm_2 \in \{12, 24, 36\}$  per la variabile  $z$ .

Questi vettori specifici permettono la definizione del più piccolo ritardo delle variabili ad alta frequenza considerate nel rispettivo modello di previsione. In R è importante specificare quest'informazione, che si traduce nell'opzione `from` della funzione `select_and_forecast`.

Oltre all'opzione `from`, la funzione è provvista anche dell'opzione `to`, che permettere di scegliere il numero minimo di ritardi ad alta frequenza che dovranno possedere il modelli di previsione ed il numero massimo di questi ultimi.

Volendo continuare l'esempio delle righe precedenti, si può decidere che per tutti i potenziali modelli con orizzonte temporale di previsione a bassa frequenza  $l \in \{1, 2, 3\}$ , dovranno possedere come minimo 10 ritardi ad alta frequenza e come massimo 15 ritardi, ovvero: per la variabile  $x$  tre coppie date da  $(lm_1 + 10, lm_1 + 15)$  e analogamente per la  $z$  tre coppie date da  $(lm_2 + 10, lm_2 + 15)$ . Se, come affermato in precedenza,  $lm_1 \in \{4, 8, 12\}$  le coppie

risultanti saranno: (14, 19), (18, 23), (22, 27) per  $l = \{1, 2, 3\}$ . Le coppie per la variabile  $z$  si possono determinare in modo analogo.

Di seguito si propone l'esempio appena descritto tramite codice R<sup>14</sup>:

```
> ms <- select_and_forecast(  
+ y ~ trend + mls(x, 0, 4) + mls(z, 0, 12),  
+ from = list(x = c(4, 8, 12), z = c(12, 24, 36)),  
+ to = list(x = rbind(c(14, 19), c(18, 23), c(22, 27)),  
+ z = rbind(c(22, 27), c(34, 39), c(46, 51))),  
+ insample = 1:200, outsample = 201:250,  
+ weights = list(x = c("nealmon", "almonp"),  
+ z = c("nealmon", "almonp")),  
+ wstart = list(nealmon = rep(1, 3), almonp = rep(1, 3)),  
+ IC = "AIC", seltype = "restricted", ftype = "fixed")
```

Similmente alla funzione `midas_r_ic_table` è possibile selezionare delle particolari righe della tabella ottenuta, per esempio:

```
ms$accuracy$individual
```

permette di distinguere le migliori specificazioni per le previsioni, sulla base dei criteri specificati e in base alla precisione delle loro previsioni out-of-sample per ogni orizzonte previsivo.

## 2.8 Applicazione e confronto

Dopo aver passato in rassegna dal punto di vista teorico i modelli MIDAS e aver analizzato funzioni e opzioni del pacchetto `midasr`, nelle prossime pagine si vuole proporre l'applicazione di quanto detto finora. In particolare

---

<sup>14</sup> Ghysels *et al.* (2016).

si vuole proporre un tentativo di stima di un modello MIDAS a partire da dati macroeconomici reali, per poi mettere a confronto quanto fatto con un classico modello ARIMA.



## Capitolo 3

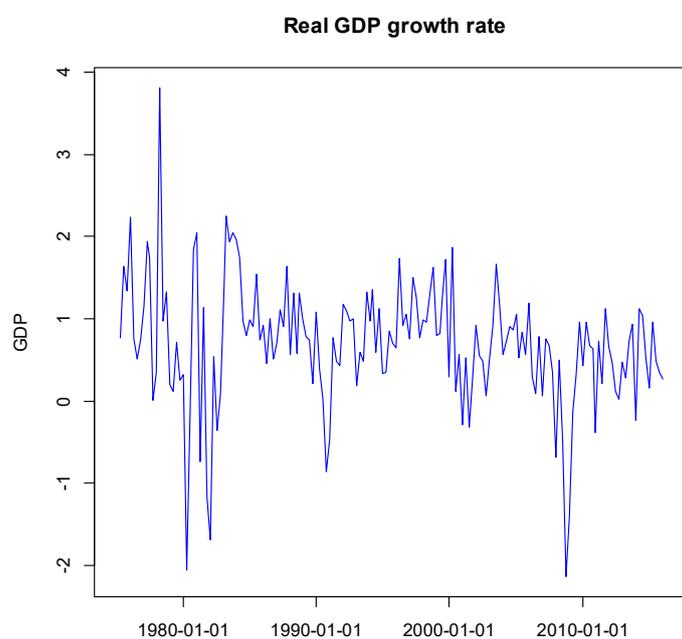
### Real GDP: stima con modelli MIDAS

#### 3.1 Il problema

Nelle prossime pagine si propone un esempio di stima di un modello MIDAS seguendo l'articolo di Armesto *et al.* e la guida: “EViews Training, *MIDAS Regression*”.

La serie d'interesse, ovvero quella a bassa frequenza è la serie trimestrale del *Real GDP Growth rate*. Per GDP (Gross Domestic Product) si intende “la misura della produzione

statunitense, il valore di mercato di beni e servizi prodotti dal lavoro e dalle



**FIGURA 2** GRAFICO DEL GDP REALE DAL 01-01-1975 AL 01-01-2016.

risorse nel territorio degli Stati Uniti”. Nel caso del *Real GDP Growth rate* si intende il tasso di crescita del GDP reale, ovvero aggiustato rispetto all’inflazione<sup>15</sup>.

Invece, le serie ad alta frequenza utilizzate come variabili esplicative per il modello saranno le serie rilevate mensilmente delle variabili indicate nella tabella di seguito.

La scelta delle variabili è stata fatta tenendo conto di quali potessero essere utili a spiegare la variabile esplicativa scelta.

<b>Variabili ad alta frequenza</b>	<b>Descrizione</b>
Industrial Production (IP) Growth Rate	È l’indicatore economico che misura la produzione reale per tutte le aziende manifatturiere, l’estrazione mineraria etc... <sup>16</sup> degli Stati Uniti.
Employment Growth Rate	Tasso di crescita dell’occupazione degli Stati Uniti.
Capacity Utilization Growth Rate	Tasso di crescita della percentuale di risorse utilizzate dalle aziende industriali per produrre beni della manifattura, dell’industria mineraria, dei servizi elettrici e gas, per tutte le strutture operanti negli Stati Uniti <sup>17</sup> .

**TABELLA 4: VARIABILI AD ALTA FREQUENZA**

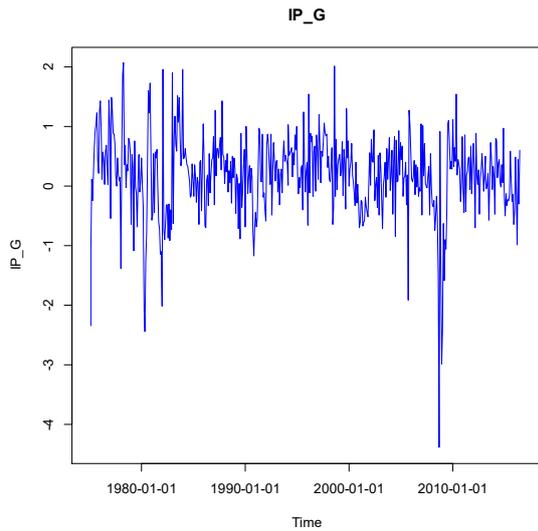
<sup>15</sup> <https://fred.stlouisfed.org/series/GDPC1>

<sup>16</sup> <https://fred.stlouisfed.org/series/INDPRO>

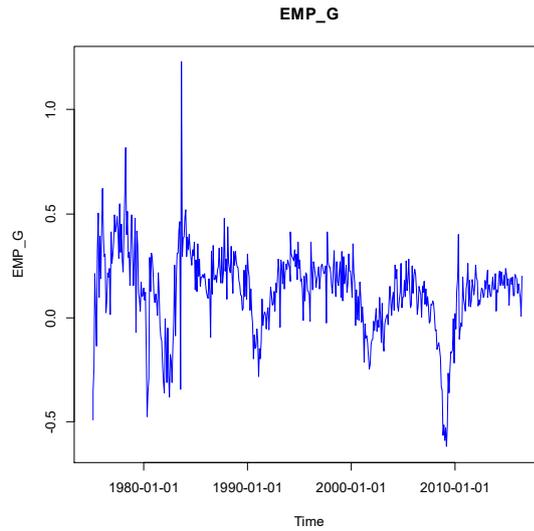
<sup>17</sup> <https://fred.stlouisfed.org/series/TCU>

Le variabili sopra elencate sono le stesse considerate nella guida “EViews Training, *MIDAS Regression*”, e i dati sono stati raccolti dal sito della Federal Reserve Economic Data (FRED)<sup>18</sup>.

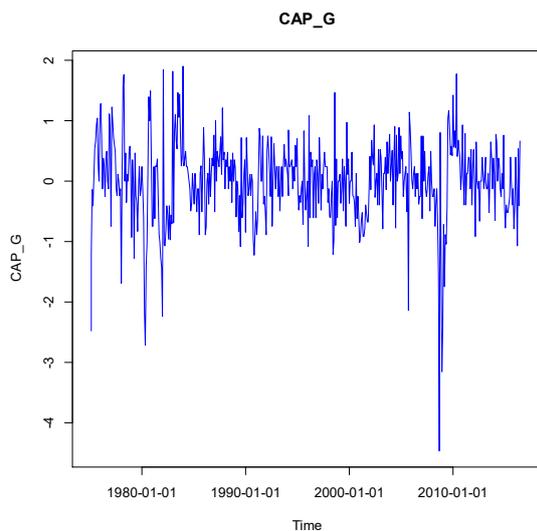
Di seguito si riportano i grafici delle serie sopraelencate.



**FIGURA 3** GRAFICO DELLA SERIE DELL'INDUSTRIAL PRODUCTION GROWTH RATE.



**FIGURA 4** GRAFICO DELLA SERIE DELL'EMPLOYMENT GROWTH RATE.



**FIGURA 5** GRAFICO DELLA SERIE DEL CAPACITY UTILIZATION GROWTH RATE.

---

<sup>18</sup> <https://fred.stlouisfed.org/>

## 3.2 La stima dei modelli MIDAS in R

Dopo aver scelto quali variabili inserire nei modelli, si procede con l'implementazione dei dati e dei modelli in R. Tutti i comandi utilizzati per ottenere i risultati di seguito esposti sono riportati nell'Appendice.

Innanzitutto si caricano i dati in R. Però è necessario trasformare i dati in formato *timeSeries*, in modo da rendere più semplice e immediata l'interpretazione del modello che successivamente verrà stimato.

È importante selezionare un campione di dati ristretto, rispetto al totale, in modo da permettere, poi, il calcolo di previsioni out-of-sample che, grazie al confronto con la serie reale dei dati, permetteranno di avere un'ulteriore informazione sulla bontà dei modelli proposti e permetterà il loro confronto.

Seguendo quanto esposto nella guida: “EViews Training, *MIDAS Regression*”, si propongono quattro modelli con le variabili sopra descritte, le cui espressioni analitiche, corredate dal rispettivo codice R vengono di seguito riportate:

Descrizione	Espressione analitica	Codice R
um	$y_t = c + \alpha_1 y_{t-1} + \sum_{j=1}^6 \beta_j x_{3t-j} + \sum_{j=1}^9 \lambda_j x_{3t-j} + \sum_{j=1}^9 \vartheta_j x_{3t-j} + \varepsilon_t$	<pre>midas_r(yy1 ~ mls(yy1, 1, 1) +mls(xempg, 1:6, 3) + mls(xipg, 1:9, 3) + mls(xcapg, 1:9, 3), start = NULL)</pre>

m1	$y_t = c + \alpha_1 y_{t-1} + \sum_{j=1}^6 \beta_j x_{3t-j} + \sum_{j=1}^9 \lambda_j x_{3t-j} + \sum_{j=1}^9 \vartheta_j x_{3t-j} + \varepsilon_t$	<pre>midas_r(yy1~mls(yy1,1,1)+mls (xempg,1:6,3,nbeta)+ mls(xipg,1:9,3,nbeta)+mls(xcapg,1:9,3,nbeta),start=list(xempg=c(1.7,1,5), xipg = c(0.5,1,2), xcapg = c(0.8,1.5,2)))</pre>
m2	$y_t = c + \alpha_1 y_{t-1} + \sum_{j=1}^6 \beta_j x_{3t-j} + \sum_{j=1}^9 \lambda_j x_{3t-j} + \sum_{j=1}^9 \vartheta_j x_{3t-j} + \varepsilon_t$	<pre>midas_r(yy1 ~ mls(yy1,1,1) + mls(xempg,1:6,3,nealmon)+mls(xipg,1:9,3,nealmon)+mls(xcapg,1:9,3,nealmon),start=list(xempg=c(0.1,0), xipg=c(0.1,0.8), xcapg= c(0.1,1)))</pre>
m3	$y_t = c + \alpha_1 y_{t-1} + \sum_{j=1}^6 \beta_j x_{3t-j} + \sum_{j=1}^9 \lambda_j x_{3t-j} + \sum_{j=1}^9 \vartheta_j x_{3t-j} + \varepsilon_t$	<pre>midas_r(yy1 ~ mls(yy1,1,1) + mls(xempg,1:6,3,almonp)+ +mls(xipg,1:9,3,almonp) + mls(xcapg,1:9,3,almonp), start=list(xempg= c(1.7,1,5), xipg= c(0.5,1,2), xcapg= c(0.8,1.5,2)))</pre>

**TABELLA 5: MODELLI STIMATI**

Per quanto riguarda le variabili esplicative ad alta frequenza, si è scelto di non utilizzare solamente i tre mesi che sarebbero sufficienti per completare l'allineamento delle frequenze, ma di utilizzare nove ritardi (per le variabili Industrial Production Growth Rate e Capacity Utilization Growth Rate) e sei ritardi (Employment Growth Rate), in modo da spiegare al meglio i dati trimestrali del GDP.

Le funzioni vincolo, ove presenti, sono state scelte in analogia rispetto alla guida sopra indicata, così come il numero di coefficienti stimati per ciascuna

variabile ad alta frequenza. Quest'ultimo viene imposto tramite l'opzione `start` della funzione `midas_r`. Per il modello vincolato con la funzione *Exponential Almon lag polynomial* sono stati inseriti due *starting values*, mentre per gli altri modelli vincolati ne sono stati inseriti tre.

Di seguito vengono riportati i coefficienti dei modelli stimati:

```
> coef(m1)
(Intercept)          yy1          xempg1          xempg2          xempg3
 0.47990350 -0.07872784  1.69460818  1.06455794 13.97278719
xipg1          xipg2
0.21049201  0.60605213
          xipg3          xcapg1          xcapg2          xcapg3
 2.01280677  0.38720983  2.66026964 11.86076978
```

```
> coef(m2)
(Intercept)          yy1          xempg1          xempg2          xipg1
 0.3940309  -0.1293724  1.4353802  -1.0022277  0.9524126
xipg2          xcapg1
-0.3063343  -0.2993226
          xcapg2
 0.3357049
```

```
> coef(m3)
(Intercept)          yy1          xempg1          xempg2          xempg3
 0.39223790 -0.11101713  1.40443608 -0.53478222  0.04598372
xipg1          xipg2
1.70143327 -0.82213832
          xipg3          xcapg1          xcapg2          xcapg3
 0.07933631 -1.36347850  0.73888830 -0.07472010
```

```
> coef(um)
```

(Intercept)	yy1	xempg1	xempg2	xempg3
0.38418330	-0.11341479	0.89006770	0.53530090	-0.26306686
xempg4	xempg5			
0.17400718	-0.15109097			
xempg6	xipg1	xipg2	xipg3	xipg4
0.12247821	1.15582739	0.87706520	-0.48180660	-1.74723221
xipg5	xipg6			
0.17195291	0.05912982			
xipg7	xipg8	xipg9	xcapg1	xcapg2
0.12333707	0.92732176	0.06105040	-0.95904903	-0.51946693
xcapg3	xcapg4			
0.71817732	1.85200633			
xcapg5	xcapg6	xcapg7	xcapg8	xcapg9
-0.29117114	-0.18850316	-0.06221404	-0.91218218	-0.04935637

Ovviamente i modelli vincolati m1, m2, m3 possiedono un numero decisamente inferiore di coefficienti rispetto al modello U-MIDAS non vincolato. Per questo i modelli vincolati sono preferibili al modello U-MIDAS dal punto di vista della parsimonia.

### 3.3 Confronto tra modelli e test di adeguatezza

Dopo aver stimato i modelli, questi ultimi possono essere valutati e confrontati mediante alcuni criteri d'informazione, come il criterio di Akaike (AIC) e il Criterio di Schwarz (BIC).

Nella tabella 4 sono riassunti i risultati ottenuti:

	<b>Modello</b>			
<b>Criterio</b>	m1	m2	m3	um
<b>AIC</b>	146.7773	136.82	138.981	148.5624
<b>BIC</b>	179.183	161.1244	171.3868	221.4754

**TABELLA 6:** CRITERI D'INFORMAZIONE

Si può notare che il modello U-MIDAS possiede BIC decisamente superiore rispetto agli altri modelli. Questo avviene poiché è presente un numero decisamente superiore di parametri rispetto agli altri modelli e perché questo specifico criterio penalizza i modelli con un maggior numero di parametri, rispetto a quelli più parsimoniosi.

Sia per il criterio di Akaike, che per quello di Schwarz, il miglior modello risulta essere m2, ovvero il modello con funzione vincolo *Exponential Almon lag polynomial*.

È possibile, inoltre, valutare l'adeguatezza dei vincoli imposti tramite le funzioni `hAh_test` e `hAhr_test`. Nella tabella 5 vengono riportati i p-value relativi ai test indicati.

	<b>Modello</b>		
<b>Test (p-value)</b>	m1	m2	m3
<code>hAh_test</code>	0.05661	0.2935	0.311
<code>hAhr_test</code>	0.004856	0.07482	0.1273

**TABELLA 7:** TEST DI ADEGUATEZZA

I test svolti evidenziano che per il modello m1 l'ipotesi nulla di adeguatezza del vincolo non può essere accettata per la versione robusta del test. Invece, per gli altri modelli, l'ipotesi non può essere rifiutata a livello di significatività 0,05, dunque i vincoli imposti risultano essere adeguati.

### 3.4 Previsioni

Un altro modo per valutare i modelli fino ad ora stimati è quello di confrontare le previsioni out-of-sample prodotte con i valori reali della serie d'interesse.

Per fare questo è possibile utilizzare la funzione R `average_forecast`, facente parte del pacchetto `midasr`, che produce questo tipo di previsioni e permette il confronto dei vari modelli tramite alcuni indici dell'affidabilità della stima, come il *Mean Squared Error* e il *Root Mean Squared Error*.

Si è scelto come campione in-sample, su cui stimare i modelli, la serie di osservazioni dal 1985-01-01 al 2013-03-01, mentre il campione out-of-sample arriva fino al secondo trimestre del 2015. Vengono prodotte, dunque, nove previsioni, che vengono riportate nella tabella 6.

	<b>Modelli</b>			
	m1	m2	m3	um
<b>2013-04-01</b>	0.6913248	0.6443269	0.6649554	0.7992956
<b>2013-07-01</b>	0.7898262	0.7274813	0.6633588	0.4222767
<b>2013-10-01</b>	0.8512770	0.7702559	0.7781928	0.7010546
<b>2014-01-01</b>	0.6777203	0.6465835	0.6445575	0.7205806
<b>2014-04-01</b>	0.9899580	0.9441859	0.9508134	1.1435922
<b>2014-07-01</b>	0.6810919	0.6540156	0.7203417	0.9579781
<b>2014-10-01</b>	0.9306073	0.8758547	0.8447173	0.7199838
<b>2015-01-01</b>	0.5244967	0.5767975	0.5883883	0.5210909
<b>2015-04-01</b>	0.5372392	0.4748948	0.4008711	0.5809067

**TABELLA 8:** PREVISIONI OUT-OF-SAMPLE

È possibile confrontare le previsioni ottenute confrontando gli RMSE calcolati sui quattro modelli stimati:

<b>Modelli</b>				
	m1	m2	m3	um
<b>RMSE</b>	0.4277945	0.4296379	0.4335125	0.4294838

**TABELLA 9:** PREVISIONI OUT-OF-SAMPLE

Il modello che presenta il minor MSE è il modello m1, ovvero quello con vincolo da funzione *Beta*. Quest'ultimo modello, dunque, potrebbe essere un buon candidato per spiegare la serie del *Real GDP Growth rate*.

### 3.5 Stima di un modello ARIMA e confronto con i modelli MIDAS

Oltre alla stima dei modelli MIDAS e all'operatività in ambiente R, si è interessati a valutare l'utilità dei modelli MIDAS rispetto ai classici modelli ARIMA per serie storiche. Per questo motivo, oltre alla stima di modelli MIDAS, è stata svolta anche la stima della serie del *Real GDP Growth rate* tramite modello ARIMA.

Il modello ARIMA è stato stimato sullo stesso campione ridotto dei modelli MIDAS per poter permettere, anche in questo caso, il calcolo di previsioni che possono essere confrontate con il valori reali della serie.

Di seguito si riportano i risultati del processo di stima del modello ARIMA:

```
> fit <- auto.arima(yy1)
> summary(fit)
```

```
Series: yy1
ARIMA(1,1,1) (0,0,1)
```

```
Coefficients:
```

```
          ar1          ma1          sma1
          0.3909   -0.9595   0.1816
s.e.    0.0977    0.0363   0.0981
```

```
sigma^2 estimated as 0.3027:  log likelihood=-91.29
AIC=190.59  AICc=190.96  BIC=201.46
```

Non è interesse di questo lavoro valutare la bontà del modello ARIMA, dunque si procederà semplicemente con il calcolo delle previsioni degli stessi nove trimestri per cui sono state calcolate utilizzando i modelli MIDAS:

```
> prev.yy1
```

```
$pred
```

```
          Qtr1          Qtr2          Qtr3          Qtr4
2013                0.4145590 0.3709119 0.3238328
2014 0.4160831 0.4185165 0.4194677 0.4198396
2015 0.4199849 0.4200417
```

```
> yy1
```

```
2013-04-01  0.277801286
2013-07-01  0.734063834
2013-10-01  0.937669216
2014-01-01 -0.232481430
2014-04-01  1.116800113
2014-07-01  1.046605926
2014-10-01  0.512722956
2015-01-01  0.160229181
2015-04-01  0.961531082
```

Rispetto alle previsioni dei modelli MIDAS si può notare come queste colgano in modo decisamente inferiore l'andamento della serie del *Real GDP Growth rate*. Infatti il RMSE delle previsioni sopra indicate è pari a: 0.4942211, che è decisamente superiore rispetto a tutti quelli calcolati per i modelli MIDAS.

## Conclusioni

L'obiettivo del lavoro svolto è stato quello di proporre una possibile soluzione al problema dell'utilizzo di dati ad alta frequenza nei modelli statistici per serie storiche. Se, normalmente, l'uso di dati a frequenza diversa in un unico modello risulta difficoltoso, l'introduzione dei modelli MIDAS può essere un'alternativa per risolvere questo problema.

In questo lavoro ci si è occupati innanzitutto della presentazione dei modelli MIDAS, tramite una rassegna concisa degli elementi formali più importanti, come la specificazione del modello, l'allineamento delle frequenze e l'aspetto predittivo.

Inoltre si è voluto dare spazio all'analisi pratica di questi modelli, proponendo l'utilizzo di un pacchetto del software R appositamente creato: `midasr`. Sono state proposte le principali funzioni disponibili nel pacchetto e si è voluta dimostrare l'attendibilità dei risultati ottenuti attraverso il confronto con quanto esposto nell'articolo di Ghysles *et al.*<sup>19</sup>.

L'aspetto più importante della soluzione del problema sopra esposto è l'attuabilità nell'analisi di serie macroeconomiche. Per questo motivo si è deciso di effettuare la stima di alcuni modelli MIDAS per la serie del *Real Gross Domestic Production Growth rate* degli Stati Uniti seguendo quanto indicato nell'articolo di Arnesto *et al.*<sup>20</sup> e nella guida "EViews Training, *MIDAS Regression*". Inoltre si è stimato anche un modello ARIMA per poter

---

<sup>19</sup> Ghysles E, Kvedaras V, Zemlys V, (2016). "Mixed Frequency Data Sampling Regression Models: The R Package `midasr`". *Journal of Statistical Software*, 72, issue 4.

<sup>20</sup> Foroni C, Marcellino M, Schumacher C (2014). "U-MIDAS: MIDAS Regressions with Unrestricted Lag Polynomials." *Journal of the Royal Statistical Society A*.

confrontare le previsioni ottenute con entrambe le tecniche. Ciò che ne è risultato è stato un *Root Mean Squared Error* inferiore per le previsioni calcolate con i modelli MIDAS rispetto a quello calcolato per il modello ARIMA.

In conclusione le previsioni calcolate con i modelli MIDAS risultano più attendibili, con ciò confermano che tali modelli rappresentano una valida soluzione del problema sopra esposto.

Tuttavia non è possibile generalizzare quanto emerso dall'analisi svolta affermando che i modelli MIDAS possano essere validi sempre. Si può dire, però, in base ai risultati ottenuti, che possono essere un'alternativa degna di essere valutata quando ci si trova di fronte al problema di dover stimare un modello con dati a frequenza mista.

# Appendice

## Report dei comandi utilizzati

```
library(fArma)
  library(timeSeries)
library(midasr)
### caricamento dei dati

dataq <- read.table("midas_eviews_Q.txt", he=T)
datam <- read.table("midas_eviews_M.txt", he=T)

### creazione delle timeSeries

charvec <- timeSequence(from = "1975-01-01", length.out = 168, by
= "quarter")

y1 <- timeSeries(dataq$rgdp_g, charvec, units = "GDP")

charvec2 <- timeSequence(from = "1975-01-01", length.out = 504, by
= "month")

x2 <- timeSeries(datam$emp_g, charvec2, units = "EMP_G")
x3 <- timeSeries(datam$ip_g, charvec2, units = "IP_G")
x4 <- timeSeries(datam$cap_g, charvec2, units = "CAP_G")

### creazione del campione su cui stimare i modelli

xipg <- window(x3, start = "1985-01-01", end = "2013-03-01" )
xcapg <- window(x4, start = "1985-01-01", end = "2013-03-01" )
xempg <- window(x2, start = "1985-01-01", end="2013-03-01")
yy1 <- window(y1, start = "1985-01-01", end="2013-03-01")
```

```

### stima dei modelli

m1 <- midas_r(yy1 ~ mls(yy1, 1, 1) + mls(xempg, 1:6, 3, nbeta)+
mls(xipg, 1:9, 3, nbeta) + mls(xcapg, 1:9, 3, nbeta),
start = list(xempg = c(1.7,1,5), xipg= c(0.5, 1, 2), xcapg= c(0.8,
1.5, 2)))
summary(m1)

##
## Formula yy1 ~ mls(yy1, 1, 1) + mls(xempg, 1:6, 3, nbeta) + mls(xip
g,
1:9, 3, nbeta) + mls(xcapg, 1:9, 3, nbeta)
##
## Parameters:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.799e-01  8.035e-02  5.972 3.68e-08 ***
## yy1          -7.873e-02  1.345e-01  -0.585  0.560
## xempg1       1.695e+00  4.028e-01  4.207 5.70e-05 ***
## xempg2       1.065e+00  7.224e-01  1.474  0.144
## xempg3       1.397e+01  1.114e+02  0.125  0.900
## xipg1        2.105e-01  3.808e-01  0.553  0.582
## xipg2        6.060e-01  2.597e+04  0.000  1.000
## xipg3        2.013e+00  1.272e+06  0.000  1.000
## xcapg1       3.872e-01  3.045e-01  1.272  0.207
## xcapg2       2.660e+00  7.348e+00  0.362  0.718
## xcapg3       1.186e+01  2.968e+01  0.400  0.690
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4457 on 99 degrees of freedom

coef(m1)

## (Intercept)          yy1          xempg1          xempg2          xempg3          x
ipg1
## 0.47990350 -0.07872784  1.69460818  1.06455794 13.97278719  0.2104
9201
##          xipg2          xipg3          xcapg1          xcapg2          xcapg3
## 0.60605213  2.01280677  0.38720983  2.66026964 11.86076978

m2 <- midas_r(yy1 ~ mls(yy1, 1, 1) + mls(xempg, 1:6, 3, nealmon)+
mls(xipg, 1:9, 3, nealmon) + mls(xcapg, 1:9, 3, nealmon),
start = list(xempg = c(0.1,0), xipg= c(0.1,0.8), xcapg= c(0.1,1)))
summary(m2)

##
## Formula yy1 ~ mls(yy1, 1, 1) + mls(xempg, 1:6, 3, nealmon) + mls(x
ipg,
1:9, 3, nealmon) + mls(xcapg, 1:9, 3, nealmon)
##
## Parameters:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.39403    0.06894  5.716 1.09e-07 ***
## yy1         -0.12937    0.15830  -0.817  0.4157
## xempg1      1.43538    0.61554  2.332  0.0217 *

```

```

## xempg2      -1.00223    1.03014   -0.973    0.3329
## xipg1       0.95241    0.39473    2.413    0.0176 *
## xipg2      -0.30633    0.20214   -1.515    0.1327
## xcapg1     -0.29932    0.27300   -1.096    0.2755
## xcapg2      0.33570    0.39990    0.839    0.4032
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4312 on 102 degrees of freedom

coef(m2)

## (Intercept)      yy1      xempg1      xempg2      xipg1      x
ipg2
##  0.3940309 -0.1293724  1.4353802 -1.0022277  0.9524126 -0.306
3343
##      xcapg1      xcapg2
## -0.2993226  0.3357049

m3 <- midas_r(yy1 ~ mls(yy1, 1, 1) + mls(xempg, 1:6, 3, almonp)+
mls(xipg, 1:9, 3, almonp) + mls(xcapg, 1:9, 3, almonp),
start = list(xempg = c(1.7,1,5), xipg= c(0.5, 1, 2), xcapg= c(0.8,
1.5, 2)))
summary(m3)

##
## Formula yy1 ~ mls(yy1, 1, 1) + mls(xempg, 1:6, 3, almonp) + mls(xi
pg,      1:9, 3, almonp) + mls(xcapg, 1:9, 3, almonp)
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.39224    0.06643   5.905 4.98e-08 ***
## yy1         -0.11102    0.16358  -0.679  0.4989
## xempg1       1.40444    0.96458   1.456  0.1486
## xempg2      -0.53478    0.66811  -0.800  0.4254
## xempg3       0.04598    0.09588   0.480  0.6326
## xipg1       1.70143    0.91751   1.854  0.0667 .
## xipg2      -0.82214    0.50322  -1.634  0.1055
## xipg3       0.07934    0.05172   1.534  0.1282
## xcapg1     -1.36348    0.86568  -1.575  0.1184
## xcapg2      0.73889    0.48091   1.536  0.1276
## xcapg3     -0.07472    0.04983  -1.499  0.1369
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4302 on 99 degrees of freedom

coef(m3)

## (Intercept)      yy1      xempg1      xempg2      xempg3      x
ipg1
##  0.39223790 -0.11101713  1.40443608 -0.53478222  0.04598372  1.7014
3327

```

```

##      xipg2      xipg3      xcapg1      xcapg2      xcapg3
## -0.82213832  0.07933631 -1.36347850  0.73888830 -0.07472010

um <- midas_r(yy1 ~ mls(yy1, 1, 1) + mls(xempg, 1:6, 3)+
  mls(xipg, 1:9, 3) + mls(xcapg, 1:9, 3), start = NULL)
summary(um)

##
## Formula yy1 ~ mls(yy1, 1, 1) + mls(xempg, 1:6, 3) + mls(xipg, 1:9,
  3) +      mls(xcapg, 1:9, 3)
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.38418    0.06386   6.016 4.49e-08 ***
## yy1          -0.11341    0.14120  -0.803  0.4241
## xempg1       0.89007    0.38672   2.302  0.0238 *
## xempg2       0.53530    0.45455   1.178  0.2423
## xempg3      -0.26307    0.42568  -0.618  0.5383
## xempg4       0.17401    0.51449   0.338  0.7360
## xempg5      -0.15109    0.43608  -0.346  0.7299
## xempg6       0.12248    0.51927   0.236  0.8141
## xipg1        1.15583    0.65858   1.755  0.0829 .
## xipg2        0.87707    0.79514   1.103  0.2732
## xipg3       -0.48181    0.91699  -0.525  0.6007
## xipg4       -1.74723    0.78413  -2.228  0.0285 *
## xipg5        0.17195    0.98410   0.175  0.8617
## xipg6        0.05913    0.77623   0.076  0.9395
## xipg7        0.12334    0.77713   0.159  0.8743
## xipg8        0.92732    1.03195   0.899  0.3714
## xipg9        0.06105    0.73906   0.083  0.9344
## xcapg1      -0.95905    0.65150  -1.472  0.1447
## xcapg2      -0.51947    0.79175  -0.656  0.5136
## xcapg3       0.71818    0.91004   0.789  0.4322
## xcapg4       1.85201    0.76189   2.431  0.0172 *
## xcapg5      -0.29117    0.96451  -0.302  0.7635
## xcapg6      -0.18850    0.76924  -0.245  0.8070
## xcapg7      -0.06221    0.75987  -0.082  0.9349
## xcapg8      -0.91218    1.01806  -0.896  0.3728
## xcapg9      -0.04936    0.73176  -0.067  0.9464
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4256 on 84 degrees of freedom

coef(um)

## (Intercept)      yy1      xempg1      xempg2      xempg3      xempg4
## 0.38418330 -0.11341479  0.89006770  0.53530090 -0.26306686  0.1740
0718
##      xempg5      xempg6      xipg1      xipg2      xipg3      xipg4
## -0.15109097  0.12247821  1.15582739  0.87706520 -0.48180660 -1.7472

```

```

3221
##      xipg5      xipg6      xipg7      xipg8      xipg9      xc
apg1
##  0.17195291  0.05912982  0.12333707  0.92732176  0.06105040 -0.9590
4903
##      xcapg2      xcapg3      xcapg4      xcapg5      xcapg6      xc
apg7
## -0.51946693  0.71817732  1.85200633 -0.29117114 -0.18850316 -0.0622
1404
##      xcapg8      xcapg9
## -0.91218218 -0.04935637

### test hAh

hAh_test(m1)

##
## hAh restriction test
##
## data:
## hAh = 24.531, df = 15, p-value = 0.05661

hAhr_test(m1)

##
## hAh restriction test (robust version)
##
## data:
## hAhr = 32.893, df = 15, p-value = 0.004856

hAh_test(m2)

##
## hAh restriction test
##
## data:
## hAh = 20.725, df = 18, p-value = 0.2935

hAhr_test(m2)

##
## hAh restriction test (robust version)
##
## data:
## hAhr = 27.228, df = 18, p-value = 0.07482

hAh_test(m3)

##
## hAh restriction test
##
## data:
## hAh = 17.133, df = 15, p-value = 0.311

```

```

hAhr_test(m3)

##
## hAh restriction test (robust version)
##
## data:
## hAhr = 21.308, df = 15, p-value = 0.1273

### AIC
AIC(m1)

## [1] 146.7773

AIC(m2)

## [1] 136.82

AIC(m3)

## [1] 138.981

AIC(um)

## [1] 148.5624

### BIC

BIC(m1)

## [1] 179.183

BIC(m2)

## [1] 161.1244

BIC(m3)

## [1] 171.3868

BIC(um)

## [1] 221.4754

### previsioni

fulldata <- list(xempg = window(x2, start = "1985-01-01", end="2015-06-01"),
xipg = window(x3, start = "1985-01-01", end="2015-06-01"),
xcapg = window(x4, start = "1985-01-01", end="2015-06-01"),
yy1=window(y1, start = "1985-01-01", end="2015-04-01"))

insample <- 1:length(yy1)
insample

## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17

```

```

## [18] 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
33 34
## [35] 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
50 51
## [52] 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66
67 68
## [69] 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83
84 85
## [86] 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 1
01 102
## [103] 103 104 105 106 107 108 109 110 111 112 113

  outsample <- (1:length(fulldata$yy1))[-insample]
  outsample

## [1] 114 115 116 117 118 119 120 121 122

  avgf <- average_forecast(list(m1, m2, m3, um), data = fulldata,
  insample = insample, outsample = outsample)

  avgf

## $forecast
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.6913248 0.6443269 0.6649554 0.7992956
## [2,] 0.7898262 0.7274813 0.6633588 0.4222767
## [3,] 0.8512770 0.7702559 0.7781928 0.7010546
## [4,] 0.6777203 0.6465835 0.6445575 0.7205806
## [5,] 0.9899580 0.9441859 0.9508134 1.1435922
## [6,] 0.6810919 0.6540156 0.7203417 0.9579781
## [7,] 0.9306073 0.8758547 0.8447173 0.7199838
## [8,] 0.5244967 0.5767975 0.5883883 0.5210909
## [9,] 0.5372392 0.4748948 0.4008711 0.5809067
##
## $avgforecast
##           EW      BICW      MSFE      DMSFE
## [1,] 0.6999757 0.6443277 0.7001309 0.7031438
## [2,] 0.6507357 0.7274790 0.6509393 0.6450990
## [3,] 0.7751951 0.7702562 0.7753330 0.7736670
## [4,] 0.6723605 0.6465834 0.6725047 0.6740722
## [5,] 1.0071374 0.9441861 1.0073759 1.0115032
## [6,] 0.7533568 0.6540179 0.7533844 0.7590674
## [7,] 0.8427908 0.8758536 0.8429501 0.8399216
## [8,] 0.5526934 0.5767979 0.5524675 0.5512034
## [9,] 0.4984780 0.4748922 0.4990118 0.5020698
##
## $accuracy
## $accuracy$individual
##
##                                     Model
## 1      yy1 ~ mls(yy1, 1, 1) + mls(xempg, 1:6, 3, nbeta) + mls(xipg
, 1:9, 3, nbeta) + mls(xcapg, 1:9, 3, nbeta)

```

```

## 2 yy1 ~ mls(yy1, 1, 1) + mls(xempg, 1:6, 3, nealmon) + mls(xipg, 1:
9, 3, nealmon) + mls(xcapg, 1:9, 3, nealmon)
## 3 yy1 ~ mls(yy1, 1, 1) + mls(xempg, 1:6, 3, almonp) + mls(xipg,
1:9, 3, almonp) + mls(xcapg, 1:9, 3, almonp)
## 4 yy1 ~ mls(yy1, 1, 1) + mls(xempg, 1:6,
3) + mls(xipg, 1:9, 3) + mls(xcapg, 1:9, 3)
## MSE.out.of.sample MAPE.out.of.sample MASE.out.of.sample MSE.in.sa
mple
## 1 0.1830081 106.2705 0.5697691 0.178
7584
## 2 0.1845887 107.0218 0.5853331 0.172
4419
## 3 0.1879331 108.8409 0.5954715 0.166
5274
## 4 0.1844563 109.0527 0.5558061 0.138
3154
## MAPE.in.sample MASE.in.sample
## 1 114.9088 0.6219355
## 2 117.5446 0.6222526
## 3 115.0049 0.6109838
## 4 108.9007 0.5515193
##
## $accuracy$average
## Scheme MSE MAPE MASE
## 1 EW 0.1784137 107.2412 0.5691634
## 2 BICW 0.1845888 107.0219 0.5853335
## 3 MSFE 0.1783780 107.2285 0.5689999
## 4 DMSFE 0.1780397 107.2420 0.5680863
##
##
## $type
## [1] "fixed"
##
## $x
## [1] 0.98887490 0.91155661 1.54415268 0.74701996 0.92278366
## [6] 0.45757871 1.00265498 0.51647724 0.69660453 1.11570403
## [11] 0.90256543 1.63765688 0.56036703 1.31279880 0.57588755
## [16] 1.31626991 1.00297803 0.78333868 0.74328660 0.21151820
## [21] 1.08893998 0.38597165 0.02449125 -0.85516945 -0.47037728
## [26] 0.77303756 0.47901742 0.43459724 1.17535560 1.09558777
## [31] 0.96781736 0.99681910 0.18692984 0.59246267 0.48615299
## [36] 1.32641493 0.97621087 1.35715495 0.58826370 1.12845207
## [41] 0.34161925 0.34835999 0.85289412 0.70668913 0.65439232
## [46] 1.73104365 0.92077678 1.05121745 0.75881626 1.49784273
## [51] 1.26470598 0.77280622 0.98440757 0.96518888 1.30023212
## [56] 1.62821085 0.79542124 0.82091517 1.25107548 1.72042678
## [61] 0.29008545 1.87087648 0.12063398 0.56630040 -0.28433118
## [66] 0.52852578 -0.31678014 0.27743423 0.91666133 0.54986859
## [71] 0.48590359 0.06327210 0.51701968 0.92349596 1.66114397
## [76] 1.16204734 0.57342650 0.73006655 0.90497041 0.86101924
## [81] 1.06027839 0.52066134 0.83683673 0.56933299 1.19436403
## [86] 0.29860297 0.08906489 0.77968982 0.06181457 0.76239939

```

```

## [91] 0.67031402 0.35616119 -0.68471184 0.49509529 -0.48099239
## [96] -2.13523515 -1.39543574 -0.13504767 0.32616926 0.96323349
## [101] 0.43161042 0.96148362 0.67317110 0.62784540 -0.38698519
## [106] 0.72512361 0.20992520 1.12011869 0.66073538 0.46585455
## [111] 0.11970128 0.02275305 0.47274053
##
## $xout
## [1] 0.2778013 0.7340638 0.9376692 -0.2324814 1.1168001 1.04660
59
## [7] 0.5127230 0.1602292 0.9615311
      sqrt(avgf$accuracy$individual$MSE.out.of.sample)
## [1] 0.4277945 0.4296379 0.4335125 0.4294838
#####modello arima
      h=9
      fit <- auto.arima(yy1)
      summary(fit)
## Series: yy1
## ARIMA(1,1,1)(0,0,1)[4]
##
## Coefficients:
##          ar1          ma1          sma1
##          0.3909   -0.9595   0.1816
## s.e.  0.0977   0.0363   0.0981
##
## sigma^2 estimated as 0.3027:  log likelihood=-91.29
## AIC=190.59   AICc=190.96   BIC=201.46
##
## Training set error measures:
##
##           ME           RMSE           MAE           MPE           MAPE
MASE
## Training set -0.05528183 0.5403862 0.4097419 -73.73974 135.0449 0.7
030009
##           ACF1
## Training set -0.08357692

```

```

AIC(fit)
## [1] 190.5877
  prev.yy1 <- predict (fit, n.ahead = h)
  prev.yy1
## $pred
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2013           0.4145590 0.3709119 0.3238328
## 2014 0.4160831 0.4185165 0.4194677 0.4198396
## 2015 0.4199849 0.4200417
##
## $se
##           Qtr1      Qtr2      Qtr3      Qtr4
## 2013           0.5502137 0.5992290 0.6101774
## 2014 0.6138740 0.6315794 0.6371911 0.6399461
## 2015 0.6418592 0.6434833

```

## Bibliografia

- Andreou E, Ghysels E, Kourtellis A (2011). “Forecasting with mixed-frequency data.” In C M, D Hendry (eds.), *Oxford Handbook of Economic Forecasting*, pp. 225–245.
- Armesto M T, Engermann K M, Owyang M T, (2010). “Forecasting with Mixed Frequencies”. *Federal Reserve Bank of St. Louis Review*, November/December, 92(6), pp.521-536.
- Clements M, Galvão A (2008). “Macroeconomic Forecasting with Mixed Frequency Data: Forecasting US output growth.” *Journal of Business and Economic Statistics*, **26**, 546–554.
- Foroni C, Marcellino M, Schumacher C (2014). “U-MIDAS: MIDAS Regressions with Unrestricted Lag Polynomials.” *Journal of the Royal Statistical Society A*.
- Ghysels E, Kvedaras V, Zemlys V, (2016). “Mixed Frequency Data Sampling Regression Models: The R Package midast”. *Journal of Statistical Software*, **72**, issue 4.
- Ghysels E, Sinko A, Valkanov R (2006). “MIDAS Regressions: Further Results and New Directions.” *Econometric Reviews*, **26**, 53–90.
- Ghysels E, Santa-Clara P, Valkanov R (2002). “The MIDAS touch: Mixed data Sampling Regression Models.” Working paper, UNC and UCLA.
- Kvedaras V, Zemlys V (2012). “Testing the Functional Constraints on Parameters in Regressions With Variables of Different Frequency.” *Economics Letters*, 116(2), 250–254.

## Sitografia

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

<https://fred.stlouisfed.org/series/GDPC1>

<https://fred.stlouisfed.org/series/INDPRO>

<https://fred.stlouisfed.org/>

<https://fred.stlouisfed.org/series/TCU>