

AN EVOLUTIONARY RECURSIVE ALGORITHM IN SELECTING STATISTICAL SUBSET NEURAL NETWORK/VDL FILTERING

ANDREW H. CHEN, JACK H. PENM, AND R. D. TERRELL

Received 26 June 2005; Accepted 4 October 2005

We propose an evolutionary recursive algorithm, for the exact windowed case, to estimate subset vector discrete lag (SVDL) filters with a forgetting factor and an intercept variable. SVDL filtering is demonstrated as a basis for constructing a multi-layered polynomial neural network by Penm et al. (2000) The new proposed time update recursions allow users to update SVDL filters at consecutive time instants, and can show evolutionary changes detected in filter structures. With this new approach we are able to more effectively analyse complex relationships where the relevant financial time series have been generated from structures subject to evolutionary changes in their environment. An illustration of these procedures is presented to examine the integration between the Australian and the Japanese bond markets, and the USA and the UK bond markets, changed over the period. The proposed algorithms are also applicable to full-order vector discrete lag (VDL) filtering with a forgetting factor and an intercept.

Copyright © 2006 Andrew H. Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Statistical filter researchers for financial time-series systems are often concerned that the coefficients of their established filters may not be constant over time, but vary when the filters are disturbed by changes arising from outside environmental factors. This concern has motivated researchers to develop sequential estimation algorithms that allow users to update subset time series filters at consecutive time instants, and allow for the coefficients to slowly evolve, and then can show evolutionary changes detected in filter structures. Hannan and Deistler [4] propose a recursive estimation of an autoregressive (AR) filter. Azimi-Sadjadi et al. [1] suggest a full-order updating procedure for the training process of a multiplayer neural network.

These studies utilise the fixed forgetting factor (henceforth called the forgetting factor) in the filtering and simulations of nonstationary time series. The forgetting factor as

2 Subset neural network/VDL filtering

described in Hannan and Deistler [4] has been widely adapted to account for nonstationarity of time series, and it is suitable to capture nonstationarity for a filter in which the underlying relationships between the variables involved change smoothly and gradually. In order to emphasise the importance of using more recent observations in filtering, the forgetting factor allocates greater weight to more recent estimated residuals and “forgets” some of the past. When the recursions are implemented in the stationary situation, the value of the forgetting factor is set to one.

The use of vector discrete lag (VDL) filtering in financial time series is versatile. As shown in Penm et al. [6], a VDL filter has been demonstrated as a basis for constructing a multi-layered polynomial neural network. Further, Holmes and Hutton [5] suggest the use of a SVDL system to assess the relationship between $z(t)$ and the set of current and lagged $y(t)$, where there is a continuous or random delay.

In order to increase the filtering and analysis power of neural networks to be applied to a time series system, Penm et al. [6] introduce inhibitor arc and switchable connection to the neural network structure. The inhibitor arc was introduced to network theory by Petri [9], and the associated connection strength for all these arcs is constrained to zero at all times. The switchable connection is obtained from switching theory, and the strength is switchable between zero and non-zero at any time. To incorporate such powerful connections the commonly used estimation algorithms for full-order filters are not applicable, because the structure of the lag coefficients is estimated without the “presence and absence” restrictions. As a result, it is necessary to develop an estimation algorithm for SVDL filters which includes full-order filters as a special case.

While there are well developed time update full-order VDL filtering algorithms [see Carayannis et al. [3]], these are not applicable to VDL filters with a subset structure. This is because the “optimal” subset filter at time instant t may become “suboptimal” at time instant $t + i$, $i = 1, \dots, n$. If one simply sets zero values to the coefficients of the missing lags, and then applies the time update algorithms for the full-order case, this leads to a loss of efficiency in the filter performance, as the subset structure of the filter is not updated accordingly. Consequently, an efficient algorithm needs to be developed for the time update SVDL filters, including full-order filters obtained as special cases.

In this paper a forward time update algorithm has been developed for the exact windowed case, to recursively estimate the SVDL filtering with a forgetting factor and an intercept variable. Compared to the residual-based algorithm proposed in Penm et al. [6], the current algorithm utilises only the available observations without any assumption on unseen observations to estimate filter coefficients. In addition, the current algorithm is a coefficient-based time update algorithm, which can detect evolutionary changes in filter structures. However the proposed algorithm in Penm et al. [6] is a residual-based order update algorithm, which undertakes recursions, moving from low-order filters to high-order filters, so no evolutionary changes are captured through parameter updating. Therefore the focus of Penm et al. [6] is not on the evolution over time of the parametric structure of the system. We now are proposing forward time update algorithms because they allow users to update SVDL filters at consecutive time instants, and can show evolutionary changes detected in filter structures.

To demonstrate the effectiveness of the proposed time update algorithm, we apply the recursive extended two-layer neural network algorithm, equivalent to time update SVDL filtering, to examine the causal linkages between government bond markets of three leading industrialised countries, namely the United States, the United Kingdom and Japan, during the period from August 1994 through December 2004. The findings are helpful in explaining linkages between the government bond markets involved.

The remainder of the paper is organised as follows. In Section 2 we present the algorithm for recursively estimating SVDL filtering with a forgetting factor and an intercept variable. An empirical example to assess whether there is any changing integration between the Australian and the Japanese bond markets, and the USA and the UK bond markets is presented in Section 3. In Section 4 a summary is given.

2. New evolutionary recursive algorithms for SVDL filtering

In this section we introduce forward time-update recursions which recursively estimate an SVDL filter for the exact-windowed case.

In SVDL filtering, it is desirable to relate $z(t)$ to present and past data for $y(t)$. We consider an SVDL filter of the form

$$z(t) + \rho + \sum_{i=1}^p h'_i(I_s) y(t+1-i) = \varepsilon^h(t), \quad \{h'_i(I_s) = 0, \text{ as } i \in I_s\}, \quad (2.1)$$

where h'_i , $i = 1, 2, \dots, p$ are $g \times r$ parameter matrices, ρ is an intercept variable, $\varepsilon^h(t)$ is a $g \times 1$ stationary process with $E\{\varepsilon^h(t)\} = 0$ and

$$E\{\varepsilon^h(t)\varepsilon^{h'}(t-\tau)\} = \begin{cases} \Omega & \tau = 0 \\ 0 & \tau \neq 0. \end{cases} \quad (2.2)$$

Equation (2.1) and properties associated with $\varepsilon^h(t)$ together constitute a VDL filter, which involves a g -dimensional regressand vector $z(t)$ and an r -dimensional regressor vector $y(t)$. The order of the system is p , $1 \leq i_1 < i_2 < \dots < i_s \leq p-1$, and $y(t+1-i)$ is a deleted lag ($i \in I_s$).

Given two finite data sample sets, $\{z(n), \dots, z(T)\}$ and $\{y(n), \dots, y(T)\}$, it is necessary to sequentially estimate all possible SVDL filters from (2.1) using the exact-windowed case. Since the actual scheme of (2.1) may not be order p , the resulting estimates of h_i is denoted by $h_{p,n,T}(i)$, where T is the sample size under examination. Then the predictor of an SVDL system of (2.1) can be described as

$$\hat{z}(i) = -H'_{p,n,T}(I_s) Y_{p-1,i}(I_s), \quad i > T, \quad (2.3)$$

where $H_{p,n,T} = [h'_{p,n,T}(1), \rho'_{p,n,T}, h'_{p,n,T}(2), \dots, h'_{p,n,T}(p)]'$, and $Y_{p,i} = [y(i), 1_r, \dots, y(i-p)]'$. 1_r denotes an $r \times 1$ vector with all 1 entries. $H_{p,n,T}(I_s)$ is formed by removing $h'_{p,n,T}(i_1), \dots, h'_{p,n,T}(i_s)$ of $H_{p,n,T}$, and $Y_{p,i}(I_s)$ is formed by removing $y'(i+1-i_1), \dots, y'(i+1-i_s)$ of $Y_{p,i}$.

4 Subset neural network/VDL filtering

The residual vector for observation i is

$$\varepsilon_{p,n,i}^h(I_s) = z(i) + H'_{p,n,T}(I_s) Y_{p-1,i}(I_s), \quad n \leq i \leq T. \quad (2.4)$$

In reality, many time-series systems present complex non-stationary features and cannot be filtered by assuming that $y(t)$ and $x(t)$ are stationary. Thus, for a VDL filter fitted to these two sample sets using the exact windowed method of forming the sample covariance matrix, we have

$$R_{p-1,n,T}(I_s) H_{p,n,T}(I_s) = -r_{p-1,n,T}(I_s), \quad (2.5)$$

where

$$\begin{aligned} R_{p-1,n,T}(I_s) &= \sum_{i=p-1+n}^T \lambda^{T-i} Y_{p-1,i}(I_s) Y'_{p-1,i}(I_s) \\ r_{p-1,n,T}(I_s) &= \sum_{i=p-1+n}^T \lambda^{T-i} Y_{p-1,i}(I_s) z'(i), \quad \Omega_{p,T}(I_s) = \sum_{i=p-1+n}^T \lambda^{T-i} \varepsilon_{p,n,i}^h(I_s) \varepsilon'_{p,n,i}(I_s), \end{aligned} \quad (2.6)$$

where λ , $0 < \lambda \leq 1$, is the forgetting factor as described in Hannan and Deistler [4]. The forgetting factor is suitable for a filter in which the underlying relationships between the variables involved change smoothly and gradually. When $\lambda = 1$, the recursions are implemented in the stationary situation.

Further, the following relations at $t = T + 1$ have been established,

$$R_{p-1,n,T+1}(I_s) H_{p,n,T+1}(I_s) = -r_{p-1,n,T+1}(I_s), \quad (2.7)$$

where

$$\begin{aligned} R_{p-1,n,T+1}(I_s) &= \lambda R_{p-1,n,T}(I_s) + H_{p-1,n,T+1}(I_s) H'_{p-1,n,T+1}(I_s), \\ r_{p-1,n,T+1}(I_s) &= \lambda r_{p-1,n,T}(I_s) + Y_{p-1,T+1}(I_s) z(T+1). \end{aligned} \quad (2.8)$$

To develop time update recursions for SVDL filtering, we consider the forward VAR(p, I_s) filter with an intercept variable of the form

$$y(t) + \tau + \sum_{i=1}^p a_i(I_s) y(t-i) = \varepsilon(t), \quad a_i(I_s) = 0, \text{ as } i \in I_s, \quad (2.9)$$

where $\varepsilon(t)$ is an independent and identically distributed random process with

$$E\{\varepsilon(t)\} = 0, \quad E\{\varepsilon(t)\varepsilon'(t-k)\} = \begin{cases} V(I_s) & \text{as } k = 0 \\ 0 & \text{as } k \neq 0. \end{cases} \quad (2.10)$$

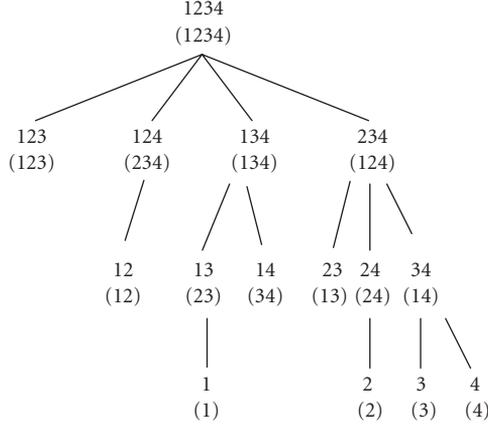


FIGURE 2.1. A four-variable tree diagram which illustrates the reciprocal integer pairs of subset VAR filters up to and including lag length $P = 4$. Of note, numerals denote particular lags in a forward VAR and numerals in italics denote such leads in a backward VAR.

We also consider a backward VAR(p) filter of the form

$$b_p(M_s)y(t) + \beta + \sum_{i=0}^{p-1} b_i(M_s)y(t-p+i) = \bar{\varepsilon}(t), \quad (2.11)$$

$$b_0(M_s) = I, \quad b_i(M_s) = 0, \quad \text{as } i \in M_s,$$

where $E\{\bar{\varepsilon}(t)\} = 0$, and the disturbance variance is $\bar{V}(M_s)$, M_s represents an integer set with elements m_1, m_2, \dots, m_s , $m_j = p - i_j$, $j = 1, 2, \dots, s$. A reciprocal integer pair for a forward subset VAR filter and a backward subset VAR filter is a pair of (2.9) and (2.11). Figure 2.1 shows a lag tree diagram which illustrates the reciprocal integer pairs of all subset VAR processes up to and including lag length, $k = 4$. Note that numerals represent particular lags in a forward VAR and underlined numerals represent such leads in a backward VAR.

We need to sequentially estimate all possible subset VAR filters from (2.9) and (2.11) using the exact-windowed case. Then we define observation i

$$\underline{Y}'_{p,i} = [I, y(i-1), \dots, y(i-p)],$$

$$A'_{p,n,T} = [\tau_{p,n,T}, a_{p,n,T}(1), \dots, a_{p,n,T}(p)], \quad B'_{p,n,T} = [b_{p,n,T}(p), \xi_{p,n,T}, \dots, b_{p,n,T}(1)]. \quad (2.12)$$

For a reciprocal integer pair of the forward VAR(p, I_s) and the backward VAR(p, M_s) filters fitted to this sample set, we have

$$R_{p,n,T}(I_s) = \sum_{i=p+n}^T \lambda^{T-i} Y_{p,i}(O_s) Y'_{p,i}(O_s), \quad (2.13)$$

6 Subset neural network/VDL filtering

where

$$Y_{p,i}(O_s) = \left[\frac{y(i)}{\underline{Y}_{p,i}(L_s)} \right] = \left[\frac{Y_{p-1,i}(O_s)}{y(i-p)} \right], \quad (2.14)$$

$$R_{p,n,T}(I_s) \left[\frac{1}{A_{p,n,T}(I_s)} \right] = \left[\frac{V_{p,n,T}(I_s)}{0} \right], \quad V_{p,n,T}(I_s) = \sum_{t=p+n}^T \lambda^{T-i} [\varepsilon_p(t) \varepsilon_p'(t)],$$

where O_s represents an integer set with elements o_j , $j = 1, \dots, s$, and $o_j = i_j + 2$. $Y_{p,i}(O_s)$ and $Y_{p-1,i}(O_s)$ are formed by removing the $(o_1), \dots, (o_s)$ 'th row of $Y_{p,i}$ and $Y_{p-1,i}$ respectively. L_s represents an integer set with elements l_j , and $l_j = i_j + 1$ and $\underline{Y}_{p,i}(L_s)$ is formed by removing the $(l_1), \dots, (l_s)$ 'th row of $\underline{Y}_{p,i}$. Also $A_{p,n,T}(I_s)$ is formed by removing $a_{p,n,T}(i_1), \dots, a_{p,n,T}(i_s)$ of $A_{p,n,T}$.

Now we define

$$D_{p,n,T}(I_s) = \sum_{i=p+n}^T \lambda^{T+1-i} \underline{Y}_{p,i}(L_s) \underline{Y}'_{p,i}(L_s), \quad K_{p,n,T}(I_s) = D_{p,n,T}^{-1}(I_s) \underline{Y}_{p,T+1}(L_s),$$

$$\tau_{p,n,T}(I_s) = 1 + \underline{Y}'_{p,T+1}(L_s) W_{p,n,T}(I_s), \quad e_{p,n,T+1}(I_s) = [I \mid A'_{p,n,T}(I_s)] \underline{Y}_{p,T+1}(L_s). \quad (2.15)$$

In addition, for the corresponding backward VAR(p, M_s), we will have

$$R_{p,n,T}(M_s) = \sum_{i=p+n}^T \lambda^{T-i} \left[\frac{Y_{p-1,i}(M_s)}{y(i-p)} \right] [Y'_{p-1,i}(M_s)],$$

$$\bar{V}_{p,n,T}(M_s) = \sum_{t=n+p}^T \lambda^{T-i} [\bar{\varepsilon}_p(t) \bar{\varepsilon}_p'(t)], \quad (2.16)$$

$$R_{p,n,T}(M_s) \left[\frac{B_{p,n,T}(M_s)}{1} \right] = \left[\frac{0}{\bar{V}_{p,n,T}(M_s)} \right],$$

where $B_{p,n,T}(M_s)$ and $Y_{p-1,i}(M_s)$ are formed by removing the $(p+2-m_1), \dots, (p+2-m_s)$ 'th row of $B_{p,n,T}$ and $Y_{p-1,i}$ respectively. We note $p+2-m_j = 2+i_j = o_j$, thus we can easily see that $Y_{p-1,i}(M_s) = Y_{p-1,i}(O_s)$ and $R_{p,n,T}(M_s) = R_{p,n,T}(I_s)$.

Next, we consider a forward VAR($p+1, I_s$) filter with a intercept variable of the form

$$y(t) + a_1(L_s)y(t-1) + \tau + \sum_{i=2}^{p+1} a_i(L_s)y(t-i) = \varepsilon(t), \quad a_i(L_s) = 0, \text{ as } i \in L_s, \quad (2.17)$$

where we have shifted the intercept variable to the third term of the filter for ease of matrix algebra operations. Suppose the filter is based on the sample set $\{y(n), y(n+1), \dots, y(T+1)\}$, we have

$$\begin{aligned} \underline{D}_{p+1,n,T+1}(L_s) &= \sum_{i=p+1+n}^{T+1} \lambda^{T+2-i} Y_{p,i-1}(O_s) Y'_{p,i-1}(O_s), \\ \tilde{K}_{p+1,n,T+1}(L_s) &= \underline{D}_{p+1,n,T+1}^{-1}(L_s) Y_{p,T+1}(O_s), \\ \tau_{p+1,n,T+1}(L_s) &= 1 + Y'_{p,T+1}(O_s) \tilde{K}_{p+1,n,T+1}(L_s). \end{aligned} \quad (2.18)$$

Again we consider a forward VAR(p, L_s) filter with intercept, that is,

$$y(t) + a_1(L_s)y(t-1) + \tau + \sum_{i=2}^p a_i(L_s)y(t-i) = \varepsilon(t), \quad a_i(L_s) = 0, \text{ as } i \in L_s, \quad (2.19)$$

where we keep the intercept variable in the third term to assist with our algebraic manipulations. Suppose the filter is fitted to the sample set $\{y(n+1), y(n+2), \dots, y(T+1)\}$, analogously we have

$$\begin{aligned} \tilde{K}_{p,n+1,T+1}(L_s) &= \underline{D}_{p,n+1,T+1}^{-1}(L_s) Y_{p-1,T+1}(O_s), \\ \tau_{p,n+1,T+1}(L_s) &= 1 + Y'_{p-1,T+1}(O_s) \tilde{K}_{p,n+1,T+1}(L_s). \end{aligned} \quad (2.20)$$

The matrix inversion of $R_{p,n,T}(I_s)$ provides

$$R_{p,n,T}^{-1}(I_s) = \begin{bmatrix} V_{p,n,T}^{-1}(I_s) & V_{p,n,T}^{-1}(I_s) A'_{p,n,T}(I_s) \\ A_{p,n,T}(I_s) V_{p,n,T}^{-1}(I_s) & D_{p,n,T}^{-1}(I_s) + A_{p,n,T}(I_s) V_{p,n,T}^{-1}(I_s) A'_{p,n,T}(I_s) \end{bmatrix}. \quad (2.21)$$

Right multiply with $\underline{Y}_{p,T+1}(L_s)$ on both sides of (2.21) and employ (2.12) and (2.15), so that we have

$$\tilde{K}_{p+1,n,T+1}(L_s) = \begin{bmatrix} 0 \\ K_{p,n,T}(I_s) \end{bmatrix} + \begin{bmatrix} I \\ A_{p,n,T}(I_s) \end{bmatrix} V_{p,n,T}^{-1}(I_s) e_{p,n,T+1}(I_s). \quad (2.22)$$

Again we right multiply the transpose of (2.22) with $\underline{Y}_{p,T+1}(L_s)$, employ (2.12), and add 1 to both sides, so that we now establish

$$\tau_{p+1,n,T+1}(L_s) = \tau_{p,n,T}(I_s) + e_{p,n,T+1}(I_s) V_{p,n,T}^{-1}(I_s) e'_{p,n,T+1}(I_s). \quad (2.23)$$

Analogously we can have

$$\begin{aligned} \tilde{K}_{p+1,n,T+1}(L_s) &= \begin{bmatrix} \tilde{K}_{p,n+1,T+1}(L_s) \\ 0 \end{bmatrix} + \begin{bmatrix} B_{p,n,T}(M_s) \\ I \end{bmatrix} \bar{V}_{p,n,T}^{-1}(M_s) \bar{e}_{p,n,T+1}(M_s), \\ \tau_{p+1,n,T+1}(L_s) &= \tau_{p,n+1,T+1}(L_s) + \bar{e}_{p,n,T+1}(M_s) \bar{V}_{p,n,T}^{-1}(M_s) \bar{e}_{p,n,T+1}(M_s), \end{aligned} \quad (2.24)$$

where $\bar{e}_{p,n,T+1}(M_s) = [B'_{p,n,T}(M_s) \mid I] Y_{p,T+1}(M_s)$.

8 Subset neural network/VDL filtering

From Penm et al. [7], if we permute the first row and the second row of the $\tilde{K}_{p+1,n,T+1}(I_s)$, the resulting vector is the $K_{p+1,n,T+1}(I_s)$ associated with a forward VAR($p+1, L_s$) filter of the form

$$y(t) + \delta + a_1(L_s)y(t-1) + \sum_{i=2}^{p+1} a_i(L_s)y(t-i) = \varepsilon(t), \quad a(i, L_s) = 0, \quad \text{as } i \in L_s, \quad (2.25)$$

which means that $K_{p+1,n,T+1}(I_s) = P_{p+1}\tilde{K}_{p+1,n,T+1}(I_s)$, where P_{p+1} is a permutation matrix of the form:

$$\begin{bmatrix} 0 & I & 0 & \vdots & 0 \\ I & 0 & 0 & \vdots & 0 \\ 0 & 0 & I & \vdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \vdots & I \end{bmatrix}. \quad (2.26)$$

Note that if there is a consecutive set of k deleted lags beginning at lag 1 in the forward VAR(p, I_s) filter fitted using the sample $\{y(n), \dots, y(T+1)\}$, we have

$$K_{p,n,T+1}(I_s) = K_{p-k,n,T+1-k}(I_k), \quad \tau_{p,n,T+1}(I_s) = \tau_{p-k,n,T+1-k}(I_k), \quad (2.27)$$

where I_s contains $i_1, \dots, i_k, \dots, i_s$, and $i_j = j$, $j = 1, 2, \dots, k$, and I_k contains i_{k+1}, \dots, i_{s-k} .

We now turn to the SVDL filtering. We use (2.6), (2.8) and (2.15), the following time update recursions for SVDL filtering can then be developed.

$$\begin{aligned} e_{p,n,T+1}^h(I_s) &= z(T+1) + H'_{p,n,T}(I_s)Y_{p-1,T+1}(I_s) \\ \varepsilon_{p,n,T+1}^h(I_s) &= e_{p,n,T+1}^h(I_s)\tau_{p,n,T+1}^{-1}(I_s) \\ H_{p,n,T+1}(I_s) &= H_{p,n,T}(I_s) - K_{p,n,T+1}(I_s)\varepsilon_{p,n,T+1}^h(I_s) \\ \Omega_{p,T+1}(I_s) &= \lambda\Omega_{p,T}(I_s) + \varepsilon_{p,n,T+1}^h(I_s)e_{p,n,T+1}^h(I_s). \end{aligned} \quad (2.28)$$

In addition, the forward-time update algorithm from T to $T+1$ is summarised in Algorithm 2.1.

To determine the optimal SVDL filter at each time instant, we utilise the filter selection criterion suggested by Hannan and Deistler [4]. From now on, we will use MHQC as an abbreviation for this criterion, which is defined by

$$\text{MHQC} = \log(\hat{\Omega}) + [2\log\log f(T)/f(T)]N, \quad (2.29)$$

where $f(T) = \sum_{t=p-1+n}^T \lambda^{T-t}$ is the effective sample size, and N the number of functionally independent parameters. The optimal filter selected has the minimum value of MHQC.

3. Applications

To demonstrate the effectiveness of the proposed recursive algorithm, we investigate whether the causal relationships between the Australian and Japanese bond markets and

$A_{p,n,T}(I_s)$, $V_{p,n,T}(I)$, $K_{p,n,T}(I_s)$, $\tau_{p,n,T}(I_s)$, $B_{p,n,T}(M_s)$, λ , $Y_{p,T+1}(O_s)$, $Y_{p-1,T+1}(I_s)$, $\tilde{V}_{p,n,T}(M_s)$, $H_{p,n,T}(I_s)$, $\Omega_{p,T}(I_s)$, $y(T+1)$ and $z(T+1)$ are available

Recursions:

$$e_{p,n,T+1}(I_s) = [1 : MA'_{p,n,T}(I_s)] Y_{p,T+1}(O_s)$$

$$\varepsilon_{p,n,T+1}(I_s) = e_{p,n,T+1}(I_s) \tau_{p,n,T}^{-1}(I_s)$$

$$A_{p,n,T+1}(I_s) = A_{p,n,T}(I_s) - K_{p,n,T}(I_s) \varepsilon'_{p,n,T+1}(I_s)$$

$$V_{p,n,T+1}(I_s) = \lambda V_{p,n,T}(I_s) + \varepsilon_{p,n,T+1}(I_s) \varepsilon'_{p,n,T+1}(I_s)$$

$$\tilde{K}_{p+1,n,T+1}(L_s) = \begin{bmatrix} 0 \\ K_{p,n,T}(I_s) \end{bmatrix} + \begin{bmatrix} 1 \\ A_{p,n,T}(I_s) \end{bmatrix} U_{p,n,T}^{-1}(I_s) e_{p,n,T+1}(I_s)$$

$$\text{partition } \tilde{K}_{p+1,n,T+1}(L_s) = [D/d]$$

$$\bar{e}_{p,n,T+1}(M_s) = \lambda \tilde{V}_{p,n,T}(M_s) d$$

$$\tilde{K}_{p,n+1,T+1}(L_s) = D - B_{p,n,T}(M_s) d$$

$$\tau_{p+1,n,T+1}(L_s) = \tau_{p,n,T}(I_s) + \varepsilon'_{p,n,T+1}(I_s) V_{p,n,T}^{-1}(I_s) e_{p,n,T+1}(I_s)$$

$$\tau_{p,n+1,T+1}(L_s) = \tau_{p+1,n,T+1}(L_s) - \bar{e}'_{p,n,T+1}(M_s) d$$

$$\bar{e}_{p,n,T+1}(M_s) = \bar{e}_{p,n,T+1}(M_s) [\tau_{p,n+1,T+1}(L_s)]^{-1}$$

$$\tilde{V}_{p,n,T+1}(M_s) = \lambda \tilde{V}_{p,n,T}(M_s) + \bar{e}_{p,n,T+1}(M_s) \bar{e}'_{p,n,T+1}(M_s)$$

$$B_{p,n,T+1}(M_s) = B_{p,n,T}(M_s) - \tilde{K}_{p,n+1,T+1}(L_s) \bar{e}'_{p,n,T+1}(M_s)$$

$$K_{p,n,T+1}(I_s) = P_p \tilde{K}_{p,n,T+1}(I_s)$$

$$e_{p,n,T+1}^h(I_s) = z(T+1) + H'_{p,n,T}(I_s) Y_{p-1,T+1}(I_s)$$

$$\varepsilon_{p,n,T+1}^h(I_s) = e_{p,n,T+1}^h(I_s) \tau_{p,n,T+1}^{-1}(I_s)$$

$$H_{p,n,T+1}(I_s) = H_{p,n,T}(I_s) - K_{p,n,T+1}(I_s) \varepsilon'^h_{p,n,T+1}(I_s)$$

$$\Omega_{p,T+1}(I_s) = \lambda \Omega_{p,T}(I_s) + \varepsilon_{p,n,T+1}^h(I_s) \varepsilon'^h_{p,n,T+1}(I_s)$$

ALGORITHM 2.1. The forward-time update recursions from T to $T+1$ for SVDL forgetting-factor inclusive filters with intercept variable.

the world's two major bond markets, those of the USA and the UK, changed over the period. The approach uses government bond indices from the Australian and the Japanese bond markets in a system that also include indices of the USA and the UK markets. The former two markets are considered to be representative of markets in the Asia Pacific Basin region, while the latter two markets are treated as a proxy of the world bond market. We undertake this research within the framework of SVDL, which provides a new approach to examining such relationships. The more traditional framework for examining these questions is through VAR. However VAR filtering does not explicitly detect evolutionary causal changes in the current and contemporary variable structures. It

only provides the relationships detected from the lagged filter structure. The current and contemporary causal relations become increasingly important in the efficient and competitive financial markets of the developed economies. Therefore the filter development should focus on VDL filtering.

The J. P. Morgan monthly bond price indices are sourced from Datastream over the period August 1994 through December 2004. Those data in US dollar terms are sampled from the government bond indices of Australia (AUS), Japan (JP), the UK (UK) and USA (USA). The sample size of each series is 125. To examine stationarity for each series, Microfit 4.0 is used to carry out the augmented Dickey-Fuller (ADF) unit root test. The 95 per cent critical values for each test computed using the response surface estimates, indicate that all series are $I(1)$.

The algorithm developed in Section 2 is used to assess the relationships between the AUS-JP pair and the UK-USA pair. In detecting the causal relationship from UK-USA to AUS-JP, the variables used are $z_1(t) = \log \text{AUS}$, $z_2(t) = \log \text{JP}$, $y_1(t) = \log \text{UK}$ and $y_2(t) = \log \text{USA}$. As discussed above, none of the logarithms of the four bond indices is stationary. Therefore, exponential forgetting is used with a forgetting factor, 0.99, to allow for the presence of non-stationarity. To begin it is assumed $P = 16$, which corresponds to a one and a quarter year period. The proposed evolutionary SVDL recursions described above are then applied to the logarithms of the data to select the “optimal” specification of the vector discrete lag filters. A SVDL filter with lags (0,8) is selected by the MHQC at $T = 120$ and 121. At $T = 122$, the lag structure selected changes to (0). Detailed filter specifications with zero constraints selected are reported in Table 3.1.

To check the adequacy of each optimal filter fit, the strategy suggested in Brailsford et al. [2] is used, with the proposed Penm and Terrell [8] algorithm applied to test each estimated residual vector series. The results in Table 3.1 support the hypothesis that each residual vector series is a white noise process. These optimal filters are then used as the benchmark filters for analysing the causal relationships. The analysis indicates that from $T = 120$ to 121, AUS is caused by the current and lagged UK; from $T = 122$ to 125, AUS is caused by the current UK only; and AUS is only caused by the current USA from $T = 120$ to 125. JP is only caused by the current UK and the current USA from $T = 120$ to 125. All these results occur when emphasis is placed on recent data. The findings show that an efficient market hypothesis does exist in the system under examination from $T = 122$. As time goes on, no arbitrage opportunity can be taken in those developed economies.

For the causal relationship from AUS-JP pair, to UK and USA, Table 3.2 shows the optimal discrete lag filters with $\lambda = 0.99$. These results strongly support the existence of current causal relationship between AUS and UK. No causality from JP to UK is detected, and no causality from either AUS or JP to US is identified. Also, the results in Table 3.2 support the hypothesis that each estimated vector residual series is a white noise process. The findings confirm that the US economy is the hub of the world economy. The UK bond market movements are not affected by the Japanese bond market movements. Since the Japanese government imposes a zero-interest rate policy to avoid deflation arising in the Japanese economy, no evidence of any causality from the Japanese bond market movements to the UK and US bond markets can be identified.

TABLE 3.1. The SVDL for the relationship, linking government bond indices from UK-USA pair to Australian and Japan by MHQC using the GLS procedure.

		$T = 120$		$T = 121$		$T = 122$			
τ	i, j	entry (standard error)		i, j	entry (s.e.)		i, j	entry (s.e.)	
0	1,1	-1.568(0.193)		1,1	-1.582(0.196)		1,1	-0.752(0.116)	
	1,2	2.011(0.281)		1,2	2.015(0.283)		1,2	1.623(0.312)	
	2,1	1.005(0.193)		2,1	1.001(0.182)		2,1	-0.1285(0.065)	
	2,2	-0.0725(0.036)		2,2	-0.0982(0.045)		2,2	0.450(0.193)	
8	1,2	0.432(0.208)		1, 2	0.448(0.213)				
ρ	ρ_1	11.535(1.201)		ρ_1	11.482(1.212)		ρ_1	8.871(1.158)	
	ρ_2	6.528(0.915)		ρ_2	6.242(0.906)		ρ_2	6.351(0.821)	
		$T = 123$		$T = 124$		$T = 125$			
τ	i, j	entry (s.e.)		i, j	entry (s.e.)		i, j	entry (s.e.)	
0	1,1	-0.765(0.123)		1,1	-0.776(0.128)		1,1	-0.788(0.142)	
	1,2	1.631(0.316)		1,2	1.636(0.316)		1, 2	1.639(0.318)	
	2,1	-0.1414(0.073)		2,1	-0.1521(0.078)		2,1	-0.1685(0.085)	
	2,2	0.458(0.196)		2,1	0.465(0.198)		2,2	0.468(0.201)	
ρ	ρ_1	8.843(1.212)		ρ_1	8.822(1.158)		ρ_1	8.773(1.203)	
	ρ_2	6.303(0.811)		ρ_2	6.281(0.812)		ρ_2	6.225(0.838)	

Variables: $z_1(t) = \log(\text{AUS})$, $z_2(t) = \log(\text{JP})$, $y_1(t) = \log(\text{UK})$, $y_2(t) = \log(\text{USA})$.

SVDL: $z(t) + \rho + \sum_{\tau=1}^p h'_\tau(I_s)y(t+1-\tau) = \varepsilon^h(t)$.

Non-zero (i, j) th entries in estimated coefficient matrices, h'_τ and ρ .

TABLE 3.2. The SVDL for the relationship, linking government bond indices from AUS-JP pair to UK and USA by MHQC using the GLS procedure.

		$T = 120$		$T = 121$		$T = 122$			
τ	i, j	entry (s.e.)		i, j	entry (s.e.)		i, j	entry (s.e.)	
0	1,1	-0.218(0.046)		1,1	-0.223(0.051)		1,1	-0.235(0.048)	
ρ	ρ_1	3.583(0.203)		ρ_1	3.566 (0.211)		ρ_1	3.532(0.221)	
	ρ_2	4.728(0.391)		ρ_2	4.725(0.388)		ρ_2	4.723(0.393)	
		$T = 123$		$T = 124$		$T = 125$			
τ	i, j	entry (s.e.)		i, j	entry (s.e.)		i, j	entry (s.e.)	
0	1,1	-0.243(0.044)		1,1	-0.246(0.042)		1,1	-0.255(0.043)	
ρ	ρ_1	3.495(0.216)		ρ_1	3.462(0.215)		ρ_1	3.425(0.212)	
	ρ_2	4.718(0.382)		ρ_2	4.716(0.395)		ρ_2	4.711(0.398)	

Variables: $z_1(t) = \log(\text{UK})$, $z_2(t) = \log(\text{USA})$, $y_1(t) = \log(\text{AUS})$, $y_2(t) = \log(\text{JP})$.

SVDL: $z(t) + \rho + \sum_{\tau=1}^p h'_\tau(I_s)y(t+1-\tau) = \varepsilon^h(t)$.

Non-zero (i, j) th entries in estimated coefficient matrices, h'_τ and ρ .

4. Conclusion

In this paper a revolutionary recursive algorithm, using the exact windowed case, has been developed to sequentially select the best specification for a statistical subset neural

network/VDL filter. The proposed construction method is simple to use, avoids cumbersome matrix inversion, and results in greater numerical accuracy. The application investigates the causal relationships between the Australian and the Japanese bond markets, and the USA and the UK bond markets. This application demonstrates the effectiveness of the proposed recursions and widens the possible use of the revolutionary recursive algorithm in financial filtering and simulations.

References

- [1] M. R. Azimi-Sadjadi, S. Sheedvash, and F. O. Trujillo, *Recursive dynamic node creation in multi-layer neural networks*, IEEE Transactions on Neural Networks **4** (1993), no. 2, 242–256.
- [2] T. J. Brailsford, J. H. W. Penm, and R. D. Terrell, *The adjustment of the Yule-Walker relations in VAR modeling: the impact of the Euro on the Hong Kong stock market*, Multinational Finance Journal **5** (2001), no. 1, 35–58.
- [3] G. Carayannis, D. Manolakis, and N. Kalouptsidis, *A unified view of parametric processing algorithms for prewindowed signals*, Signal Processing **10** (1986), no. 4, 335–368.
- [4] E. J. Hannan and M. Deistler, *The Statistical Theory of Linear Systems*, John Wiley & Sons, New York, 1988.
- [5] J. M. Holmes and P. A. Hutton, *‘Optimal’ model selection when the true relationship is weak and occurs with a delay*, Economics Letters **30** (1989), no. 4, 333–339.
- [6] J. H. W. Penm, T. J. Brailsford, and R. D. Terrell, *A robust algorithm in sequentially selecting subset time series systems using neural networks*, Journal of Time Series Analysis **21** (2000), no. 4, 389–412.
- [7] J. H. W. Penm, J. H. Penm, and R. D. Terrell, *A note on the sequential fitting of multichannel subset autoregressions using the prewindowed case*, IEEE Transactions on Signal Processing **43** (1995), no. 1, 322–327.
- [8] J. H. W. Penm and R. D. Terrell, *Multivariate subset autoregressive modelling with zero constraints for detecting ‘overall causality’*, Journal of Econometrics **24** (1984), no. 3, 311–330.
- [9] C. A. Petri, *Introduction to general net theory*, Net Theory and Applications (Proceedings of Adv. Course General Net Theory Processes Systems, Hamburg, 1979), Lecture Notes in Computer Science, vol. 84, Springer, Berlin, 1980, pp. 1–19.

Andrew H. Chen: Edwin L. Cox School of Business, Southern Methodist University, USA
E-mail address: achen@mail.cox.smu.edu

Jack H. Penm: School of Finance and Applied Statistics, The Australian National University, Canberra, Australia
E-mail address: jack.penm@anu.edu.au

R. D. Terrell: National Graduate School of Management, The Australian National University, Canberra, Australia
E-mail address: deane.terrell@anu.edu.au