Prediction Based on a Multiscale Decomposition

O. Renaud
Faculté de Psychologie et Sciences de l'Education, Université de Genève
40, Bd du Pont d'Arve, 1211 Genève 4, Switzerland

J.-L. Starck
DAPNIA/SEI-SAP, CEA-Saclay, 91191 Gif sur Yvette, France

F. Murtagh
School of Computer Science, Queen's University Belfast
Belfast BT7 1NN, Northern Ireland

Corresponding author: O. Renaud, `Olivier.Renaud@pse.unige.ch`

Abstract

A wavelet-based forecasting method for time series is introduced. It is based on a multiple resolution decomposition of the signal, using the redundant "à trous" wavelet transform which has the advantage of being shift-invariant.

The result is a decomposition of the signal into a range of frequency scales. The prediction is based on a small number of coefficients on each of these scales. In its simplest form it is a linear prediction based on a wavelet transform of the signal. This method uses sparse modelling, but can be based on coefficients that are summaries or characteristics of large parts of the signal. The lower level of the decomposition can capture the long-range dependencies with only a few coefficients, while the higher levels capture the usual short-term dependencies.

We show the convergence of the method towards the optimal prediction in the autoregressive case. The method works well, as shown in simulation studies, and studies involving financial data.

Keywords

Wavelet transform, forecasting, resolution, scale, autoregression, time series, model

1 Introduction

The wavelet transform has been proposed for time series analysis in many papers in recent years. Much of this work has focused on periodogram or scalogram analysis of periodicities and cycles. For financial time series prediction, Bjorn (1995), Moody and Lizhong (1997) and Soltani et al. (2000) discussed the use of the wavelet transform in the case where the market can be modelled by a fractional Brownian motion (fBm), a $1/f$ fractal process, which implies the presence of correlations across time. Wavelets would appear to be very appropriate for analyzing non-stationary signals (Swee and Elangovan, 1999), and a link between wavelets and the difference operator was made in Xizheng et al. (1999).

Several approaches have been proposed for time-series filtering and prediction by the wavelet transform, based on a neural network (Zheng et al., 1999; Bashir and El-Hawary, 2000), Kalman filtering (Cristi and Tummula, 2000; Hong et al., 1998), or an AR (autoregressive) model (Soltani et al., 2000). In Zheng et al. (1999) and Soltani et al. (2000), the undecimated Haar transform was used. This choice of the Haar transform was motivated by the fact that the wavelet coefficients are calculated only from data obtained previously in time, and the choice of an undecimated wavelet transform avoids aliasing problems. See also Daoudi et al. (1999) which relates the wavelet transform to a multiscale autoregressive type of transform.

Section 2 presents the à trous Haar wavelet transform, and section 3 shows how this transform is well-suited to design a Multiresolution AR model (MAR). A set of experiments illustrating the method is discussed in section 5.

2 Wavelets and Prediction

The continuous wavelet transform of a continuous function produces a continuum of scales as output. Input data, however, is usually discretely sampled, and furthermore a "dyadic" or two-fold relationship between resolution scales is both practical and adequate. The latter two issues lead to the discrete wavelet transform.

The output of a discrete wavelet transform can take various forms. Traditionally, a triangle (or pyramid in the case of 2-dimensional images) is often used to represent all that we have to consider in the sequence of resolution scales. Such a triangle comes about as a result of "decimation" or the retaining of one sample out of every two. The major advantage of decimation is that just enough information is kept to allow exact reconstruction of the input data. See for instance Chui (1992), Daubechies (1992), Mallat and Falzon (1998) and Vidakovic (1999) for more details about the wavelet transform. Therefore decimation is ideal for an application such as compression. A major disadvantage of the decimated form of output is that we cannot simply – visually or graphically – relate information at a given time point at the different scales. With somewhat greater difficulty, however, this goal is possible. What is not possible is to have shift invariance. This means that if we had deleted the first few values of our input time series, then the output wavelet transformed, and decimated, data would not be the same as heretofore. We can get around this problem at the expense of a greater storage requirement, by means of a redundant or non-decimated wavelet transform.

A redundant transform based on an N-length input time series, then, has an N-length resolution scale for each of the resolution levels that we consider. It is easy, under these circumstances, to relate information at each resolution scale for the same time point. We do have shift invariance. Finally, the extra storage requirement is by no means excessive.

The à trous wavelet transform (Shensa, 1992; Dutilleux, 1987; Percival and Walden, 2000) decomposes a signal $X = (X_1, \ldots, X_N)$ as a superposition of the form

$$X_t = c_{J,t} + \sum_{j=1}^{J} w_{j,t}$$

where c_J is a coarse or smooth version of the original signal X and w_j represents "the details of X" at scale 2^{-j}. See Starck et al. (1998) for more information. Thus, the algorithm outputs $J+1$ subbands of size N. The indexing is such that, here, $j = 1$ corresponds to the finest scale (high frequencies).

The non-decimated Haar algorithm uses the simple filter $h = (\frac{1}{2}, \frac{1}{2})$. Consider the creation of the first wavelet resolution level. We derive it from the input data by convolving the latter with h. Then:

$$c_{j+1,t} = 0.5(c_{j,t-2^j} + c_{j,t}) \tag{1}$$

and

$$w_{j+1,t} = c_{j,t} - c_{j+1,t} \tag{2}$$

At any time point, t, we never use information after t in calculating the wavelet coefficient.

This algorithm is different from the invariant discrete wavelet transform (Coifman and Donoho, 1995), and from the implementation of the Maximal Overlap Discrete Wavelet Transform (MODWT) described in Cohen et al. (1997). It is similar to the implementation described in Percival and Walden (2000). It has the following advantages:

- It is simple to implement. The computational requirement is $O(N)$ per scale, and in practice the number of scales is set as a constant.

- Because we do not shift the signal, the wavelet coefficients at any scale j of the signal $(X_1, \ldots, X_t)$ are strictly equal to the first t wavelet coefficients at scale j of the signal $(X_1, \ldots, X_N)$ $(N > t)$.

This second point is very convenient in practice. For instance, if the data are regularly updated (i.e. we get new measurements), we do not have to recompute the wavelet of the full signal. Figure 1 shows which pixels of the input signal are used to calculate the last wavelet coefficient in the different scales. A wavelet coefficient at a position t is calculated from the signal samples at positions less than or equal to t, but never larger.

3 Linear and Non-Linear Multiscale Prediction

In this section, we use the above decomposition of the signal for prediction. Instead of using the vector of past observations $X = (X_1, \ldots, X_N)$ to predict X_{N+1}, we will use its wavelet transform.

The first point is to know how many and which wavelet coefficients will be used at each scale. A sparse representation of the information contained in the decomposition is the key.

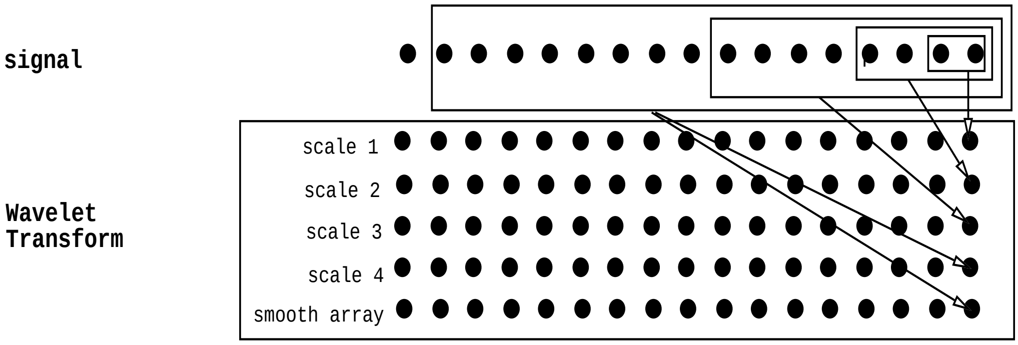

Figure 1: This figure shows which pixels of the input signal are used to calculate the last wavelet coefficient in the different scales.

After some simulations and for theoretical reasons that will become clear, the wavelet and scaling function coefficients that will be used for the prediction at time $N+1$ have the form $w_{j,N-2^j(k-1)}$ and $c_{J,N-2^J(k-1)}$ for positive value of k, as depicted in Figure 2. Note that for each N this subgroup of coefficients is part of an orthogonal transform.

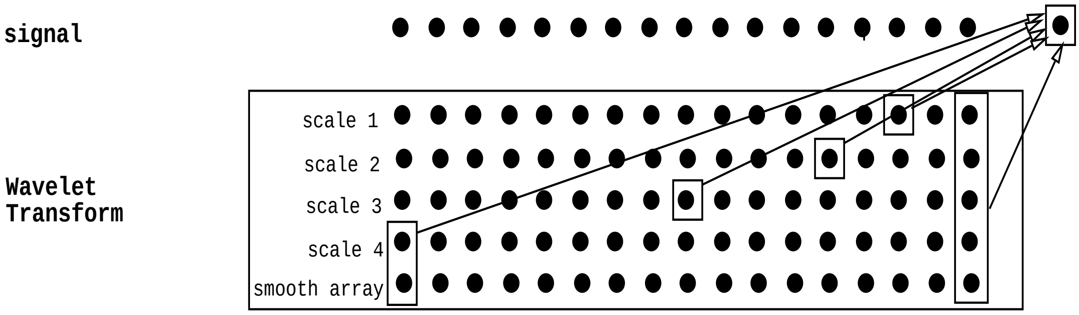

Figure 2: Wavelet coefficients that are used for the prediction of the next value.

3.1 Stationary Signal

Assume a stationary signal $X = (X_1, \ldots, X_N)$ and assume we want to predict X_{N+1}. The basic idea is to use the coefficients $w_{j,N-2^j(k-1)}$ for $k = 1, \ldots, A_j$ and $j = 1, \ldots, J$ and $c_{J,N-2^J(k-1)}$ for $k = 1, \ldots, A_{j+1}$ (see Figure 2) for this task. One example is to feed a neural network with these coefficients w and c as inputs, one or more hidden layer(s) and X_{N+1} as the output. With one hidden layer with P perceptrons, this writes as

$$\hat{X}_{N+1} = g_2 \left(\sum_{p=1}^{P} \hat{b}_p g_1 \left(\sum_{j=1}^{J} \sum_{k=1}^{A_j} \hat{a}_{j,k,p} w_{j,N-2^j(k-1)} + \sum_{k=1}^{A_{J+1}} \hat{a}_{J+1,k,p} c_{J,N-2^J(k-1)} \right) \right). \tag{3}$$

The prediction of financial future in Section 5 is based on this kind of model. We can imagine using virtually any type of prediction that use the previous data $X_N, \ldots, X_{N-q}$ and generalize through the use of the coefficients w and c instead. One of the simplest model for prediction beeing autoregressive, we show how to proceed in this case.

Recall that to minimise its mean square error, the one-step forward prediction of an AR(p) process is written $\hat{X}_{N+1} = \sum_{k=1}^{p} \hat{\phi}_k X_{N-(k-1)}$. Estimating $\hat{\phi}_k$ by MLE, by Yule-Walker, or by least squares, has the same asymptotic efficiency. In order to use the decomposition, we modify the prediction to the AR multiscale prediction:

$$\hat{X}_{N+1} = \sum_{j=1}^{J} \sum_{k=1}^{A_j} \hat{a}_{j,k} w_{j,N-2^j(k-1)} + \sum_{k=1}^{A_{J+1}} \hat{a}_{J+1,k} c_{J,N-2^J(k-1)} \tag{4}$$

where $\mathcal{W} = w_1, \ldots, w_J, c_J$ represents the Haar à trous wavelet transform of X ($X = \sum_{j=1}^{J} w_j + c_J$). For example, choosing $A_j = 1$ for all resolution levels, j, leads to the prediction

$$\hat{X}_{N+1} = \sum_{j=1}^{J} \hat{a}_j w_{j,N} + \hat{a}_{J+1} c_{J,N}. \tag{5}$$

Figure 2 shows which wavelet coefficients are used for the prediction using $A_j = 2$ for all resolution levels j, and a wavelet transform with five scales (four wavelet scales + the smoothed array). In this case, we can see that only ten coefficients are used, including coefficients that take into account low-resolution information. This means that a long-term prediction can easily be introduced, either by increasing the number of scales in the wavelet transform, or by increasing the AR order in the last scales, but with a very small additional number of parameters.

To further link this method with a prediction based on a regular AR, note that if on each scale the lagged coefficients follow an AR(A_j), the addition of the predictions on each level would lead to the same prediction formula (4).

This Multiresolution AR prediction model is actually linear. However, we can easily extend this to any model, linear or non-linear, that uses the coefficients $w_{j,N}$ and $c_{J,N}$ to predict the future signal. As an example, in the following section, a neural network with these coefficients as input and with X_{N+1} as output is discussed.

To estimate the $Q = \sum_{j=1}^{J+1} A_j$ unknown parameters grouped in a vector $\boldsymbol{\alpha}$, we solve the normal equations $A'A\boldsymbol{\alpha} = A'S$ that follow from least squares, with:

$$\begin{aligned}
A' &= (L_{N-1}, \ldots, L_{N-M}) \\
L'_t &= (w_{1,t}, \ldots, w_{1,t-2A_1}, \ldots, w_{2,t}, \ldots, w_{2,t-2^2A_2}, \ldots, w_{J,t}, \ldots, w_{J,t-2^JA_J}, c_{J,t}, \ldots, c_{J,t-2^JA_{J+1}}) \\
\boldsymbol{\alpha}' &= (a_{1,1}, \ldots, a_{1,A_1}, a_{2,1}, \ldots, a_{2,A_2}, \ldots, a_{J,1}, \ldots, a_{J,A_j}, \ldots, a_{J+1,1}, \ldots, a_{J+1,A_{J+1}}) \\
S' &= (X_N, \ldots, X_{t+1}, \ldots, X_{N-M+1})
\end{aligned}$$

Note that A is a $Q \times M$ matrix (M rows L_t, each with Q elements), $\boldsymbol{\alpha}$ and S are respectively Q- and M-size vectors, and Q is larger than M.

Interval prediction can also be obtained from the multiresolution decomposition by mimicking the method used on each scale. In the Multiscale AR case, following the prediction equation (4), and assuming Gaussian innovation, the $1 - \alpha$ confidence interval for X_{N+1} is given by $[\hat{X}_{N+1} \pm z(1 - \alpha/2)\hat{\sigma}]$, where z is the quantile of the standard normal distribution and $\hat{\sigma}$ is an estimation of the innovation standard deviation. One choice for $\hat{\sigma}$ is given by the square root of $(A\boldsymbol{\alpha} - S)'(A\boldsymbol{\alpha} - S)/(M - Q)$.

3.2 Signal with a Piecewise Smooth Trend

The previous prediction is valid for a zero mean signal. When a trend is present, several methods exist to remove the trend before conducting the analysis. By virtue of the multiscale decomposition, we can take advantage of the fact that the multiscale decomposition automatically separates the trend from the signal. We thus propose to predict both the trend and the stochastic part within the multiscale decomposition. The idea is that, in many instances, the trend affects the low frequency components, while the high frequencies may still be purely stochastic. Therefore we can separate our signal X into two parts, the low and the high frequencies L and H:

$$\begin{aligned} L &= c_J \\ H &= X - L = \sum_{j=1}^{J} w_j \\ X_{N+1} &= L_{N+1} + H_{N+1} \end{aligned}$$

The smoothed vector of the wavelet transform is first subtracted from the data, and we have now that the signal H is zero-mean. Our prediction will be the coaddition of two predicted values, one on the signal H by the AR Multiscale model, and the second on the low frequency component by another method. The AR-Multiscale model gives:

$$\hat{H}_{N+1} = \sum_{j=1}^{J} \sum_{k=1}^{A_j} a_{j,k} w_{j,N-2^j(k-1)} \tag{6}$$

The estimation of the $Q = \sum_{j=1}^{J} A_j$ unknown parameters proceeds as previously, except that the coefficients c are not used in L_i and that S is based on H_{t+1}.

Many methods may be used for the prediction of L_{N+1}. For a smooth trend, the problem is simplified by the fact that L is very smooth. We use polynomial fitting of degree 3 in our experiments.

The AR order at the different scales must now be defined. A global optimisation of all A_j parameters would be the ideal method, but is too computer intensive. However, by the relative non-overlapping frequencies used in each scale, we can consider selecting the parameters A_j independently on each scale. This can be done by standard methods, based on AIC, AICC or BIC methods (Shumway and Stoffer, 1999).

4 Convergence

The proposed method can be viewed as a generalisation principle that can be applied to virtually any time series model. For example, instead of fitting an AR model to the raw data, we propose to fit an AR model to each scale of the multiresolution transform. The following theorem shows that if the true process is actually AR, our forecasting procedure will converge to the optimal procedure and that it is even asymptotically equivalent to the best forecast.

Theorem 1 *Suppose that $\{X_t\}$ follows a causal AR process of order p with parameter $\boldsymbol{\phi}' = (\phi_1, \ldots, \phi_p)$, i.e. $X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \epsilon_t$, where $\{\epsilon_t\}$ are IID$(0, \sigma^2)$. If the selected* *orders A_j on each scale are greater than or equal to $p/2^j$ for $j = 1, \ldots, J$ and $A_{J+1} \geq p/2^J$, then the multiresolution model is such that, with increasing sample size, $\hat{\boldsymbol{\alpha}}$ has the following asymptotic property:*

$$N^{1/2}(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}) \Rightarrow \mathcal{N}(\mathbf{0}, \sigma^2(R'\mathcal{W}_B'\Gamma_B\mathcal{W}_BR)^{-1}),$$

where $\boldsymbol{\alpha}$, defined in Section 3 is equal to $\Omega\boldsymbol{\phi}$, $\Gamma_B = [\gamma(t-k)]_{t,k=1,\ldots,B}$ is the autocovariance matrix where $\gamma(l)$ is the autocovariance of the series at lag l. Related to the wavelet transform, $\Omega, R, \mathcal{W}_B$ and B are defined in the proof.

The parameter $\boldsymbol{\alpha}$ is the coefficient of the best linear predictor of X_{t+1} based on previous observations.

The proof may be found in the Appendix. This theorem shows that the estimator converges towards the parameter that allows the best prediction, if the underlying process is simply autoregressive. However, if this model is not true, only a few coefficients at the lower resolution scales can capture autocovariances at much higher lags, as shown by the Γ_B matrix resulting from this theorem. Note that B can be very large while the number of coefficients in the model stays reasonable.

5 Experiments

In this section, we will compare our methodology to different benchmark methods on simulated and real data. We will first use three different types of time dependencies. The first is a Gaussian white noise model with unit variance, where the signal at any time-point is independent of the past. The optimal prediction for this signal is the constant prediction. This experiment allows us to check whether a given method is able to avoid overfitting the empirical dependencies that will be present.

The second noise structure tested is a pure AR(4) noise, with parameter $\boldsymbol{\phi}'$ = (0.5, -0.5, -0.1, 0.3) and with a unit Gaussian innovation. The optimal prediction is of course a model with an AR(4) structure. For these two noise structures, we can compare our procedure with the optimal one.

The last noise structure is fractional ARIMA$(1, 0.49, 0)$, with parameter $\phi_1 = -0.5$ and with a unit Gaussian innovation. This type of signal is more difficult to predict for any method, due to its long-range dependency.

The simulation process runs as follows: For a given type of noise, we generate 50 signals of size 1000. For each signal, we use the 500 first points to estimate the model parameters, which are then kept fixed. We forecast the 500 last points one by one, based on all the previous ones, with the given estimate. The forecast values are then compared to the true values and the standard deviation of the prediction error is computed on the 500 differences.

Figure 3 shows the boxplots of the 50 standard deviations of the 50 samples. A boxplot gives in the centre the median value for the response and the box contains 50% of the responses, giving an idea of the variability between the samples. Hence the best method is the one for which the boxplot is closest as possible to 1, and as compact as possible. The different methods displayed are: the proposed method with 3 levels of wavelet coefficients and the low resolution scale (multires), a constant forecast (mean of the points, constant), an AR model, where the order is selected with a BIC selection criterion (AR BIC), and an AR(4) forecast (AR4). The

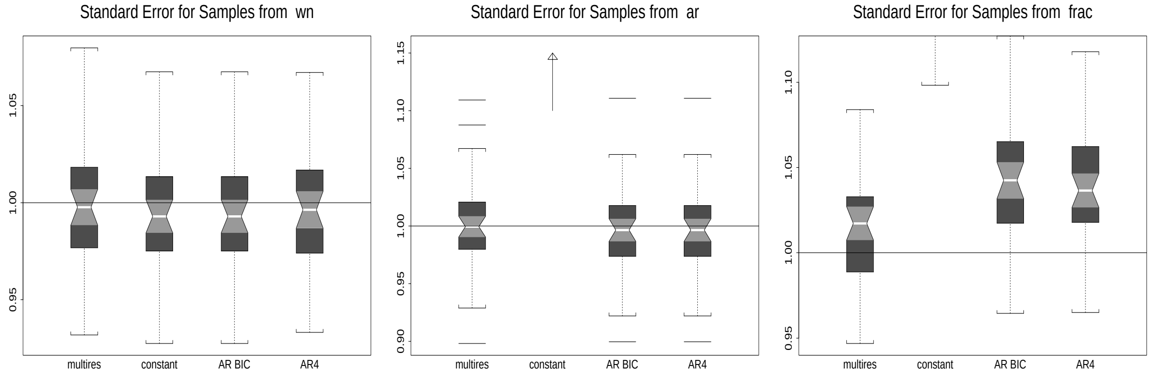

Figure 3: Left panel: boxplots of the error of prediction based on 500 predictions on 50 samples when the signal is pure white noise for 4 different methods: the proposed method (multires), a constant forecast (constant), an AR model, with a BIC selection criterion for the order (AR BIC) and an AR(4) forecast (AR4). Centre and right panels: same methods when the signals are AR(4) and fractional ARIMA.

left panel of Figure 3 is for samples that follow the white noise model, the centre panel is for the AR(4) model and the right panel is for the fractional ARIMA. The constant estimation is optimal for the white noise and the AR(4) estimation is optimal in the second case.

We see from the left panel of Figure 3 that the forecast capabilities of all methods are basically the same for the white noise model. No method is trapped in overfitting the possible sample dependencies. Both the multiresolution method and the AR-BIC model can select the number of parameters, and seem to have selected very few. Their predictions are merely slightly better than the AR(4) method that has to keep 4 parameters.

For the AR noise, both AR methods are merely slightly better than the multiresolution approach. This shows that although the latter has many parameters to choose from, it is able to downplay all unnecessary information and to use only a few important parameters to forecast these simple signals. As expected, the constant approach is inadequate.

Concerning the fractional noise, the proposed method is better than the two AR methods, while the constant approach is again inadequate. Note that both the AR-BIC and the proposed method have to select the number of parameters in their respective models. Due to a more flexible model, the latter seems to catch better the long-range particularities of the signal, and gives better forecasts. See McCoy and Walden (1996) for another view of the same phenomenon.

A close look at the average number of coefficients kept by the multiresolution method (not shown) reveals that it is adaptive to the nature of the signal: for the white noise and the AR(4) signals, few if any coefficients are kept in the lower scales while for the fractional noise, more coefficients are kept. This confirms that this method can adapt to the nature of the autocorrelation function.

Concerning interval prediction, we use the same data and the same methods to do our comparison. In each case, we compute how many times the true values X_{N+1} are within the interval $[\hat{X}_{N+1} \pm 2\hat{\sigma}]$. For the best method, this should be as close as possible to the actual level of confidence, which is 95.45. For the three different types of noise, the results are similar to the ones for the error of prediction. This shows again the robustness of the Multiresolution

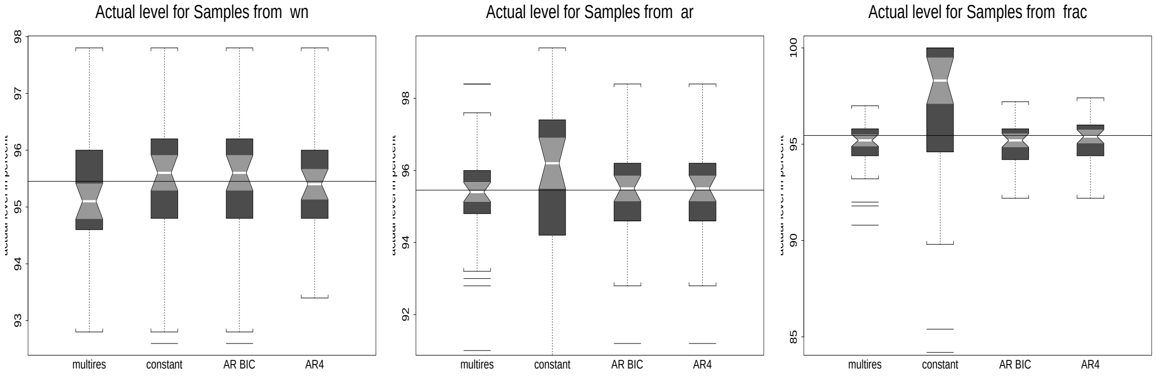

Figure 4: Accuracy of the interval prediction of the form $[\hat{X}_{N+1} \pm 2\hat{\sigma}]$ for the same data and the same methods as for Figure 3. The nominal level is 95.45 (horizontal line) and the best method is the one which boxplot of actual level is as close to this line and as compact as possible.

AR model to adapt to different noise types.

It is important to note that the performance of the proposed technique is only indicative and we have tested it with the simplest approach consisting of using an AR model for each scale. One can of course plug more advanced methods into it, over and above AR, that are more suitable for the signal at hand.

We now turn to two appraisals with real data. Yearly minimal water levels of the Nile River for the years 622 to 1281, measured at the Roda gauge near Cairo, provide a standard benchmark data set. These data are known to exhibit long-range dependence, and do not appear to have a trend. A wide range of parameters was used. The best AR model found, AR(2), gives an error standard deviation of 67.6497. A multilayer perceptron (MLP, back-propagation, 3 layers, 4 input units) was found to give an error standard deviation of 66.1845. An MAR(1) model, with 5 wavelet resolution scales, gave an error standard deviation of 64.9241. Figure 5 (top panels) show the Nile data (upper left) and the difference between known value and one-step ahead predictions on the rightmost half of the data (upper right).

Local area network traffic has been frequently used to exemplify long-memory processes. Assuming a similar process to hold for web access data, we used a set of 34,727 successive hourly numbers of bytes transferred from a web server. An AR(30) model gave a standard deviation error close to a very wide range of other AR models, 150595. An MLP based on an input window size of 6 values gave a standard deviation error of 148351. An MAR(2) model gave a standard deviation error of 149600. The latter, a little worse than the non-linear MLP and better than any non-multiresolution AR fit, took a few seconds computational effort, compared to many hours on a multiple processor machine for the MLP. Figure 5 (bottom panels) shows the web access data (lower left), with the ordinate rescaled for clarity, and with the final 2000 values of the data set used only. The lower right panel shows the difference between known value and one-step ahead predictions (again with ordinate rescaled for clarity).

In summary, these examples of time series exemplify lack of trend, lack of seasonality, and stationarity. We find MAR to be superior to an AR model in both cases. In the first case, MAR out-scores the non-linear MLP, while in the second case a non-linear model is found to be somewhat superior to MAR but at the cost of vastly increased computational requirements.

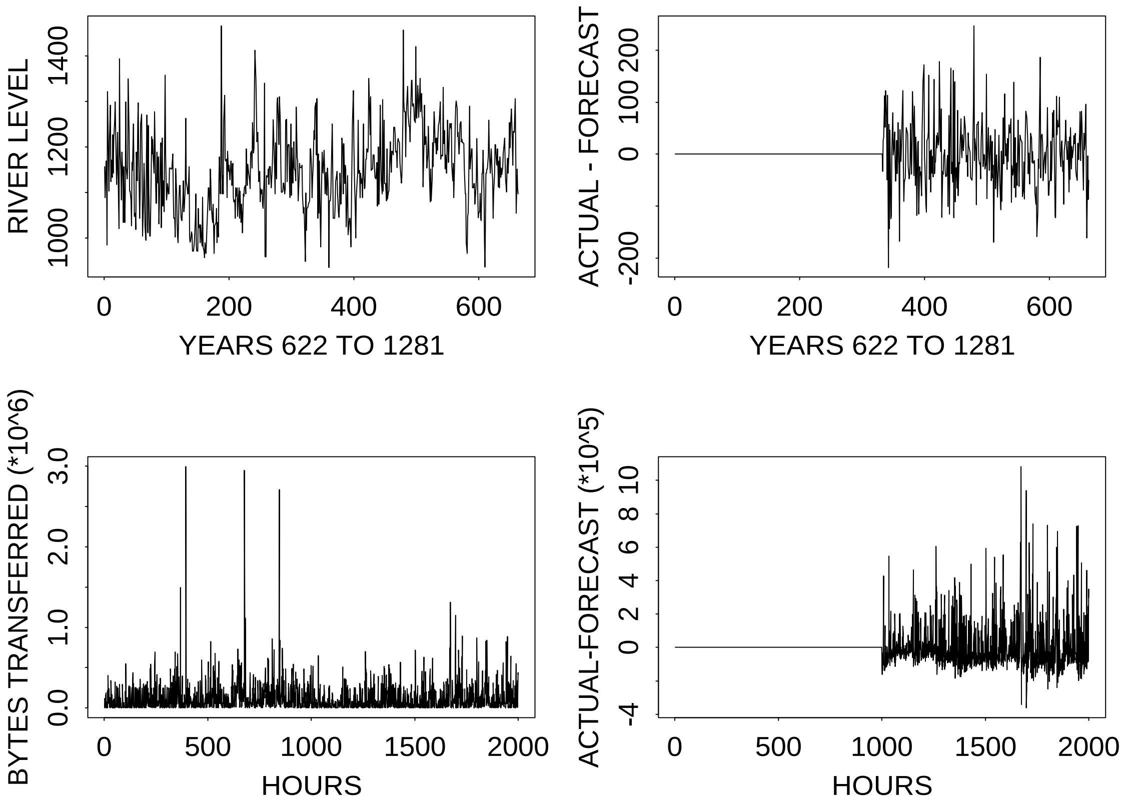

Figure 5: Top: Nile river level. Bottom: hourly rates of data access on a web site. Left panels: original data. Right panels: right parts show differences between target values and one-step ahead predictions.

For a set of financial futures data (daily highs, 6160 values), we used a non-multiresolution autoregressive AR(1) model to provide a baseline. An MAR(2), i.e. multiresolution AR(2), model provided better results. Respectively, standard deviation prediction error results of 13.96 and 13.90 were obtained. Using the final 1000 values, alone, provided a nearly identical result. Figure 6 illustrates the fit between predicted and target.

Using the first 4901 data values further improved this linear MAR(2) prediction to 11.32. Expecting to find even better results with nonlinear prediction based on the multiscale decomposition, we experimented with a range of neural network approaches. Surprisingly, we found worse results consistently across a range of neural network approaches. Backpropagation with different training algorithms gave rise to local optima with consequent problems in training. In these experiments we took a training set of size just over 1000 data values, and a test set of nearly 5000 (providing therefore a very demanding job for training; one reason for having a relatively small training set was to economize on training time). Consistent with an MAR(2) model, and also with Figure 2, the number of input data values was 10, and the number of output data values was 1. The Matlab Neural Network Toolbox was used. We next used a radial basis function network, and a generalized regression neural network. Standard deviation prediction errors of 82.78 and 71.93, respectively, were found for the test set. An

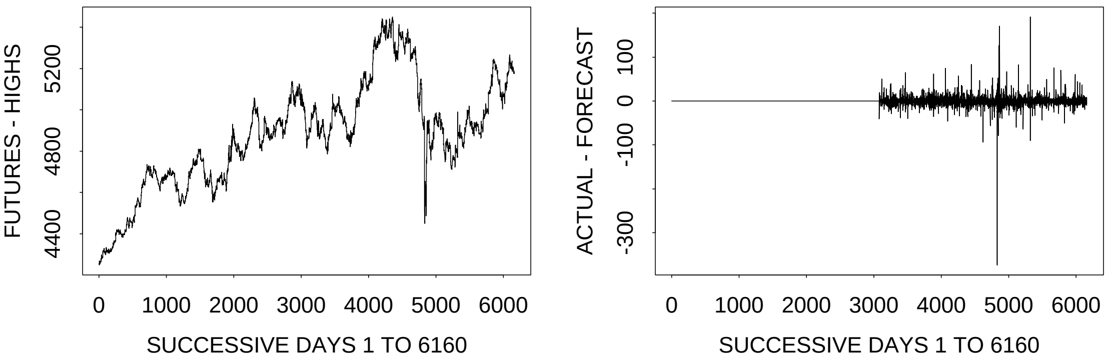

Figure 6: Financial futures, 6160 successive daily highs. Right: forecast minus actual for the second half of the data set.

explanation for these poor results was clear from a plot of output predicted values: they were relatively very flat and did not differ greatly from the mean data value. Note that this is not overtraining, which would imply excellent results on the training set, and poor generalization when used on the test set. To further explain these difficulties in training, we carried out a principal components analysis on the 10-dimensional input data. Variances and covariances were used, i.e. the data were centred. The principal component explained 97.27% of the variance. This was not surprising given the presence, without any normalization, of large-valued smooth coefficients and low-valued wavelet coefficients (cf. Figure 2). The multiple regression correlation was 0.993.

We conclude that for these financial futures, in the absence of input data normalization, using order 2, 4-band, multiresolution data (cf. Figure 2), a linear mapping proved far more stable than more sophisticated nonlinear alternatives.

Our discussion in terms of linearity and nonlinearity has been in terms of the mapping of inputs defined by wavelet coefficients vis à vis the output target value (cf. again Figure 2). We conclude with a remark on the linearity or nonlinearity of the mapping of a window of original time series data onto the output target value. Clearly in the case of the MAR model using the Haar à trous wavelet transform, our overall method is linear. But two distinct linear mappings are used. The wavelet transform is a particular set of moving averages and moving differences. One could envisage an alternative and perhaps nonlinear mapping here. The mapping from wavelet space to target predicted value is a Euclidean least squares mapping in the case of the MAR model. Our motivation is clear, viz. to avail of resolution scale information in our data signal. In the case of MAR Haar à trous the overall mapping is linear, but the least squares fit criterion is not one which can be simply expressed.

6 Conclusion

In this article, we propose a prediction method that is based on a time-frequency decomposition of the signal called the "à trous" wavelet transform. This very flexible procedure permits capturing of short-range as well as long-range dependencies with only a few parameters. In its simplest form this method is a generalisation of the standard AR method: instead of a linear prediction based on past values, we use a linear prediction based on some coefficients of the decomposition of the past values. Our proposal has been extended to neural networks and can easily be extended to generalise other sophisticated methods such as GARCH. The concept is very simple and easy to implement, while the potential is very significant.

Acknowledgements

The Matlab experiments were carried out by Dr X. Qiao, Queen's University Belfast.

A Proof of the Theorem

First note that the estimator can be written as $\hat{\boldsymbol{\alpha}} = (A'A)^{-1}A'S$. Define $T_p = (S_1, \ldots, S_p)$ where $S_t' = (X_{N-t}, \ldots, X_{N-M-t+1})$. Since the signal follows an AR(p) process, we can write $S = T_p\boldsymbol{\phi} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}' = (\epsilon_N, \ldots, \epsilon_{N-M+1})$ are the uncorrelated errors (innovations) of the series.

Let B be the smallest power of two that is larger than or equal to $2^j A_j$ for all $j = 1, \ldots, J$ and to $2^J A_{J+1}$. Let $\mathcal{W}_B$ be the $B \times B$ matrix of the orthogonal wavelet transform with the Haar basis with J wavelet scales. For any given time t, this represents the smallest orthogonal transform that contains all the coefficients in L_t.

The matrix Ω is a (usually small) part of $\mathcal{W}_B$. Define Π to be the first p columns of $\mathcal{W}_B$. At each time t, Π allows us to recover the p values $(X_t, \ldots, X_{t-p+1})$ from the B coefficients of the orthogonal transform. However, we do not need all B coefficients, and by the condition that $A_j \geq p/2^j$, the coefficients in L_t are sufficient (all other coefficients have only zeros in the Π matrix). We can therefore remove all rows of Π that do not correspond to elements in L_t to obtain the $Q \times p$ matrix Ω. We have that $(X_t, \ldots, X_{t-p+1})' = \Omega' L_t$ for all t. By inspection of A and T_p, the previous equation can be written as $T_p = A\Omega$. This Ω is also used to define the parameter $\boldsymbol{\alpha}$. We have

$$\begin{aligned} N^{1/2}(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}) &= N^{1/2}\left[(A'A)^{-1}A'(T_p\boldsymbol{\phi} + \boldsymbol{\epsilon}) - \boldsymbol{\alpha}\right] \\ &= N^{1/2}\left[(A'A)^{-1}A'(A\Omega\boldsymbol{\phi} + \boldsymbol{\epsilon}) - \Omega\boldsymbol{\phi}\right] \\ &= N(A'A)^{-1}(N^{-1/2}A'\boldsymbol{\epsilon}). \end{aligned}$$

Define $\boldsymbol{U}_t = \epsilon_t L_t$. Clearly,

$$N^{-1/2} A' \boldsymbol{\epsilon} = N^{-1/2} \sum_{t=1}^{N} \boldsymbol{U}_t.$$

Note that all the elements in L_t depend only on the observations prior to X_t and are therefore independent of ϵ_t. Then $E(\boldsymbol{U}_t) = 0$ and $E(\boldsymbol{U}_t \boldsymbol{U}_l') = 0$ for $t \neq l$. To compute the covariances of $\boldsymbol{U}_t$, we first define the $B \times Q$ matrix R that selects the rows of $\mathcal{W}_B$ corresponding to elements of L_t. It is composed of ones and zeros. Letting $\boldsymbol{X}_t' = (X_t, \ldots, X_{t-B+1})$, we have that $L_t' = \boldsymbol{X}_t' \mathcal{W}_B R$. Now,

$$\begin{aligned} E(\boldsymbol{U}_t \boldsymbol{U}_t') &= \sigma^2 E(L_t L_t') \\ &= \sigma^2 E(R' \mathcal{W}_B' \boldsymbol{X}_t' \boldsymbol{X}_t \mathcal{W}_B R) \\ &= \sigma^2 R' \mathcal{W}_B' \Gamma_B \mathcal{W}_B R. \end{aligned}$$

The proof of the asymptotic normality of the sum of $\boldsymbol{U}_t$ is the same as in the proof of Proposition 8.10.1 in Brockwell and Davis (1991), which implies that

$$N^{-1/2} A' \boldsymbol{\epsilon} \Rightarrow \mathcal{N}(\boldsymbol{0}, \sigma^2 R' \mathcal{W}_B' \Gamma_B \mathcal{W}_B R).$$

If we define T_B the same way as T_p but with B columns, we have that $A = T_B \mathcal{W}_B R$ and

$$\begin{aligned} N^{-1} A' A &= N^{-1} R' \mathcal{W}_B' T_B' T_B \mathcal{W}_B R \\ &\to R' \mathcal{W}_B' \Gamma_B \mathcal{W}_B R, \end{aligned}$$

in probability and thus $N(A'A)^{-1}$ converges in probability to $(R' \mathcal{W}_B' \Gamma_B \mathcal{W}_B R)^{-1}$. By one of Slutsky's theorems, we obtain the stated result. Finally, $\boldsymbol{\alpha}$ is the coefficient that leads to the same prediction as with the true model parameter $\boldsymbol{\phi}$.

References

Bashir, Z. and El-Hawary, M.E., Short term load forecasting by using wavelet neural networks, *Canadian Conference on Electrical and Computer Engineering*, (2000) 163–166.

Bjorn, V., Multiresolution methods for financial time series prediction, *Proceedings of the IEEE/IAFE 1995 Conference on Computational Intelligence for Financial Engineering*, **97** (1995).

Brockwell, P.J. and Davis, R.A. *Time Series: Theory and Methods* (Springer-Verlag, New York, 1991).

Chui, C.H. *Wavelet Analysis and Its Applications* (Academic Press, San Diego, 1992).

Cohen, I., Raz, S. and Malah, D., Orthonormal and shift-invariant wavelet packet decompositions, *Signal Processing* **57** (1997) 251–270.

Coifman, R.R. and Donoho, D.L., Translation invariant de-noising, In: A. Antoniadis and G. Oppenheim (Eds.), *Wavelets and Statistics*, (Springer-Verlag, New York, 1995) 125–150.

Cristi, R. and Tummula, M., Multirate, multiresolution, recursive Kalman filter, *Signal Processing* **80** (2000) 1945–1958.

Daoudi, K., Frakt, A.B. and Willsky, A.S., Multiscale autoregressive models and wavelets, *IEEE Transactions on Information Theory*, **45** (1999) 828–845.

Daubechies, I. *Ten Lectures on Wavelets* (Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1992).

Dutilleux, P., An implementation of the "algorithme à trous" to compute the wavelet transform, In J.M. Combes, A. Grossmann, and Ph. Tchamitchian (Eds.), *Wavelets: Time-Frequency Methods and Phase-Space*, (Springer-Verlag, New York, 1987).

Hong, L., Chen, G. and Chui, C.K., A filter-bank based Kalman filter technique for wavelet estimation and decomposition of random signals, *IEEE Transactions on Circuits and Systems – II Analog and Digital Signal Processing*, **45** (1998) 237–241.

Mallat, S. and Falzon, F., Analysis of low bit rate image transform coding, *IEEE Transactions on Signal Processing*, **46** (1998) 1027–42.

McCoy, E.J. and Walden, A.T., Wavelet analysis and synthesis of stationary long-memory processes, *Journal of Computational and Graphical Statistics*, **5** (1996) 26–56.

Moody, J. and Lizhong, W., What is the "true price"? State space models for high frequency FX data, *Proc. IEEE/IAFE 1997 Conference on Computational Intelligence for Financial Engineering (CIFEr)*, (1997) 150–156.

Percival, D.B. and Walden, A.T. *Wavelet Methods for Time Series Analysis* (Cambridge University Press, Cambridge, 2000).

Shensa, M.J., Discrete wavelet transforms: Wedding the à trous and Mallat algorithms, *IEEE Transactions on Signal Processing*, **40** (1992) 2464–2482.

Shumway, R.H. and Stoffer, D.S. *Time Series Analysis and Its Applications* (Springer-Verlag, New York, 1999).

Soltani, S., Boichu, D., Simard, P. and Canu, S., The long-term memory prediction by multiscale decomposition, *Signal Processing*, **80** (2000) 2195–2205.

Starck, J.L., Murtagh, F. and Bijaoui, A. *Image Processing and Data Analysis: The Multiscale Approach* (Cambridge University Press, Cambridge, 1998).

Swee, E.G.T. and Elangovan, S., Applications of symmlets for denoising and load forecasting, *Proceedings of the IEEE Signal Processing Workshop on Higher-Order Statistics*, (1999) 165–169.

Vidakovic, B. *Statistical Modeling by Wavelets.* (Wiley, New York, 1999).

Xizheng, K., Licheng, J., Tinggao, Y. and Zhensen, W., Wavelet model for the time scale, in: *Proceedings of the 1999 Joint Meeting of the European Frequency and Time Forum, 1999 and the IEEE International Frequency Control Symposium, 1999*, Vol. 1 (1999), 177–181.

Zheng, G., Starck, J.L., Campbell, J. and Murtagh, F., The wavelet transform for filtering financial data streams, *Journal of Computational Intelligence in Finance*, **7** (1999) 18–35.