Penalized Least Squares for Smoothing Financial Time Series

Adrian Letchford, Junbin Gao, and Lihong Zheng

School of Computing and Mathematics, Charles Sturt University, Australia

Abstract. Modeling of financial time series data by methods of artificial intelligence is difficult because of the extremely noisy nature of the data. A common and simple form of filter to reduce the noise originated in signal processing, the finite impulse response (FIR) filter. There are several of these noise reduction methods used throughout the financial instrument trading community. The major issue with these filters is the delay between the filtered data and the noisy data. This delay only increases as more noise reduction is desired. In the present marketplace, where investors are competing for quality and timely information, this delay can be a hindrance. This paper proposes a new FIR filter derived with the aim of maximizing the level of noise reduction and minimizing the delay. The model is modified from the old problem of time series graduation by penalized least squares. Comparison between five different methods has been done and experiment results have shown that our method is significantly superior to the alternatives in both delay and smoothness over short and middle range delay periods.

Keywords: Penalized least squares, Time series analysis, Financial analysis, Finite impulse response, Time series data mining.

1 Introduction

The presence of noise in time series data severely limits the ability to extract useful information [21]. Two types of noise have been identified, dynamical [8, 22] and measurement [8, 22, 24] noise. Dynamical noise is within the system and measurement noise is the result of less than perfect measurement instruments. Noise reduction is a broad term where the goal is to remove from a time series some unwanted component which itself is made up of noise [8].

Noise reduction of time series can be placed into four groups; graduation, prediction, filtering and smoothing. Graduation assumes that the signal has finished, thus allowing the use of all the data to be used to reduce the noise. This has been a very big area of research with models such as wavelets [2, 3], singular value decomposition [29, 34], empirical mode decomposition [1, 4], particle filters [10, 20], and singular spectrum analysis [12, 13]. Prediction involves estimating the future noise free values using old data. A very common and simple series of models for this purpose are the exponential smoothing models [9]. Filter models estimate the current noise free price using all available information. A famous

D. Wang and M. Reynolds (Eds.): AI 2011, LNAI 7106, pp. 72–81, 2011.
© Springer-Verlag Berlin Heidelberg 2011

filter, the Kalman filter [17], has been used since it's derivation in 1960. Smoothing models are identical to filters with the exception of an added delay [26], they use some future data to reduce the noise such as in [32]. The models provide more accurate estimates at the cost of using some future data (the delay).

For noise reduction of financial data, it would appear that smoothing models are the most ideal. They are calculated in real time as the financial data stream is received, and they provide the best estimate in comparison to filters or predictors. The problem with these models, however, is the obvious lag. For example, the smoothed value at time t reflects the correct smoothed value for time $t - l$, where l is the lag. With the reduction of lag comes reduction of smoothness.

It has been shown that perfectly reducing the noise of the streaming time series increases the performance of data mining and forecasting methods [19, 31]. Investors use various combinations of these filters to produce trading rules, based on the reduced level of noise, to assist with buy and sell decisions. A comparison of two types of filters for this purpose was performed by [7] while [15] optimized the rules with a particle swarm algorithm. This paper will be concerned with a form of filter that is in wide spread use for security price analysis, finite impulse response (FIR) filters – or more commonly known in the financial industry as moving averages. The current methods will be presented and a new method with theoretical basis will be proposed to address the issue of lag within the limitation of the FIR filter.

The rest of this paper is outlined as follows, Sect. 2 will show the variations on the finite impulse response filter that are used. Section 3 proposes a new model with a theoretical basis. Section 4 describes the experiment performed to compare the various models and the results are presented in Sect. 5. Finally, Sect. 6 discusses the conclusions.

2 Current Methods

Finite impulse response (FIR) filters in the financial literature are more commonly known as moving averages, they can be generalized as:

$$\hat{y}_t = \sum_{i=1}^{n} \alpha_i y_{t-n+i} \tag{1}$$

Where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_n]$ is the set of model coefficients. The number of coefficients is denoted n, otherwise known as the FIR window size. There are only a handful of different methods of selecting these coefficients. The **simple moving average** (SMA) [5] sets $\alpha_1 = \alpha_2 = \ldots = \alpha_n = 1/n$. Analysts changed the coefficients to increase the weight on the most recent data with the aim of reducing the lag. The **weighted moving average** (WMA) [5] is one of these changes which sets the vector $\boldsymbol{\alpha} = [1, 2, \ldots, n] \cdot [n(n+1)2^{-1}]^{-1}$. The **hull moving average** (HMA) [16] is a modification of the WMA, which has less lag. Given that $\mathbf{WMA}(y, n)$ is the WMA of series $\mathbf{y}$ with n coefficients, the HMA is calculated as:

$$\hat{\mathbf{y}} = \mathbf{WMA}(2 \cdot \mathbf{WMA}(\mathbf{y}, \frac{n}{2}) - \mathbf{WMA}(\mathbf{y}, n), \sqrt{n}) \tag{2}$$

A Gaussian implementation, where $\boldsymbol{\alpha}$ is selected from a Gaussian kernel [11], is commonly known as the **Arnaud Legoux moving average** (ALMA) [23] which uses an offset O:

$$\hat{y}_t = \frac{\sum_{i=0}^{n-1} K_\sigma(i - O) y_{t-i}}{\sum_{i=0}^{n-1} K_\sigma(i - O)}, K_\sigma(x) = e^{-\frac{x^2}{2\sigma^2}} \tag{3}$$

There are other methods for selecting the coefficient vector $\boldsymbol{\alpha}$, however, they are unsuitable for financial data. For example, the least mean squares filter [30] is an adaptive moving average, the coefficients change with time. To calculate $\boldsymbol{\alpha}$, one must first know the smoothed series, quite impossible in finance and economics. The FIR wiener filter [18] also requires knowledge of the smoothed series.

Each of these FIR designs aims to maintain a smooth output while attempting to reduce lag. The following section shows a derivation of the coefficient vector $\boldsymbol{\alpha}$ which is optimized to give the smoothest curve on a training data set after specifying the FIR window size.

3 Our Proposed Method

Our proposed method for real time noise reduction is based on the **penalized least squares** (PLS) graduation method [6, 14, 33]. The PLS method balances two conflicting attributes of the final curve: (1) the accuracy of the curve to the original series and (2) the smoothness of the curve. The accuracy is expressed in matrix notation with the normal least squares method $||\mathbf{y} - \hat{\mathbf{y}}||^2$. The smoothness can be measured with differencing where $\nabla \hat{y}_x = \hat{y}_x - \hat{y}_{x-1}$ and $\nabla^2 \hat{y}_x = \nabla(\nabla \hat{y}_x)$. The differencing can be expressed in matrix notation where D is a matrix such that $\mathbf{D}_d \hat{\mathbf{y}} = \nabla^d \hat{\mathbf{y}}$ where $d \in \mathbb{Z}$. For example, if the size of the $\mathbf{y}$ vector is 5 and $d = 1$ then:

$$\mathbf{D}_1 = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{4}$$

The problem is then expressed in least squares form as:

$$\mathbf{Q} = ||\mathbf{y} - \hat{\mathbf{y}}||^2 + \lambda ||\mathbf{D}_d \hat{\mathbf{y}}||^2 \tag{5}$$

Where λ is a smoothing factor. Differentiating both sides with respect to $\hat{\mathbf{y}}$ and setting to zero leads to the following solution where $\hat{\mathbf{y}}$ is a graduation of $\mathbf{y}$:

$$\hat{\mathbf{y}} = (\mathbf{I} + \lambda \mathbf{D}_d^T \mathbf{D}_d)^{-1} \mathbf{y} \tag{6}$$

Penalized least squares moving average (PLSMA) is the proposed model which modifies the PLS method to calculate optimal moving average coefficients. To change the problem to a moving average model the underlying time series needs to be represented in a trajectory matrix $\bar{\mathbf{y}}$ and the corresponding time series vector $\mathbf{y}$ needs to be adjusted to match. The trajectory matrix is calculated as follows, considering the time series $\mathbf{y} = [y_1, y_2, \ldots, y_N]$ let n be the number of coefficients in the model, then:

$$\bar{\mathbf{y}} = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \\ y_2 & y_3 & \cdots & y_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N-n+1} & y_{N-n+2} & \cdots & y_N \end{bmatrix} \tag{7}$$

While the corresponding time series vector $\mathbf{y}$ is the last column of $\bar{\mathbf{y}}$.

The model coefficients are represented in a column vector $\boldsymbol{\alpha}$, consistent with (1), and $\hat{\mathbf{y}}$ is then replaced by $\bar{\mathbf{y}}\boldsymbol{\alpha}$ in (5):

$$\mathbf{Q} = ||\mathbf{y} - \bar{\mathbf{y}}\boldsymbol{\alpha}||^{\mathbf{2}} + \lambda||\mathbf{D_d}\bar{\mathbf{y}}\boldsymbol{\alpha}||^{\mathbf{2}} \tag{8}$$

Differentiating both sides with respect to $\boldsymbol{\alpha}$ and setting to zero gives the solution:

$$\boldsymbol{\alpha} = [\bar{\mathbf{y}}^T\bar{\mathbf{y}} + \lambda(\mathbf{D_d}\bar{\mathbf{y}})^{\mathbf{T}}\mathbf{D_d}\bar{\mathbf{y}}]^{-\mathbf{1}}\bar{\mathbf{y}}^{\mathbf{T}}\mathbf{y} \tag{9}$$

Now, $\boldsymbol{\alpha}$ are FIR coefficients. While training data is needed to compute these coefficients, they can be used to smooth future data in an online fashion with increased smoothness (reduced lag) over the given data.

This raw method does come with some problems. (1) As λ increases the curve gets smoother until a point is reached where it cannot be any smoother and still remain on the same scale as $\mathbf{y}$. Then, $\hat{\mathbf{y}} \to 0$ as $\lambda \to \infty$. (2) As $\lambda \to \infty$ the matrix $\bar{\mathbf{y}}^T\bar{\mathbf{y}} + \lambda(\mathbf{D_d}\bar{\mathbf{y}})^{\mathbf{T}}\mathbf{D_d}\bar{\mathbf{y}}$ becomes singular – non-invertible. (3) most of the current filters have one or two inputs, this method has three inputs, FIR size, d, and λ.

The first problem is solved by normalizing $\boldsymbol{\alpha}$ by the sum of $\boldsymbol{\alpha}$. The second problem is rectified by noting that λ is used to change the proportion of the least squares equation by increasing the smoothness penalty. This ratio is maintained if the error part of the equation is multiplied by λ^{-1} and the smoothness penalty is left without a multiplier. Thus, (8) & (9) become:

$$\mathbf{Q} = \lambda^{-1}||\mathbf{y} - \bar{\mathbf{y}}\boldsymbol{\alpha}||^{\mathbf{2}} + ||\mathbf{D_d}\bar{\mathbf{y}}\boldsymbol{\alpha}||^{\mathbf{2}} \tag{10}$$

$$\boldsymbol{\alpha} = [\lambda^{-1}\bar{\mathbf{y}}^T\bar{\mathbf{y}} + (\mathbf{D_d}\bar{\mathbf{y}})^{\mathbf{T}}\mathbf{D_d}\bar{\mathbf{y}}]^{-\mathbf{1}}\lambda^{-\mathbf{1}}\bar{\mathbf{y}}^{\mathbf{T}}\mathbf{y} \tag{11}$$

Because of normalization, (11) can drop the second λ^{-1}:

$$\boldsymbol{\alpha} = [\lambda^{-1}\bar{\mathbf{y}}^T\bar{\mathbf{y}} + (\mathbf{D_d}\bar{\mathbf{y}})^{\mathbf{T}}\mathbf{D_d}\bar{\mathbf{y}}]^{-\mathbf{1}}\bar{\mathbf{y}}^{\mathbf{T}}\mathbf{y} \tag{12}$$

The third problem is overcome by noting that the goal is to achieve the greatest smoothing. Thus, λ ought to be maximized. Taking the limit:

$$\boldsymbol{\alpha} = \lim_{\lambda \to \infty} [\lambda^{-1} \bar{\mathbf{y}}^T \bar{\mathbf{y}} + (\mathbf{D_d} \bar{\mathbf{y}})^{\mathbf{T}} \mathbf{D_d} \bar{\mathbf{y}}]^{-\mathbf{1}} \bar{\mathbf{y}}^{\mathbf{T}} \mathbf{y} \tag{13}$$

$$= [(\mathbf{D_d} \bar{\mathbf{y}})^{\mathbf{T}} \mathbf{D_d} \bar{\mathbf{y}}]^{-1} \bar{\mathbf{y}}^{\mathbf{T}} \mathbf{y} \tag{14}$$

4 Experiment Description

The models were compared over several data sets with a cross validation method. To calculate the performance of each model two measures were developed. One for measuring how smooth the new time series is and the other to calculate how much lag it has. The rest of this section presents the details of this experiment and these statistical measures.

4.1 Data

Six real world time series were used for these experiments; AUD/USD, EUR/USD, GOOG, INDU, NASDAQ, and XAU/USD, all daily prices each around 2000 samples. Table 1 shows the range and description of each series. In addition, two randomly generated series were also used. Both have 2000 random prices with returns generated from the standard normal distribution.

Table 1. Names and description of the time series used in the experiment

Series Name	Range	Description
AUD/USD	20/10/2003 - 14/06/2011	Australian Dollar to U.S.A. Dollar
EUR/USD	15/10/2003 - 31/05/2011	Euro to the U.S.A. Dollar
GOOG	25/10/2004 - 14/06/2011	Stock for Google
INDU	21/07/2003 - 14/06/2011	Index for Dow Jones Industrial Average
NASDAQ	10/04/2003 - 14/06/2011	NASDAQ market index
XAU/USD	22/10/2003 - 14/06/2011	Gold to U.S.A. Dollar
Random 1		2000 random prices
Random 2		2000 random prices

4.2 Smoothness (Noise) Function

Previously, to calculate the level of noise reduction, the signal to noise ratio (SNR) would be used [27]. However, it assumes that the clean signal is known, and assumes that $\hat{\mathbf{y}}$ has no delay. In previous research, measures have been used which do not hold these assumptions. For example, autocorrelation and power spectrum are used in [25]. Unfortunately, these methods output the result in a large dimension resulting in comparison issues when processing thousands of comparisons. The measure for smoothness used here builds upon $||\mathbf{D_d}\hat{\mathbf{y}}||^{\mathbf{2}}$ used in the PLS equation. Some considerations are made, if $d = 1$, then the error stems from using the previous value of $\hat{\mathbf{y}}$, similarly, $d = 2$ is using the previous rate of change (ROC) to forecast. However, the ROC may be smooth, where

$d = 3$ would result in a smaller error. Thus, the smoothness of $\hat{\mathbf{y}}$ is the minimum of the following function with respect to d normalized by the smoothness of $\mathbf{y}$:

$$S(\hat{\mathbf{y}}) = min\left\{||\mathbf{D_d}\hat{\mathbf{y}}||\right\}, d \in \mathbb{N} \tag{15}$$

$$S(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{S(\hat{\mathbf{y}})}{S(\mathbf{y})} \tag{16}$$

Which can be interpreted as the percentage of noise filtered from the original series $\mathbf{y}$ to produce the smooth curve $\hat{\mathbf{y}}$. Unlike the SNR, the S function does not assume that the clean signal is known, and does not make assumptions about the lag.

Usually, noise is measured as an error between values such as in prediction problems or when using the SNR. However, as this paper is not dealing with estimating exact unknown quantities, this is redundant. Instead, the aim is the online reduction of noise in known noisy data. Thus, a natural conclusion would be to reduce the variance between values. As this would result in producing a straight line, instead of following the time series, the smoothness function extends this to reducing the variance at the best derivative level. As a result, small values for $S(\mathbf{y})$ means that $\mathbf{y}$ is smooth and takes on the form of a curve.

4.3 Lag Function

Cross correlation is adapted to calculate the lag between $\mathbf{y}$ and a given $\hat{\mathbf{y}}$. After calculating the smoothed series $\hat{\mathbf{y}}$ of a testing data, the entire training-testing window ($\mathbf{y}$) and $\hat{\mathbf{y}}$ are lined up by their right side. This is lag 0 and the correlation is calculated between $\hat{\mathbf{y}}$ and the adjacent values in $\mathbf{y}$. Then $\hat{\mathbf{y}}$ is shifted left by 1, corresponding to lag 1, and the correlation is again calculated. This process is continued and the lag with the highest correlation is taken to be the lag of the smoothed series.

4.4 Cross Validation

The five models in Sect. 2 and 3 were compared by using a cross validation method. The best window size for the training data was 800 and the testing data was 400. Due to the large size of this combined window (1,200), it was shifted by 100 rather than 400 to maintain enough sample optimizations.

The aim of the experiment was to find out which model has greater smoothing for a given amount of lag. There is no direct input for lag, however, as FIR filters, the lag is related to the size of the filter. Thus, the size (n) was iterated between 2-150 and the remaining variables were optimized at each iteration.

As has been shown, the SMA, WMA, and HMA have a single input, the FIR window size (n). As a result, these three models do not need optimization. The smoothed series is simply calculated over each testing set, and the smoothness and lag are averaged for each value of n.

For each $n \in [2-150]$, the ALMA is optimized 5 times over each set of training-testing data using $1 - S(\mathbf{y}, \hat{\mathbf{y}})$. The best parameters out of the 5 are chosen for that data set. The variables σ and O are optimized over the ranges 1 to 50 and -50 to 50 respectively. The optimization algorithm is simulated annealing, see [28]. Put simply, simulated annealing takes an initial starting point and “jitters” it around the error surface with a tendency to move around local minima. The jittering gradually comes to a halt where the point is expected to be in a local minima. The standard MatLab algorithm with default parameters was used.

The PLSMA model optimization is performed differently. The only parameter to be optimized is d and this is an integer. After a few trials of different FIR window sizes up to 150 it seemed that the optimal d did not go over 10. d was evaluated over the range [1-10] and the best value according to the smoothness measure was selected.

Figure 1 is the pseudocode of the cross validation algorithm.

```
foreach model
    foreach time series
        for n = 2-150
            foreach CV window
                Optimize model on training data
                Apply model to testing data
                calculate smoothness and lag
            Calculate average smoothness and lag over the CV windows
        Calculate average smoothness for each lag
```

Fig. 1. Pseudocode for the cross-validation algorithm

Once the average smoothness for each lag had been obtained for each of the models on each of the time series, summary statistics were compiled. The percentage of superior lags in comparison to the other models on each time series is calculated. The model with the highest percentage of superior lags is considered to be the best model.

5 Results

A clear indication of each model's performance is shown in Tbl. 2. The %Lags column shows the percentage of lags for which that model is superior, and the Range column shows the range of those lags. These comparisons were different for each model, as each model spans a different range in relation to the others. For example, the HMA only goes as far as 17 delay periods on the AUD/USD data. Thus, these figures are for the comparable range of each model on each data set.

The SMA is the worst model with no suitable lag periods except for the XAU/USD series where it is superior for lag 1. The WMA falls next being only

Table 2. Percent improvement, lag range, and algorithm complexity

	SMA		WMA		HMA		ALMA		PLSMA	
	%Lags	Range	%Lags	Range	%Lags	Range	%Lags	Range	%Lags	Range
Random 1	0.0%	[]	3.0%	[1]	5.3%	[2]	31.8%	[31-44]	58.3%	[3-30]
Random 2	0.0%	[]	2.4%	[1]	11.1%	[2-3]	24.0%	[39-50]	57.4%	[4-38]
INDU	0.0%	[]	2.6%	[1]	11.8%	[2-3]	28.0%	[37-50]	49.3%	[5-36]
AUD/USD	0.0%	[]	2.4%	[1]	11.8%	[2-3]	30.0%	[36-50]	50.8%	[4-35]
EUR/USD	0.0%	[]	0.0%	[]	23.5%	[1-4]	28.0%	[37-50]	47.1%	[5-36]
GOOG	0.0%	[]	0.0%	[]	11.8%	[2-3]	24.0%	[39-50]	48.6%	[1, 4-38]
NASDAQ	0.0%	[]	2.4%	[1]	11.8%	[2-3]	26.0%	[38-50]	47.2%	[4-37]
XAU/USD	2.1%	[1]	0.0%	[]	11.1%	[2-3]	19.0%	[35-42]	66.0%	[4-34]
Average	**0.3%**		**1.6%**		**12.3%**		**26.4%**		**53.1%**	
Complexity	$O(n)$		$O(n)$		$O(n^2)$		$O(n)$		$O(n^3)$	

superior on average by lag 1. The HMA is approximately on the range 2-3. The ALMA takes a much wider range of about 37-48 lag periods. The PLSMA model (our proposed model) is shown to be the best smoother. Being the most smoothest model for 48.6%+ of the lag periods. It appears that the PLSMA is superior over short to middle term lag periods of about 4-36 while the ALMA smoother is best for longer term lag periods.

Once the FIR coefficients for each model has been calculated, applying the filter to the financial data stream is of $O(n)$ complexity. However, the models do have varying degrees of complexity for the calculation of the FIR coefficients. PLSMA excluded, the best model is the ALMA which is of complexity $O(n)$. The improvement that PLSMA brings comes at a complexity cost, with the model sitting at $O(n^3)$. However, this is not a setback in online applications as the FIR coefficients are calculated offline. The complexity for each model is shown in Tbl 2.

6 Conclusions

In this paper, we have shown some of the different FIR filters used by investors to smooth security prices. It is noted that the output of a FIR filter is delayed with respect to the underlying time series. In addition, there is a positive relationship between the smoothness of the resulting curve and the lag which is undesirable. A method was proposed to derive an impulse response which maximizes the smoothness and minimizes the delay. As there is no assurance of optimality over any future data the filter may be applied to, this model was compared against five common models with a cross validation process. It was discovered that the proposed model achieves greater overall smoothing, more specifically for the short to middle range lag periods. While the very short term (4 or less periods) and longer term (37+ periods) were dominated by other models.

Future research will expand the analysis in this paper to include noise reduction models that are not otherwise used for financial pre-processing. Further experiments will also be conducted to discover the level of improvement for data mining and forecasting algorithms as previous research implies.

References

1. Boudraa, A.O., Cexus, J.C.: EMD-based signal filtering. IEEE Transactions on Instrumentation and Measurement 56(6), 2196–2202 (2007)
2. Donoho, D.L., Johnstone, I.M.: Adapting to unknown smoothness via wavelet shrinkage. Journal of the American Statistical Association 90(432), 1200–1224 (1995)
3. Donoho, D.L., Johnstone, I.M.: Minimax estimation via wavelet shrinkage. The Annals of Statistics 26(3), 879–921 (1998)
4. Drakakis, K.: Empirical mode decomposition of financial data. International Mathematical Forum 3(25), 1191–1202 (2008)
5. Ehlers, J.F.: Rocket science for traders: Digital signal processing applications. John Wiley & Sons, Inc., New York (2001)
6. Eilers, P.H.C.: A perfect smoother. Analytical Chemistry 75(14), 3631–3636 (2003)
7. Ellis, C.A., Parbery, S.A.: Is smarter better? a comparison of adaptive, and simple moving average trading strategies. Research in International Business and Finance 19(3), 399–411 (2005)
8. Farmer, D.J., Sidorowich, J.J.: Optimal shadowing and noise reduction. Physica D: Nonlinear Phenomena 47(3), 373–392 (1991)
9. Gardner, J.E.S.: Exponential smoothing: The state of the art–part II. International Journal of Forecasting 22(4), 637–666 (2006)
10. Godsill, S.J., Doucet, A., West, M.: Monte Carlo smoothing for nonlinear time series. Journal of the American Statistical Association 99(465), 156–168 (2004)
11. Hale, D.: Recursive gaussian filters. Tech. rep., Center for Wave Phenomena (2006)
12. Hassani, H.: Singular spectrum analysis: Methodology and comparison. Journal of Data Sciences 5, 239–257 (2007), mPRA Paper
13. Hassani, H., Soofi, A.S., Zhigljavsky, A.A.: Predicting daily exchange rate with singular spectrum analysis. Nonlinear Analysis: Real World Applications 11(3), 2023–2034 (2010)
14. Hodrick, R.J., Prescott, E.C.: Postwar u.s. business cycles: An empirical investigation. Journal of Money, Credit and Banking 29(1), 1–16 (1997)
15. Hsu, L.-Y., Horng, S.-J., He, M., Fan, P., Kao, T.-W., Khan, M.K., Run, R.-S., Lai, J.-L., Chen, R.-J.: Mutual funds trading strategy based on particle swarm optimization. Expert Systems with Applications 38(6), 7582–7602 (2011)
16. Hull, A.: Hull moving average HMA (2011), `http://www.justdata.com.au/Journals/AlanHull/hull_ma.htm`
17. Kalman, R.E.: A new approach to linear filtering and prediction problems. Transactions of the ASME Journal of Basic Engineering 82(Series D), 35–45 (1960)
18. Kamen, E.W., Su, J.K.: Introduction to optimal estimation. Springer, London (1999)
19. Karunasingha, D.S.K., Liong, S.Y.: Enhancement of chaotic time series prediction with real-time noise reduction. In: International Conference on Small Hydropower - Hydro Sri Lanka (2007)
20. Klaas, M., Briers, M., Freitas, N.d., Doucet, A., Maskell, S., Lang, D.: Fast particle smoothing: if I had a million particles. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 481–488 (2006)
21. Kostelich, E.J., Yorke, J.A.: Noise reduction in dynamical systems. Physical Review A 38(3), 1649 (1988)
22. Kostelich, E.J., Yorke, J.A.: Noise reduction: Finding the simplest dynamical system consistent with the data. Physica D: Nonlinear Phenomena 41(2), 183–196 (1990)

23. Legoux, A.: ALMA Arnaud Legous Moving Average (2009), http://www.arnaudlegoux.com/wp-content/uploads/2011/03/ALMA-Arnaud-Legoux-Moving-Average.pdf (2011)
24. Li, T., Li, Q., Zhu, S., Ogihara, M.: A survey on wavelet applications in data mining. SIGKDD Explor. Newsl. 4(2), 49–68 (2002), 772870
25. Liu, Y., Liao, X.: Adaptive chaotic noise reduction method based on dual-lifting wavelet. Expert Systems with Applications 38(3), 1346–1355 (2011)
26. Moore, J.B.: Discrete-time fixed-lag smoothing algorithms. Automatica 9(2), 163–173 (1973)
27. Nikpour, M., Nadernejad, E., Ashtiani, H., Hassanpour, H.: Using pde's for noise reduction in time series. International Journal of Computing and ICT Research 3(1), 42–48 (2009)
28. Russell, S.J., Norvig, P.: Artificial intelligence: A modern approach. Pretice Hall series in artificial intelligence. Prentice Hall, N.J (2010)
29. Sadasivan, P.K., Dutt, D.N.: SVD based technique for noise reduction in electroencephalographic signals. Signal Processing 55(2), 179–189 (1996)
30. Shynk, J.J.: Frequency-domain and multirate adaptive filtering. IEEE Signal Processing Magazine 9(1), 14–37 (1992)
31. Soofi, A.S., Cao, L.: Nonlinear forecasting of noisy financial data. In: Soofi, A.S., Cao, L. (eds.) Modeling and Forecasting Financial Data: Techniques of Nonlinear Dynamics, pp. 455–465. Kluwer Academic Publishers, Boston (2002)
32. Vandewalle, N., Ausloos, M., Boveroux, P.: The moving averages demystified. Physica A: Statistical Mechanics and its Applications 269(1), 170–176 (1999)
33. Whittaker, E.T.: On a new method of graduation. Proceedings of the Edinburgh Mathematical Society 41(-1), 63–75 (1923)
34. Zehtabian, A., Hassanpour, H.: A non-destructive approach for noise reduction in time domain. World Applied Sciences Journal 6(1), 53–63 (2009)