OPTIMISING THE SMOOTHNESS AND ACCURACY OF MOVING AVERAGE FOR STOCK PRICE DATA

Aistis RAUDYS[1*], Židrina PABARŠKAITĖ[2]

[1]*Faculty of Mathematics and Informatics, Vilnius University, Naugarduko g. 24, LT-03225 Vilnius, Lithuania*
[2]*Faculty of Electrical and Electronics Engineering, Kaunas University of Technology, Studentų g. 48, LT-51367 Kaunas, Lithuania*

Abstract. Smoothing time series allows removing noise. Moving averages are used in finance to smooth stock price series and forecast trend direction. We propose optimised custom moving average that is the most suitable for stock time series smoothing. Suitability criteria are defined by smoothness and accuracy. Previous research focused only on one of the two criteria in isolation. We define this as multi-criteria Pareto optimisation problem and compare the proposed method to the five most popular moving average methods on synthetic and real world stock data. The comparison was performed using unseen data. The new method outperforms other methods in 99.5% of cases on synthetic and in 91% on real world data. The method allows better time series smoothing with the same level of accuracy as traditional methods, or better accuracy with the same smoothness. Weights optimised on one stock are very similar to weights optimised for any other stock and can be used interchangeably. Traders can use the new method to detect trends earlier and increase the profitability of their strategies. The concept is also applicable to sensors, weather forecasting, and traffic prediction where both the smoothness and accuracy of the filtered signal are important.

Keywords: moving average filter, smoothness and accuracy, weight optimisation, triple smoothed exponential moving average (TSEMA), custom moving average.

JEL Classification: C22, C6, C63.

Introduction

In various disciplines, moving averages (MAs) are used to smooth and remove noise from time series data (Fiodor 2014; Sakalauskiene 2003; Kriaučiūnienė *et al.* 2007). They are very popular in finance (Kononovicius, Gontis 2013, 2015), and especially in trend following where traders use moving averages for trend forecasting. It is a popular trading strategy

---

*Corresponding author. E-mail: *aistis@raudys.com*

and the author in (Grebenkov, Serror 2014) states that "85% of CTA returns are explained by simple trend following". Moving averages smooth stock market data and make trend detection easier. Traders buy or sell (Raudys, Matkenaite 2016) depending on the direction of the smoothed stock market data. In this process two factors influence trader's success: the number of ill-detected trends and the promptness of trend detection. Poorly detected trends lead to unnecessary trading costs, slippage and commissions. Promptness of trend detection allows traders to enter a trend earlier and increase their profit. Ideally one needs to make few trades but enter the market at the beginning of the trend. Unfortunately, these requirements conflict. If the data are more smoothed then there are fewer error trades but trends are detected after a delay, and, vice versa, if the data are less smoothed then trends are detected earlier but there are many erroneous trades. Smooth time series cannot change direction often enough (otherwise they would not be smooth) to be close to the most recent price, so accuracy decreases as a series become smoother. For more information, please refer to Section II and illustrations in Figure 1.

The trading community has been trying to invent the "perfect moving average" for a long time, and many new methods have been introduced. Unfortunately, the comparison approach was not scientific, "proofs" were in many cases visual graphs with few moving averages plotted, and authors typically claimed that the new moving average was smoother than the others or lagged (was more accurate or differed from the actual price) less than the others. In this paper we solve this problem scientifically. We define two quality measures mathematically, name them "smoothness" and "accuracy" and try to find the best ratio between them.

Moving averages date to 1901, although the name was applied later. The method of smoothing data points was used for decades, until the general term came into use. In statistics moving averages are called "autoregressive models" where model parameters correspond to moving average weights. The statistical moving average model and the moving average used in finance to smooth the data must not be confused – they are different. Moving average is also a type of real time filter that removes unwanted frequencies (usually high) from the data. In signal processing, moving averages are thus also called "low pass filters" see (Orlowski 2010). Another name for moving averages is "smoothers" and "exponential smoothing" is the best known method. In the financial trading community however, the term "moving average" is the most popular. In a nutshell, moving average is a simple weighted sum calculated over a selected historical price range. Financial trading price data are usually noisy, so by using weighted sums we reduce noise. By averaging more data (increasing the lookback period), we can achieve a more smoothed price that could forecast the trend (Wang, Wu 2013; Chikhi *et al.* 2013), despite the price fluctuations. Let's define c_i as a price at the time i. Let $c = \{c_i\}$, $i = 1...p$ be the time series where p is the time series length. So, the simplest moving average of the period n at the time l would be:

$$ma_l^n = \frac{1}{n}\sum_{i=1}^{n}c_{l-i+1}. \qquad (1)$$

We define a series of moving averages as $ma^n = \{ma_i^n\}, i = 1...p$. The alternative form of simple moving average can be transformed using weights $w = w_i$, $i = 1 ... n$:

$$ma_l^n = \sum_{i=1}^{n}c_{l-i+1}w_i, \quad w_i = \frac{1}{n}, \quad \sum_{i=1}^{n}w_i = 1. \qquad (2)$$

Here each historical price c_l is multiplied with the appropriate weights element w_i, and n is the period (number of days or data periods to look back), w_1 is the weight for the oldest data member and w_n is the weight for the newest (most recent) data member. More complex moving averages have a more complex weight vector, $w = (w_1, w_1, ..., w_n)$. Some have linearly declining weights, such as weighted moving average (Kaufman 2013), some have exponentially declining weights – such as an exponential moving average (Kaufman 2013), and some can have negative weights but the w_i sum must always be 1. More exotic distributions of the weight components are also possible.

Moving averages are predominantly used to forecast trends in noisy data (Wang, Wu 2013). As the period (the window width), n, of a moving average is increased, more noise can be filtered from the financial data and better smoothness can be achieved. Sometimes price fluctuation is a trend reversal and not noise (Bury 2014). Since a moving average combines historical prices and a current price to obtain the filtered price, for some time, depending on the period, it will show the previous trend direction instead of the new one. This is called "lag". Lag is the distance in days that the smoothed signal needs to be shifted into the future. Lag is easily calculated for simple moving averages but is not so easily done for more exotic methods. Lag is sometimes called inertia. An easier and more mathematically proved method is to measure the quality as the accuracy – the difference between price and moving average values. A raw price (unsmoothed data) is perfectly accurate. The important question is how to improve accuracy.

Moving averages can have different properties. The simplest moving average has only positive $w_i > 0, i = 1...n$ and non-changing weights. We only analyse this type of moving averages in this paper, however, there are other types. Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) are two terms used in signal processing (Oppenheim *et al.* 1989) and indicate the size of the weight vector w. With FIR $w_i, i = 1...n, n < p$ and in IIR case $w_i, i = 1...n, n = p$, we note that n is the size of the weight vector and p is the size of time series. Adaptive moving averages (Makridakis, Wheelwright 1977) change weights depending on the circumstances – so smoothness and accuracy changes too. The adaptive moving average weight w^j used to calculate jth value is not equal to w^i used to calculate ith value, where $j \neq i$. The idea of adaptive moving averages has been extensively discussed in (Kaufman 2013) and some trading strategies involving adaptive element has been assessed in (Ellis, Parbery 2005). A fuzzy logic-based approach was discussed in (Raudys, S., Raudys, A. 2010). Another type of moving averages is corrective moving averages, which have positive and negative weights $w_i \in [-1,1], i = 1...n$ but still have to sum up to one: $\sum_{i=1}^{n} w_i = 1$. Some data (usually the most recent) has greater weights and this is compensated by negative weights (usually the oldest). For example, $\sum_{i=1}^{n} w_i, w_i > 0 = 1.2$ and $\sum_{i=1}^{n} w_i, w_i < 0 = -0.2$. This way error is reduced at the expense of an "overshooting" effect during trend reversals. These are special cases of Kalman filter (Welch, Bishop 1995). The most famous examples are the Hull moving average (Hull 2004) introduced by Allan Hull and T3 moving average (Tillson 1998) introduced by Tim Tillson. The Hull moving average has been widely used in trading applications (Pereira, Mello 2014). Once again, we analyse only moving averages with positive, non-adaptive, non-correcting weights in this paper.

Comparing moving averages based on smoothness and accuracy is very new. A comparison of 19 moving averages was presented in (Raudys *et al.* 2013). Only recently this topic has been addressed by authors and in some cases indirectly, as a side effect. It is worth mentioning Yager (Yager 2013, 2008) and Letchford (Letchford *et al.* 2011, 2012, 2013). In this paper, contrary to other works, we directly aim to optimise weights and create custom moving averages that would give the best smoothness and accuracy ratio. Global (Hendrix, Lancinskas 2015) or local methods can be used here. Our proposed method would give the highest accuracy for the given level of smoothness.

1. Moving averages

We evaluate two opposing properties of moving averages: smoothness and accuracy. Traders desire optimal filters so trend following would be accurate and whipsaw trades could be avoided. Better accuracy for a trader means that they can detect a trend earlier and profit more. Smoothness means that it avoids changing trading direction too often and at the same time reduces transaction costs. Usually moving averages take one parameter, the period *n* of the past prices, for use. As the period increases, accuracy decreases but smoothness increases. For different moving average periods *n* means something different. For the same *n*, two moving averages can generate enormously different smoothness and accuracy profile. In Figure 1 we see two different moving averages (simple moving average and TSEMA) with the same period of 10 days.

When the trend changes, the black one responds very slowly, value is far from the real price, but the line is smooth. Red follows price more aggressively, and remains close to the price, but is bumpier. The same period generates vastly different moving averages.
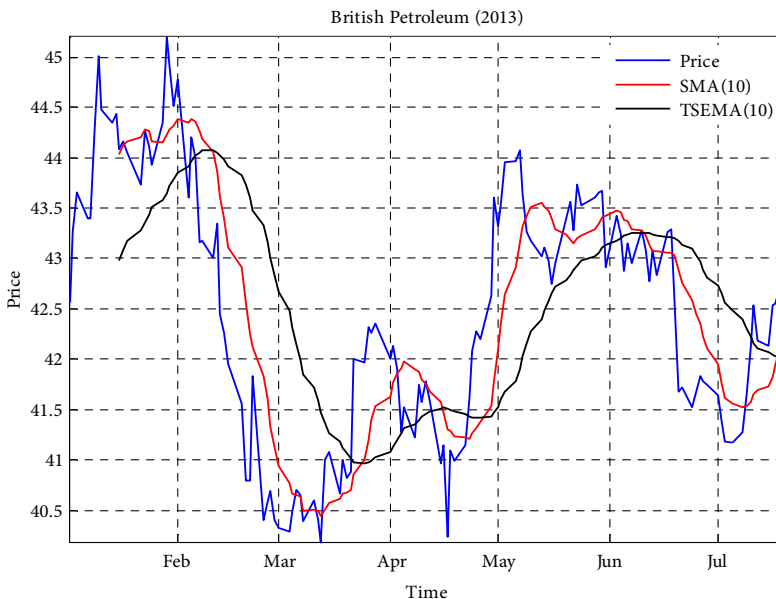

Fig. 1. The equal periods of 10 days for different MA generate different curves

We cannot compare moving averages by period as a similarity measure. To solve this problem we introduce a new comparison measure – *accuracy*. We define accuracy as the difference between current value and moving average value at that point. At the time t accuracy is: $acc_t = \left| c_t - ma_t^n \right|$. For the entire dataset, average accuracy is calculated as follows:

$$acc^{ma^n} = \frac{1}{p} \sum_{i=1}^{p} \left| c_i - ma_i^n \right|. \tag{3}$$

Here p is the time series length and n is a moving average period. This estimate explains how the moving average is missing the price on average. We will call "average accuracy" simply "accuracy" in this paper. Smaller accuracy values are better.

The other comparative measure is *smoothness*. It is defined like this:

$$smo^{ma^n} = \frac{1}{p} \sum_{i=3}^{p} \left| \left(ma_i^n - ma_{i-1}^n \right) - \left(ma_{i-1}^n - ma_{i-2}^n \right) \right| = \frac{1}{p} \sum_{i=3}^{p} \left| ma_i^n - 2ma_{i-1}^n + ma_{i-2}^n \right|. \tag{4}$$

In other words smoothness represents the extent to which the moving average is smooth, and how much it changes direction. The smaller the smoothness value, the better it is. Characteristic (4) is similar to the second derivative of the time series. In order to compare comparable things, we considered moving averages with similar smoothness. Our aim overall is to create a moving average that has a small accuracy value and a small smoothness value that corresponds to smooth and accurate results.

To illustrate how these metrics change we computed simple moving average smoothness and accuracy of the periods from 1 to 30. The results are depicted in Figure 2. Short period moving averages are very bumpy but quite accurate. Long period MA has a high smoothness but poor accuracy. As the period increases, moving average becomes smoother, but at the same time accuracy decreases. Alternatively, smoothness and accuracy can be plotted against each other. Different moving averages characteristics are plotted in Figure 3.
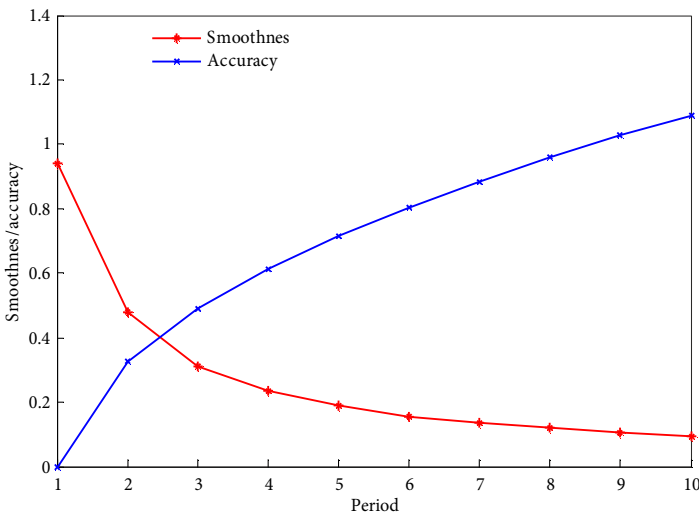

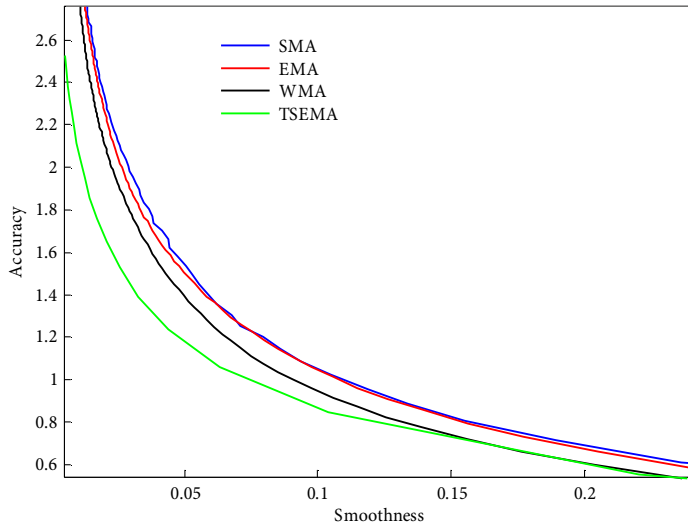Fig. 2. Smoothness and accuracy of SMA plotted as two lines

Fig. 3. Smoothness vs. accuracy plotted as one line

It can be seen that as the accuracy decreases, smoothness improves and vice versa. Each MA changes periods from 2 to 30 over the same dataset (British Petrol daily stock price series). These two simple measurements can help to answer questions: which moving average has the best smoothness/accuracy ratio? Or is smoothest for a given accuracy? Or is the most accurate for a given smoothness? As can be seen, the green (TSEMA) moving average is slightly better than the others.

Some researches propose fuzzy moving averages (Zeng *et al.* 2016). They are very interesting and potentially powerful methods. However they differ in the mechanics how they are calculated and used. Such models assume that values in time series are fuzzy. This is rather different approach, thus fair comparison of such methods becomes impossible. We plan to expand our investigation in the future where we will consider fuzzy moving averages. The principle of multi criteria optimization of proposed custom moving average can be applied to fuzzy moving averages as well.

2. Description of the moving averages

Simple moving average (SMA) is well known and widespread. It gives equal weights to all past prices and by definition is simply their average. Although it is very simple, it can solve serious problems and is widely used. It will be used as a benchmark to compare against other averages. The formula can be found in the introduction.

Exponential moving average (EMA) gives exponentially diminishing weights to all past prices. This moving average is very well known and used.

$$EMA_n(X)_i = \alpha X_i + \left(1-\alpha\right)EMA_n(X)_{i-1}, \ \alpha = \frac{2}{n+1}.$$

Weighted moving average (WMA) gives arithmetically diminishing weights for past prices, depending on the length of the average $WMA_n(X)_i = \sum_{l=1}^{n} \frac{X_{i-l+1}}{(n^2-n)/2+n}(n-l+1)$.

Sinus weighted moving average (SWMA) is a weighted average, based on motivation, that price fluctuates following some unknown wave. As a model, the Sine wave is used to adjust price weights. SWMA is calculated using this formula:

$$SWMA_n(X)_i = \frac{\sum_{i=1}^{n} \sin\left(i\frac{180}{6}\right)X_i}{\sum_{i=1}^{n} \sin\left(i\frac{180}{6}\right)} \; .$$

Where n is a period of moving averages, X is a list of prices, with X_0 being the most recent.

Double exponential moving average (DEMA) is completely different from those described above. It is a composite moving average and uses other moving averages to get the final result (Kaufman 2013). In the case of DEMA, the EMA is used. DEMA also employs mechanisms to adapt to price swings dynamically. DEMA uses tricks to improve smoothness by running a moving average on itself, but this operation increases lag, so to counter this problem it uses the so called "twicing" technique. Formula:

$$DEMA_n(X) = 2 \times EMA_n(X) - EMA_n(EMA_n(X)),$$

where n is the length of the moving average and X are actual prices.

Triple smoothed exponential moving average (TSEMA) applies the exponential moving average three times on the result:

$$TSEMA_n(X) = EMA_n\Big(EMA_n\Big(EMA_n\big(X\big)\Big)\Big).$$

Zero lag moving average (ZMA) sounds like a perfect moving average (Ehlers 2001), but the only thing without lag is the price, which this adaptive and composite moving average uses to correct itself. In a nutshell, ZMA adds a portion of price to EMA to counter lag, while giving up some smoothness. Formula (n – period, X – prices) if $n > 1$, $\beta = 0.2$.

$$ZMA_{\downarrow}n(X) = \alpha * (X_{\downarrow}(n-1) + \beta * (X_{\downarrow}n - ZMA_{\downarrow}(n-1)(X_{\downarrow}(n-1))) +$$
$$(1-\alpha) * ZMA_{\downarrow}(n-1)(X_{\downarrow}(n-1))).$$

Kaufman moving average (KAMA) alters alpha of EMA using smoothing constant C to achieve adaptability (Kaufman 2013: 731). Formula (n – period, X – prices, X_i – past price i bars back):

$$ER = \frac{|X_i - X_{i-n}|}{n * \sum_{i=1}^{n} |X_n - X_{n-1}|} \; ;$$

$$C = \left(ER * \left(0,6667 - 0,0645\right) + 0,645\right)^2 \; ;$$

$$KAMA(i) = KAMA(i-1) + C * \left(X_i - KAMA(i-1)\right).$$

KAMA adjusts alpha using a market efficiency ratio, which is the ratio between direction and volatility. Constants 0.6667 and 0.0645 represent adaptability ranging from 2 to 30 bars of the EMA alpha value. These constants are suggested by the author, so we will not change them.

3. Custom moving average

We propose using a custom moving average (CMA) where weights w are optimised to achieve the best accuracy on a given smoothness level. We emphasise that weights were not optimised and evaluated on the same data. Weights were optimised on one dataset and used for all stock and artificial time series. We computed 200 smoothness levels, equivalent to 200 SMA levels, and maximised accuracy for each of these smoothness levels by fixing it:

$$w = \min_{w} acc(ma(w)), sm(ma(w)) < sm^x.$$

Here w is a weight vector, acc is an accuracy function, sm is a smoothness function, ma is a moving average generated using weights w and sm^x is fixed to the specific smoothness level.

To achieve our goal, we used two methods - the heuristic and gradient descent type of optimisations and compared the results. Heuristic optimisation is slower but requires less tuning (Fiedor 2014), while gradient is very fast but optimisation parameters require calibration. Other faster optimisation methods are also possible, such as semi-definite-quadratic-linear programming.

3.1. Heuristic weight optimisation

We tried several heuristic methods for optimisation. Finally we used simple random weights modification optimisation to obtain optimal weights. Here we start with equal weights (the same as simple moving average weights) and randomly modify weights, and if accuracy is better than previously we retain the modification. This procedure is repeated iteratively (circa 200000 times). The method is simple, but relatively slow for use in practical tasks. To speed up calculations we developed a fast version of the gradient descent method. Our proposed gradient descent method is described in the next section.

3.2. Gradient weight optimisation

The more advanced method to optimise weights is the gradient descent method. It is widely used in neural network optimisations (Zhang *et al.* 2014). A cost function is created here and minimised. Denote weights $W = (w_1, w_2, \ldots, w_n)$, and multiple delayed time series $C_i^n = (c_i, c_{i-1}, c_{i-2}, \ldots, c_{i-n+1})$. Then

$$ma_l^n = \frac{1}{n}\sum_{i=1}^{n} c_{l+1-i} w_i = C_i^n \, W^T \text{ , and } acc^{ma^n} = \frac{1}{p}\sum_{i=1}^{p} |c_i - ma_i^n| = \frac{1}{p}\sum_{i=1}^{p} |C_i^n W^T - c_i|, \text{ and}$$

$$smooth^{ma^n} = \frac{1}{p-2}\sum_{i=1}^{p-2} | \left(ma_i^n - ma_{i-1}^n\right) - (ma_{i-1}^n - ma_{i-2}^n) | =$$

$$\frac{1}{p-2}\sum_{i=1}^{p-2}\mid ma_i^n - 2ma_{i-1}^n + ma_{i-2}^n \mid.$$

We use a modulus error criterion, hence we have:

1. Minimise following modulus: Cost (loss) function:

$$\text{Cost}_{\text{mod}}^{nl} = acc^{ma^n} = \left|\frac{1}{p}\sum_{i=1}^{p}ma_i^n - c_i\right| = \frac{1}{p}\sum_{i=1}^{p}\left|C_i^n W^T - c_i\right|, \qquad (5)$$

2. and fulfil the requirement:

$$smo^{ma^n} = const. \qquad (6)$$

To satisfy requirement (6) instead of minimising Cost (5) we can minimise modified cost function with an additional "weight decay" term:

$$\text{Cost}_{\text{mod}}^{nl} = \frac{1}{p}\sum_{i=1}^{p}\left|C_i^n W^T - c_i\right| + \Lambda \times \left|smooth^{ma^n} - const_1\right|,$$

where Λ is a large positive constant introduced in order to ensure fulfilment of the requirement (6). It penalises the cost function if the requirement is not completely fulfilled.

The smoothness term (6) can be rewritten in the following form:

$$smooth^{ma^n} = \frac{1}{p-2}\sum_{i=1}^{p-2}\mid C_i^n W^T - 2C_{i-1}^n W^T + C_{i-2}^n W^T \mid = \frac{1}{p-2}\sum_{i=1}^{p-2}\mid \Delta C_i^n W^T \mid, \qquad (7)$$

where we denote $\Delta C_i^n = C_i^n - 2C_{i-1}^n + C_{i-2}^n$. In this way

$$\text{Cost}_{\text{mod}}^{nl} = \frac{1}{p}\sum_{i=1}^{p}\left|C_i^n W - c_i\right| + \Lambda \times \left|(\frac{1}{p-2}\sum_{i=1}^{p-2}\left|\Delta C_i^n W^T\right|) - const_1\right|. \qquad (8)$$

The n-dimensional vector-row of the gradient of the cost function (8) is:

$$\frac{\partial \text{Cost}^{nl}}{\partial W} = \frac{1}{p}\sum_{i=1}^{p}\text{sign}\left(C_i^n W^T - c_i\right)C_i^n +$$

$$\frac{\Lambda}{p-2}\text{sign}\left(\frac{1}{p-2}\sum_{i=1}^{p-2}\left|\Delta C_i^n W^T\right| - const_1\right) \times \left(\sum_{i=1}^{p-2}\text{sign}(\Delta C_i^n W^T)\Delta C_i^n\right). \qquad (9)$$

The iterative total gradient descent training procedure adapts n-dimensional weight vector W as

$$W_{t+1} = W_t - n \times \frac{\partial \text{Cost}_{\text{mod}}^{nl}}{\partial W}, \qquad (10)$$

where n is a learning step parameter.

The components of the correction term $\dfrac{\partial \text{Cost}_{\text{modified}}^{nl}}{\partial W}$, in gradient descent training do not depend on the current values of the weight vector, or on both distances to the desired values, $C_i^n W_i^n - c_i$ and $\left|C_i^n W^T\right| - const_1$. To realise smooth minimisation of the cost function, learning step size should decrease with an increase in the number of iterations. In Figure 4 we present weights of the $p + n = 2000 + 49$ random values generated by $x_i \in N(1,0)$, $c_i = c_{i-1} + x_i$ x, $const_1 = \gamma_1 \dfrac{1}{p}\sum_{i=1}^{p}\mid C_i^n W^T - c_i \mid$, where γ_1 is a chosen constant, e.g. $\gamma_1 = 1$.

In comparison with a direct heuristic search of the best set of the weights, $w_1, w_2, \ldots, w_n$, training is very fast, but it requires the tedious selection of training parameters, a number of iterations, Λ, η, and a style of reduction of η with the increase in the number of iterations. A successful example is presented in Figure 4.
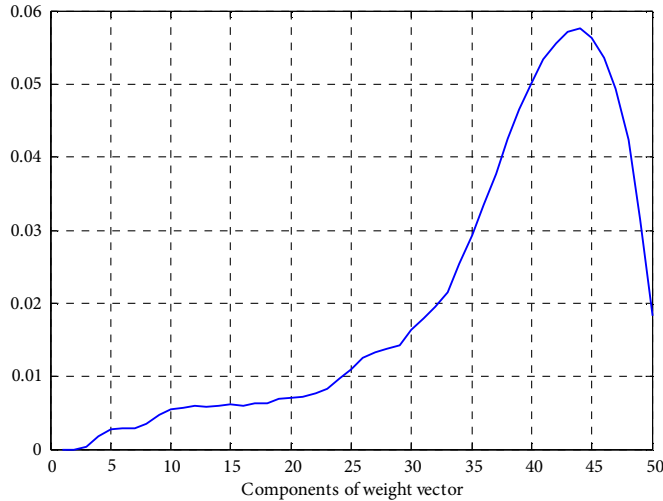

Fig. 4. Weights of the CMA $n = 50$ method after optimisation

3.3. Optimisation results

Initially we thought that optimised weights would be different for different stocks but weights are near optimal for any stock data, as well as for randomly generated (independent and uncorrelated) data (Ruseckas *et al.* 2012). The difference cannot be distinguished visually. We generated random data using $x_i \in N(1,0)$, $c_i = c_{i-1} + x_i$, $c_0 = 0, i = 1 \ldots p$ normal distribution and later cumulatively summed them. Triple smoothed exponential moving average (TSEMA) has the weights most similar to our method. This is seen in the Figure 5.

Here we can see that weights optimised for British Petroleum (BP), S&P 500 (SPY) and cumulatively summed normally distributed random data (random) and TSEMA method weights are very similar. The difference between weights is more visible for the longer averaging periods. In Figure 6 we present $n = 63$ day weights.

Random and BP (British Petroleum) weights are initially (on the right) very similar but diverge as they go further into the past. TSEMA weights are shifted to the left in comparison to others. This confirms that recent history is the most important.

3.4. Beta distribution

This weighting scheme is similar to the beta probability density function where $\alpha = 1.9$; $\beta = 6$:

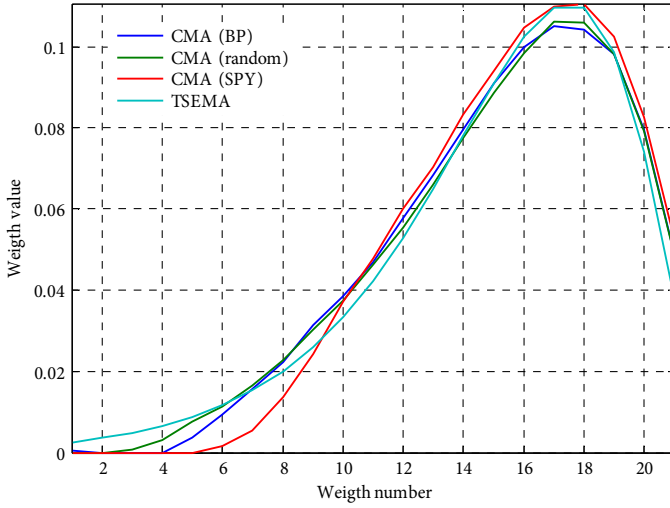$$f(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathrm{B}(\alpha,\beta)}.$$ (11)

Fig. 5. Weights of various methods generated using various data (data name in brackets)
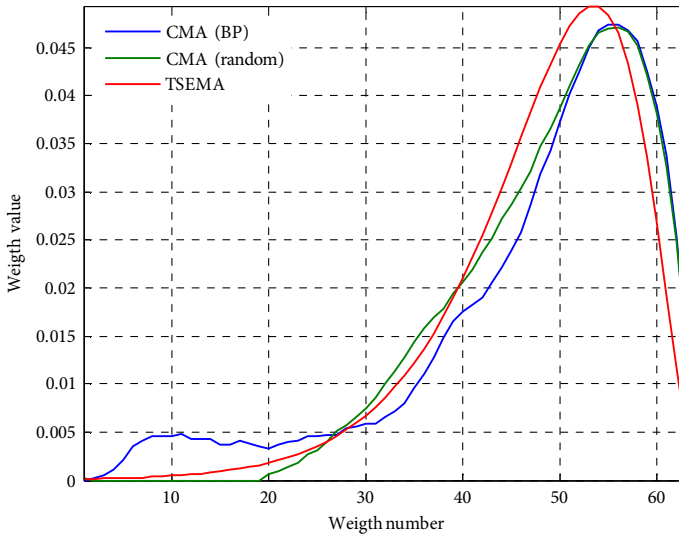

Fig. 6. CMA weights generated using two different datasets and weights of TSEMA

In Eq. (11) B is the beta function. This weighting method has to be investigated further and is the subject of future research. Preliminary investigation has demonstrated very similar results. See Figure 7.

4. Discussion

Weighs that correspond to the oldest data points has a very minor impact on the accuracy and smoothness, so for longer periods noise in the time series data distorts weights for the

oldest weights (left side of the graphs). This is illustrated in Figure 8 BP (British Petroleum) optimised weights for a 123 day period using the latest 4000 days of data (from 16-Sep-1997 to 19-Jul-2013). Weights for the oldest data (left side) are distorted.

In the experiments we used the average absolute smoothness measure defined in (4). We also tried different smoothness measures such as mean square smoothness:

$$smo^{ma^n} = \frac{1}{p}\sum_{i=3}^{p}\left[\left(ma_i^n - ma_{i-1}^n\right)-\left(ma_{i-1}^n - ma_{i-2}^n\right)\right]^2 .$$
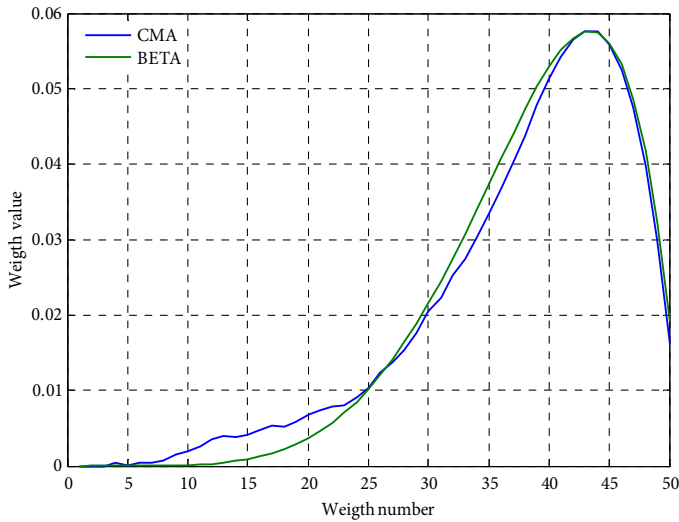

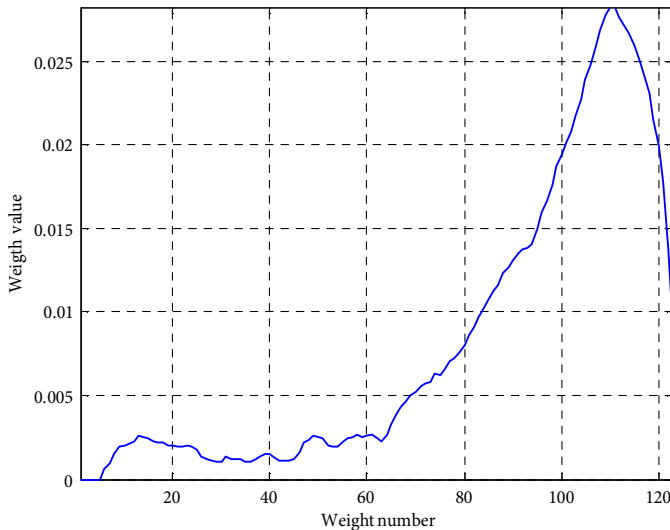Fig. 7. Beta distribution weights alongside CMA weights

Fig. 8. Very long (123) CMA weight vector

The preliminary results showed that weights are highly skewed to the most recent data and visual smoothness is not as good as absolute smoothness. We therefore left the investigation of this topic for the future.

Other metrics for smoothness have been investigated by other authors. For example, in (Yager 2008) the authors use variance as a smoothness measure $smo^{ma^n} = \text{var}(ma^n)$.

The reasoning of the authors remains unknown. We plan to investigate and compare these and other smoothness measures in the future.

5. Experiments

We compared our new method to other methods with real and artificial data in the experiments. We found that our new method is usually superior. Comparison methodology was a challenge as we needed to compare moving averages by two criteria. More example of multi criteria optimisation can be found in (Ulubeyli, Kazaz 2016).

5.1. Comparison methodology

Comparing two moving averages is not easy due to the *two* criteria: smoothness and accuracy. Either the volume of a two dimensional shape or values at specific points can be measured. We decided to test the four smoothness levels that are most common in practice, derived from simple moving average of 5, 10, 21 and 63 days ($n = 5,10,21,63$). These periods are commonly used by practitioners, and represent one week, two weeks, one month and a quarter of the year. Each period has its associated smoothness level computed according to a formula (4). For random data these smoothness levels correspond to the values $smo^{ma^n} = 0.22, 0.11, 0.052$ and 0.017. For each moving average we then computed all smoothness and accuracy values in the range from 1 to 500, so that we had all possible versions of moving averages. We found the smallest smoothness value, not less than P5, P10, P21, P63, and then we measured the accuracy: that is, we found the MA with the same smoothness level and then measured the accuracy. This way we compared "apples to apples". We repeated this procedure for all moving averages and finally we compared accuracies between each MA. At this point we found the MA that for a given smoothness has the best accuracy. For each time series there were four winners for different smoothness levels (P5, P10, P21, P63). Finally, we counted how often each moving average was most accurate. The results can be seen in Results section. The summary of moving averages used in our experiments is presented in Table 1.

Table 1. List of moving averages

Abbreviation	Full Name
cmavg	Custom Moving Average (New Method)
tsemavg	Triple Smoothed Exponential Moving Average
zeromavg	Zero Lag Moving Average
mavg	Simple Moving Average
kamaavg	Kaufman Adaptive Moving Average

End of Table 1

Abbreviation	Full Name
emavg	Exponential Moving Average
wmavg	Weighted Moving Average
smavg	Sine Weighted Moving Average
demavg	Double Exponential Moving Average

5.2. Real world time series data

For the real world data we used ~2000 United States traded stocks from NASDAQ and NYSE exchanges. Initially we had 2082 stocks, and after filtering stock with less than 1000 days of history we had 1952 stocks. We obtained time series data from Google finance. Data points go as old as 13-Aug-1993 and the newest record is 24-Jul-2013. Date range is 4524 days and across all stocks we performed experiments on 6643501 data points.

Stock data are a series of prices, obtained from the exchange where the stock was traded. Historical stock prices are adjusted at the point they pay a dividend. On the day after dividend payment, historical data are moved down according to the amount of dividend stocks paid. Stocks that paid an unusually big dividend at some point in its history may thus have negative historical prices. Adjustment is necessary to make stock price look like a long and continuous investment, and dividends are accumulated into the stock's growth. In our study we selected the most popular and liquid stocks. We filtered stocks with the highest trading volume at the time of writing and with a recent price above 10 USD. An example of stocks used in this study is presented in Table 2.

Table 2. Used stocks

Ticker	Name
AAPL	Apple Inc.
PFE	Pfizer, Inc.
GE	General Electric
MSFT	Microsoft Corp.

6. Results and discussion

To demonstrate the benefits of the proposed method we generated 1000 random normally distributed synthetic time series datasets and cumulatively summed the values. This resulted in visually similar time series as typical stock price time series. We compared our method to other moving averages techniques on these artificial time series and the results are listed in Table 3 where numbers indicate number of times moving average was the best. The very right column is percentage representation. All moving averages under investigation were compared using four smoothness levels. For every smoothness level we select a winning

moving average. We repeated this procedure for all time series. Finally we counted how often each moving average was most accurate. We can see that the novel method is the best choice 99.5 percent of the time.

  Experiments with the stock data were performed in a similar way as in experiments with the random data. We compared every moving average for every stock and every smoothness level. In each category we selected the most accurate. The results are presented in Table 4 where numbers indicate number of times moving average was the best. The very right column is percentage representation.

  To demonstrate a snapshot of results that have not been summarised we included a subset of stock comparison in Table 5. Every row represents stocks and each column represents a smoothness level equivalent to a simple MA smoothness of 5, 10, 21 and 63 days. Cells contain the name of the most accurate moving average.

Table 3. Summary of experiments with random data

Data	P5	P10	P21	P63	Total	percent
cmavg	1000	1000	990	989	3979	99.5%
tsemavg	0	0	10	11	21	0.5%
mavg	0	0	0	0	0	0.0%
emavg	0	0	0	0	0	0.0%
wmavg	0	0	0	0	0	0.0%
zeromavg	0	0	0	0	0	0.0%
smavg	0	0	0	0	0	0.0%
kamaavg	0	0	0	0	0	0.0%
demavg	0	0	0	0	0	0.0%

Table 4. Summary of stock experiments

Data	P5	P10	P21	P63	Total	percent
cmavg	1968	1972	1813	1434	7187	91.0%
tsemavg	0	1	160	539	700	8.9%
mavg	5	0	0	0	5	0.1%
emavg	0	1	1	1	3	0.0%
wmavg	1	0	0	0	1	0.0%
zeromavg	0	0	0	0	0	0.0%
smavg	0	0	0	0	0	0.0%
kamaavg	0	0	0	0	0	0.0%
demavg	0	0	0	0	0	0.0%

Table 5. Sample of non-summarised stock experiments. Table body cells contain the name of the most accurate moving average for that smoothness category

Stock	P5	P10	P21	P63
A	cmavg	cmavg	cmavg	cmavg
AAN	cmavg	tsemavg	cmavg	cmavg
AAON	cmavg	cmavg	cmavg	tsemavg
AAP	cmavg	cmavg	tsema	tsemavg
AAPL	cmavg	cmavg	cmavg	tsemavg
AAXJ	cmavg	cmavg	cmavg	cmavg
AB	cmavg	cmavg	cmavg	tsemavg
ABAX	cmavg	cmavg	cmavg	tsemavg
ABB	cmavg	cmavg	cmavg	cmavg
ABC	cmavg	cmavg	cmavg	tsemavg
ABCO	cmavg	cmavg	cmavg	cmavg
ABG	cmavg	cmavg	cmavg	cmavg

For P63 smoothness, TSEMA method is superior circa 10% of the time with stocks data. We believe that in the future we could improve our method and reduce this number by improving the longer weight optimisation process. This subject is reserved for future research.

Visual comparison of CMA and the next best method, TSEMA, are depicted in Figure 9 and Figure 10. The visual difference between them is small but consistent. Similar behaviour was observed in all the experiments.

6.1. Numerical example

In this example we numerically demonstrate the superiority of CMA over other methods and visually illustrate on one specific stock. For comparison we chose only TSEMA method as other methods are much worse. We compare CMA only with the best of available methods: TSEMA. We select SPY dataset closing prices of the day (S&P 500 index traded on NYSE). This one of the most popular indexes in the world and is widely traded and used. We select Simple MA of 20 days. We smoothed the data and measured smoothness and lag of the resulting time series: $S_{SMA} = 0.0669$ $A_{SMA} = 2.5180$.

Next, in order to compare to other methods we need to fix one of the two metrics. We fix smoothness, so smoothness of other methods cannot be higher than $S = 0.0669$. We selected best smoothness not exceeding given value 0.0669 for CMA and TSEMA methods: $S_{CMA} = 0.0668$ $A_{CMA} = 1.8952$ and $S_{TSEMA} = 0.0614$ $A_{TSEMA} = 1.9890$.

We can see that CMA is more accurate than TSEMA: $L_{CMA} = 1.8952 < L_{TSEMA} = 1.9890$. CMA is almost always produces better accuracy than TSEMA and other methods, regardless of smoothness level or stock data series we chose. This is illustrated in the Figure 9 where we depict smoothness and lag of two methods. CMA red line is always below TSEMA cyan line. In the next Figure 10 we illustrate numerical example. Smoothness of all 3 methods is not exceeding $S = 0.0668$ value. CMA (red) is always closer to the price (blue) than other methods, thus is the most accurate.
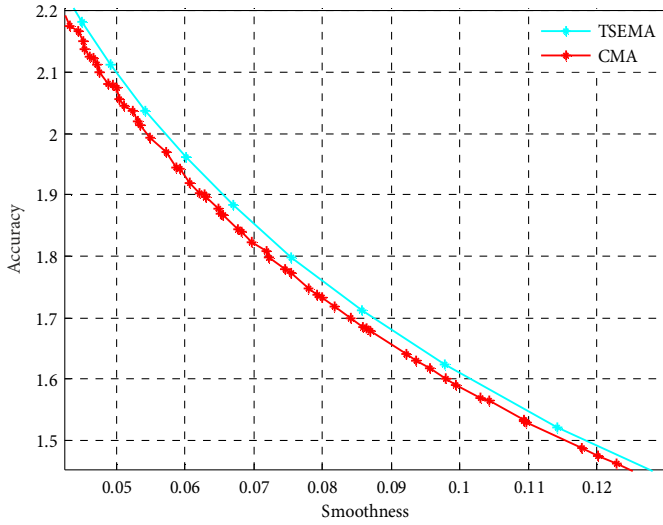
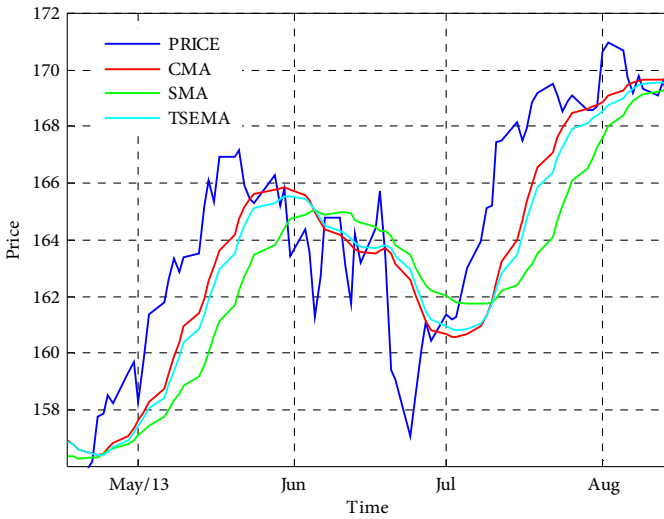Fig. 9. Smoothness vs. Accuracy of CMA and TSEMA

Fig. 10. Smoothed price curve using CMA, SMA and TSEMA.
CMA is more accurate than other methods

Conclusions

In the past, technical traders have debated whether some moving averages are smoother or have less lag (are more accurate) than the others. No one, however, analysed these two criteria together and often analysis was purely visual, in a chart. We define smoothness and accuracy mathematically and propose a new Pareto optimised custom moving average (CMA) method, which optimises weights to obtain the most accuracy for a given smoothness or vice versa. The originality of CMA method is in simultaneous optimisation of *two* quality

measures – smoothness *and* accuracy. Typical moving average has fixed weights and aim to smooth the data or produce least delayed (most accurate) results. In other words we optimise one of the parameters in isolation – smoothness *or* accuracy. The novelty of this paper is in weights optimisation based on *dual* criteria. This was not done in the past. The study showed that our new method was more accurate than all other popular moving averages for stock price series smoothing. Custom moving average is more accurate than other moving averages in 99.5% of cases on synthetic data and in 91% of cases on real world stock data. By definition it is the best possible weighting scheme (subject to the optimisation quality) as we optimise accuracy and smoothness. The evaluation is performed using out of sample, unseen data from 1000 randomly generated datasets and ~2000 the most liquid NASDAQ and NYSE stocks. We demonstrate empirically that the new approach outperforms all tested moving averages and has both the best smoothness and the best accuracy. For the majority of stocks and artificial data, for the four most popular smoothness levels, the suggested moving average generates the best accuracy in comparison to all other moving averages. In a minority of cases the new method is outperformed by other methods and we plan to address this issue in future research. The concept is applicable not only to financial data but also to sensors, weather forecasting, and traffic prediction where both the smoothness and accuracy of the filtered signal are important.

More importantly we obtained very similar smoothing weights for both stocks and for random data. Weights optimised on one stock can be interchangeably used for another and still outperform all other techniques. The same is true for random data. Weights optimised on the random data can be successfully used for stock data and still outperform all other techniques. For all experiments we used the same weights generated on random data. No overfitting is possible here. This means that stock data are random in nature and is not auto-correlated. We will investigate this phenomenon in more detail in future research.

CMA has fixed length weights (a finite impulse response) compared to other methods, which have infinite weights (such as Exponential MA or any composite method using EMA). This feature can be beneficial in certain situations where a fast processing time is required.

We analysed moving averages according to two criteria: smoothness and accuracy. These are, of course, important criteria, but the ultimate criterion for a trader is profit. In future research we will try to use CMA for trading applications using "one moving average" or "two moving average crossover" or other trading logic. Alternatively, instead of comparing MA value to today's price we may compare MA to future value (tomorrow). Preliminary research showed that weights are very similar. This change will make MA similar to the autoregressive model with one difference. Autoregression pays no attention to the smoothness of the forecast; it simply tries to optimise one criterion – accuracy (or mean square error). In parallel to the other accuracy criterion we plan to investigate different smoothness definitions including squared smoothness and variance of the moving average.

We also plan to improve the weight optimisation procedure. We already used the heuristic and gradient descent method but think that it is possible to solve the weight optimisation problem analytically. We also plan to investigate beta distribution for weight generation. This would speed up the optimisation process substantially. Overall this method is very promising and is worth further investigation.

Acknowledgements

References

Bury, T. 2014. Predicting trend reversals using market instantaneous state, *Physica A: Statistical Mechanics and its Applications* 404: 79–91. https://doi.org/10.1016/j.physa.2014.02.044

Chikhi, M.; Péguin-Feissolle, A.; Terraza, M. 2013. SEMIFARMA-HYGARCH modeling of Dow Jones return persistence, *Computational Economics* 41(2): 249–265. https://doi.org/10.1007/s10614-012-9328-9

Ehlers, J. 2001. *Rocket science for traders*. New York: JohnWiley.

Ellis, C. A.; Parbery, S. A. 2005. Is smarter better? A comparison of adaptive, and simple moving average trading strategies, *Research in International Business and Finance* 19(3): 399–411. https://doi.org/10.1016/j.ribaf.2004.12.009

Fiedor, P. 2014. Sector strength and efficiency on developed and emerging financial markets, *Physica A: Statistical Mechanics and its Applications* 413: 180–188. https://doi.org/10.1016/j.physa.2014.06.066

Grebenkov, D. S.; Serror, J. 2014. Following a trend with an exponential moving average: analytical results for a Gaussian model, *Physica A: Statistical Mechanics and its Applications* 394: 288–303. https://doi.org/10.1016/j.physa.2013.10.007

Hendrix, E. M. T.; Lancinskas, A. 2015. On benchmarking stochastic global optimization algorithms, *INFORMATICA* 26(4): 649–662. https://doi.org/10.15388/Informatica.2015.69

Hull, A. 2004. *Hull moving average (HMA)* [online], [cited 24 March 2011]. Available from Internet: http://www.justdata.com.au/Journals/AlanHull/hull_ma.htm

Kaufman, P. J. 2013. *Trading systems and methods*. John Wiley & Sons.

Kononovicius, A.; Gontis, V. 2013. Three-state herding model of the financial markets, *EPL (Europhysics Letters)* 101(2): 28001. https://doi.org/10.1209/0295-5075/101/28001

Kononovicius, A.; Gontis, V. 2015. Herding interactions as an opportunity to prevent extreme events in financial markets, *The European Physical Journal B* 88(7): 1–6. https://doi.org/10.1140/epjb/e2015-60160-0

Kriaučiūnienė, J.; Kovalenkovienė, M.; Meilutytė-Barauskienė, D. 2007. Changes of the Low Flow in Lithuanian Rivers, *Environmental Research, Engineering & Management* 42(4): 5–12.

Letchford, A.; Gao, J.; Zheng, L. 2011. Penalized least squares for smoothing financial time series, in D. Wang, M. Reynolds (Eds.). *Advances in Artificial Intelligence. AI 2011. Lecture Notes in Computer Science*. Vol 7106. Berlin, Heidelberg: Springer, 72–81.

Letchford, A.; Gao, J.; Zheng, L. 2012. Optimizing the moving average, in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 10–15 June 2012, Brisbane, Australia, 1–8.

Letchford, A.; Gao, J.; Zheng, L. 2013. Filtering financial time series by least squares, *International Journal of Machine Learning and Cybernetics* 4(2): 149–154. https://doi.org/10.1007/s13042-012-0081-0

Makridakis, S.; Wheelwright, S. C. 1977. Adaptive filtering: an an integrated autoregressive/moving average filter for time series forecasting, *Journal of the Operational Research Society* 28(2): 425–437. https://doi.org/10.1057/jors.1977.76

Oppenheim, A. V.; Schafer, R. W.; Buck, J. R. 1989. *Discrete-time signal processing*. Prentice-Hall, Inc.

Orłowski, P. 2010. Simplified design of low-pass, linear parameter-varying, finite impulse response filters, *Information Technology and Control* 39(2): 130–137.

Pereira, C. M.; de Mello, R. F. 2014. TS-stream: clustering time series on data streams, *Journal of Intelligent Information Systems* 42(3): 531–566.

Raudys, A.; Lenčiauskas, V.; Malčius, E. 2013. Moving averages for financial data smoothing, in T. Skersys, R. Butleris, R. Butkiene (Eds.). *Information and Software Technologies. ICIST 2013. Communications in Computer and Information Science*. Vol 403. Berlin, Heidelberg: Springer, 34–45. https://doi.org/10.1007/978-3-642-41947-8_4

Raudys, A.; Matkenaite, S. 2016. Analysis of execution methods in US and European futures, *The Journal of Trading* 11(1): 38–52. https://doi.org/10.3905/jot.2016.11.1.038

Raudys, S.; Raudys, A. 2010. Pairwise costs in multiclass perceptrons, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32(7): 1324–1328.

Ruseckas, J.; Gontis, V.; Kaulakys, B. 2012. Nonextensive statistical mechanics distributions and dynamics of financial observables from the nonlinear stochastic differential equations, *Advances in Complex Systems* 55(2): 161–167. https://doi.org/10.1142/s0219525912500737

Sakalauskienė, G. 2003. The Hurst phenomenon in hydrology, *Environmental Research, Engineering and Management* 3(25): 16–20.

Tillson, T. 1998. Smoothing techniques for more accurate signals, *Technical Analysis of Stocks and Commodities* 16: 57–59.

Ulubeyli, S.; Kazaz, A. 2016. Fuzzy multi-criteria decision making model for subcontractor selection in international construction projects, *Technological and Economic Development of Economy* 22(2): 210–234. https://doi.org/10.3846/20294913.2014.984363

Wang, Y.; Wu, C. 2013. Efficiency of crude oil futures markets: new evidence from multifractal detrending moving average analysis, *Computational Economics* 42(4): 393–414. https://doi.org/10.1007/s10614-012-9347-6

Welch, G.; Bishop, G. 1995. *An introduction to the Kalman filter*. Technical paper. Unpublished.

Yager, R. R. 2008. Time series smoothing and OWA aggregation, *Fuzzy Systems, IEEE Transactions on Fuzzy Systems* 16(4): 994–1007. https://doi.org/10.1109/TFUZZ.2008.917299

Yager, R. R. 2013. Exponential smoothing with credibility weighted observations, *Information Sciences* 252: 96–105. https://doi.org/10.1016/j.ins.2013.07.008

Zeng, S. Z.; Su, W. H.; Zhang, C. 2016. Intuitionistic fuzzy generalized probabilistic ordered weighted averaging operator and its application to group decision making, *Technological and Economic Development of Economy* 22(2): 177–193. https://doi.org/10.3846/20294913.2014.984253

Zhang, B.; Wei, Y.; Yu, J.; Lai, X.; Peng, Z. 2014. Forecasting VaR and ES of stock index portfolio: a Vine copula method, *Physica A: Statistical Mechanics and its Applications* 416: 112–124. https://doi.org/10.1016/j.physa.2014.08.043