Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 652653, 5 pages
http://dx.doi.org/10.1155/2013/652653

Research Article

On the Use of Backward Difference Formulae to Improve the Prediction of Direction in Market Related Data

E. Momoniat,[1] C. Harley,[1] and M. Berman[2]

[1] *Centre for Differential Equations, Continuum Mechanics and Applications, School of Computational and Applied Mathematics, University of the Witwatersrand, Private Bag 3, Wits 2050, Johannesburg, South Africa*
[2] *RAPA Pty Ltd., Level 10, 2 Bligh Street, Sydney, NSW 2000, Australia*

Correspondence should be addressed to E. Momoniat; ebrahim.momoniat@wits.ac.za

Received 10 August 2013; Revised 13 October 2013; Accepted 14 October 2013

Academic Editor: Fenghua Wen

The use of a BDF method as a tool to correct the direction of predictions made using curve fitting techniques is investigated. Random data is generated in such a fashion that it has the same properties as the data we are modelling. The data is assumed to have "memory" such that certain information imbedded in the data will remain within a certain range of points. Data within this period where "memory" exists—say at time steps $t_1, t_2, \ldots, t_n$—is curve-fitted to produce a prediction at the next discrete time step, t_{n+1}. In this manner a vector of predictions is generated and converted into a discrete ordinary differential representing the gradient of the data. The BDF method implemented with this lower order approximation is used as a means of improving upon the direction of the generated predictions. The use of the BDF method in this manner improves the prediction of the direction of the time series by approximately 30%.

1. Introduction

In this brief note we show how a BDF method can be used as a corrector for predictions. BDF methods are backward differentiation formulae which are a family of multistep implicit methods. They are designed to solve initial value ordinary differential equations. The derivative of a function is approximated using information computed from earlier time steps, thereby increasing the accuracy of the approximation. This characteristic makes BDF methods ideal for our purposes where we seek to improve upon already existing data in the form of predictions. The predictions aim at accurately reflecting the direction of random data and are made using curve fitting techniques. This research comes out of a project undertaken to predict the direction of a subset of the South African market. The approach taken in analysing the data assumes that the data has a "memory." More precisely this presupposes that a time series will have certain periods when the data has the same inherent information and dynamics. This allows us to conclude that the same information embedded in a previous selection of data points is still contained within the data point to be predicted from that set. While the data we generate in this paper is random with zero mean we are still able to show how an application of a BDF method improves the degree of accuracy in predicting the direction the data takes. The BDF formulae are constructed by satisfying the differential equation exactly at one point t_n and then interpolating k previous points. A Lagrangian interpolation is typically used. The initial prediction of direction is made using linear/spline curve fitting. The implementation of the BDF is not done directly; rather it is combined with a lower order approximation of the gradient of the data vector which leads to the difference equation we aim to use. The novelty of the approach taken here is that we iterate the difference equation structured from the BDF formula and the lower order approximation of the gradient to convergence.

The random walk hypothesis has had its fair share of attention as a means of explaining stock price movements. This financial theory states that stock market prices evolve according to a random walk and thus the prices of the stock market cannot be predicted. While the work undertaken in this paper concurs with this theory with regard to the random walk followed by the relevant data, we still maintain an assumption that there exists embedded information within the relevant data which can be modelled. Thus we assume a consistent underlying dynamic which can be identified and used as a means of extrapolating beyond known data points, that is, predict future movements in market prices. Theoretical developments in mathematics of finance have centred around the random walk hypothesis [1, 2] and the fact that the market cannot be predicted when using this hypothesis. Various modifications to this hypothesis have been proposed with the development of the theory of Martingales [3, 4]. The validity of the random walk hypothesis has been questioned in many studies. Most prominent among these is the book by Lo and MacKinlay [5].

While the mathematical theory used to develop the mathematical aspects of finance has not really focused on predicting returns there has been a strong interest in developing tools which can give a sense of the direction the market is going to take or when possible turning points will occur. The inclusion of ideas from the social sciences in financial mathematics has heralded the potential development of tools that can be used to aid the prediction of market trends. Among these ideas are aspects of behavioral science [6, 7] which studies the influence of psychology on the behaviour of financial practitioners and the subsequent effect on markets [8]. This theory suggests that, since the irrational behaviours of traders impact price movements, a time series of prices contains information which does not reflect what could be termed logical or mathematical dynamics. Behavioural science brings our attention to the possibility that "noise" may have been incorporated into the data obtained from price movements which makes it difficult to determine some identifiable characteristics which can be used to predict future movements. This theory is related to the random walk hypothesis in the sense that both indicate some irrational behaviour in the data. In this paper we have assumed that there is however some rational underlying dynamics which can be investigated mathematically which allows us to make predictions of future price movements. In some sense the impact of "noise," due to the irrational behaviour of traders, on price movements is an obstacle which we believe we have overcome by being able to improve upon the direction of our predictions. Other tools considered as aids for predicting market trends are notions of overreaction, underreaction, and contrarian strategies [7, 9–12]. Berman [13] has attempted one of the first studies to analyse the Global Real Estate Securities market using aspects of these contrarian ideas.

The paper is set out as follows. In Section 2 we develop and motivate the algorithm. Convergence properties of the algorithm are discussed in Section 3. Results and concluding remarks are presented in Section 4.

2. Algorithm Description

The first part of this analysis is to generate random data that may be used to simulate an actual financial time series. Here we use the MATLAB function `randn` that generates pseudorandom scalar values drawn from a normal distribution with mean zero and standard deviation one. The code used to generate this data is presented in Algorithm 1. We start out with an element that has value zero and then start stepping

```
repeat
for $j$ from 1 to 100
   $x(1) = 0$;
   for $i$ from 1 to 99;
      $z$ = randn;
     if $z < 0.5$
         $x(i + 1) = x(i) + z$;
     else
         $x(i + 1) = x(i) - z$;
     end
  end
  $y(j) = x(n)$;
end
```

Algorithm 1: MATLAB code implemented to generate data capable of simulating an actual time series of returns.

along the x-axis. The direction in which we step is dependent on whether the number outputted by `randn` is greater or less than a half. The magnitude of the step is determined by the magnitude of the number outputted by `randn`. We continue stepping in this way a finite number of times; 100 steps were chosen in this instance. The last data point is put into a vector which we use to simulate our times series of returns. Only the last point is chosen since the vector which has just been created consists of points with relative small errors between them; that is, this vector is not a good representation of the data we are trying to simulate. We continue running the random walk until we have a vector of returns. We subtract the mean and divide by the variance in order to produce data that is more reflective of financial returns obtained from price movements, that is, data within the range $[0, 1]$. The return data generated in this way has zero mean with standard deviation smaller than one. The standard deviation is not one since the data has not been normalised; that is, we divided by the variance and not the standard deviation. This was done in order to maintain the strict range of $[0, 1]$ for the data. In Figure 1 we plot three simulations of return data obtained in this way. When we compare the generated data to an actual times series of returns from the South African market we find that our simulation is fairly accurate since the movements exhibited by the returns of the South African market are similar to those depicted in Figure 1.

We then choose an appropriate length of data to indicate what we term "memory" in the data. For the purposes of this paper we assume that using four initial data points is sufficient to account for the memory. Given the fact that the behaviour observed in Figure 1 is highly oscillatory a short memory span of four points seems sensible for a data series of 100 points as was chosen here. This indicates that any rational underlying dynamics are not observed in the long term but rather in the short term. We then curve-fit through the initial four points of the vector which represents our actual known data points, y_1, y_2, y_3, and y_4. We use this fitted curve to predict the value of the fifth point, y_5. We then use data values two to five, y_2, y_3, y_4, and y_5, to predict the sixth value, y_6. Continuing in this way we end up with a vector of actual known values and

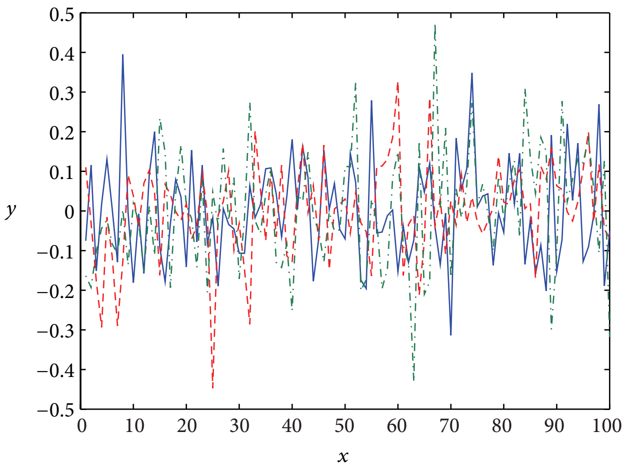

Figure 1: Plot of three different simulated data series.

a vector of predicted values—four data points shorter than the original.

Our aim is to predict—and improve upon—the direction of the data, that is, whether the quantitative value is positive or negative, which is indicative of whether market prices are moving up or down. Since most forecasting systems are far from accurate when predicting the quantitative value, predicting direction is far easier and can be equally profitable. For instance, for a system that draws its conclusion of how to trade tomorrow from the closing price of today's action, getting the direction is vitally important. We are not looking at this from a multisignal/asset point of view which would require the determination of how much to invest in each asset. This would be along the lines of an efficient frontier in Modern Portfolio Theory which would take factors like standard deviation and error in the forecast into account. Rather, we are simply wishing to trade on the back of successfully predicting direction. Thus while considering the distribution of the time series itself and the relevant mean and standard deviation may be more accurate quantitatively, trading on the predictions of the direction the prices are moving in can in itself be very profitable.

As a consequence the success of our methodology is calculated by considering the percentage of times the sign of the predicted data matches the sign of the actual originally generated data. To improve the accuracy of predicting direction we make use of the fact that the direction is just the gradient of the data. By creating a vector of gradients we have the numerical representation of an ordinary differential equation. We then use the structure of a BDF method to numerically solve the ordinary differential equation. BDF methods are appropriate because they depend on previous values. Some examples of BDF methods for solving the first order ordinary differential equation

$$y' = f(t, y) \tag{1}$$

are given by

$$y_{n+1} = y_n + hf_{n+1}, \tag{2}$$

$$y_{n+1} = \frac{4}{3}y_n - \frac{1}{3}y_{n-1} + \frac{2}{3}hf_{n+1}, \tag{3}$$

$$y_{n+1} = \frac{18}{11}y_n - \frac{9}{11}y_{n-1} + \frac{2}{11}y_{n-2} + \frac{6}{11}hf_{n+1}, \tag{4}$$

where we have used the conventions $y_n = y(t_n)$ and $f_{n+1} = f(t_{n+1}, y_{n+1})$.

Using a forward difference approximation to the derivative we find that

$$f(t_{n+1}, y_{n+1}) = \frac{y^*_{n+1} - y_n}{h}, \tag{5}$$

where y^* is the value we are trying to improve. The BDF method (3) becomes

$$y^{*(j+1)}_{n+1} = \frac{2}{3}y^{*(j)}_{n+1} + \frac{2}{3}y_n - \frac{1}{3}y_{n-1}. \tag{6}$$

Similarly, by approximating the derivative by a central difference approximation we obtain

$$y^{*(j+1)}_{n+1} = \frac{1}{3}y^{*(j)}_{n+1} + \frac{4}{3}y_n - \frac{2}{3}y_{n-1}. \tag{7}$$

As stated in their names, BDF methods are backward approximations of the first order derivative in a first order ordinary differential equation. In this instance, however, we are not applying the BDF method to an ODE but rather to actual discrete data points. Equations (2)–(4), being discrete, are applicable when considering actual data instead of an ODE as is usually the case. Since we do not have a function f, as per (1), we take an approximation of the first order derivative as per (5). The BDF method approximates a differential equation whereas finite difference approximations could be said to be a means of approximating the gradient making it a convenient way of approximating f. This means that a lower order approximation of a gradient, that is, the finite difference approximation of f, is incorporated into a higher order approximation of what we can term our discrete ODE.

Thus (6) mixes first and second order approximations of the first order derivative. Equation (7) mixes a centred approximation at position x_n and a backward approximation at position x_{n+1} which is equivalent to a 0 order approximation. The reason why all the terms do not cancel out in formulae (6) and (7) and why this approach is still relevant is due to the fact that the formulae mix different order approximations. Our purpose is to improve upon already obtained predictions—or an already obtained solution. More precisely our predictions are our y function and it is exactly the y function as discrete points which we aim to improve upon in the same way predictor correctors are used to improve upon solutions obtained via other numerical methods such as Euler's method. Hence we are in fact not solving the ODE again, only improving on already known data in a pointwise fashion.

It is important to note that we are not implementing the BDF method in a direct fashion. We are incorporating a lower order approximation into the method in order to obtain a means of improving on already generated data. In the computational implementation of this work, the BDF method given by (6) is iterated to convergence. The initial value $y_{n+1}^{*(0)} = y_{n+1}$ is obtained from either a linear or spline interpolation. The value of y_{n+1} can be obtained using a neural network as discussed in other literature [14–16].

3. Convergence Properties

In this section we investigate the convergence properties of the difference equations (6) and (7). We want to show that

$$\left| y_{n+1}^{*(j+1)} - y_{n+1}^{*(j)} \right| \longrightarrow 0, \tag{8}$$

by considering the general solution to (6) which is

$$y_{n+1}^{*(j)} = 2y_n - y_{n-1} + \left(\frac{2}{3}\right)^j \left(y_{n+1}^{*(0)} - 2y_n + y_{n-1} \right). \tag{9}$$

By considering (9) as follows

$$\left| y_{n+1}^{*(j+1)} - y_{n+1}^{*(j)} \right| = -\left(\frac{1}{3}\right)^j \left(y_{n+1}^{*(0)} - 2y_n + y_{n-1} \right), \tag{10}$$

we find that as $j \rightarrow \infty$ (8) holds. In a similar fashion it is relatively easy to show that (8) is satisfied for (7), indicating convergence. In fact, scheme (7) converges faster than (6) since the coefficient of $y_{n+1}^{*(j)}$ is smaller.

When either (6) or (7) is iterated to convergence we have $y_{n+1}^{*(j+1)} = y_{n+1}^{*(j)} \pm \epsilon = y_{n+1}^{*} \pm \epsilon$, where $\epsilon \ll 1$ is the tolerance. For both (6) and (7) we find that at convergence

$$y_{n+1}^{*} = 2y_n - y_{n-1} \mp \epsilon. \tag{11}$$

It is in fact the right hand side of (11) which indicates the "correction" made to the original prediction. By implementing a lower order approximation within a higher order approximation as we have done in the previous section we have been able to obtain this necessary difference equation which is the manner in which the predictions are improved upon.

The results and concluding remarks are presented in the next section.

4. Results and Concluding Remarks

As a means of evaluating the effectiveness of the BDF method as implemented above to improve upon the prediction of the direction of data we compare the accuracy of the originally fitted data and the corrected data. Direction success rate can be seen as a simple bimodal result. If we let y_{n+1}^{T} be the true value then

$$y_{n+1}^{T} y_{n+1}^{*} = \begin{cases} \text{positive} \Longrightarrow \text{success}, \\ \text{negative} \Longrightarrow \text{failure}. \end{cases} \tag{12}$$

Choosing a sign based criterion is a standard method in financial analysis. The criteria are appropriate within the context of stock market modelling given the growing interest in developing tools which can give a sense of the *direction* in which the market may move or when possible turning points or shocks in returns will occur. This information is critical irrespective of the actual numerical value assigned to the movement given that the numerical values of the data used and analysed within this context are not necessarily useful in identifying characteristics which can be used to predict future movements. Hence, if the return on a stock decreases from 0.15 to 0.14, that is, by 0.01, for example, a prediction of −1 is of more value than a prediction of +0.01. This is due to the fact that the former prediction indicates the market trend whereas the latter only indicates a numerical value which within the context of market modelling loses value since it indicates a rise in the return at the next time step which is false. Thus the numerical value does not hold as much value in market trend trading as does the *direction* indicated by the sign of our prediction. In terms of our criterion, we have simply followed convention based upon methods currently employed by traders who trade according to market directions or trends.

In Table 1 we compare the percentage accuracy of predictions obtained through fitting a linear curve or spline through four points and the improved predictions found via (6). On average, this simple approach ensures that we get the direction correct 87% of the time in comparison to 48% for a linear curve and 86% when compared to 52% for a spline curve. These percentages have been calculated by simply counting the amount of times the sign of the actual point is generated, such as the three samples represented in Figure 1, and either the fitted or corrected point is the same and divided by the overall number of points. The number of points used—as indicated before—was 100. This is an arbitrary choice simply done for convenience.

Table 1 reflects the implementation of (6) obtained from the BDF method (3) and by using a forward difference approximation for the first derivative. If we instead incorporate a central difference approximation as per (7) we are able to improve a direction success rate of 50% for a linear curve to 80%. A comparison done against the spline fitting showed an improvement from 50% to 79%. We have also considered (4) with a forward and central difference approximation for the first derivative, respectively. The former choice showed an increase from approximately 48% to 80% for the linear comparison and 49% to 79% for the spline comparison. The latter case indicated an increase in the accuracy of the direction from 50% to 81% when the original predictions are obtained via a linear fitting and 49% to 81% when a spline is implemented.

In this paper we have shown how the implementation of the BDF method with a lower order approximation of the gradient of discrete data can improve the accuracy of predicting the direction of random data. The motivation for using a numerical ordinary differential equation solver comes from the fact that direction is just a gradient. We form a discrete ordinary differential equation from our vector of predictions. This discrete ordinary differential equation is solved with the implementation of a lower order approximation of the gradient with a BDF method. We find that the accuracy of our predictions improves from an accuracy of approximately 50% to an accuracy of approximately 87%.

Table 1: Table comparing percentage accuracy of predicting direction.

Linear		Spline	
Fitted	Corrected	Fitted	Corrected
51.0417%	80.0000%	56.2500%	88.4211%
52.0833%	92.6316%	50.0000%	86.3158%
52.0833%	90.5263%	52.0833%	86.3158%
45.8333%	87.3684%	47.9167%	88.4211%
55.2083%	92.6316%	41.6667%	85.2632%
46.8750%	85.2632%	53.1250%	91.5789%
48.9583%	84.2105%	48.9583%	85.2632%
42.7083%	85.2632%	59.3750%	86.3158%
35.4167%	87.3684%	44.7917%	76.8421%
48.9583%	88.4211%	52.0833%	85.2632%

An advantage of the approach taken in this paper is that the BDF scheme is a marching scheme. Irrespective of how big the data set is, the scheme will march through the data accordingly. The "speed" of the algorithm on very large data sets can be improved upon by using a computer with a faster CPU.

Acknowledgment

E. Momoniat and C. Harley acknowledge support from the National Research Foundation of South Africa.

References

[1] S. M. Keane, *Stock Market Efficiency*, Philip Allan Limited, Oxford, UK, 1983.

[2] B. G. Malkiel, *A Random Walk Down Wall Street*, W. W. Norton & Company, Inc., New York, NY, USA, 6th edition, 1973.

[3] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 1, John Wiley & Sons, New York, NY, USA, 3rd edition, 1968.

[4] D. Williams, *Probability with Martingales*, Cambridge University Press, Cambridge, UK, 1991.

[5] A. W. Lo and A. C. Mackinlay, *A Non-Random Walk Down Wall Street*, Princeton University Press, Princeton, NJ, USA, 5th edition, 2002.

[6] H. Fromlet, "Behavioral finance-theory and practical application," *Business Economics*, vol. 36, no. 3, pp. 63–69, 2001.

[7] J. Lakonishok, A. Shleifer, and R. W. Vishny, "Contrarian investment, extrapolation, and risk," *The Journal of Finance*, vol. 49, no. 5, pp. 1541–1578, 1994.

[8] M. Sewell, "Behavioural Finance," University of Cambridge, 2007, http://www.behaviouralfinance.net/.

[9] W. F. M. de Bondt and R. Thaler, "Does the stock market overreact?" *The Journal of Finance*, vol. 40, no. 3, pp. 793–805, 1985.

[10] L. K. C. Chan, N. Jegadeesh, and J. Lakonishok, "Momentum strategies," *The Journal of Finance*, vol. 51, no. 5, pp. 1681–1713, 1996.

[11] K. Daniel, D. Hirshleifer, and A. Subrahmanyam, "Investor psychology and security market under- and overreactions," *The Journal of Finance*, vol. 53, no. 6, pp. 1839–1885, 1998.

[12] H. Hong and J. C. Stein, "A unified theory of underreaction, momentum trading, and overreaction in asset markets," *The Journal of Finance*, vol. 54, no. 6, pp. 2143–2184, 1999.

[13] M. Berman, *Cum Dividend Irrational Pricing Behaviour in US REITs [Ph.D. thesis]*, Rushmore University, 2007.

[14] K. Chakraborty, K. Mehrotra, C. K. Mohan, and S. Ranka, "Forecasting the behavior of multivariate time series using neural networks," *Neural Networks*, vol. 5, no. 6, pp. 961–970, 1992.

[15] G. Zhang, M. Y. Hu, B. E. Patuwo, and D. C. Indro, "Artificial neural networks in bankruptcy prediction: general framework and cross-validation analysis," *European Journal of Operational Research*, vol. 116, no. 1, pp. 16–32, 1999.

[16] Z. Tang and P. A. Fishwick, "Feed-forward neural nets as models for time series forecasting," *ORSA Journal of Computing*, vol. 5, pp. 374–385, 1993.