International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:8, No:8, 2014

Novel GPU Approach in Predicting the Directional Trend of the S&P 500

A. J. Regan, F. J. Lidgey, M. Betteridge, P. Georgiou, C. Toumazou, K. Hayatleh, J. R. Dibble

***Abstract*—**Our goal is development of an algorithm capable of predicting the directional trend of the Standard and Poor's 500 index (S&P 500). Extensive research has been published attempting to predict different financial markets using historical data testing on an in-sample and trend basis, with many authors employing excessively complex mathematical techniques. In reviewing and evaluating these in-sample methodologies, it became evident that this approach was unable to achieve sufficiently reliable prediction performance for commercial exploitation. For these reasons, we moved to an out-of-sample strategy based on linear regression analysis of an extensive set of financial data correlated with historical closing prices of the S&P 500. We are pleased to report a directional trend accuracy of greater than 55% for tomorrow (t+1) in predicting the S&P 500.

***Keywords*—**Financial algorithm, GPU, S&P 500, stock market prediction.

I. Introduction

CURRENT methods of forecasting financial markets require computationally intense algorithms because the parameter inputs needed to create a meaningful prediction are extremely large. Two of the most popular methods used are Artificial Neural Networks (ANNs) [1] and Support Vector Machines (SVMs) [2], which are able to predict financial markets with some success. Both approaches have inherent advantages and disadvantages. An ANN involves a network of processing artificial neurons that can exhibit complex global behavior, determined by the connections between the processing elements and element parameters [3]. The disadvantage of the ANN approach lies in scalability; whilst an ANN is possible to implement in hardware, it is computationally inefficient and power hungry.

The standard SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the input, making the SVM a non-probabilistic binary linear classifier [4]. SVMs are not readily scalable and cannot be easily implemented onto a dedicated hardware [5], and hence we believe are unsuitable for predicting the directional trend of the S&P 500.

The aim of this work is to develop a forecasting algorithm for financial markets, which overcomes the scalability limitations inherent in both ANNs and SVMs. The expectation is that this should be able to process extremely efficiently in parallel, and at great speed, to obtain meaningful out-of-sample results. The S&P 500, which comprises the 500 largest US companies based on market capitalisation, was chosen as the test index. The scale and complexity of the hugely liquid S&P 500 creates an incredibly difficult entity to predict.

The implementation is capable of significant gains when choosing optimal parameters for forecasting future financial data. We have adopted a radically different approach in our work, namely to predict tomorrow's value of the index by looking closely at a wide range of financial factors that influence the financial performance of the member companies within the index, and link the predicted values back to the past values of the index. The values of these financial market data (FMD inputs) are readily available and include items such as currency pairs, key commodity indices and other financial indices. These FMD inputs are then used in a linear regression algorithm to compute tomorrow's (t+1) value of the S&P 500 index. Changing the number of FMD inputs affects the directional accuracy of the prediction result; increasing the number of FMD inputs significantly increases the computational complexity and hence the time taken to compute a daily result. Another key aspect of our algorithm is that its architecture is intended for commercial application. Therefore, it is imperative that each prediction result must be computed in less than 24 hours. This is so a result can be obtained today (t) for tomorrows (t+1) US market open.

II. Architecture of the Algorithm

Development of financial forecasting algorithms that depend entirely on historical data are referred to as in-sample modeling [6]. Algorithms that rely entirely on in-sample modeling are effectively trend based. However, as Granger [7] stated, "one of the main worries about the present methods of model formulation is that the specification search procedure produces models that fit the data spuriously well, and also makes standard techniques of inference unreliable". Out of sample testing is essential to guard against curve fitting [8]. Many authors have attempted to use an in-sample approach with moderate success. However, the fundamental assumption that the future is entirely determined by the past misses the facts that an index such as the S&P 500 is an aggregation of the performance of 500 individual companies. In recognizing this limitation, we have decided to develop an algorithm based on an alternative methodology.

The key elements that underpin our system are:

1. Identification of 51 financial market data (FMD) inputs,

A. J. Regan is the Chief Executive Officer of Corvus Capital. He is also studying for a PhD at Oxford Brookes University, UK (e-mail: ar@corvus.com).
F. J. Lidgey is a Professor Emeritus at Oxford Brookes University, UK.
M. Betteridge is the Chief Executive Officer of Preciousbluedot Ltd., UK.
P. Georgiou is a Lecturer at Imperial College London, UK.
C. Toumazou is a Professor at Imperial College London, UK.
K. Hayatleh is a Professor at Oxford Brookes, UK.
J. R. Dibble is an Analyst at Corvus Capital.

including other indexes, currency pairs, swap rates, etc., that we have proved influence the movement of the S&P 500.

2. The use of an extensive historical data set (actual daily closing prices of the chosen 51 FMD inputs and S&P 500).
3. The ability to compute this large data set (comprising more than 12.7 billion combinations) in a time frame of less than 24 hours.

The data set is fed into a linear regression algorithm to determine the predicted value of tomorrow's (t+1) S&P 500 closing price.

III. Distributed Algorithm

What we require is to rapidly compute the best results of a given subset of parameters. In order to reduce the complexity of the algorithms architecture, the available data was inspected and reduced down to 51 key FMD inputs. Having identified these 51 significant market parameters for forecasting the directional trend of the S&P 500, the result is a mathematical function that is computationally intense. Fig. 1 displays the number of possible combinations increasing as each FMD input is added. Increasing the number of FMD inputs above 10 produces an unmanageably large data set for even the fastest computational technology to process within our stringent 24-hour time limit.

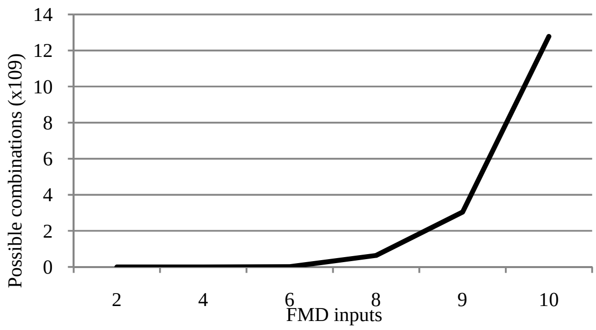

Fig. 1 Possible number of RPA combinations vs. FMD inputs

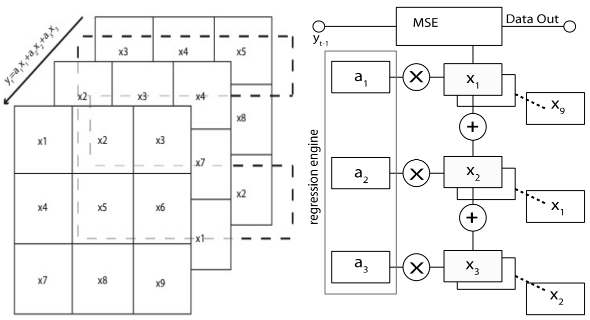

Fig. 2 Distributed architecture for forecasting the S&P 500 for 3 FMD inputs. (left) Regression core, (right) data bus for comparison

The requirements for a distributed implementation of the S&P 500 consist of two key components, a regression core, which forecasts using a given set of parameters, and a data bus, which is used to shift information in a daisy chain fashion to compare all the results. This architecture can be seen in Fig. 2.

The regression core computes a simple linear regression based on the parameters described by (1):

$$y_t = a_1x_1 + a_2x_2 + a_3x_3 \tag{1}$$

where y_t is our forecast, x_n is the input data and an are the weights which are optimized by computing the minimum mean squared error (MSE) of y_t with y_{t-1}.

The aim is then to select the 'best' FMD input parameters x_1, x_2, x_3 (in this case) from a given subset of FMD input parameters, 51, which affect S&P 500. The algorithm works on the basis that it can compute the best MSE for that given set. To select the best parameters, a data bus is used which shifts the input parameters such that a regression is computed for all possible combinations.

As is shown in Fig. 1, the initial architecture will compute in parallel 10 possible combinations at a time, and then sequentially cycle through the FMD input parameters until all possible combinations are evaluated. Throughout this process the optimum set of parameters yielding the minimum MSE is constantly saved. These are then used to conduct the forecast.

IV. Proof of Concept

In order to test the feasibility of the algorithm, initially it was programmed in MATLAB and demonstrated on a 12-core Central Processing Unit (CPU). Results shown in Table I confirm the functionality of the algorithm achieving a directional efficiency of greater than 50% for a minimum of six inputs. However, the time to find the best six inputs was extremely long given the limitations of processing in MATLAB. In order to fully exploit the potential of the algorithm and to conduct a more exhaustive study of the optimum number of input parameters in less than our 24-hour time constraint, a more computationally powerful hardware platform was needed.

TABLE I
Directional Results, Accuracy and Speed of Algorithm on a Pc Based System

51 FMD Inputs data	1	2	4	6
Directional Result	11/30	14/30	15/30	17/30
RMSE	28.57	33.02	24.784	19.446
Ave. Time Elapsed	0.02760	.66196	16.3922	36096.65

V. GPU Based Computational Engine

In order to successfully meet the requirement of handling such a large data set to complete the prediction target on time, we decided to adopt a novel Graphics Processing Unit (GPU) based computational engine which we anticipated would overcome the limitations of using MatLab on a conventional CPU system.

To support this, we decided to build a bespoke hardware unit, which consisted of a 6-core CPU (i73930K) and 3 GPU cards (EVGA GeForce GTX TITAN [9]), supported with 32GB of RAM. Each of the 3 GPU cards houses 2688 cores. The software is designed so that the CPU first initialises and organises the data, before feeding to the 8064 GPU cores, which undertake the extremely intensive processing in parallel. The algorithm is particularly suited to a GPU approach as we were able to re-engineer the algorithm from a linear execution path (as it was in MatLab), to a parallel execution path. This enabled us to utilise the full 8064 cores, all of which calculate in parallel, before passing the results back to the CPU for analysis.

In order to fully utilise the performance of the 8064 cores, a number of items were critical and needed to be thoroughly explored: for instance it is extremely important to organise the pipeline and the memory for optimum performance because severe cache misses will stall the pipeline and be extremely detrimental to performance. The modern GPU has been designed around handling very specific data, namely vertices, textures and shading models, and is unlike a purely CPU architecture, which has been designed to efficiently handle a huge variety of memory accesses. A deep understanding of this process allowed us to optimise and reconfigure the data set to be 'GPU friendly' in a format that perfectly suits its architecture for use in financial prediction. To highlight the importance of organising the data, it should be noted that in early tests almost 90% of theoretical performance was lost due to the GPU pipeline stalling caused by poor memory configuration.

Through this extensive optimisation of our GPU architecture, we were able to achieve a sufficient speed up to meet key element (3), reducing the initial MatLab processing time which was considerably over 24 hours, by approximately 95% with a non-optimised GPU and further reducing the procession time by a further 87% on the fully optimised GPU, as shown in Fig. 3.

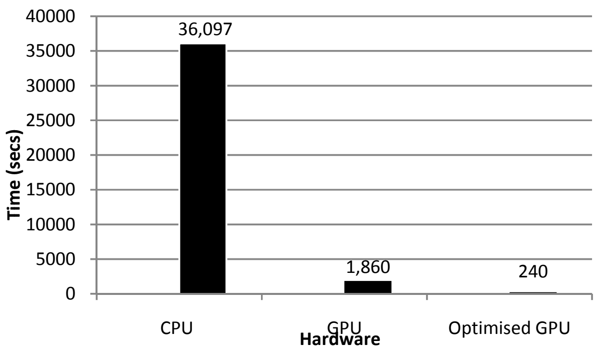

Fig. 3 Processing time of algorithm – 6 FMD inputs

VI. Results and Evaluation

The In achieving our final and most accurate directional trend result in predicting the S&P 500, it was necessary to conduct an extensive range of tests. We explored a number of trade-offs which were tested using an 8-year data set of both the S&P 500 closing price and the chosen 51 FMD inputs. We focused on three significant trade-offs that we believed would influence the result of our algorithm to establish an optimum result in predicting the S&P 500:

1) Testing to determine the optimum number of historical days data that is used to create each prediction cycle. The optimum was established at 60 days.
2) Testing to determine the optimum number of FMD inputs to be selected when computing the algorithm. This varied from 1 to 10 and we established 9 to be optimal.
3) Testing whether a lag exists between the movement in value of each FMD input and the price of the S&P 500. We therefore tested establishing a prediction for tomorrow (t+1) to four trading days later (t+4). We concluded that no lag exists, hence (t+1) remains the optimum.

Having established the optimum parameters for our algorithm and in doing so generating a successful directional trend of greater than 55%, it is important to explain the architecture of each individual test cycle. Depending on market conditions at the time, the FMD inputs to form the basis of the prediction change. Each daily prediction cycle produces a ‘Figure of Merit’ (FoM), a weighted total of the chosen FMD inputs for that individual (daily) cycle. Fig. 4 displays the FoM in the form of a pie chart.

Fig. 4 represents one 60-day cycle computed by our algorithm. This cycle used the prior 60 days ending at time = t = 24th November 2008 to predict (t+1) = 25th November 2008. This particular test was carried out selecting 6 FMD points (of the total set of 51), these are the 6 inputs with the lowest MSE against the daily movement of the S&P 500 closing price.

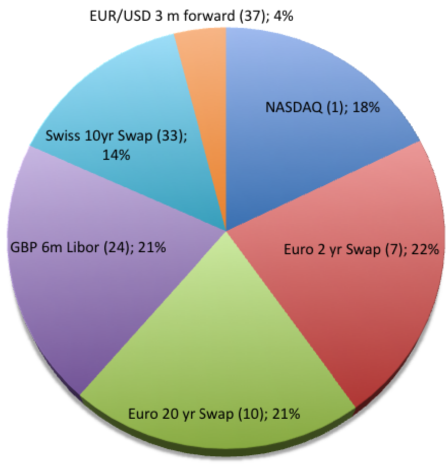

Fig. 4 Figure of Merit weighted total – 6 FMD inputs

Fig. 4 represents one 60-day cycle computed by our algorithm. This cycle used the prior 60 days ending at time = t = 24th November 2008 to predict (t+1) = 25th November 2008. This particular test was carried out selecting 6 FMD points (of the total set of 51), these are the 6 inputs with the lowest MSE against the daily movement of the S&P 500 closing price. It is important to note that depending on daily changing economic conditions, this weighted total will change; different inputs representing different weightings will emerge as the economic climate alters. This dynamic and automatic feature of our algorithm is imperative and ensures that the algorithm is constantly adapting to the ever-changing economic environment by producing an entirely different weighting of FMD inputs each day (cycle). This approach is particularly important when predicting the complex aggregated S&P 500 index.

As a result of establishing the optimum parameters for the algorithm, we can confirm that we have obtained a directional trend of greater than 55% in predicting the S&P 500 at (t+1).

VII. Conclusion and Future Work

A novel architecture algorithm for predicting S&P 500 has been designed and implemented. The data that needs to be processed within a 24-hour period demands a very high speed and highly parallel computation engine which was realized using an advanced GPU design, optimised to meet our particular requirements. The work we have presented here is on-going. The authors plan to refine the design further to improve accuracy and also and adapt it for other financial market prediction tasks, such as movement of currencies and other key stock indices.

Acknowledgments

The authors gratefully acknowledge the support received from Corvus Capital and Preciousbluedot Ltd for their contribution to this work.

References

[1] Hamid, S. “Primer on Using Neural Networks for Forecasting Market Variables” Working Paper No. 2004-03
[2] Sapankevych N. and Sankar R., “Time Series Prediction Using Support Vector Machines: A Survey,” Computational Intelligence Magazine, IEEE, vol. 4, no. 2, pp. 24-38, May 2009
[3] Veerabhadram, P., Lombard, A. and Conradie, P. “Artificial Neural Network” International Journal of Scientific & Engineering Research, vol. 3, no. 2, February 2012
[4] Yin, Y., Han, D. and Cai, Z. "Explore data classification algorithm based on SVM and PSO for education decision." Journal of Convergence Information Technology 6.10, 122-128, 2010
[5] Prosser, B et al. "Person Re-Identification by Support Vector Ranking." BMVC. Vol. 1. No. 3. 2010.
[6] Khan, A., Bandopadhyaya, T. K., & Sharma, S. “Genetic algorithm based backpropagation neural network performs better than backpropagation neural network in stock rates prediction” Journal of Computer Science and Network Security, 8(7), 162-166. 2008
[7] Granger C. “Modeling economic time series: Readings in econometric methodology”, Oxford University Press: Oxford, UK, 1990
[8] Inoue, A. &Kilian, L. “In-sample or out-of-sample tests of predictability: Which one should we use?” Econometric Reviews, 23(4), 371-402, 2005
[9] EVGA, 2014. EVGA GeForce GTX TITAN BLACK Superclockedhttp://www.evga.com/Products/ProductList.aspx?type=0&family=GeForce+TITAN+Series+Family&chipset=GTX+TITAN+BLACK [Accessed 26.06.14]