Temporal Pattern Matching for the Prediction of Stock Prices

Richi Nayak & Paul te Braak

Faculty of Information Technology
Queensland University of Technology, Brisbane

r.nayak@qut.edu.au

Abstract

Time series data poses a significant variation to the traditional segmentation techniques of data mining because the observation is derived from multiple instances of the same underlying record. Additionally, the standard segmentation methods employed in traditional clustering require instances to be classified exactly by attaching an event to a specific cluster at the exclusion of other clusters. This paper is an investigation into the predictive power of the clustering technique on stock market data and its ability to provide stock predictions that can be utilised in strategies that outperform the underlying market. This uses a brute force approach to the prediction of stock prices based on the formation of a cluster around the query sequence. The prediction is then applied in a model designed to capitalise on the derived prediction. The predictive accuracy of minimum distance clusters produced promising results with a prediction error incorporated into the forecast strategy.

1 Introduction

Time series data captures changes to an object that is typically measured at equal time intervals [2]. This type of representation is common in everyday applications, so is not surprising to read that a study on random sample of 4,000 graphics from 15 of the world's news papers published between 1974 and 1989 found that more than 75% of all graphics were time series [1]. The use of time series data is of growing importance [3] and is common in many applications ranging from scientific databases with sensor data (weather, geological, environmental, astrophysics) [9] to financial applications.

'Temporal data mining' can loosely be defined as any data mining application executed over temporal (time series) data. Lin, Orgun and Williams [8] classify these applications as two fundamental exercises. Firstly, the matching of sequence patterns and secondly, the recognition of periodical patterns in time series databases. This paper is focused at applying the first of these mining applications over stock market data in order to predict the future price of a sequenced set of stock price information. Specifically, it is the goal of this paper to investigate the use of cluster prediction as an efficient mechanism of trading strategy.

This approach employs several existing concepts that are found in the traditional data mining suite of definitions by applying clustering to improve reliability around prediction. The proposed method can be summarised as the formation of a cluster around the query (sequence) based on a minimum distance. It is proposed that the use of this approach will offer a measure of strength to the prediction that is not available with traditional cluster based data mining. This allows querying and decision rules to become somewhat fuzzy. Additionally, in testing the algorithm on actual Australian Stock Exchange (ASX) data, the commercial reality of the method is appraised. The ability of the model to 'outperform' the market is also examined as an evaluation method.

This paper is organised as follows. Section 2 examines previous work. Section 3 covers the proposed method and output of data mining by introducing the definitions, concepts and algorithms used in this study. Section 4 analyses the experiments and their outcomes. Section 5 examines the future application and improvement of the study.

2 Previous Work

Hellstrom in [10] identifies that the prediction of stock market data is a very difficult task. The Efficient Market Theory (EMT) was proposed by Eugene Fama (1965) and specifies that a price of a stock reflects all known information about it at that point in time. One of the implications of this theory is that prediction based on prior trading patterns is not possible since no stock can encompass the same information and have the market receive it in the same manner. It is argued by Fama that the price of a stock follows a random walk (unpredictable movement).

Notwithstanding this, entire trading schemes have been developed based on the stock history. This discipline is commonly referred to as technical analysis and is believed to have its foundations in candlestick charting and the Dow Theory [11, 12]. Some authors [2, 11] have commented that the typical techniques for the prediction of time series (stock market data) lie in the analysis of trend lines and therefore relationship of values immediately prior to the prediction point. A common approach is to use a moving average (simple or weighted) as the predictor, indicating the expected future result is directly relational to past performance.

There is no universal acceptance of the ability to predict stock market data. Povinelli [12] in work with others [13]

has claimed that temporal event matching exceeded baseline returns, and Marketos et al [14] have proposed significant pattern extraction for trading strategies. There are also numerous commercially available products (eg incredible charts [15], eASCTrading [16]) that claim to predict stock returns based on trend analysis techniques.

Agrawal, Faloutsos and Swami [17] have identified whole and subsequence matching. Whole matching compares two sequences of the same length (to determine a measure of similarity) whereas subsequence matching compares a smaller query sequence within a larger sequence to find the closest match(s). These concepts have also been identified by others [2, 4, 18]. Keogh et al [18] also has expanded the subsequence matching methodology to include a sliding window over time series data which efficiently derives subsequence.

The methods proposed in this project therefore have its foundations in whole sequence matching [7] by utilising a sliding window [18] to derive sequences for comparison. The sequence index and length conventions discussed above were also used by Faloutsos, Ranganathan and Manolopoulos [9].

Advanced and dynamic methods of time series prediction [19] have incorporated data streams and prediction outcomes in a continuous method by utilising prediction error in the result. The approach at [19] incorporates a nearest neighbour methodology in the prediction of the (streamed) time series event in providing fast responses and Fast Fourier Transformation (FFT) in the indexing of results. Other authors [20] incorporate the need for dynamic prediction formed around continuous streaming inputs. The authors at [20] also propose the recording of results for addition and future reference and incorporate weighted subsequence distance measures in the determination of similarity.

In 2003 Povinelli and Feng [21] have extended the previous work by Povenelli (and others [12, 13]) to form predictions that are based loosely around a desired outcome. That is, given a significant event, determine if the input stream (current query) is likely to produce that event. It may be said that the approach adopted by Povinelli and others focus on the identification of event(s) and not the continuous prediction of a time series. Last et al in [22] discuss the use of important feature recognition in the prediction of time series data (albeit neural networks) which may justify the event recognition proposed by Povinelli ([12], [13], [21]).

3 Proposed Method

This paper seeks to accurately predict stock price changes through the temporal clustering of (immediately) prior price sequence. That is, given a stocks consecutive price information (day 0 – N), predict the price at the next day (day N+1).

The proposal in this paper centres on the idea that a cluster formed around an event (query) should be a good predictor to the future of the event. The cluster is formed around each query on the basis of a 'best fit error margin' which specifies it must be formed by using the least distance possible. Reference to minimum distance implies a quasi nearest neighbour (KNN) methodology. However, unlike KNN the number of neighbours is not relevant to the classification of fit. In this situation the distance to the query sequence is considered most important and the intention is to forecast based on this proximity. Additionally, the actual minimum distance of the results obtained should indicate the predictive power of those results.

The distinctions between these two methods may best be demonstrated by an example. If we consider a query(Q) and employ the traditional KNN clustering technique, the user stipulates that they require an exact number of matches (ie the K in KNN) before a prediction can be made. The user may specify the number of observations required (K) is five and therefore prediction will be based on the five closest matches. If three of the matches are almost exact and the other two only slackly based around the query, then the three exact matches would be the best indicator of the expected outcome, not the entire five. Furthermore the average distance of the three 'exact' matches will imply a better and more precise fit than the average distance of the five matches. The distance of the results can therefore be used in rational decision making (example, only act on predictions where the average distance for results is below a certain threshold).

The standard data mining naming conventions of a training data set (being known history) and a validation dataset (set requiring prediction) are used in this paper. Further definitions are defined as follows.

Window (W) specifies the complete series of trading information for all stocks in a given dataset. The window can further be defined by its source (eg *Window(Training)* which represents a window formed over training data whereas *Window(Validation)* represents a window formed over validation data). The window has a *Length* which represents the number of discrete observations (values from *0 to (Length-1)).* The horizontal position in the window (its x axis position) is referred to by the *Index* and is in the range of 0 to (Length – 1). All trading information (trading date, stock code and price) is captured in the widow and can be returned by specifying a single position *(Index)* and the required attribute (*Window.TradingDate,Window.StockCode, Window.TradingPrice*)

Sequence (S) specifies a consecutive series of normalised trading information for a particular stock. The *Sequence* can also be defined by its *Length* and therefore, the *Sequence* is a subset of a window that satisfies the following conditions: (1) The Sequence represents data for only a single stock, and (2) The time intervals represented by the Sequence Index position (*Sequece[i]*) which are equidistant and consecutively incremental. Therefore, the sequence cannot contain 'missing blocks' of trading days. A Sequence represents a snapshot of a stocks uninterrupted trading and all information can be derived through the following attributes.

Sequence.Valid specifies that the sequence meets the conditions above. *Sequence.StockCode* is the name of the stock that the sequence is formed over.

Sequence.StartDate is the initial date that the sequence is reflected from. *Sequence.Length* is the number of observations (length) of the sequence.

Sequence[n] is the normalised value of the position *n* in the sequence (where *n* is a number from 0 to *Sequence.Length* - 1. The value is normalised over the opening *Sequence.StartDate*'s value. The sequence also holds an additional value (*Sequence.Day1*) being the normalised value of the sequence for the first day after the sequence finishes. This value is logically *Sequence[Sequence.Length]* but is stored as a distinct variable for predictive purposes. This is because, in a predictive model *Sequence.Day1* is not known.

Error Margin(ε) specifies the acceptable margin of error that defines one *Sequence* similar to another. That is, two sequences *S1, S2* are considered close if the distance between them is lower than the error margin ($D(S1,S2) < \varepsilon$).

Graphically this relationship is represented by in figure 1. In this figure, the window is formed over an entire dataset and holds all the stock information in it. It is shaped by stock code and trading date which means that the horizontal positions are ordered around stock code and trading date. That is, all information for one stock code is added, then the information for another and so on. The stock information is added by date order. The data point represented above show the stocks price on that day, and, moving along the horizontal axis shows the day by day price movements for the stock (note this window is formed over a single stock). The Sequence is formed by taking a subset of the window and normalising the prices around the opening day. Hence, in figure 1, an index position in the window (9) is used as a starting point for forming the sequence represented by the red box. The index position of the window is used to supply the {stock code x trading date} → $ price relationship for all elements (price values) in the sequence. The (normalised) price values can be retrieved by their position (index) in the sequence.

3.1 Search Algorithm

As discussed above, the prediction of a value is determined by forming a cluster around a query sequence. The value that is the target for prediction is the first value after the end of the sequence, that is, the theoretical value held in the sequence position Sequence[Sequence.Length].

This is achieved through the use of two datasets. Firstly, a validation dataset is used to hold query sequences (the data targeted for prediction). In addition to the query sequences, the validation dataset also contains the actual future value (those we are trying to predict) so that it may be compared to the predicted value for accuracy. Note that in a dynamic prediction environment, this would not be the case and there would be no known future values. Secondly, a training dataset holds a large amount of historical information so that each query sequence can be compared to every sequence in the training data set. The sequences in the training dataset hold the 'future' values (the next logical position in the sequence) and these values are used as a basis for prediction.

Window("Validation") → A Window formed on Validation Data

Window("Training") → A Window formed on Training Data

Sequence("Validation") → A sequence formed on Validation Data

Sequence("Training") → A Sequence formed on Training Data

Error Margin → A minimum measure of similarity that must be satisfied to consider

Algorithm:

```
For each Sequence that exists in the Window("Validation")
    Get Sequence("Validation")
    Let Error Margin be a minimum value
    While Sequence("Validation") results = nothing do
        For each Sequence that exists in the Window("Training")
            Get Sequence("Training")
            If Sequence("Validation") Is similar to Sequence("Training") then record result
            Next Sequence
        If there are results then output
        Else Increase the Error Margin by a small value
```

Table 1: The data mining operation

Diagrammatically, the method of querying is shown in figure 2. Note that the query sequence and a training sequence should be of the same length in order to determine the distance between the two. This paper does not incorporate varied length sequence matching. In pseudo code the data mining operation employed in this paper are simply defined in Table 1.

3.2 Defining Similarity

The Euclidean distance is a commonly used distance measure to define similarity between multidimensional vectors [4, 7,14] by recognising the distance between the two vectors in space. The Euclidean Distance may be defined as

$$Distance(A->B) = \sqrt{\sum (a_i - b_i)^2}$$

However, as a quantitative similarity measure similarity the Euclidean distance (and the Manhattan variation) is not without flaw. Lee, Kwon and Lee [6] discuss the imperfections of the measure resulting from vertical

offsets when two vectors do not start from the same vertical position or are not positioned in the same vertical region of a vector space graph. The authors note that 'similar' sequences (vectors) will not be realised where a vertical difference exists between two. They propose the use of a Minimum Euclidean Distance (MED) as a method of negating the differences caused through vertical axis offsets. The method utilises amplitude reduction to reduce the empty space between the two sequences which effectively brings them to the same vertical region of a vector plot. This is also recognised by others [4, 5] as an acceptable method of neutralising the effects of an offset. The formula for determining the Minimum Euclidean distance is given by [6];

$$D\min imum(A,B) = \left(\sum \left|a_i - b_i - m\right|^2\right)^{1/2}$$

Where $m = \sum_{i=1}^{n} \frac{(a_i - b_i)}{n}$

However, the use of the Minimum Euclidean Distance as a distance measure is further flawed in the financial domain because it fails to recognise the relative value movement and the associated returns experienced by a stock. That is, amplitude reduction does not recognise the relative movement of a stock as a return on initial investment. Price movements should be thought of relatively and not an absolute value because returns are more effectively represented as a change in underlying value. For example, a 10c gain on a stock that costs $20 is not as impressive as a 5c gain on a stock that costs $2 (0.5% verse 2.5% return).

Additionally, the use of amplitude reduction on non normalised data creates a bias for finding a match on stocks which have a lower value. This is because the distance between stocks of a lower value will be considered closer despite the relative shape of these stocks. The low value stocks movement approximates a 'straight line' and are less volatile that higher value stocks because the movements in price are not as excessive as those with a high value. In layman's terms, they don't jump around as much as higher value stocks. The price of low valued stocks approximate a straight line because they do not move as much along the vertical (price) axis. Using amplitude reduction as the sole method of offset neutralisation results in more low valued stocks matches because the prices are considered more consistent. It also means that the relative shape of the vector will not be as important for low value stocks as they approximate a straight line when compared to the price movement of a higher valued stock.

To overcome these inefficiencies, the sequence must be normalised around the sequence opening price before the Euclidean Distance can be used to determine similarity. This removes the need for an amplitude offset and allows for the shape of the sequence to retain its graphical representation. The Euclidean Distance can then be used as an effective distance measure

An alternative to normalisation around the opening price is normalisation over the prior day's value. However, normalisation around opening price was chosen in this project because the sequenced values still retains the diagrammatic representation of the underlying stock and therefore enhances the visible attributes of the sequence.

3.3 The Data and Measures Used

This study is conducted on Australian stock market data obtained through a commercial supplier (www.netquote.com.au) and represents end of day (EOD) trading information. Each stock traded for the day is represented with the opening, closing, high and low price. When price is referred to in this document, the average price is implied. This is because the average price is theoretically more attainable than any other price given in the data.

The validation dataset represented all stock market data for the month of November 2006 and comprised 49,102 unique EOD observations from 2,324 stocks. Despite the possibility of 2,234 theoretical validation sequences, only 2,115 on these sequences were valid. The training dataset represented all stock market data from 1-Jan-2004 through to 31-Aug-2006. This comprised 1,073,855 EOD observations from 2,256 stocks. There were 628,908 valid sequences in the training data. Sequence length was arbitrarily set to 20 days with an initial error margin set to 0.001 and incremental values of 0.001.

4 Experimental Results

The objective of this paper is to use temporal pattern matching to predict the values of stock prices. The reader is also reminded that one of the implications of Efficient Market Theory (EMT) is that stock prediction is not possible when the prediction is based on prior price information. The other fundamental implication of EMT is that trading strategies (where the predicted price is based on prior price) can not consistently outperform the market.

Should this method of price prediction proposed in this paper be accurate, effective trading strategies can be formed around the price expectation and returns greater than those of the general market should be possible. Additionally the objective of any data mining operation is the accurate prediction results, that is, overall accuracy fall within some acceptable limits. This paper therefore evaluates the results in two ways;

Firstly, the ability to outperform the market by using trading strategies based on the predicted value of stocks is tested. The strategy employed allows for price increases and decreases through a purchase-sell or sell-purchase discipline. If the stock is expected to rise significantly (greater than 1%) it is purchased on one day and sold on the next. Additionally if, the stock is expected to fall it is sold on the first day and purchased the day after (the prediction day).

Secondly, the prediction accuracy of the model is examined. In this evaluation, each prediction is referenced by its error margin (predicted result – actual result) and classified. An analysis of error rates is conducted in addition to the ability to make 'perfect' predictions. This form of testing will determine the

overall accuracy of the model. The (absolute) error is also mapped against the error margin in order to visually represent the model and its output when compared to the 'comfort' gained by considering the distance value for the prediction(s).

4.1 Results by Using a Trading Strategy

The purpose of a trading strategy is to use knowledge in a rational manner in order to achieve a desired outcome. In this project, the knowledge used is the expected change in value for each query sequence and this change is used as a basis for making a buy-sell or a sell-buy decision. Because the actual changes for the sequences are known, the expected returns (based on predicted results) can be compared to the actual results (for both specific all sequences).

The trading strategy used (based on model output) is as follows;

1. If the predicted change in value is more than 1% the stock is traded.
2. If the stock is expected to rise (that is, the model predicts a rise) the stock is purchased on one day and sold on the next (buy low sell high).
3. If the stock is expected to fall, it is sold on one day and purchase on the next (sell high buy low).

Of the 810 stocks (used in testing) that produced results, the total increase was 363%, implying that investing in every stock would have given an overall increase of 363%. Using the strategy described above and the predicted price movement that resulted from the data mining operation, the expected gain was an increase of 1,382%. When these theoretical trades were applied to the actual stock data there was an overall loss of 38%. Had an investment occurred, a net loss would have occurred. This is a disappointing result given that the overall rise of the market of 363%. Investing in every stock would have made gains of 363% and proven much more successful than using a strategy based on the data mining application. The trading strategy has therefore proven unsuccessful in providing returns that exceed the overall market.

These results are not directly comparable to Povinelli's [12, 13] studies. Povinelli focused on techniques that recognised temporal patterns where high gains were found whereas this study focused on prediction regardless of the prior pattern. Povinelli also comments on gains that were made without reference to the underlying market.

There are two underlying assumptions that are applied to the strategy applied above. Firstly, the assumption is that all predictions are valid and therefore used. Secondly, the assumption that only stocks with an absolute change of 1% or more is traded.

One should question the validity and reasonableness of the first assumption in light of the distance measure provided with a result set. Prima facie trading should be geared towards predictions that are most certain. A more realistic trading strategy is to apply a greedy consideration to the stock trading decision based on each result distance. That is, only trade stocks that we feel most confident in predicting. This can be achieved by considering predictions that have a low distance (error margin) first.

The graphical application of this type of strategy is shown in Figure3. In this graph, the same 1% buy-sell/sell-buy strategy is used however the expected and actual returns are displayed with reference to distance. Only those stocks that are predicted within the horizontal axis (distance) are considered in the trading strategy.

The expected gains returned through the model are incremental over the entire distance. This is not surprising given the generic condition that we expect to make a gain is we know what the expected outcome is. In contrast the actual returns appear to have an optimum value in the range of 0.14 to 0.18 which could arguably be extended until 0.32 distance when the actual return decreases significantly. This implies that there is an optimal distance that should be considered when deciding on whether a stock prediction should be used in the trading strategy.

If trading the previously mentioned strategy is altered to allow trades on results obtained with distance less than 0.32 from the query sequences, the expected result would be a change in value to 1,215% verses and the actual change in value of 38%. This is an improvement in the trading strategy and contributes positive gains however, gains that are still significantly lower than the 363% returned by the 'total market'.

4.2 Prediction Accuracy

A scatter plot of the prediction absolute error verse average distance for that prediction is illustrated in Figure 4. For visual clarity, it has been restricted to only contain observations with an error of 0 to 10% on the vertical axis and distances from 0 to 0.32 on the horizontal axis. The plot reinforces the relationship theoretical return, actual return and distance that was discussed above and shows how the distribution of error increases with the average distance of result compared to query sequence.

It is also worthy to highlight the following points;

- There appears is a general diffusing of prediction error as the distance increases.
- There are observations with high error value (~9%, 5%, 2% and 1%) where the error margin is extremely low (~ 0). This implies that the distance measure is not a perfect predictor.
- There are instances of larger distances producing zero errors.

Due to the in-ability to draw a direct correlation between the error distance and the expected error, the cumulative plot of distance verse average error is shown in Figure 5.

The accumulated error graph in Figure 5 clearly shows a correlation to error to distance. That is, the lower the error margin the lower the error of estimation. If this correlation exists, one may question why the expected returns were so different from the actual return for stocks that met the trading requirements discussed above? One

reason for this lies in the trading rules that were dictated, principally that the stock must move by more than 1% if it is considered for trading. Of the 84 predictions with a zero distance, only 3 were included in a trading strategy.

An additional way to summarise the predictive power of the model is to plot the accumulative error rate against the number of predictions meeting that error rate. This is shown in Figure 6.

Of the 800 predictions, 400 had an error rate of 1.5% or less. Based on the average error v distance plot (the complete data) these observations had a distance of 0.04 or less.

This signifies that trading strategies should include an allowance for the average distance in determining wether or not to trade a stock. If we are willing to accept a 1% error margin, a distance of 0.04 should be used and an actual return of 8.5% would be generated.

An important observation on the result data is that there are accurate predictions of stocks that do not meet trading strategies. This will occur when the change in price change is less than 1% and under the strategy defined above the stock is not considered for trade. It is proposed that the model is accurate (or at least a degree of confidence can be applied to the prediction through the distance measure) but the ability to earn returns in excess of the general market is unduly influenced by low stock movements.

5 Conclusions and Further Work

This paper was targeted at the prediction of stock market values using temporal pattern recognition by reference to a minimum distance. Compared to previous work, the method employed allowed for a degree of prediction confidence to be supplied for each transaction so that the accuracy of the prediction could be taken into consideration when developing trading strategies based. The model did not target specific expected returns but rather sought to predict an expected value for all events that existed in the validation data set.

The data mining algorithm was a time intensive operation. For the events that were predicted, the total return experienced by the market was 368%. Using a trading strategy based on predicted events the return would have been 38%. Notwithstanding this the model displayed some positive prediction statistics. Thirty seven percent of all predictions were in a 1% error margin of actual outcome with an average distance of 0.038. As expected, when the distance from result increased, so did the margin of error associated with that prediction.

The models performance would be improved by a reduction in prediction (processing) time as this was the most inhibiting factor in prediction. The results obtained were only based on one prediction per stock (for one day) which may indicate that the model is not truly reflective of the predictive power of such a model. Thus, the model did not examine the likely hood of specific stocks that follow reoccurring pattern in their own movement or exhibit lead/ lag characteristics when compared to other stocks. Further investigation into the subject should consider such scenarios.

The immediate solution to the run time problem coined a two scan approach. The first scan would retrieve the minimum distance of the query to all possible results and the second would predict based on some rounded up value of the first. Indexing and pre-processed training would also allow for a much larger knowledge base to be applied in the prediction methodology.

6 References

[1] Tufte, E, 1983, The visual display of quantitative information, Cheshire Connecticut, Graphics Press, as cited by Ratanamahatana et al [4].

[2] Han J, Kamber M, 2001, *Data Mining Concepts and Techniques*, San Francisco, Morgan Kaufmann Publishers

[3] Rafiei D, 1997, On Similarity-Based Queries for Time Series Data, SIGMOD Record, 26(2):13--25, May 1997

[4] Ratanamahatana C, Lin J, Gunopulos D, Keogh E *Mining Time Series Data*, *The Data Mining and Knowledge Discovery Handbook*, Springer Science and Business Media, 2005

[5] Negi T, Bansal V *Time Series: Similarity Search and its Applications* in Proceedings - International Conference on Systemics, Cybernetics and Informatics: ICSCI-04,

[6] Lee S, Kwon D, Lee S, 2002, *Efficient Pattern Matching of Time Series Data*, LNCS 2358 Springer.

[7] Bhagat P, 2005, *Pattern Recognition in Industry*, Elsevier, Oxford, UK

[8] Lin, W., Orgun, M., Williams, G, 2002: *An Overview of Temporal Data Mining*. ADM02, Sydney, Australia (2002) 83-90.

[9] Faloutsos C, Ranganathan M, Manolopoulos, Y, 1994, *Fast Subsequence Matching in Time-Series Databases*, ACM SIGMOD, Volume 23, Issue 2, Pages 419-429,.

[10] Hellstrom T, Holmstrom K, 1998, *Predicting the Stock Market,* Marardalen University, **Technical report** ISRN HEV-BIB-OP-26-SE 26 (1998), IMa-TOM-1997-07,

[11]Wikipedia
http://en.wikipedia.org/wiki/Technical_analysis#History

[12] Povinelli R, 2000, *Indentifing Temporal Patterns for Characterization and Prediction of Financial Time Series Events,* Temporal, Spatial and Spatio-Temporal Data Mining: First International Workshop; TSDM2000, Lyon, France, 46-61

[13] Povinelli R, Higgs D, 2001, *A Temporal Pattern Approach for Predicting Weekly Financial Time Series*, Artificial Neural Networks in Engineering, St. Louis, Missouri, 707-712

[14] Marketos G, Pediaditakis K, Theodoridis Y Theodoulidis B *Intelligent Stock Market Assistant using Temporal Data Mining,* MSc Thesis, UMIST, 2004.

[15] Incredible Charts http://www.incrediblecharts.com/ [Last Accessed 11-Feb-07]

[16] eASCTrend http://www.wintick.com/6_0/home.asp [Last Accessed 11-Feb-07]

[17] Agrawal R, Faloutsos C, Swami A, 1994, Efficient Similarity Search In Sequence Databases, Proceedings of the 4th International Conference of Foundations of Data Organization and Algorithms (FODO)

[18] Keogh E, Lin J, Truppel W, 2005 Clustering of Time Series Subsequence is Meaningless: Implication for Previous and Future Research, KIS 8 (2), 154- 177.

[19] Gao L, Wang X, 2005, *Continuous Similarity-Based Queries on Streaming Time Series*, IEEE Transactions on Knowledge and Data Engineering, Vol 17, No 10.

[20] Wu H, Slazberg B, Sharp G, Jiang S, Shirato H, Kaeli D, 2005, *Subsequence Matching on Structured Time Series Data*, SIGMOD 2005, June 14-16, Baltimore, Maryland, USA

[21] Povinelli R, Feng X, 2003, *A New Temporal Pattern Identification Method for Characterization and Prediction of Complex Time Series Events,* IEEE Transactions on Knowledge and Data Engineering, Vol 15, No 2, March/April 2003

[22] Last M, Klein M, Kandel A, 2001, *Knowledge Discovery in Time Series,* Databases¸Systems, Man and Cybernetics, IEEE Transactions on Knowledge and Data Engineering, Feb 2001, Volume 31 Issue 1.

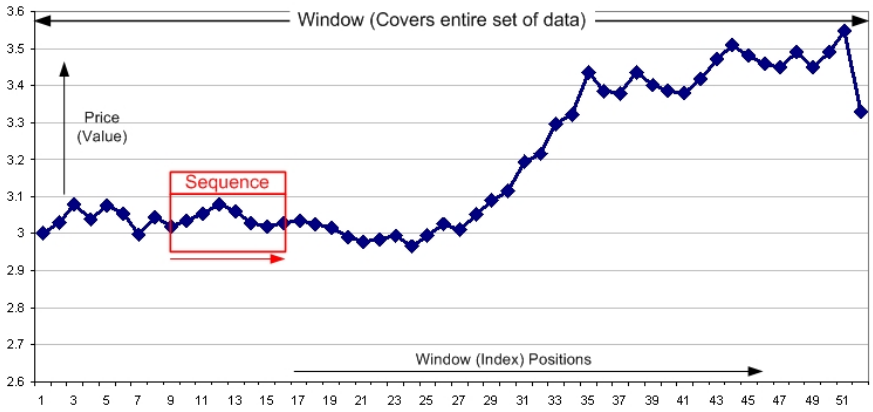

Figure 1: A graphical representation of window and series in the time series data

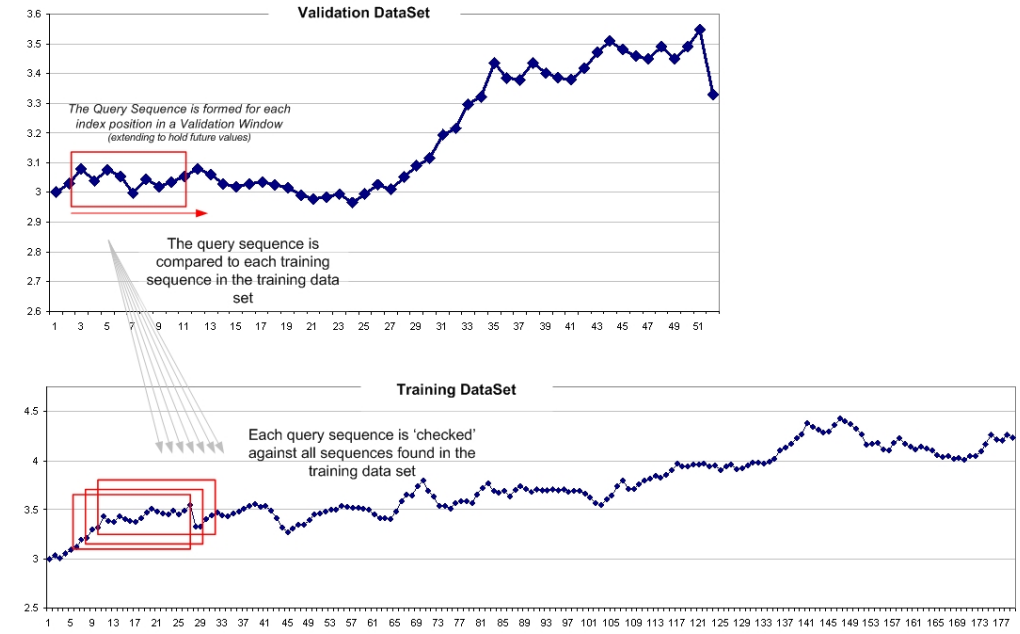

Figure 2: Process of querying over the validation data based on training data

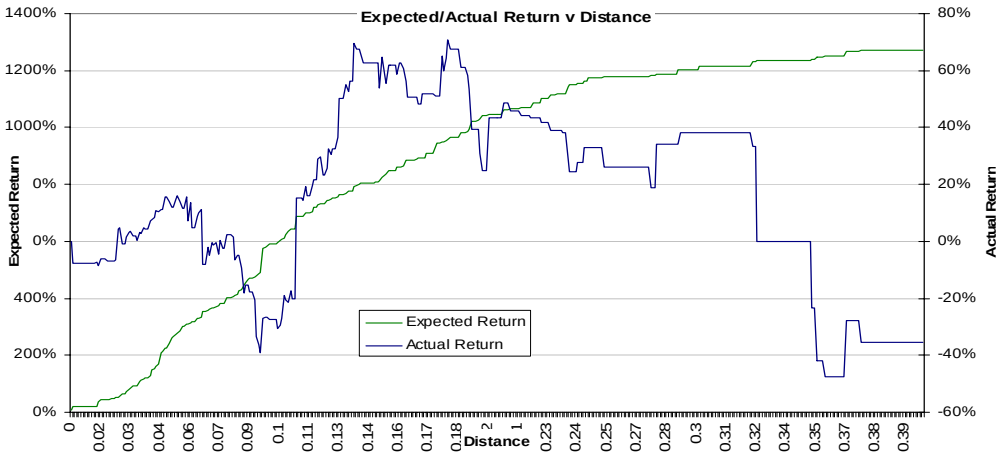

Figure 3: Expected and actual return using the 1% buy-sell/sell-buy strategy

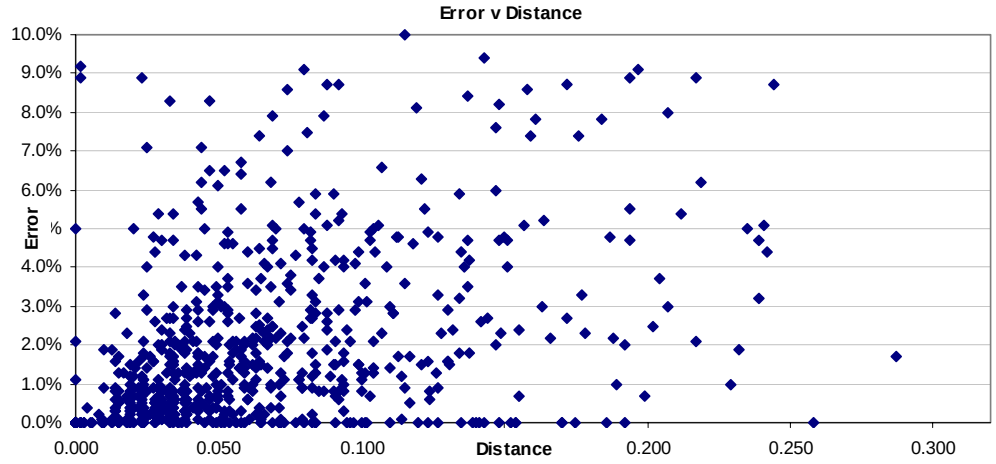

Figure 4: Prediction error versus average distance for that prediction

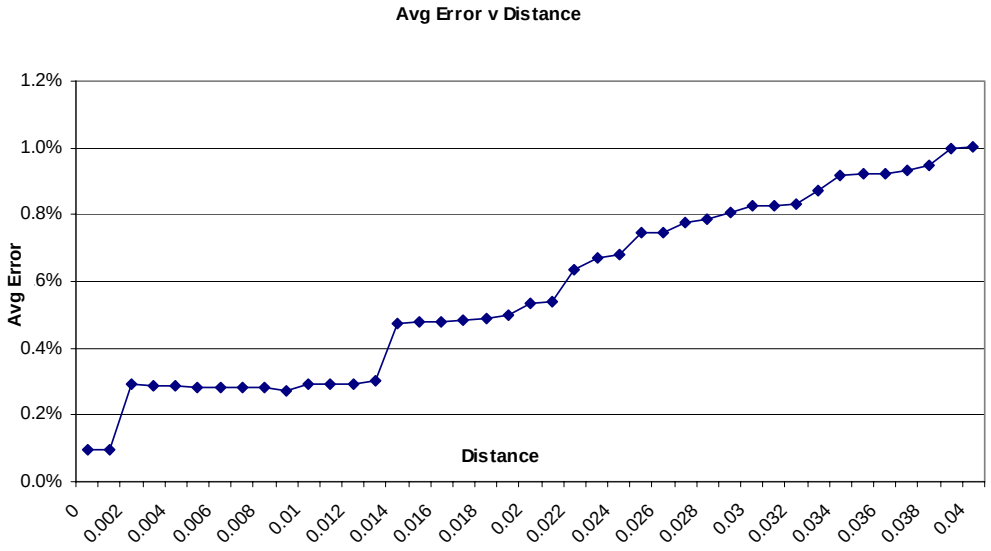

Figure 5: Direct correlation between the error distance and expected error

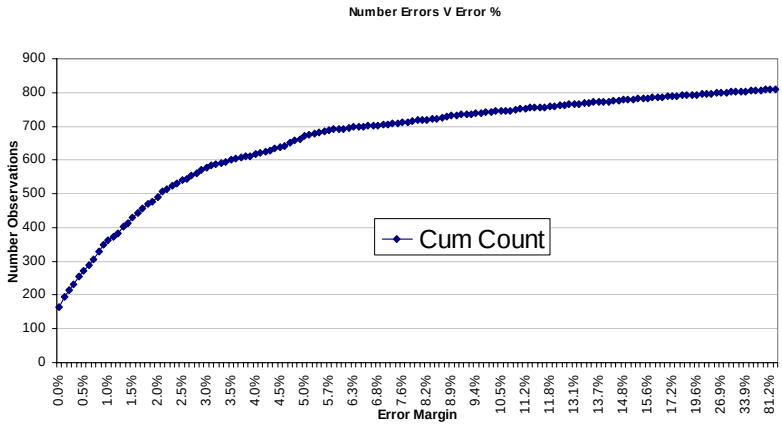

Figure 6: Number of predictions meeting the error rate