Financial Time Series Forecasting Using K-Nearest Neighbors Classification*

M. Maggini [a], C.L. Giles [b,c], and B. Horne [b]

[a] Dipartimento di Sistemi e Informatica, Universita di Firenze, Firenze, Italy

[b] NEC Research Institute, Princeton, NJ 08540

[c] UMIACS, U. of Maryland, College Park, MD 20742

Abstract

Deriving a relationship that allows to predict future values of a time series is a challenging task when the underlying law is highly non linear. Usually, when facing with a problem of non-linear prediction, we are provided with the past history of the time series and we want to extract from that set of data a mathematical function that relates a certain window of past values with the value T time steps ahead in the future. In practical problems the discovering of a forecasting function is hard since we deal with processes corrupted by noise due to an inaccurate modeling of the system and to the measurement procedure. Another important characteristic of the signal that has to be considered is its stationarety. An adaptive forecasting technique should be devised for non-stationary processes.

This is particularly true for the analysis and for all the attempts in forecasting financial time series. It seems that such series are intrinsecally non stationary and that a complete model requires not only the knowledge of past values of the series but also some other information regarding the environment. The results in this field are contradictory but it is clear that trying to infer a global unchanging model from the historical information leads to results comparable with a random walk [8].

We try to use a different approach for prediction rather than dynamical system modeling that is usually the basis for time series characterization. We transform the prediction problem in a classification task and we use a statistical approach based on the k-nearest neighbors algorithm to obtain the most probable variation with respect to the present price value.

Index terms: Nearest-neighbors, Financial time series.

1 Introduction

The forecasting of the future values of a time series requires the discovery of the laws that govern the phenomenon from which the series is measured. The ideal situation is that in which the

*Published in *Nonlinear Financial Forecasting: Proceedings of the First INFFC*, Ed. R.B. Caldwell, p. 169-181, Finance & Technology Publishing, Haymarket, VA. 1997.

correct equations can be written so that the evolution of the system can be predicted with high accuracy once the initial conditions and the inputs are known. Instead, very often we deal with the case of systems for which the equations are unknown and we want to infer the rules that produce their response extracting every regularity from samples of the past behavior. Obviously this approach relies on the hypothesis that the information contained in the available data is sufficient to characterize with high accuracy the dynamics of the system. This means that we are assuming that all the variables that influence the trajectory of the measured output can be used or that they vary suffiently slowly in order to build a model adapting in time. Moreover, another important assumption is that the system has reached some kind of stationary evolution, i.e. its trajectory is moving on a well defined surface in the state space. These results are founded on the embedding theory that states that the evolution of a deterministic system can be described by means of a surface in the space of delayed values once it has reached a stationary behavior. This surface describes the system *attractor* and takes into account all the possible complex trajectories that can be realized with a non-linear system, from limit cycles (periodical solutions) to deterministic chaos (aperiodical solutions) [10].

When we want to build a model to predictict the output of a given system we have two different possible choices: the first one is to develop a *strong model*, i.e. choosing a well defined set of equations (ex. from an analysis of the problem) depending on a small set of parameters; the second one is to use a *weak model* that has wide approximation capabilities but relies on the use of many parameters. Each method has its own adavantages and disavantages; the choice of the approach to follow is related to the *bias-variance dilemma*. Using a very weak model can be extremely flexible when dealing with problems for which a direct analysis to make an hypothesis on the form of the governing equations is very difficult; however, this kind of models require a lot of examples to find a set of values for the parameters in order to reach a good generalization without overfitting the time series. From a statistical point of view we say that these models have a *low bias* (the model can fit a large variety of functions) but *high variance* in the parameters (if the behavior of the model is not well constrained, the parameters will be estimated with large errors). To obtain accurate results using a weak model, it is necessary to use a dense and complete sampling of the state space behavior to build the model.

Neural networks are an example of a powerful weak model. A result by Cybenko [5] states that using a neural network with an hidden layer of sigmoidal neurons and linear outputs, it is possible to approximate any given vectorial multidimensional function $f : D \subset \Re^n \rightarrow C \subset \Re^m$, where D and C are compact sets. The weakness of this result is that it does not provide any knowledge on the size of the network that is needed to solve a given problem, neither it tells how many examples are necessary to train the network in order to obtain a good generalization. Moreover, the learning process, based on Backpropagation and gradient descent, can be seriously plagued by the presence of local minima in the cost function used to evaluate the performance of the neural network on the sample set [1, 6], thus producing suboptimal solutions.

Despite all of these problems, neural networks have been proven to be effective in learning quite complex problems in time series forecasting [3] when dealing with non linear systems even in the presence of chaotic trajectories. Neural networks can be used to generalize the linear models like AR e ARMA, and can show even more powerful capabilities when fully recurrent connections are allowed [3]. Many interesting algorithms, based on extended Kalman filtering techniques, have been proposed to deal with the presence of outliers and trends in the training data [3].

The availability of powerful non-linear forecasting techniques has suggested their application on a complex problem: the prediction of future values in economic series, like stock prices and currency exchange rates. These series have a very complex behaviour since their evolution in time is the result of the cooperation of a huge number of different factors and agents. You can think as the whole set of stocks quoted in a market as a state vector of a dynamical system whose transition function depends on external inputs such as interest rates, governament decisions, international events, industrial production and so on, and on the activity of all the brookers, speculators, and operators acting on the market. The actual evolution rules, if they exist, are produced by the emergent behavior of the single actions, like pressure in a gas is produced by the forces of the single molecules. If such a parallel is true, despite the unpredictability of the single actions of each operator in the market, some kind of general rules should govern the overall market behavior. A well established theory on the market evolution has not been developed yet. The *efficient market* theory states that the evolution of the prices for each economic variable is a *random walk*, i.e. the variations in price are completely independent from one time step to the next one. This idea derives from the observation that if the market presents some kind of regularity then there is someone who takes advantage of that regularity so that it disappears at once. Actually some regularities must exist, but they are well hidden and they change very fast in time. The task of discovering these regularities is very hard and requires a continuos adaptation because when you discover a regularity and you take advantage from it, consequently it tends to disappear. As an example, a regularity that has been discovered by means of a statistical analysis on historical data is the *weekend effect*; the result of this study is that it seems that the share prices do worse on Mondays than on any other day of the week, at least in America. This shows how an information like the day of the week can be important for the prediction of the price evolution (i.e. it is a variable correlated to the changes in the prices).

When dealing with prediction tasks in economics there are essentially two different approaches. The *fundamental analysis* tries to determine all the econometric variables that may influence the evolution of a given stock price or exchange rate. Many factors can be taken into account as economic indicators (unemployement, internal product, etc.), company-related informations (assets, debt, productivity, type of production, announcements), monetary parameters, interest rates, relevant political events (international wars, governament directives, etc.). Very often it is very hard to extablish which variables are actually relevant and in many cases it is

difficult to evaluate a particular enviromental condition (ex. political elections, international events). It is clear that an accurate understanding of these influences requires the help of a human expert, who knows exactly which factors are taken into account while planning market strategies.
The *technical analysis* states that when the sampling rate of a given economic variable is high, all the information necessary to predict the future values is contained in the series itself. Actually this may be true only for very short term predictions when we can assume that no great changes occurred in the economical environment between one tick and the successive one. However, an accurate prediction is very hard since successive ticks correspond to bids from different sources and the correlation between the price variations may be very low.
A good approach should combine both the previous points of view, using all the information that is available in order to produce the prediction. It seems quite obvious that the more information is available for the decision, the more accurate and easy the prediction will be. So a multivariate model should achive an higher performance.

Essentially when analyzing a financial time series the assumption of stationariety is hardly met, as there is always a clear trend in mean and in variance. Thus we may expect that the parameters in any model, we are trying to build, will be varying in time both slowly or rapidly. We will show which problems arise when we want to analyze a process that is intrinsically non stationary and we have not an ensamble of different measurements for it (different realizations of the random process). Basically it is hard to derive good statistical indicators from a single series and in general there will be not enough examples to infer accurately the evolution rules while they are changing in time. This is reflected by the quite unsuccesful attempts of applying neural networks to such tasks; neural network training is quite hard because of the presence of many contradictory examples and, even when convergence can be reached using only a short period of time as training set, the resulting generalization is always bad due to overfitting and to the incompleteness of the information contained in the learning set. In this case the overfitting problem is due not only to the fact that the system adapts to noise because it has too many parameters, but to the change of rules in time. While dealing with this problem is hard to distinguish between noise in the signal and noise due to the drift in the rules. For examples all non-modelled inputs to the system are actually a source of randomness, unless they have regularities in themselves.

Our approach is different from dynamical system modeling since it transforms the time series prediction into a classification problem. The main idea is that a high accuracy in the predicted value is impossible to be achived and that we can do a good job just considering *classes* of values. Thus the values are quantized using a specified resolution and a finite number of classes are defined. The prediction is performed finding the most probable class of price variation given the past variations and other variables that might be correlated. The classification is based on a set of examples extracted from the history of the series and it is adaptive in the sense that the new values are inserted in the prototype set when they are available and the old ones are discarded.

Another solution is to use the real nature of the stock price series that is made up of bids occurring at random time steps rather than a series evolving uniformly in time. If the exact time of each bid was available, it would be better to use a system that tries to discover the correlation between these events (i.e. considered as a couple of variables $(value, time)$ [1]) and the next event. A model of this kind is closer to the real structure of the market.

2 Financial data

The recent diffusion of electronic trading systems has made available high frequency financial time series. It is quite common to have data corresponding to each minute of the trading day, so that a lot of information can be used to try to devise a system capable to produce short term predictions.

The first problem to face when we want to use such data to build a forecasting system is to choose the variables that will be used as input. It seems there is not a rigorous way to do that, but all the choices are driven by heuristic considerations. In all the applications of forecasting techniques to financial time series, the first transformation that is applied to data is the Δ_k-difference operator. This operator is defined as follows:

$$\Delta_k(y_t) = y_t - y_{t-k}. \tag{1}$$

The idea of applying this operator is derived from the assumption that they are the variations in price that move the market rather than the actual value. In other words this filtering of data should remove the non stationariety due to the trend in price. However the resulting signal is produced by an high pass filtering and so the noise will be amplified. Another problem is that any dependence on the actual value will be lost, even if in practice this dependence would have been hard to discover due to the small number of examples available for each price range. By the way, we can assume that if we are using an adaptive system, the actual dependence on the current price will be reflected by the model.

Usually $k = 1$; any other value for k corresponds to a downsampling of the data. In this case it is more correct to use these differences delayed by k steps as inputs to the predictor.

When $k = 1$, we are applying a filter whose transfer function is $H(z) = 1 - z^{-1}$. If we assume that $y_t = y_{t-1} + d_t$, then the operation performed by the previous filtering is to cancel the pole produced by the integration. As stated before this operation corresponds to model the series of price variations rather than the price itself.

[1] The value used here may be the actual price or the difference with respect to the previous bid. The time can be resonably chosen as the gap between the two consecutive bids.

The second step is a low pass filtering in order to reduce the high frequency noise present in the difference sequence. A simple average on a given window, can produce this effect. If T_w is the width of the averaging window, the filtered signal is obtained as:

$$\overline{\Delta_k(y_t)} = \frac{1}{T_w} \sum_{\tau=0}^{T_w-1} \Delta_k(y_{t-\tau}) \tag{2}$$

If we choose $k > 1$, $T_w = k$ and we use only one sample every k time steps, actually we are downsampling the series, and the lowpass filtering is necessary to reduce the aliasing effect. More complex lowpass filters than the average may be used. Actually, keeping in mind the genesis of this series (i.e. it is obtained considering the successive bids that happen at discrete times with a non-uniform distribution), oversampling and downsampling techniques do not have a clear meaning. In fact the values for each minutes are not obtained from the sampling of a continuos-time signal having specific spectral characteristics. Usually a frequency analysis of such series produces a large band spectrum. Both downsampling and oversampling introduce artificial regularities due to the filtering of the signal. These regularties may help the learning process but can also hide other regularities that we hope are present in the signal.

A different transformation that can be applied to the original sequence instead of the difference operator is computing ratios of successive values. In this case the actual input to the system will be the sequence:

$$r_k(t) = \frac{y_t}{y_{t-k}}, \tag{3}$$

where usually $k = 1$ and obviuosly we are assuming $y_t > 0$.

Other authors suggest to apply also a logarithmic transformation to these ratios in order to compress the escursion of the signal. It is quite unclear which is the effect of these additional trasformations because it is impossible to establish a priori which is the best preprocessing for the data. The data-preprocessing should be a procedure to extract the relevant information from the data in order to make easier the task of the classifier or predictor. In general the benefits of a particular transformation should be evalutated with respect to the effects produced on the performance of the particular algorithm used for prediction. The application of an additional non-linear transformation on the inputs is an additional degree of freedom, but it needs evaluation before deciding if it is useful and not harmful for the system.

Beside the sequence values, other sources of information related to the sequence itself can be available and can be used as input to the predictor. Possible inputs are the day of the week, the hour of the day, and informations related to the volume of bids (*tick frequency*). As described in the introduction, the *weekend effect* is one of these possible correlations.

Once we have chosen the input values for the prediction system, we have to decide how long back in time there is correlation between the value to predict and the input sequence. We can try to compute the autocorrelation function for the sequence, but if the sequence is non-stationary the result may be meaningless. Usually heuristics and the trial and error procedure are applied.

As there is not a clear recipe to choose the inputs, neither there is a method to evaluate if we are using a good architecture for the predictor, unless we were lucky with our choices obtaining a system that shows a good performance on a *significant* test set, we will always have the doubt that some change somewhere in our procedure will improve the performance. Unless there is the possibility to formulate the problem in a more solid and rigorous theoretical framework, this question will remain open. For example, analyzing his not very encouraging results, Mozer [8] addresses this problem.

2.1 The INNFC data set

For the INFFC competition the train data provided spans more than 1000 trading days. Each trading day is assumed starting at 10:31 and ending at 15:00. For each minute of the trading day five values are available:

- Open value : the value of the first bid in that minute;
- High value : the value of the highest bid in that minute;
- Low value : the value of the lowest bid in that minute;
- Close value : the value of the last bid in that minute;
- Tick volume : the number of bids in the minute.

Obviously there is information on the current hour but there is no information on the day of the week so that it is impossible to detect the regularities correlated for example to weekend gaps. The organizators give a rule in order to make the data uniform; they suggest to fill in the missing minutes by copying the previous available close value and setting to zero the tick volume. This is quite reasonable and reflects the fact that if there are no bids the price should be considered constant. This way the resulting series is uniformly sampled in time. The average number of filled minutes per day is 192.7 (71.1%); the average width of the filled gap is 15.3 minutes.

The competition requires to devise a non-linear predictor in order to minimize the normalized root mean square error (nRMSE) for the prediction of the close value in the two cases of two hours ahead prediction and one day ahead prediction. In formula the task is to minimize:

$$nRMSE = \sqrt{\frac{1}{m\sigma_y^2}\sum_{t=1}^{m}(y_{t+T} - \hat{y}_{t+T})^2} \tag{4}$$

where y_t is the sequence of close values, m is the total number of predictions, σ_y is the standard deviation of the actual sequence, $\hat{y}_{t+T}$ is the value predicted by the system, and T is the prediction horizon. For the competition the two cases $T = 120$ and $T = 270$ are considered.

Another interesting index is used to compare the prediction with respect to the random walk prediction (it corresponds to assume the present value as the best prediction; this follows from the assumption that the change in value is zero mean and uncorrelated with the past history). The index has the following expression:

$$Q = \frac{\sum_{t=0}^{m}(y_{t+T} - \hat{y}_{t+T})^2}{\sum_{t=0}^{m}(y_t - y_{t+T})^2}. \tag{5}$$

This index is 1 when the performance is comparable with the random walk and less than 1 when the system has a better performance.

For the current system the following three input values are used:

- the mean value of the variation of the close price as defined in eq. (2), $\overline{\Delta_k}(y_t)$;
- the mean value of the tick volume, $\overline{TV}(t)$;
- the fraction of hour from the last tick, $TG(t)$.

The idea is to use these variables sampled every T_g minutes and use this interval to average the variables. In other words the width of window to compute $\overline{\Delta_k}(y_t)$ will be $T_w = T_g$, and $\overline{TV}(t) = \sum_{k=0}^{T_g-1} TV(t-k)$. The inputs to the predictor will be $\overline{\Delta_k}(y_{kT_g}))$ and $\overline{TV}(kT_g)$ for $k = 1, 2, \ldots, E$.

All the other available values for the price (Open, High, Low) are discarded as they are not very different for the close value, and their use will produce actually an increase in the sampling rate but they will also introduce more noise.

2.2 Statistical analysis

Performing a statistical analysis on the data may be useful in order to get some information to build the predictor (ex. to guess an embedding dimension). The main difficulty is that as noted by other authors [7] financial time series are an highly non-stationary problem. Even if we can think to reduce the non stationarety applying the operator Δ_k to the sequence, this is not sufficient and this is reflected by the difficulties encountered by other reserchers while trying to build a global and invariant model for the market. The analysis of a non stationary sequence is quite complex especially in this case because only a single realization of the process is available. Thus in this case an ensamble of sequences corresponding to the same process is not available in order to extract all the possible correlations and statistical indicators. Unless we can assume that the same correlations are valid for a set of different stocks, we have only one example and we can extract only those regularities that are showing for a sufficient time along the sequence. As the sequence presents a lot of contradictory information due to noise, insufficient data sources, and perhaps to a bad choice of the preprocessing and inputs, it is very hard to decide if a certain relationship is a rule or just produced by chance. For this reason

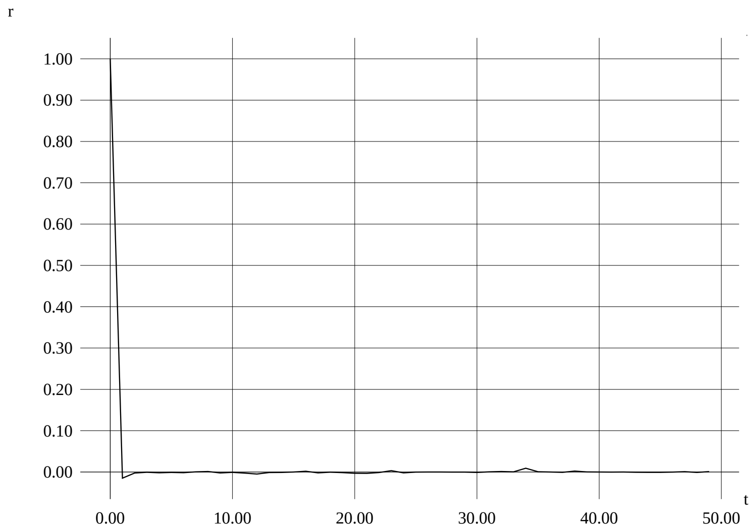

Figure 1: Autocorrelation of $\Delta_1(y_t)$ computed on the training data (1035 days).

all lerning methods have an hard time with this kind of data and very often the random walk prediction performance is the best that we can obtain.

In fig. 1 the plot of the autocorrelation of the first order differences, $\Delta_1(y_t)$ is reported. The plot shows that the correlation between the $\Delta_1(y_t)$'s is very low even when the time gap is 1. The first autocorrelation coefficient is -0.015, while all the others are smaller than this one by more than one order of magnitude. As a consequence we can not get any hint from this analysis to guess an embedding dimension because the forecasting horizons are $T = 120$ and $T = 270$. It seems that for these lags the variables are virtually independent.

Because of the quantized nature of the data, it is easy to build the hystogram corresponding to the distribution of the increments between the present value and the value to predict. In figure 2 the hystogram of the variation in price for a forecasting horizon $T = 120$ is shown for a sequence of 250 days. The tails of the distribution were cut because only few outliers were present. The distribution is far from being gaussian and it also presents a strange clustering of the quantized values around particular values (i.e. some values are more likely than others). The high peak in zero is due to the rule used to fill the missing minutes.

From a statistical point of view the prediction can be done if the conditional probability distribution $p(y_t|y_{t-1}, \ldots, y_{t-E})$ can be approximated. Actually we are not interested in the exact value, that is quite hard to predict, but we can think to divide the range of the increments into intervals, identifying each of these intervals by its average value. Using this approach we can define N different classes of price variations and try to approximate the conditional probability of each class. We tried to apply this approach for prediction using the nearest-neighbors algorithm described in the next section.

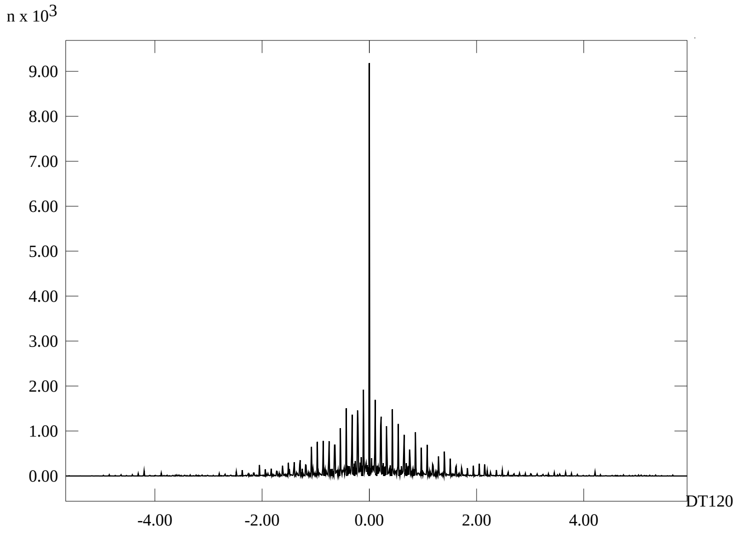

Figure 2: Hystogram of the differences in close price for $T = 120$ (250 days).

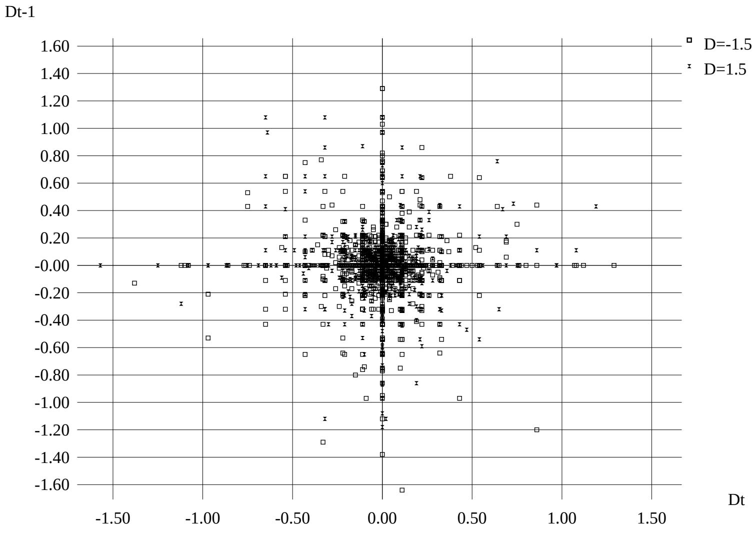

Figure 3: Distribution of the points in the two-dimensional embedding space for the two classes corresponding to a difference in price of 1.5 and -1.5 respectively for $T = 120$. The two axes represent the first order difference $\Delta_1(y_t)$ for two consecutive time steps.

It is possible to visualize the points corresponding to two different classes when we choose an embedding dimension $E = 2$. In figure 3 it is shown the distribution of the points corresponding to the classes related to a variation in price of 1.5 and -1.5. This plot shows that the two classes are completely mixed. As a consequence it is impossible to distinguish between the two conditions using only these two differences as inputs. It is hard to decide if this would be possible using an embedding vector with a bigger dimension but the difficulty of the task is clear even from this plot.

3 The prediction algorithm

The algorithm we are using for prediction is a sligth modification of the *k-nearest neighbors algorithm* defined for classification tasks. In financial time series the differences in price are quantized such that these variables could be described by a discrete conditional probability distribution. We make the hypothesis that the variation in price at a certain time step depends on the previous variations (or from a set of variables derived using a preprocessing algorithm from the price trajectory) and on other variables that are correlated with the series (like the tick volume and the tick gap). Actually the task of finding the variables that are influencing the variation is not very simple.

The k-nearest neighbors algorithm is related to this probability distribution as it basically relies on an approximation of that distribution around the present configuration of the known variables. The original algorithm [4] is applied to classification problems where a unknown object is to be assigned to one of C different classes on the basis of a vector $\mathbf{b}$ of predictor measurements. The algorithm uses N training observations, the prototypes, which consist of predictor measurements $\mathbf{p}_k$ and the associated class membership. The classification is obtained finding the k nearest neighbors in the training set to the present predictor vector $\mathbf{b}$ and then classifying $\mathbf{b}$ as the most frequent class among the k neighbors.

The input vector we can use to identify the *class of price variation* contains both a set of past values of the preprocessed price sequence and other information like the average tick volume and the time gap between successive ticks. The price variations can be divided into classes using their original quantization or defining larger classes thus using a smaller resolution. Classes will be centered around the origin. For example if we use a resolution equal to 0.5, all the price variations in the interval $[-0.25, 0.25]$ will belong to the same class (that containing 0).

Let us suppose to memorize a window of N_f past values of the time series (*fitting* set). If we choose an embedding dimension E and the prediction horizon is T, we can define a set of $N_f - T - E + 1$ delayed vectors with respect to the present time t:

$$\mathbf{Y}_j = [y_j, y_{j-1}, \ldots, y_{j-E}], \quad t - N_f + E \leq j \leq t - T. \tag{6}$$

The *target* value y_{j+T} is associated to each delayed vector. We are assuming that the fitting set is updated at each time step; in fact, when a new value is available, the oldest value in the fitting set is discarded while the new one is inserted; this way the size of the fitting set is kept fixed. This techinique may be useful to deal with non stationary systems. Otherwise we can decide to keep the initial fitting set unchanged.

The predictor vector for our problem is the following:

$$\mathbf{p} = [\overline{\Delta_k}(x(t)), \ldots, \overline{\Delta_k}(x(t - ET_g)), \overline{TV}(t), \ldots, \overline{TV}(t - ET_g), TG(t), \ldots, TG(t - ET_g)], \quad (7)$$

where E is the number of past values used for each variable, T_g is the downsampling rate and k the time gap used to compute input price variations.

The algorithm requires to compute the distances between vectors and we used a *normalized euclidean distance*. The normalization is introduced to weigh the different groups of components in the predictor in order to make their contribution similar. We normalized each group using the variance of the corresponding variable computed on the training vectors. As the training vectors are updated during prediction, discarding the older values and inserting the new available information, also these normalization factors are updated. Thus the distance used to find the neighbors to the present predictor is:

$$\begin{aligned} d_h &= \frac{1}{E}\left[\frac{\sum_{i=0}^{E}(\overline{\Delta_k}(x(t-iT_g)) - \overline{\Delta_k}(x(h-iT_g)))^2}{\sigma^2_{\Delta x}} + \frac{\sum_{i=0}^{E}(\overline{TV}(t-iT_g) - \overline{TV}(h-iT_g))^2}{\sigma^2_{TV}} + \right. \\ &\left. \frac{\sum_{i=0}^{E}((\overline{TG}(t-iT_g) - \overline{TG}(h-iT_g))^2}{\sigma^2_{TG}}\right]^{\frac{1}{2}} \end{aligned}$$

where t identifies the current vector and h the prototype. Instead of finding exactly k neighbors, our algorithm defines a ball around the present input vector and analyzes the points that lie inside this ball. If the data is considered insufficient the ball is stretched until enough information is gained. The algorithm builds the hystogram of the variation in price in this ball assuming that the corresponding distribution approximates the unknown conditional distribution around the input vector. Thus the algorithm to produce a prediction is the following:

1. compute the distance d_i between the prototype $\mathbf{p}_i$ and the input vector for each prototype in the training set;

2. find all the prototypes such that $d_i < \rho$; let N_p be the number of prototypes satisfying this condition;

3. if $N_p < N_{min}$ the data is considered not very significant and the ball radius ρ is enlarged by a given quantity $\Delta\rho$; then we repeat step 2 if the number of repetitions is less than a given number of maximum tries [2];

4. find the class having the maximum normalized frequency $n_c/(1 + \overline{d})$; n_c is the number of vectors of the class c found in the ball defined in step 2; $\overline{d}$ is the average distance between these vectors and the input vector;

[2]This is only used to avoid waste of time due to a sparse distribution of the points.

5. use the price variation associated to the class with maximum normalized frequency.

The use of the normalized frequency introduces a weight to distinguish between classes with similar frequency but different distribution of points. Actually this measure favors the class that is nearer to the input vector.

The prediction obtained with this method should be robust with respect to the presence of outliers in the training data as it uses the median as prediction rather than the mean. Moreover the hypothesis of using quantized classes should reduce the noise in the data. In fact it is impossible to predict a price variation with high accuracy and the reduction in variability due to quantization should make the task easier.

Our system keeps a prototype buffer containing examples from 30 days. The downsampling factor is 15 minutes and the embedding dimension is 8.

4 Conclusions

We tried to build non-linear autoregressive models of financial time series. The results showed that this approach does not produce an encouraging performance. We addressed that some of the possible reasons for this failure are in the non-stationariety of the series we want to model, in the weakness of the model with respect to the poorness of data available to compute its parameters, and in the lack of many inputs that heavily influence the price variations. These results are similar to those reported by other researchers, and they seem quite independent from the choice of the preprocessing of the input data and from the architecture used to produce the predictions.

We propose a different approach to the problem that is based on a classification procedure. This method is very fast in learning because it suffices to store in memory all the values contained in the fitting set. Even retraining is straightforward, as the new values are used to update the fitting test. The burden of this technique lies in the forecasting phase as it is necessary to compute the distances of the new delayed vector with respect to all the delayed vectors in the fitting set.

References

[1] M. Bianchini, M. Gori, and M. Maggini, "On the Problem of Local Minima in Recurrent Neural Networks," *IEEE Transaction on Neural Networks*, Special Issue on Dynamic Recurrent Neural Networks, March 1994, pp. 167–177.

[2] M. C. Casdagli and A. S. Weigend, "Exploring the Continuum Between Deterministic and Stochastic Modeling," in *Predicting the Future and Understanding the Past,* A. Weigend and N. Gershenfeld, Eds. redwood City, CA: Addison-Wesley.

[3] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent Neural Networks and Robust Time Series Prediction," *IEEE Transaction on Neural Networks*, Special Issue on Dynamic Recurrent Neural Networks, March 1994, pp. 167–177.

[4] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Proc. IEEE Trans. Information Theory,* pp. 21–27, 1967.

[5] G. Cybenko "Approximation by Superpositions of a Sigmoidal function," Mathematics of Control, Signals and Systems, n. 2, 1989, pp. 303–314.

[6] M. Gori and A. Tesi, "On the problem of local minima in Backpropagation," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-14, no. 1, pp. 76-86, 1992.

[7] X. Zhang and J. Hutchinson, "Simple Architectures on Fast Machines: Pratical Issues in Nonlinear Time Series Prediction," in A. S. Weigend and N.A. Gershenfeld, *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, 1993.

[8] M. C. Mozer, "Neural Net Architectures for Temporal Sequence Processing," in *Predicting the Future and Understanding the Past,* A. Weigend and N. Gershenfeld, Eds. redwood City, CA: Addison-Wesley.

[9] A.N. Refenes and M. Azena-Barac, "Neural network Applications in Financial Asset Management," *Neural Computing and Applications*, Vol. 2, n. 1, 1994, pp 13–39, Springer Verlag.

[10] T. Sauer, "Time series Prediction by Using Delay Coordinate Embedding," in *Predicting the Future and Understanding the Past,* A. Weigend and N. Gershenfeld, Eds. redwood City, CA: Addison-Wesley.