A Computational Intelligence Based Framework for One-Subsequence-Ahead Forecasting of Nonstationary Time Series

Vasile Georgescu

Department of Mathematical Economics, University of Craiova, A.I.Cuza str. 13, 200585 Craiova, Romania
vgeo@central.ucv.ro

Abstract. This paper proposes a mix of noise filtering, fuzzy clustering, neural mapping and predictive techniques for one-subsequence-ahead forecasting of nonstationary time series. Optionally, we may start with de-noising the time series by wavelet decomposition. A non-overlapping subsequence time series clustering procedure with a sliding window is next addressed, by using a lower-bound of the Dynamic Time Warping distance as a dissimilarity measure, when applying the Fuzzy C-Means algorithm. Afterwards, the subsequence time series transition function is learned by neural mapping, consisting of deriving, for each subsequence time series, the degrees to which it belongs to the c cluster prototypes, when the $p \cdot c$ membership degrees of the previous p subsequences are presented as inputs to the neural network. Finally, this transition function is applied to forecasting one-subsequence-ahead time series, as a weighted mean of the c cluster prototypes to which it belongs, and the S&P 500 data are used for testing.

Keywords: Computational intelligence, Subsequence time series fuzzy clustering, Neural mapping, One-subsequence-ahead forecasting of time series.

1 Introduction

The prediction of financial markets is a very complex task, because the financial time series are inherently noisy, non-stationary, and deterministically chaotic (i.e., short-term random but long-term deterministic). In principle, stock trading can be profitable if the direction of price movement can be predicted consistently. However, due to the "near-random-walk" behavior of stock prices, many experimental works show little evidence of predictability when out-of-sample forecasts are considered.

In order to ameliorate the stock market forecasting accuracy, numerous computational intelligence based techniques have been proposed previously. Among them, feedforward and recurrent neural networks (NNs) gained increasing popularity. Hybridizations of NNs and genetic algorithms (GAs) have also been proposed in an attempt to avoid the local convergence of the gradient descent algorithms and thus to accurately predict the stock price index and the direction of its change. They, however, did not bear outstanding prediction accuracy partly because of the tremendous noise and non-stationary characteristics in stock market data.

V. Torra, Y. Narukawa, and M. Daumas (Eds.): MDAI 2010, LNAI 6408, pp. 187–194, 2010.

On the other hand, the presence of short-term randomness suggests that larger profits can be consistently generated if long-term movements in the stock are accurately predicted rather than short-term movements. Unfortunately, most of the proposed models focused on the accurate forecasting of the levels (i.e. value) of the underlying stock index (e.g., the next day's closing price forecast). Actually, the absolute value of a stock price is usually not as interesting as the shape of up and down movements. As an alternative to *one-value-ahead forecast*, the approach in this paper proposes a novel *one-subsequence-ahead forecasting technique*, which focuses on the predictability of the direction of stock index movement. Our approach also differs from other studies that consider the sign of movements and thus convert the prediction problem into a classification task, which can be carried out with classification tools, such as Support Vector Machines, random forest, logit models and so on.

The proposed framework consists of four stages: the preprocessing stage; the subsequence time series fuzzy clustering stage; the neural mapping based learning stage of the subsequence time series fuzzy transition function; the one-subsequence-ahead time series forecasting stage.

2 Time Series Preprocessing

This stage consists of de-noising data by wavelet decomposition and some other transformations that rely heavily on the selection of a distance measure for clustering.

The Discrete Wavelet Transform (DWT, [8]) uses scaled and shifted versions of a mother wavelet function, usually with compact support, to form either an orthonormal basis (Haar wavelet, Daubechies) or a bi-orthonormal basis (Symlets, Coiflets). Wavelets allow cutting up data into different frequency components (called approximations and details), and then studying each component with a resolution matched to its scale. They can help de-noise inherently noisy data such as financial time series through wavelet shrinkage and thresholding methods, developed by David Donoho ([2]). The idea is to set to zero all wavelet coefficients corresponding to details in the data set that are less than a particular threshold. These coefficients are used in an inverse wavelet transformation to reconstruct the data set. An important advantage is that the de-noising is carried out without smoothing out the sharp structures and thus can help to increase both the clustering accuracy and predictive performance.

Care has to be taken in choosing suitable transformations such that the time series distance measure chosen in the clustering stage is meaningful to the application. Normalization of data is common practice when using Fuzzy C-Means, which means applying scaling and vertical translation to the time series as a whole. Moreover, as we already mentioned, the absolute value of a stock price is not as interesting as the shape of up and down movements. Thus, for allowing stock prices comparisons subsequence by subsequence, a local translation is also necessary, in such a way to have each subsequence starting from zero. A subset of 8192 daily closing prices drawn from the S&P 500 stock index data and used for training, as well as the normalized and de-noised data are shown in Fig. 1, were a level 5 decomposition with Sym8 wavelets and a fixed form soft thresholding were used.

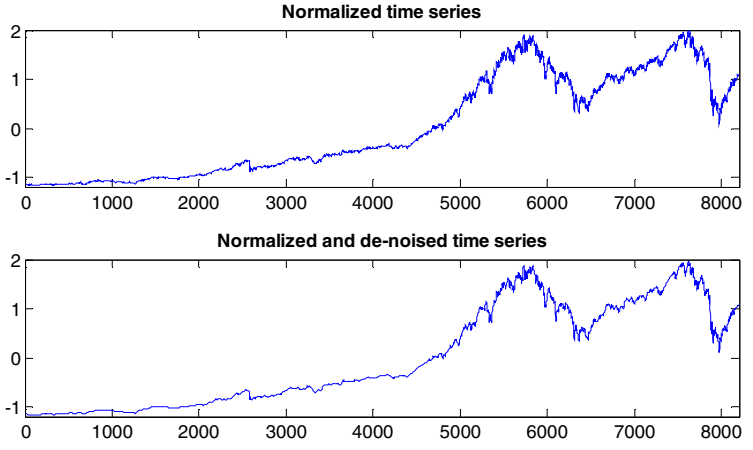

Fig. 1. A normalized and de-noised data subset, drawn from the S&P 500 stock index data

3 Subsequence Time Series Fuzzy Clustering

The idea in subsequence time series (STS) clustering is as follows. Just a single long time series is given at the start of the clustering process, from which we extract short series with a sliding window. The resulting set of subsequences are then clustered, such that each time series is allowed to belong to each cluster to a certain degree, because of the fuzzy nature of the fuzzy c-means algorithm we use. The window width and the time delay between consecutive windows are two key choices. The window width depends on the application; it could be some larger time unit (e.g., 32 days for time series sampled as daily S&P 500 stock index, in our application). Overlapping or non-overlapping windows can be used. If the delay is equal to the window width, the problem is essentially converted to non-overlapping subsequence time series clustering. We will follow this approach, being motivated by the Keogh's criticism presented in [6], where using overlapping windows has been shown to produce meaningless results, due to a surprising anomaly: cluster centers obtained using STS clustering closely resemble "sine waves", irrespective of the nature of original time series itself, being caused by the superposition of slightly shifted subsequences.

Using larger time delays for placing the windows does not really solve the problem as long as there is some overlap. Also, the less overlap, the more problematic the choice of the offsets becomes.

Since clustering relies strongly on a good choice of the dissimilarity measure, this leads to adopting an appropriate distance, depending on the very nature of the subsequence time series.

Let $S = y_m, \ldots, y_{m+w-1}$ be a subsequence with length w of time series $Y = y_1, \ldots, y_n$, where $1 \le m \le n - w + 1$. Subsequences will be represented as vectors in a w-dimensional vector space. For relatively short time series, shape-based distances, such as L_p norms, are commonly used to compare their overall appearance. The Euclidean distance (L_2) is the most widely used shape-based

distance. Other L_p norms can be used as well, such as Manhattan (L_1) and Maximum (L_∞), putting different emphasis on large deviations.

There are several pitfalls when using an L_p distance on time series: it does not allow for different *baselines* in the time sequences; it is very sensitive to *phase shifts* in time; it does not allow for *acceleration and deceleration* along the time axis (time warping). Another problem with L_p distances of time series is when scaling and translation of the amplitudes or the time axis are considered, or when outliers and noisy regions are present.

A number of non-metric distance measures have been defined to overcome some of these problems. Small distortions of the time axis are commonly addressed with non-uniform time warping, more precisely with Dynamic Time Warping (DTW, [1], [5]). The DTW distance is an extensively used technique in speech recognition and allows warping of the time axes (acceleration–deceleration of signals along the time dimension) in order to align the shapes of the two time series better. The two series can also be of different lengths. The optimal alignment is found by calculating the shortest warping path in the matrix of distances between all pairs of time points under several constraints (boundary conditions, continuity, monotonicity).

The warping path is also constrained in a global sense by limiting how far it may stray from the diagonal. The subset of the matrix that the warping path is allowed to visit is called the warping window. The two most common constraints in the literature are the Sakoe-Chiba band and the Itakura parallelogram. We can view a global or local constraint as constraining the indices of the warping path $w_k = (i, j)_k$, such that $j - r \le i \le j + r$, where r is a term defining the allowed range of warping, for a given point in a sequence. In the case of the Sakoe-Chiba band (see Fig. 2), r is independent of i; for the Itakura parallelogram, r is a function of i.

DTW is a much more robust distance measure for time series than L_2, allowing similar shapes to match even if they are out of phase in the time axis. Unfortunately, however, DWT is calculated using dynamic programming with time complexity $O(n^2)$. Recent approaches focus more on approximating the DTW distance by bounding it from below. For example, a novel, linear time (i.e., with complexity reduced to $O(n)$), lower bound of the DTW distance, was proposed in [7]. The

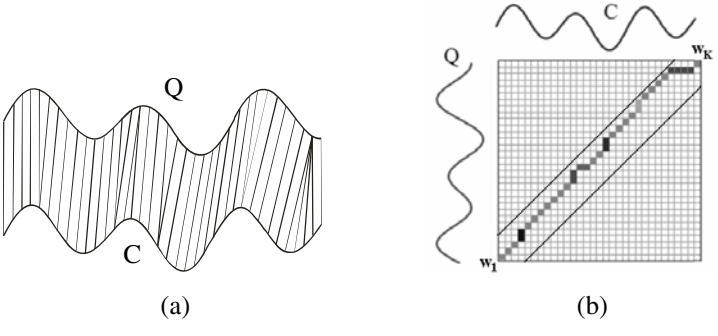

Fig. 2. (a) Aligning two time sequences using DTW. (b) Optimal warping path with the Sakoe-Chiba band as global constraints.

intuition behind the approach is the construction of a special "envelope" around the query. It can be shown that the Euclidean distance between a potential match and the nearest orthogonal point on the envelope lower bounds the DTW distance. To index this representation, an approximate bounding envelope is created.

Let $Q = \{q_1, \ldots, q_n\}$ and $C = \{c_1, \ldots, , c_m\}$ be two subsequences and $w_k = (i, j)_k$ be the warping path, such that $j - r \le i \le j + r$, where r is a term defining the range of warping for a given point in a sequence. The term r can be used to define two new sequences, L and U, where $L_i = \min(q_{i-r} : q_{i+r})$, $U_i = \max(q_{i-r} : q_{i+r})$, with L and U standing for *Lower* and *Upper*, respectively. An obvious but important property of L and U is the following: $\forall i, U_i \ge q_i \ge L_i$. Given L and U, a lower bounding measure for DTW can now be defined (see Fig. 3):

$$LB\text{-}Keogh(Q, C) = \sqrt{\begin{cases} \sum_{i=1}^{n} (c_i - U_i)^2 & if \quad c_i > U_i \\ \sum_{i=1}^{n} (c_i - U_i)^2 & if \quad c_i < U_i \\ 0 & otherwise \end{cases}}. \tag{1}$$

We are now going to generalize the fuzzy c-means algorithm to subsequence time series clustering. In this particular context, the entities to be clustered, denoted by x_k, and the cluster prototypes (centroids), denoted by v_i, are both set-defined objects, i.e. subsequence time series. The centroids are computed as weighted means, where the weights, denoted by u_{ik}, are the fuzzy membership degrees to which each subsequence belongs to a cluster. Both the *DTW* and *LB-Keogh* distances outperform L_2 and thus qualify better to be used with the fuzzy c-means algorithm. However, the *LB-Keogh*'s lower bound of DTW distance has been preferred, due to its linear time complexity. Fig. 4 plots the cluster centroids (prototypes) and the subsequence time series grouped around each centroid.

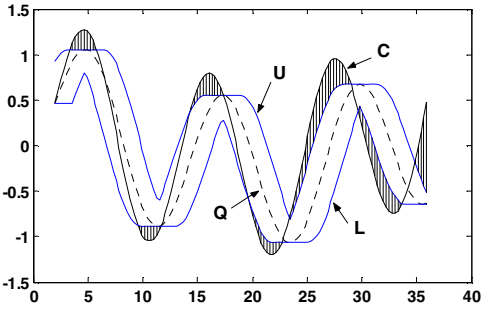

Fig. 3. The lower bounding function LB-Keogh(Q,C). The original sequence Q is enclosed in the bounding envelope of U and L.

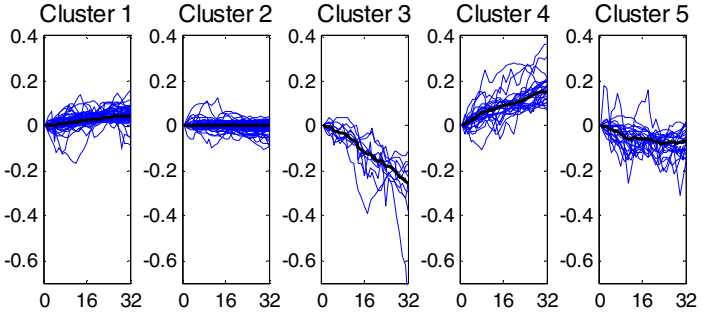

Fig. 4. The cluster centroids and the subsequence time series grouped around each centroid

4 Estimation of the Fuzzy Transition Function between Clusters by Neural Mapping

At this stage, a fuzzy transition function between clusters must be learned, which is a nonlinear vector function mapping a number of p c-dimensional membership degree vectors $\mu(STS_{t-j+1})$, $j=1,\ldots,p$, into a c-dimensional membership degree vector $\mu(STS_{t+1})$, i.e., $\mu(STS_{t+1}) = f\left(\mu(STS_t),\ldots,\mu(STS_{t-p+1})\right)$, where STS_{t+1} is the subsequence time series to be predicted. In our experiment, $p=2$ and $c=5$.

Neural networks are well known for their capability to be universal approximators (i.e., to estimate almost any computable function on a compact set arbitrarily closely, provided that enough experimental data are available). Actually, we use a *multilayer perceptron* network with two layers: one hidden layer with the tan-sigmoid transfer function and one output layer with the log-sigmoid transfer function (the latter allows constraining the output of the network between 0 and 1). The dimensions of input and

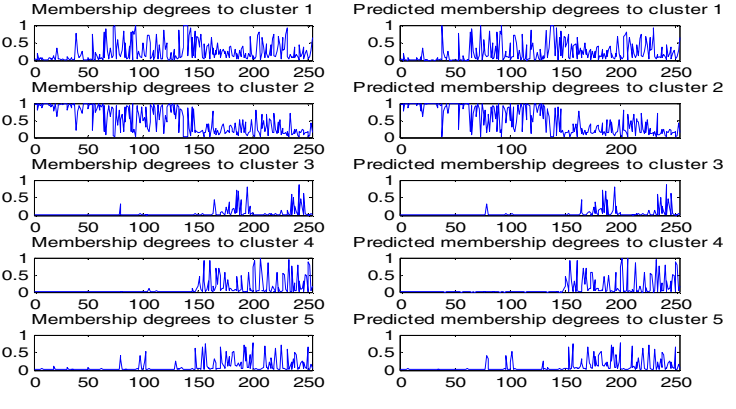

Fig. 5. Accurate neural mapping: actual and predicted membership degrees to which each of the 256 subsequence time series belongs to one of the 5 clusters

output spaces are $p \cdot c$ and c, respectively. This neural architecture is known to have the capability of approximating any nonlinear function with a finite number of discontinuities arbitrarily well, given sufficient neurons in the hidden layer.

5 One-Subsequence-Ahead Forecasting of Time Series

The one-subsequence-ahead forecast can then be obtained as a weighted mean of the c cluster prototypes (v_i), each one representing a subsequence time series:

$$STS_{t+1} = \sum_{i=1}^{c} v_i \cdot \mu\left(STS_{t+1}\right). \tag{2}$$

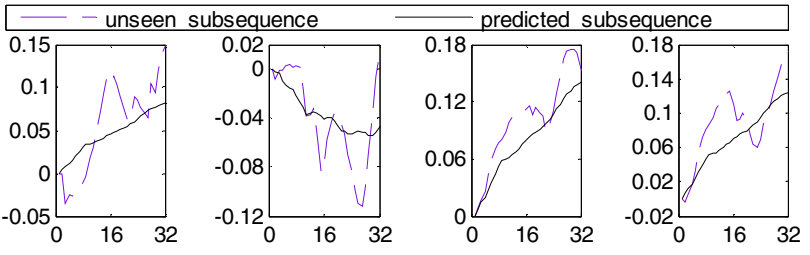

Fig. 6. One-subsequence-ahead forecasting of 4 unseen subsequences (i.e., not included in the training dataset). This covers 4*32=128 daily stock price index value forecasts.

Each forecast is a 32-length subsequence, obtained as follows: the membership degrees of the previous p subsequences to each of the c clusters are first computed; the membership degrees to which the next subsequence belongs to each cluster are then found by neural mapping; these membership degrees are finally used to compute the weighted mean of the c cluster prototypes (v_i), representing the forecast.

6 Conclusion

Predicting price levels is an intriguing, challenging, and admittedly risky endeavor. Technical analysis uses trend following strategies to forecast future price movements and to infer trading decision rules, based on the assertion that price changes have inertia. Although experimental works show little evidence of predictability, many traders consider accuracy rates of about (or greater than) 55% to be consistently profitable. However, one-value-ahead forecasting of price levels is not as useful as the shape of long-term up and down movements, due to their inherent short-term randomness. The approach in this paper proposed a novel one-subsequence-ahead forecasting framework, based on a mix of computational intelligence techniques that allow the prediction of stock index movements in a more robust way, focusing on predicting one price subsequence rather than one price level at a time.

References

1. Berndt, D.J., Clifford, J.: Finding patterns in time series: A dynamic programming approach. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 229–248. AAAI Press, Menlo Park (1996)
2. Donoho, D.: Nonlinear Wavelet Methods for Recovery of Signals, Densities and Spectra from Indirect and Noisy Data. In: Daubechies, I. (ed.) Different Perspectives on Wavelets, Proceeding of Symposia in Applied Mathematics, vol. 47, pp. 173–205. Amer. Math. Soc., Providence (1993)
3. Georgescu, V.: Generalizations of Fuzzy C-Means Algorithm to Granular Feature Spaces, based on Underlying Fuzzy Metrics: Issues and Related Works. In: 13th IFSA World Congress and 6th Conference of EUSFLAT, Lisbon, Portugal, pp. 1791–1796 (2009)
4. Georgescu, V.: A Time Series Knowledge Mining Framework Exploiting the Synergy between Subsequence Clustering and Predictive Markovian Models. Fuzzy Economic Review XIV(1), 41–66 (2009)
5. Keogh, E., Pazzani, M.J.: Scaling up dynamic time warping to massive datasets. In: Żytkow, J.M., Rauch, J. (eds.) PKDD 1999. LNCS (LNAI), vol. 1704, pp. 1–11. Springer, Heidelberg (1999)
6. Keogh, E., Lin, J., Truppel, W.: Clustering of time series subsequences is meaningless: implications for previous and future research. In: 3rd IEEE International Conference on Data Mining, pp. 115–122 (2003)
7. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowledge and Information Systems 7, 358–386 (2005)
8. Mallat, S.G., Peyré, G.: A Wavelet Tour of Signal Processing: The Sparse Way, 3rd edn. Academic Press, London (2009)