Machine Learning Strategies for Time Series Forecasting

Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne

Machine Learning Group
Computer Science Department, Faculty of Sciences
ULB, Université Libre de Bruxelles
Bd Triomphe, 1050, Brussels, Belgium
{gbonte,sbentaieb,ylebORGn}@ulb.ac.be
http://mlg.ulb.ac.be

Abstract. The increasing availability of large amounts of historical data and the need of performing accurate forecasting of future behavior in several scientific and applied domains demands the definition of robust and efficient techniques able to infer from observations the stochastic dependency between past and future. The forecasting domain has been influenced, from the 1960s on, by linear statistical methods such as ARIMA models. More recently, machine learning models have drawn attention and have established themselves as serious contenders to classical statistical models in the forecasting community. This chapter presents an overview of machine learning techniques in time series forecasting by focusing on three aspects: the formalization of one-step forecasting problems as supervised learning tasks, the discussion of local learning techniques as an effective tool for dealing with temporal data and the role of the forecasting strategy when we move from one-step to multiple-step forecasting.

Keywords: Time series forecasting, machine learning, local learning, lazy learning, MIMO.

1 Introduction

A *time series* is a sequence S of historical measurements y_t of an observable variable y at equal time intervals. Time series are studied for several purposes such as the forecasting of the future based on knowledge of the past, the understanding of the phenomenon underlying the measures, or simply a succinct description of the salient features of the series. In this chapter we shall confine ourselves to the problem of forecasting. Forecasting future values of an observed time series plays an important role in nearly all fields of science and engineering, such as economics, finance, business intelligence, meteorology and telecommunication [43]. An important aspect of the forecasting task is represented by the size of the horizon. If the one-step forecasting of a time series is already a challenging task, performing multi-step forecasting is more difficult [53] because of

M.-A. Aufaure and E. Zimányi (Eds.): eBISS 2012, LNBIP 138, pp. 62–77, 2013.

additional complications, like accumulation of errors, reduced accuracy, and increased uncertainty [58,49].

The forecasting domain has been influenced, for a long time, by linear statistical methods such as ARIMA models. However, in the late 1970s and early 1980s, it became increasingly clear that linear models are not adapted to many real applications [25]. In the same period, several useful nonlinear time series models were proposed such as the bilinear model [44], the threshold autoregressive model [56,54,55] and the autoregressive conditional heteroscedastic (ARCH) model [22] (see [25] and [26] for a review). However, the analytical study of nonlinear time series analysis and forecasting is still in its infancy compared to linear time series [25].

In the last two decades, machine learning models have drawn attention and have established themselves as serious contenders to classical statistical models in the forecasting community [1,43,61]. These models, also called black-box or data-driven models [40], are examples of nonparametric nonlinear models which use only historical data to learn the stochastic dependency between the past and the future. For instance, Werbos found that Artificial Neural Networks (ANNs) outperform the classical statistical methods such as linear regression and Box-Jenkins approaches [59,60]. A similar study has been conducted by Lapedes and Farber [33] who conclude that ANNs can be successfully used for modeling and forecasting nonlinear time series. Later, other models appeared such as decision trees, support vector machines and nearest neighbor regression [29,3]. Moreover, the empirical accuracy of several machine learning models has been explored in a number of forecasting competitions under different data conditions (e.g. the NN3, NN5, and the annual ESTSP competitions [19,20,34,35]) creating interesting scientific debates in the area of data mining and forecasting [28,45,21].

This chapter aims to present an overview of the role of machine learning techniques in time series forecasting by focusing on three aspects: the formalization of one-step forecasting problems as supervised learning tasks, the discussion of local learning techniques as an effective tool for dealing with temporal data and the role of the forecasting strategy when we move from one-step to multi-step forecasting.

The outline of the chapter is as follows. Section 2 introduces some basic notions of time series modeling and the formalization of the forecasting task as an input-output problem. Section 3 discusses the role of machine learning techniques in inferring accurate predictors from observed data and introduces the local learning paradigm. Section 4 presents several strategies for multi-step forecasting which have been proposed so far in literature. Section 5 reviews how local learning techniques have been integrated with multiple-step strategies to perform accurate multi-step forecasts.

2 Forecasting and Modeling

Two main interpretations of the forecasting problem on the basis of historical dataset exist. Statistical forecasting theory assumes that an observed sequence is a specific realization of a random process, where the randomness arises from many independent degrees of freedom interacting linearly [4]. However, the emergent view in dynamical systems theory [23,17] is that apparently random behavior may be generated by deterministic systems with only a small number of degrees of freedom, interacting nonlinearly. This complicated and aperiodic behavior is also called *deterministic chaos* [48].

We adopt the working hypothesis that many classes of experimental time series may be analyzed within the framework of a dynamical systems approach. Therefore the time series is interpreted as the observable of a dynamical system whose state s evolves in a state space $\varGamma \subset \Re^g$, according to the law

$$s(t) = \mathcal{F}^t(s(0)) \tag{1}$$

where $\mathcal{F} : \varGamma \to \varGamma$ is the map representing the dynamics, $\mathcal{F}^t$ is its iterated versions and $s(t) \in \varGamma$ denotes the value of the state at time t.

In the absence of noise the time series is related to the dynamical system by the relation

$$y_t = \mathcal{G}(s(t)) \tag{2}$$

where $\mathcal{G} : \varGamma \to \Re^D$ is called the *measurement function* and D is the dimension of the series. In the following we will restrict to the case $D = 1$ (*univariate time series*).

Both the function $\mathcal{F}$ and $\mathcal{G}$ are unknown, so in general we cannot hope to reconstruct the state in its original form. However, we may be able to recreate a state space that is in some sense equivalent to the original.

The *state space reconstruction problem* consists in reconstructing the state when the only available information is contained in the sequence of observations y_t. State space reconstruction was introduced into dynamical systems theory independently by Packard *et al.* [42] and Takens [52]. The Takens theorem implies that for a wide class of deterministic systems, there exists a mapping (*delay reconstruction map*) $\varPhi : \varGamma \to \Re^n$

$$\varPhi(s(t)) = \{\mathcal{G}(\mathcal{F}^{-d}(s(t))), \dots, \mathcal{G}(\mathcal{F}^{-d-n+1}(s(t)))\} = \{y_{t-d}, \dots, y_{t-d-n+1}\} \tag{3}$$

between a finite window of the time series $\{y_{t-d}, \dots, y_{t-d-n+1}\}$ (*embedding vector*) and the state of the dynamic system underlying the series, where d is called the *lag time* and n (*order*) is the number of past values taken into consideration. Takens showed that generically $\varPhi$ is an *embedding* when $n \geq 2g + 1$, where embedding stays for a smooth one-to-one differential mapping with a smooth inverse [17]. The main consequence is that, if $\varPhi$ is an embedding then a smooth dynamics $f : \Re^n \to \Re$ is induced in the space of reconstructed vectors

$$y_t = f(y_{t-d}, y_{t-d-1}, \dots, y_{t-d-n+1}) \tag{4}$$

This implies that the reconstructed states can be used to estimate f and consequently f can be used in alternative to F and G, for any purpose concerning time series analysis, qualitative description, forecasting, etc.

The representation (4) does not take into account any noise component, since it assumes that a deterministic process f can accurately describe the time series. Note, however, that this is simply one possible way of representing the time series phenomenon and that any alternative representation should not be discarded a priori. In fact, once we assume that we have not access to an accurate model of the function f, it is perfectly reasonable to extend the deterministic formulation (4) to a statistical Nonlinear Auto Regressive (NAR) formulation

$$y_t = f\left(y_{t-d}, y_{t-d-1}, \ldots, y_{t-d-n+1}\right) + w(t) \tag{5}$$

where the missing information is lumped into a noise term $\mathbf{w}$. In the rest of the chapter, we will then refer to the formulation (5) as a general representation of the time series which includes as particular instance also the case (4).

The success of a reconstruction approach starting from a set of observed data depends on the choice of the hypothesis that approximates f, the choice of the order n and the lag time d.

In the following section we will address only the problem of the modeling of f, assuming that the values of n and d are available a priori. A good reference on the order selection is given in Casdagli *et al.* [17].

3 Machine Learning Approaches to Model Time Dependencies

3.1 Supervised Learning Setting

The embedding formulation in (5) suggests that, once a historical record S is available, the problem of one-step forecasting can be tackled as a problem of supervised learning. Supervised learning consists in modeling, on the basis of a finite set of observations, the relation between a set of *input* variables and one or more *output* variables, which are considered somewhat dependent on the inputs. Once a model of the mapping (5) is available, it can be used for one-step forecasting. In one-step forecasting, the n previous values of the series are available and the forecasting problem can be cast in the form of a generic regression problem as shown in Fig. 1.

The general approach to model an input/output phenomenon, with a scalar output and a vectorial input, relies on the availability of a collection of observed pairs typically referred to as *training set*.

In the forecasting setting, the training set is derived by the historical series S by creating the $[(N-n-1) \times n]$ input data matrix

$$X = \begin{bmatrix} y_{N-1} & y_{N-2} & \cdots & y_{N-n-1} \\ y_{N-2} & y_{N-3} & \cdots & y_{N-n-2} \\ \vdots & \vdots & \vdots & \vdots \\ y_n & y_{n-1} & \cdots & y_1 \end{bmatrix} \tag{6}$$

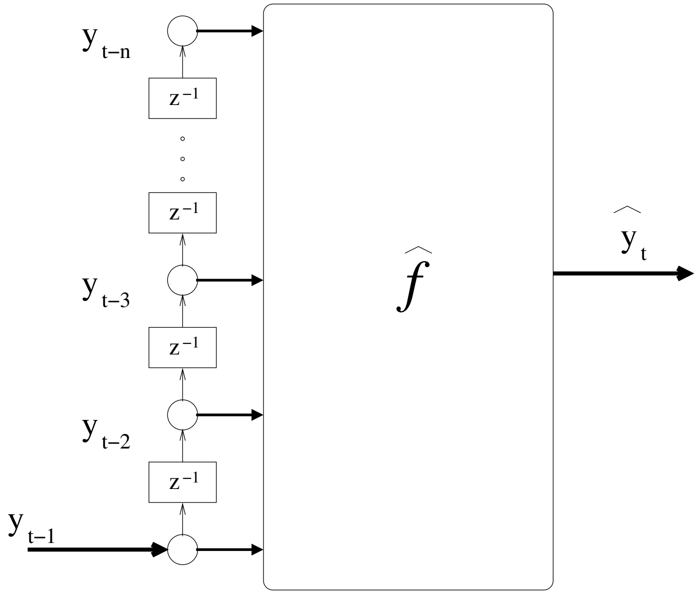

Fig. 1. One-step forecasting. The approximator $\hat{f}$ returns the prediction of the value of the time series at time $t+1$ as a function of the n previous values (the rectangular box containing z^{-1} represents a unit delay operator, i.e., $y_{t-1} = z^{-1} y_t$).

and the $[(N-n-1) \times 1]$ output vector

$$Y = \begin{bmatrix} y_N \\ y_{N-1} \\ \vdots \\ y_{n+1} \end{bmatrix} \tag{7}$$

For the sake of simplicity, we assume here a $d = 0$ lag time. Henceforth, in this chapter we will refer to the i^{th} row of X, which is essentially a temporal pattern of the series, as to the (reconstructed) *state* of the series at time $t-i+1$.

3.2 Instantiation with Local Learning

Forecasting one-step-ahead consists then in predicting the value of the output when a subset of past observed values (also denoted as query) is given. Machine learning provides a theoretical framework to estimate from observed data a suitable model of the time dependency f. Because of the impossibility of reviewing here the entire state-of-the-art of machine learning in time series forecasting, we will more specifically consider local learning techniques [12,31,29] in the following section. This choice is motivated by the following reasons:

- Reduced number of assumptions: local learning assumes no a priori knowledge on the process underlying the data. For example, it makes no assumption on the existence of a global function describing the data and no assumptions on the properties of the noise. The only available information is represented by a finite set of input/output observations. This feature is particularly relevant in real datasets where problems of missing features, non- stationarity and measurement errors make appealing a data-driven and assumption-free approach.

– On-line learning capability: The local learning method can easily deal with on-line learning tasks where the number of training samples increases with time. In this case, local learning simply adds new points to the dataset and does not need time-consuming re-training when new data become available.
– Modelling non-stationarity: The local learning method can deal with time-varying configurations where the stochastic process underlying the data is non-stationary. In this case, it is sufficient to interpret the notion of neighbourhood not in a spatial way but both in a spatial and temporal sense. For each query point, the neighbours are no more the samples that have similar inputs but the ones that both have similar inputs and have been collected recently in time. Therefore, the time variable becomes a further precious feature to consider for accurate prediction.

We describe in the following two instances of local learning techniques, namely Nearest Neighbor [36,29] and Lazy Learning [12,5].

Nearest Neighbor. The Nearest Neighbor method is the most trivial example of local approximation applied to the problem of time series forecasting. This method consists in looking through the data set for the nearest neighbor of the current state and predicting that the current state will evolve in the same manner as the neighbor did.

Figure 2 represents an example of nearest-neighbor one-step forecasting. Suppose we have available a time series y_t up to time $\bar{t}-1$ and we want to predict the next value of the series. Once selected a certain dimension n, for example $n = 6$, the nearest neighbor approach searches for the pattern in the past which is the most similar, in a given metric, to the pattern $\{y_{\bar{t}-6}, y_{\bar{t}-5}, \dots, y_{\bar{t}-1}\}$ (the dashed line). If the nearest pattern is, for instance, $\{y_{\bar{t}-16}, y_{\bar{t}-15}, \dots, y_{\bar{t}-11}\}$, then the forecasts $\hat{y}_{\bar{t}}$ returned by the NN method is the value $y_{\bar{t}-10}$ (black dot).

This approach was first proposed by Lorenz [36] to examine weather maps. Imagine that we want to predict tomorrow's weather in Bruxelles and that we choose a dimension $n = 1$. The nearest neighbor approach suggests (i) to search the historical database of the meteorological conditions in Bruxelles, (ii) to find the weather pattern most similar to that of today (for example the weather pattern on March 5th, 1999, by chance a rainy day!) and (iii) to predict that tomorrow's weather will be the same as March 6th, 1999 (just by chance another rainy day!!).

Natural extensions of the Nearest Neighbor approach consider more neighbors [31] or higher order approximations. Piecewise linear approximation in time series analysis was introduced by Tong and Lim [56]. Priestley [46] suggested the importance of higher order approximations. Farmer and Sidorowich [23,24]

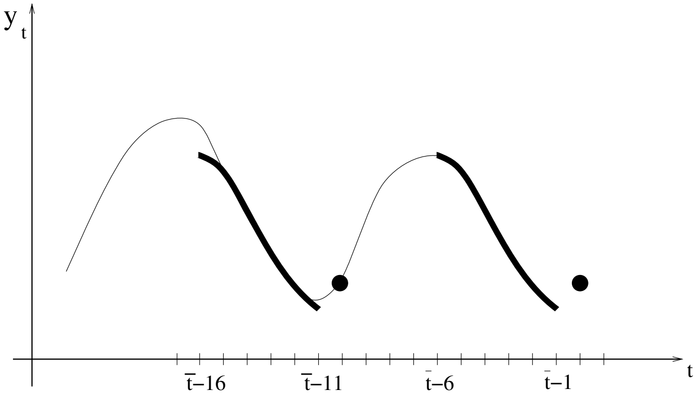

Fig. 2. Nearest-neighbor one-step-ahead forecasts. We want to predict at time $\bar{t}-1$ the next value of the series y of order $n = 6$. The pattern $y_{\bar{t}-16}, y_{\bar{t}-15}, \dots, y_{\bar{t}-11}$ is the most similar to the pattern $\{y_{\bar{t}-6}, y_{\bar{t}-5}, \dots, y_{\hat{t}-1}\}$. Then, the prediction $\hat{y}_{\bar{t}} = y_{\bar{t}-10}$ is returned.

studied local approximation in time series and demonstrated its effectiveness on several experiments and numerical time series analysis. In particular they applied local learning techniques to predict the behavior of chaotic time series, sequences which, although deterministic, are characterized by second-order persistent statistics with random properties.

Lazy Learning. The Lazy Learning (LL) is a lazy and local learning machine [12,11] which automatically adapts the size of the neighborhood on the basis of a cross-validation criterion. The major appeal of Lazy Learning is its divide-and-conquer nature: Lazy Learning reduces a complex and nonlinear modeling problem into a sequence of easily manageable local linear problems, one for each query. This allows to exploit, on a local basis, the whole range of linear identification and validation techniques which are fast, reliable, and come with a wealth of theoretical analyses, justifications, and guarantees. The Lazy Learning procedure essentially consists of the following steps once the matrix X in (6) and Y in (7) and a query point $\mathbf{x}_q$ are given:

1. Sort increasingly the set of vectors in X with respect to the distance (e.g. Euclidean) to $\mathbf{x}_q$.
2. Determine the optimal number of neighbors.
3. Calculate, given the number of neighbors, the prediction for the query point by using a local model (e.g. constant or linear).

Let us consider a time series $\{y_1, \dots, y_t\}$ composed of t observations for which we intend to predict the next one.

The forecasting problem boils down to estimating the output $\hat{y}_{t+1}$ when the latest window of observations is represented by the vector $\mathbf{x}_q = \{y_t, \dots, y_{t-n+1}\}$. Algorithm 1 illustrates how constant local learning techniques return the output associated to a query point $\mathbf{x}_q$, for a given number of neighbors k. The notation $[j]$ is used to designate the index of the jth closest neighbor of $\mathbf{x}_q$. Note that also the local linear version of the algorithm is commonly used, as discussed in [11].

Algorithm 1. LL

```
Input  : $D = \{(\mathbf{x}_i, y_i) \in (\mathbb{R}^n \times \mathbb{R})\}$, dataset.
Input  : $\mathbf{x}_q \in \mathbb{R}^d$, query point.
Input  : $k$= the number of neighbors.
Output: $\hat{y}_{t+1}$, the estimation of the output of the query point $\mathbf{x}_q$
        (obtained with $k$ neighbors).
Sort increasingly the set of vectors $\{\mathbf{x}_i\}$ with respect to the distance to $\mathbf{x}_q$.
$\hat{y}_{t+1} = \frac{1}{k}\sum_{j=1}^{k} y_{[j]}$.
return $\hat{y}_{t+1}$.
```

This algorithm requires the choice of a set of model parameters (e.g. the number k of neighbors, the kernel function, the distance metric) [5]. We will discuss here an automatic method based on a Leave-One-Out (LOO) criterion to determine the number of neighbor [11,12]. The main idea is to assess the quality of each local model by using a LOO measure and to select the best neighborhood size according to such measure.

A computationally efficient way to perform LOO cross-validation and to assess the performance in generalization of local linear models is the PRESS statistic, proposed in 1974 by Allen [2]. By assessing the performance of each local model, alternative configurations can be tested and compared in order to select the best one in terms of expected prediction. The idea consists in associating an LOO error $e_{LOO}(k)$ to the estimation

$$\hat{y}_q^{(k)} = \frac{1}{k}\sum_{j=1}^{k} y_{[j]}, \tag{8}$$

associated to the query point $\mathbf{x}_q$ and returned by k neighbors. In case of a constant model, the LOO term can be derived as follows [12]:

$$e_{LOO}(k) = \frac{1}{k}\sum_{j=1}^{k}(e_j(k))^2, \tag{9}$$

where

$$e_j(k) = y_{[j]} - \frac{\sum_{i=1(i\neq j)}^{k} y_{[i]}}{k-1} = k\frac{y_{[j]} - \hat{y}_k}{k-1}. \tag{10}$$

The best number of neighbors is then defined as the number

$$k^* = \arg\min_{k\in\{2,\ldots,K\}} e_{LOO}(k), \tag{11}$$

which minimizes the LOO error.

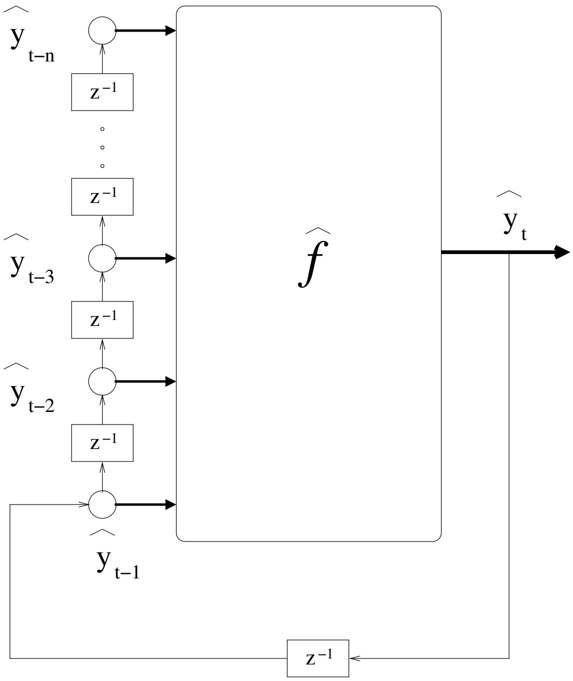

Fig. 3. Iterated prediction. The approximator $\hat{f}$ returns the prediction of the value of the time series at time $t+1$ by iterating the predictions obtained in the previous steps (the rectangular box containing z^{-1} represents a unit delay operator, i.e., $\hat{y}^{t-1} = z^{-1}\hat{y}^{t}$).

Lazy learning was applied with success to several regression and one-step forecasting tasks [14]. More details on the LL technique and its applications can be found in [11,12].

4 Strategies for Multi-step Time Series Forecasting

The previous section showed that one-step forecasting can be cast in a conventional supervised learning framework by having recourse to conventional learning techniques such as Local Learning. In this section, we extend the framework to show how learning techniques can be used to tackle the multi-step forecasting problem. Three strategies can be considered, namely recursive, direct and multiple output strategies.

A multi-step time series forecasting task consists of predicting the next H values $[y_{N+1}, \ldots, y_{N+H}]$ of a historical time series $[y_1, \ldots, y_N]$ composed of N observations, where $H > 1$ denotes the forecasting horizon.

This section will give a presentation of the three existing strategies to adopt machine learning in multi-step forecasting. We will use a common notation where f and F denote the functional dependency between past and future observations,

n refers to the embedding dimension [17] of the time series, that is the number of past values used to predict future values and w represents the term that includes modeling error, disturbances and/or noise.

4.1 Recursive Strategy

The *Recursive* strategy [58,49,18] trains first a one-step model f

$$y_{t+1} = f(y_t, \dots, y_{t-n+1}) + w_{t+1}, \tag{12}$$

with $t \in \{n, \dots, N-1\}$ and then uses it recursively for returning a multi-step prediction (Figure 3). A well-known drawback of the recursive method is its sensitivity to the estimation error, since estimated values, instead of actual ones, are more and more used when we get further in the future.

In spite of these limitations, the Recursive strategy has been successfully used to forecast many real-world time series by using different machine learning models, like recurrent neural networks [47] and nearest-neighbors [38,15].

4.2 Direct Strategy

The *Direct* strategy [58,49,18] learns independently H models f_h

$$y_{t+h} = f_h(y_t, \dots, y_{t-n+1}) + w_{t+h}, \tag{13}$$

with $t \in \{n, \dots, N-H\}$ and $h \in \{1, \dots, H\}$ and returns a multi-step forecast by concatenating the H predictions.

Since the Direct strategy does not use any approximated values to compute the forecasts (Equation 13), it is not prone to any accumulation of errors. Notwithstanding, it has some weaknesses. First, since the H models are learned independently no statistical dependencies between the predictions $\hat{y}_{N+h}$[13,16,32] is considered. Second direct methods often require higher functional complexity [54] than iterated ones in order to model the stochastic dependency between two series values at two distant instants [27]. Last but not least, this strategy demands a large computational time since the number of models to learn is equal to the size of the horizon.

Different machine learning models have been used to implement the Direct strategy for multi-step forecasting tasks, for instance neural networks [32], nearest neighbors [49] and decision trees [57].

4.3 DirRec Strategy

The *DirRec* strategy [50] combines the architectures and the principles underlying the Direct and the Recursive strategies. DirRec computes the forecasts with different models for every horizon (like the Direct strategy) and, at each time step, it enlarges the set of inputs by adding variables corresponding to the forecasts of the previous step (like the Recursive strategy). However, note that unlike the two previous strategies, the embedding size n is not the same for all the horizons. In other terms, the DirRec strategy learns H models f_h from the time series $[y_1, \dots, y_N]$ where

$$y_{t+h} = f_h(y_{t+h-1}, \dots, y_{t-n+1}) + w_{t+h}, \tag{14}$$

with $t \in \{n, \dots, N-H\}$ and $h \in \{1, \dots, H\}$.

4.4 Multiple Output Strategies

In spite of their diversity, iterated and direct techniques for multiple-step forecasting share a common feature: they model from data a multi-input single-output mapping whose output is the variable y_{t+1} in the iterated case and the variable y_{t+k} in the direct case, respectively. When a very long term prediction is at stake and a stochastic setting is assumed, the modeling of a single-output mapping neglects the existence of stochastic dependencies between future values, (e.g. between y_{t+k} and y_{t+k+1}) and consequently biases the prediction accuracy. A possible way to remedy to this shortcoming is to move from the modeling of single-output mappings to the modeling of multi-output dependencies. This requires the adoption of multi-output techniques where the predicted value is no more a scalar quantity but a vector of future values of the time series.

The MIMO Strategy. The *Multi-Input Multi-Output* (MIMO) strategy [13,16] (also known as Joint strategy [32]) avoids the simplistic assumption of conditional independence between future values made by the Direct strategy [13,16] by learning a single multiple-output model

$$[y_{t+H}, \dots, y_{t+1}] = F(y_t, \dots, y_{t-n+1}) + \mathbf{w}, \tag{15}$$

where $t \in \{n, \dots, N-H\}$, $F : \mathbb{R}^d \to \mathbb{R}^H$ is a vector-valued function [39], and $\mathbf{w} \in \mathbb{R}^H$ is a noise vector with a covariance that is not necessarily diagonal [37].

The forecasts are returned in one step by a multiple-output model $\hat{F}$ where

$$[\hat{y}_{t+H}, \dots, \hat{y}_{t+1}] = \hat{F}(y_N, \dots, y_{N-n+1}). \tag{16}$$

The rationale of the MIMO strategy is to model, between the predicted values, the stochastic dependency characterizing the time series. This strategy avoids the conditional independence assumption made by the Direct strategy as well as the accumulation of errors which plagues the Recursive strategy. So far, this strategy has been successfully applied to several real-world multi-step time series forecasting tasks [13,16,10,9].

However, the wish to preserve the stochastic dependencies constrains all the horizons to be forecasted with the same model structure. Since this constraint could reduce the flexibility of the forecasting approach [10], a variant of the MIMO strategy is discussed in the following section.

The DIRMO Strategy. The DIRMO strategy [10,9] aims to preserve the most appealing aspects of DIRect and MIMO strategies by partitioning the horizon H in several blocks, and using MIMO to forecast the values inside each block. This means that the H-step forecast requires m multiple-output forecasting tasks ($m = \frac{H}{s}$), each having an output of size s ($s \in \{1, \ldots, H\}$).

Note that for $s = 1$, the DIRMO coincides with the conventional Direct strategy, while for $s = H$ it corresponds to the MIMO strategy. The tuning of the parameter s allows us to improve the flexibility of the MIMO strategy by calibrating the dimensionality of the outputs (no dependency in the case $s = 1$ and maximal dependency for $s = H$). This provides a beneficial trade off between the preserving a larger degree of the stochastic dependency between future values and having a greater flexibility of the predictor.

5 Local Learning for Multi-step Forecasting

Local learning appears to be an effective algorithm not only for one-step but also for multi-step forecasting. This section discusses some works which used local learning techniques to deal specifically with the long term forecasting problem.

In [38,15] the authors proposed a modification of the local learning technique to take into account the temporal behavior of the multi-step forecasting problem and consequently improve the results of the recursive strategies. In particular [15] modified the PRESS criterion (10) by introducing an iterated version of the leave-one-out statistic. They showed that the iterated PRESS outperforms a non-iterated criterion by assessing the generalization performance of a local one-step predictor on a horizon longer than a single step, yet preserving nice properties of computational efficiency. It is worth noting that the two techniques proposed by [38] and [15] ranked respectively first and second in the 1998 Leuven time series prediction.

A recent improvement of the recursive strategy based again on local learning is RECNOISY [6], which perturbs the initial dataset at each step of the forecasting process to handle more properly the approximated values in the prediction process. The rationale of the RECNOISY method is that the training examples used by the recursive strategy, though observed, are not necessarily representative of the forecasting tasks which will be required later all along the forecasting process. To remedy to this problem, this strategy exploits the particular nature of the forecasting tasks induced by the recursive strategy and incorporates it in the local learning phase in order to improve the results.

Two improvements of Lazy Learning to deal with long-term prediction of time series are presented in [51]. The first method is based on an iterative pruning of the inputs; the second one performs a brute force search in the possible set of inputs using a k-NN approximator.

The use of local learning for multi-input multi-output prediction was proposed in [13] where a multi-output extension of the algorithm 1 is discussed as well as an averaging strategy of several long term predictors to improve the resulting accuracy.

The use of the local learning approximator to implement a DIRMO strategy is presented in [8,9]. The DIRMO strategy based on local learning has been successfully applied to two forecasting competitions: ESTSP'07 [10] and NN3 [9].

A detailed review and comparison of strategies for multi-step time series forecasting based on the local learning algorithm is presented in [7].

6 Conclusion

Predicting the future is one of the most relevant and challenging tasks in applied sciences. Building effective predictors form historical data demands computational and statistical methods for inferring dependencies between past and short-term future values of observed values as well as appropriate strategies to deal with longer horizons. This chapter discussed the role of machine learning in adapting supervised learning techniques to deal with forecasting problems. In particular we stressed the role played by local learning approximators in dealing with important issued in forecasting, like nonlinearity, nonstationarity and error accumulation. Future research should be concerned with the extension of these techniques to some recent directions in business intelligence, like the parallel mining of huge amount of data (big data) [41] and the application to spatio-temporal tasks [30].

Acknowledgments. Gianluca Bontempi acknowledges the support of the ARC project "Discovery of the molecular pathways regulating pancreatic beta cell dysfunction and apoptosis in diabetes using functional genomics and bioinformatics" funded by the Communauté Francaise de Belgique.

References

1. Ahmed, N.K., Atiya, A.F., El Gayar, N., El-Shishiny, H.: An empirical comparison of machine learning models for time series forecasting. Econometric Reviews 29(5-6) (2010)
2. Allen, D.M.: The relationship between variable selection and data agumentation and a method for prediction. Technometrics 16(1), 125–127 (1974)
3. Alpaydin, E.: Introduction to Machine Learning, 2nd edn. Adaptive Computation and Machine Learning. The MIT Press (February 2010)
4. Anderson, T.W.: The statistical analysis of time series. J. Wiley and Sons (1971)
5. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally weighted learning. AIR 11(1-5), 11–73 (1997)
6. Ben Taieb, S., Bontempi, G.: Recursive multi-step time series forecasting by perturbing data. In: Proceedings of IEEE-ICDM 2011(2011)
7. Ben Taieb, S., Bontempi, G., Atiya, A., Sorjamaa, A.: A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. ArXiv e-prints (August 2011)
8. Ben Taieb, S., Bontempi, G., Sorjamaa, A., Lendasse, A.: Long-term prediction of time series by combining direct and mimo strategies. In: Proceedings of the 2009 IEEE International Joint Conference on Neural Networks, Atlanta, U.S.A., pp. 3054–3061 (June 2009)
9. Ben Taieb, S., Sorjamaa, A., Bontempi, G.: Multiple-output modelling for multi-step-ahead forecasting. Neurocomputing 73, 1950–1957 (2010)

10. Ben Taieb, S., Bontempi, G., Sorjamaa, A., Lendasse, A.: Long-term prediction of time series by combining direct and mimo strategies. In: International Joint Conference on Neural Networks (2009)
11. Birattari, M., Bontempi, G., Bersini, H.: Lazy learning meets the recursive least-squares algorithm. In: Kearns, M.S., Solla, S.A., Cohn, D.A. (eds.) NIPS 11, pp. 375–381. MIT Press, Cambridge (1999)
12. Bontempi, G.: Local Learning Techniques for Modeling, Prediction and Control. PhD thesis, IRIDIA- Université Libre de Bruxelles (1999)
13. Bontempi, G.: Long term time series prediction with multi-input multi-output local learning. In: Proceedings of the 2nd European Symposium on Time Series Prediction (TSP), ESTSP 2008, Helsinki, Finland, pp. 145–154 (February 2008)
14. Bontempi, G., Birattari, M., Bersini, H.: Lazy learners at work: the lazy learning toolbox. In: Proceeding of the 7th European Congress on Intelligent Techniques and Soft Computing, EUFIT 1999 (1999)
15. Bontempi, G., Birattari, M., Bersini, H.: Local learning for iterated time-series prediction. In: Bratko, I., Dzeroski, S. (eds.) Machine Learning: Proceedings of the Sixteenth International Conference, pp. 32–38. Morgan Kaufmann Publishers, San Francisco (1999)
16. Bontempi, G., Ben Taieb, S.: Conditionally dependent strategies for multiple-step-ahead prediction in local learning. International Journal of Forecasting (2011) (in press, corrected proof)
17. Casdagli, M., Eubank, S., Farmer, J.D., Gibson, J.: State space reconstruction in the presence of noise. PHYD 51, 52–98 (1991)
18. Cheng, H., Tan, P.-N., Gao, J., Scripps, J.: Multistep-Ahead Time Series Prediction. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) PAKDD 2006. LNCS (LNAI), vol. 3918, pp. 765–774. Springer, Heidelberg (2006)
19. Crone, S.F.: NN3 Forecasting Competition, `http://www.neural-forecasting-competition.com/NN3/index.html` (last update May 26, 2009) (visited on July 05, 2010)
20. Crone, S.F.: NN5 Forecasting Competition, `http://www.neural-forecasting-competition.com/NN5/index.html` (last update May 27, 2009) (visited on July 05, 2010)
21. Crone, S.F.: Mining the past to determine the future: Comments. International Journal of Forecasting 5(3), 456–460 (2009); Special Section: Time Series Monitoring
22. Engle, R.F.: Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. Econometrica 50(4), 987–1007 (1982)
23. Farmer, J.D., Sidorowich, J.J.: Predicting chaotic time series. Physical Review Letters 8(59), 845–848 (1987)
24. Farmer, J.D., Sidorowich, J.J.: Exploiting chaos to predict the future and reduce noise. Technical report, Los Alamos National Laboratory (1988)
25. De Gooijer, J.G., Hyndman, R.J.: 25 years of time series forecasting. International Journal of Forecasting 22(3), 443–473 (2006)
26. De Gooijer, J.G., Kumar, K.: Some recent developments in non-linear time series modelling, testing, and forecasting. International Journal of Forecasting 8(2), 135–156 (1992)
27. Guo, M., Bai, Z., An, H.Z.: Multi-step prediction for nonlinear autoregressive models based on empirical distributions. In: Statistica Sinica, pp. 559–570 (1999)
28. Hand, D.: Mining the past to determine the future: Problems and possibilities. International Journal of Forecasting (October 2008)

29. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference and prediction, 2nd edn. Springer (2009)
30. Hsu, W., Lee, M.L., Wang, J.: Temporal and spatio-temporal data mining. IGI Pub. (2008)
31. Ikeguchi, T., Aihara, K.: Prediction of chaotic time series with noise. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E78-A(10) (1995)
32. Kline, D.M.: Methods for multi-step time series forecasting with neural networks. In: Peter Zhang, G. (ed.) Neural Networks in Business Forecasting, pp. 226–250. Information Science Publishing (2004)
33. Lapedes, A., Farber, R.: Nonlinear signal processing using neural networks: prediction and system modelling. Technical Report LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, NM (1987)
34. Lendasse, A. (ed.): ESTSP 2007: Proceedings (2007)
35. Lendasse, A. (ed.): ESTSP 2008: Proceedings. Multiprint Oy/Otamedia (2008) ISBN: 978-951-22-9544-9
36. Lorenz, E.N.: Atmospheric predictability as revealed by naturally occurring analogues. Journal of the Atmospheric Sciences 26, 636–646 (1969)
37. Matías, J.M.: Multi-output Nonparametric Regression. In: Bento, C., Cardoso, A., Dias, G. (eds.) EPIA 2005. LNCS (LNAI), vol. 3808, pp. 288–292. Springer, Heidelberg (2005)
38. McNames, J.: A nearest trajectory strategy for time series prediction. In: Proceedings of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling, pp. 112–128. K.U. Leuven, Belgium (1998)
39. Micchelli, C.A., Pontil, M.A.: On learning vector-valued functions. Neural Comput. 17(1), 177–204 (2005)
40. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
41. Owen, S.: Mahout in action. Manning (2012)
42. Packard, N.H., Crutchfeld, J.P., Farmer, J.D., Shaw, R.S.: Geometry from a time series. Physical Review Letters 45(9), 712–716 (1980)
43. Palit, A.K., Popovic, D.: Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications. Advances in Industrial Control. Springer-Verlag New York, Inc., Secaucus (2005)
44. Poskitt, D.S., Tremayne, A.R.: The selection and use of linear and bilinear time series models. International Journal of Forecasting 2(1), 101–114 (1986)
45. Price, S.: Mining the past to determine the future: Comments. International Journal of Forecasting 25(3), 452–455 (2009)
46. Priestley, M.B.: Non-linear and Non-stationary time series analysis. Academic Press (1988)
47. Saad, E., Prokhorov, D., Wunsch, D.: Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. IEEE Transactions on Neural Networks 9(6), 1456–1470 (1998)
48. Schuster, H.G.: Deterministic Chaos: An Introduction. Weinheim Physik (1988)
49. Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., Lendasse, A.: Methodology for long-term prediction of time series. Neurocomputing 70(16-18), 2861–2869 (2007)
50. Sorjamaa, A., Lendasse, A.: Time series prediction using dirrec strategy. In: Verleysen, M. (ed.) European Symposium on Artificial Neural Networks, ESANN 2006, Bruges, Belgium, April 26-28, pp. 143–148 (2006)
51. Sorjamaa, A., Lendasse, A., Verleysen, M.: Pruned lazy learning models for time series prediction. In: European Symposium on Artificial Neural Networks, ESANN 2005, pp. 509–514 (2005)

52. Takens, F.: Detecting strange attractors in fluid turbulence. In: Dynamical Systems and Turbulence. Springer, Berlin (1981)
53. Tiao, G.C., Tsay, R.S.: Some advances in non-linear and adaptive modelling in time-series. Journal of Forecasting 13(2), 109–131 (1994)
54. Tong, H.: Threshold models in Nonlinear Time Series Analysis. Springer, Berlin (1983)
55. Tong, H.: Non-linear Time Series: A Dynamical System Approach. Oxford University Press (1990)
56. Tong, H., Lim, K.S.: Thresold autoregression, limit cycles and cyclical data. JRSS_B 42, 245–292 (1980)
57. Tran, T.V., Yang, B.-S., Tan, A.C.C.: Multi-step ahead direct prediction for the machine condition prognosis using regression trees and neuro-fuzzy systems. Expert Syst. Appl. 36(5), 9378–9387 (2009)
58. Weigend, A.S., Gershenfeld, N.A.: Time Series Prediction: forecasting the future and understanding the past. Addison Wesley, Harlow (1994)
59. Werbos, P.J.: Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD thesis, Harvard University, Cambridge, MA (1974)
60. Werbos, P.J.: Generalization of backpropagation with application to a recurrent gas market model. Neural Networks 1(4), 339–356 (1988)
61. Zhang, G., Eddy Patuwo, B., Hu, M.Y.: Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting 14(1), 35–62 (1998)