Forecasting the NN5 Time Series

Jörg D. Wichard, *Member, IEEE,*

***Abstract*— We propose a simple way of predicting time series with reoccurring seasonal periods. We combine several forecasting methods by taking the samplewise weighted mean of those forecasts that were generated with models showing low prediction errors on left-out parts of the time-series. We show the application of this approach to the NN5 Time Series Competition data set.**

I. Introduction

Time series forecasting is a growing field of interest with applications in nearly any field of science. Today, there is a huge interest to build models that can give indication of trends and changes in economy or climatology.

In the NN5 Time Series Competition [1] there is the day of the year as additional information, reflecting the underlying yearly calendar with reoccurring seasonal periods like the changing date of Easter and special events like holidays. In order to forecast the periodic parts of the time series, nearest neighbor models could be used, that were developed to forecast time series from nonlinear systems [2], [3] where conventional linear methods fail. Since the discovery of deterministic chaos, many methods for nonlinear time series modeling and prediction have been suggested and refined (see Kantz et al. for and overview [4]). In particular the nearest neighbor models and nearest trajectory models are of special interest in the case of time series with a latent periodicity [5]. A common characteristic of these models is the reconstruction of the system's state space based on the embedding theorems given by Takens [6], Sauer et al. [7] and Stark [8]. From an equally sampled time series $\{x_t\}_{t=1,\ldots,N}$ we can construct the d-dimensional state space vector $\vec{x}_n = (x_{(n-\lambda(d-1))}, x_{(n-\lambda(d-2))}, \ldots, x_n)$ where λ denotes the time lag. We consider a model $f(\vec{x})$ for time series prediction of the form

$$\begin{aligned} f : \mathbf{R}^d &\rightarrow \mathbf{R} \\ f(\vec{x}_n) &= x_{n+\tau} \end{aligned} \quad (1)$$

where τ is the time horizon of the prediction. In the case $\tau = 1$ this is called the *one step ahead prediction* and it is the common choice in the case of iterated predictions, wherein the predicted value x_{n+1} is used to construct the next state space vector $\vec{x}_{n+1}$ which is used to predict the next time series sample x_{n+2} and so on.

The difficulties of long term prediction arise from the observation, that the uncertainty increases with the horizon of prediction. In the case of seasonal data, we can make use of our knowledge concerning the period of the system under investigation. In the following sections, we like to introduce a simple forecasting strategy based on the a simple weighted mean of those forecasts that were generated by models showing low prediction errors on left-out parts of the time-series.

J. Wichard is with the FMP Berlin, Molecular Modelling Group, Robert-Roessle-Str. 10, 13125 Berlin, Germany, (email: wichard@fmp-berlin.de)

II. Data Preprocessing

The data was taken from the website of the NN5 Competition [1]. The task was to forecast a set of 111 daily time series of cash money withdrawals from cash-machines at different locations in England. The time series consists of 2 years of daily cash money demand and the forecasting horizon was 56 days. The time series show two types of gabs or singularities: Observations of zero withdrawals, which indicate that no money was withdrawn at that day and missing observations without any values, which indicate days in which no value was recorded. We decided to close these gaps with a simple but robust method. Let $\{x_t\}_{t=1,\ldots,N}$ denote the time series and x_n denote the gap sample. In order to substitute the gap sample we take the mean over the following values (in case they exist):

- The one year shift: Either x_{n+364} or x_{n+364}
- The double sided one week shift: x_{n+6} and x_{n-6}

III. Forecasting Models

We used several models and combined them in a weighted average, wherein the model weights were defined by the inverse forcasting errors. The error measure was the Symmetric Mean Absolute Percent Error (SMAPE) across an contiguous 56 day out-of-training set of the time series (in general the last 56 days of the data).

A. The weekly cycle

All time series show a strong weekly cycle. For the further process we define the weekly cycle $c_7(t)$ as the mean over all weeks.

B. The one year shift

Almost all time series show a one year periodicity. Of particular interest are the movable feasts like Easter holidays. For the time span of the prediction, the Easter sunday is 12-April-1998. The time series include two Easter dates: 7-April-1996 and 30-March-1997. We decided to use the 56 days around these two Easter events and combined it with the time series from the weekly cycle in order to handle the singularities. Let $e_1(t)$ and $e_2(t)$ denote the time series around the two Easter events and $c_7(t)$ the weekly cycle.The shifted time series $s(t)$ to cover the Easter date is then

$$s(t) = median\{e_1(t), e_2(t), c_7(t)\}$$

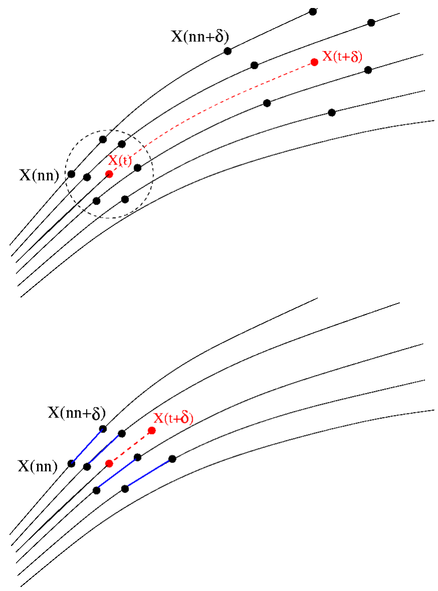

Fig. 1. A nearest neighbor model (on the top) takes the mean over neighboring states to make a forecast. A nearest trajectory model (on the bottom) takes the average over the neighboring trajectories.

C. *Nearest Trajectory Models*

The nearest trajectory model is based on a strategy for time series prediction introduced by McNames [5]. It is based on the assumption that the time series stems from a dynamical system and the states can be reconstructed with a time delay embedding, which is possible for a large class of systems [6]–[8]. In our case the strong weekly periodicity is a hint, that a nearest trajectory model could lead to proper forecasting results. The nearest trajectory model looks for the nearest trajectory segments in the reconstructed state space instead of the nearest neighbors. The prediction is done with a local linear model of the closest trajectory points as described in [5]. The number of neighboring trajectories is a free parameter. We decided to use up to three neighboring trajectories, the embedding dimension was $d = 70$ time lag $\lambda = 1$ and prediction horizon $\tau = 1$. Figure 1 shows the principle of this forecasting method.

D. *Neural Networks*

We trained multilayer feed-forward neural networks with the $\tanh(\vec{x})$ as nonlinear element. The number of hidden layers is chosen at random to be one or two and the numbers of neurons in also random (4-32 in the first layer and 3-9 in the second layer). The training is based on the Rprop Algorithm [9], which is fast and robust. As regularization method we use the common weight decay with the penalty term

$$P(\vec{w}) = \lambda \sum_{i=1}^{N} \frac{w_i^2}{1 + w_i^2} \, , \tag{2}$$

where $\vec{w}$ denotes the N-dimensional weight vector of the MLP and the regularization parameter is small $\lambda = 0.001$.
The neural networks are trained to perform a one step ahead prediction model which is the base model for an iterated prediction. The embedding dimension was $d = 42$ time lag $\lambda = 1$.

IV. Combining Forecasts

In order to generate the final forecast, we propose the following strategy:

1) Leave out a fraction $L = 56$ of samples from the end of the time series and keep those as a test set.
2) Compute forecasts for a range of different parameter settings $\alpha_j, j = 1, \ldots, J$ for the neural network and the nearest trajectory model. The model for the one year shift has no variable parameters, so the computation is straight forward
3) Compute the forecasting error $E_L(\alpha_j)$ on the left-out test set for each parameter setting α_j of the neural network, the nearest trajectory model and the forecasting error of the one year shift.
4) Select the best neural network and the best nearest trajectory model according to the forecasting error $E_L(\alpha_j)$
5) Compute model weights $w(\alpha_j) = 1/E(\alpha_j)$. Weights are thus chosen inversely proportional to forecasting error for the best α_j.
6) Forecast the full time series and build the combined forecast $\hat{y}_{T+1}, \ldots, \hat{y}_{T+M}$ by taking the samplewise mean over all three models (neural network, nearest trajectory and one year shift)

$$\hat{y}_{T+m} = w(\alpha_j) y_{T+m}(\alpha_j). \tag{3}$$

The choice of the weighted average increases the robustness of the approach. Combining the output of several forecasts has the additional advantage of easily deriving lower and upper bounds by computing the samplewise lower and upper quartile of the forecasts.

Acknowledgment

The authors would like to thank the members of the Molecular Modelling Group at FMP Berlin for fruitful discussions.

References

[1] S. Crone, 2008 Time series forecasting competition for computational intelligence, http://www.neural-forecasting-competition.com (2008).
[2] J. Farmer, J. Sidorowich, Predicting chaotic time series, Phys. Rev. Lett. 59(8) (1987) 845 – 848.
[3] M. Casdagli, Nonlinear prediction of chaotic time series, Physica D 35 (1989) 335–356.
[4] H. Kantz, T. Schreiber, Nonlinear time series analysis, Cambridge University Press, Cambridge UK, 1997.
[5] J. McNames, A nearest trajectory strategy for time series prediction, in: Proceedings of the International Workshop on Advanced Black-Box Techniques for N onlinear Modeling, K. U. Leuven Belgium, 1998.
[6] F. Takens, Detecting strange attractors in turbulence, in: Dynamical Systems and Turbulence, Lect. Notes Math., Sringer-Verlag, Berlin, 1981.
[7] T. Sauer, J. Yorke, M. Casdagli, Embedology, J.Stat.Phys. 65 (1991) 579–618.
[8] J. Stark, D. Broomhead, M. Davies, J. Huke, Takens embedding theorems for forced and stochastic systems, Nonlinear Analysis 30 (1997) 5303–5314.
[9] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: The RPROP algorithm, in: Proc. of the IEEE Intl. Conf. on Neural Networks, San Francisco, CA, 1993, pp. 586–591.