Chaotic Feature Selection and Reconstruction in Time Series Prediction

Shamina Hussein[1] and Rohitash Chandra[2(✉)]

[1] School of Mathematical and Computing Sciences,
Fiji National University, Suva, Fiji
shamina.hussein@gmail.com

[2] Artificial Intelligence and Cybernetics Research Group Software Foundation,
Nausori, Fiji
c.rohitash@gmail.com
http://aicrg.softwarefoundationfiji.org

Abstract. The challenge in feature selection for time series lies in achieving similar prediction performance when compared with the original dataset. The method has to ensure that important information has not been lost by with feature selection for data reduction. We present a chaotic feature selection and reconstruction method based on statistical analysis for time series prediction. The method can also be viewed as a way for reduction of data through selection of most relevant features with the hope of reducing training time for learning algorithms. We employ cooperative neuro-evolution as a machine learning tool to evaluate the performance of the proposed method. The results show that our method gives a data reduction of up to 42 % with a similar performance when compared to the literature.

1 Introduction

Time series prediction can be cumbersome for big data related problems. The volume of data and the associated computational complexity can yield higher prediction inaccuracies due to learning irrelevant information [1]. The volume of data and the associated computational complexity can make the application very challenging and therefore, robust feature selection methods are important for removing redundant features [2].

Feature selection methods identify features that are most relevant for a time series in order to achieve faster training performance and with the hope of improving prediction performance as noisy features can be eliminated [1]. The major categories of feature selection methods include the wrapper [3], filter [4] and embedded [4] methods. In a wrapper method, the selection criterion is dependent on the learning algorithm as a part of the fitness function. The selection criterion of filtering methods are independent of the learning algorithm and the selection of feature relies on the relevance score of the feature. The embedded method is specific to a learning algorithm and searches for an optimal subset of features by estimating changes in the objective function value incurred by

A. Hirose et al. (Eds.): ICONIP 2016, Part III, LNCS 9949, pp. 3–11, 2016.
DOI: 10.1007/978-3-319-46675-0_1

making moves in the variable subset. Although wrapper and filter methods are the most commonly used methods for feature selection, their drawbacks initiate a need for a simpler and less expensive method. Wrapper methods have reported superior performance but it is computationally expensive when compared to filter methods [5]. A drawback of filters is that they cannot scale for high dimensional data [5]. Some commonly used feature extraction techniques include statistical methods that feature mean and standard deviation [6], frequency count summations, KarhunenLoeve transformations, Fourier transformations and wavelet transformations [7].

Recently, big data related problems have gained much attention that highlighted further challenges in learning algorithms as they have to deal with enormous amounts of data. However, there has not been much focus on time series problems. The increase in the implementation of technologies such as internet-of-things [8] would result in enhanced data collection and time series analysis will become more difficult as it would need to deal with big data challenges. Hence, we aim to present an approach to address this for upcoming challenges in big data related time series problems [9].

We present a chaotic feature selection and reconstruction method for time series prediction with the hope to reduce the size of the original time series while retaining important information. We employ cooperative neuro-evolution to evaluate the performance of the proposed method. In principle, any machine learning method could be used, however, we selected cooperative neuro-evolution due to its promising performance for time series prediction in previous work [10].

The paper is organized as follows. Section 2 presents the proposed method and Sect. 3 presents the experiments with results and discussion. Section 4 concludes the paper with insights for future work.

2 Chaotic Feature Selection and Reconstruction

We present the details of chaotic feature selection and reconstruction (CFSR) method for chaotic time series. It essentially eliminates the smooth regions of the time series and selects the noisy and chaotic regions. We first divide the time series into subsets known as feature windows and employ simple statistical evaluations to determine if the feature window contains smooth or noisy data points. Note that statistical measurements such as the mean and the standard deviation have been used in the past in feature extraction methods [6]. In our case, they are used to identify the chaotic and noisy regions in the feature window.

In Algorithm 1, the *length* of the *feature window* define the subsets in the time series. The feature window length must be determined experimentally to find the optimal value for best prediction performance. In Step 1, feature window is used to partition the entire time series (Step 2). For each feature window until the entire time series has been considered, the *upper boundary* (Eq. 3) and *lower boundary* (Eq. 4) is defined using the *standard deviation* (Eq. 1) and the *mean* (Eq. 2). The values which falls between the boundaries are selected as the features for reconstruction.

Algorithm 1. Chaotic Feature Selection and Reconstruction

```
Step 1: Define chaotic feature window length
Step 2: Partition the time series into $n$ feature windows according to its length
foreach feature window do
    i. Calculate the Mean (Eq. 2)
    ii. Calculate the Standard Deviation (Eq. 1)
    iii. Identify the Upper-boundary (Eq. 3)
    iv. Identify the Lower-boundary (Eq. 4)
    v. Select features that fall within Upper and Lower boundaries
end
```

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2} \qquad (1)$$

$$\mu = \frac{1}{n}\sum_{k=1}^{n} x_k \qquad (2)$$

$$B_u = \mu + \sigma \qquad (3)$$

$$B_l = \mu - \sigma \qquad (4)$$

We then apply Takens' embedding theorem [11] to the selected chaotic features in order to reconstruct the dataset for a one-step prediction. Given an observed time series $x(t)$, an embedded phase space $Y(t) = [x(t), x(t - T), ..., x(t(D - 1)T)]$ can be generated. T is the time delay, D is the embedding dimension and N is the actual length of the observed time series [11]. This resulting dataset is then used as the input vector for training the model, which in our case is the feedforward neural network.

2.1 Cooperative Neuro-Evolution

Cooperative coevolution(CC) that was initially proposed for function optimization [12], has gained success in neuro-evolution for time series prediction [10]. CC decomposes a problem into subcomponents that are implemented as subpopulations. Much work has been done in the past that focus on problem decomposition that are based on architectural properties of the network [13].

We employ cooperative neuro-evolution (CNE) to demonstrate the effectiveness of proposed feature selection method and it has shown promising results in chaotic time series prediction [10]. CNE used for training feedforward neural networks is given in Algorithm 2. It employs neuron-level decomposition for decomposing the neural network into k subcomponents [13]. The number of subcomponents k is determined by the total number of hidden and output neurons.

In the initialization stage, each sub-population is assigned random numbers in a range and evaluated cooperatively. This is implemented by concatenating the current individual that needs to be evaluated with the fittest individual from the rest of the sub-populations. The concatenated individual is then encoded into the neural network which returns the fitness defined by the root-mean-squared-error.

The main part of the algorithm begins by evolving each of the sub-populations in a *round-robin* fashion for a certain number of generations called the *depth of search.* Any evolutionary algorithm can be chosen for evolution of the sub-populations that feature operations such as crossover, selection and mutation. However, fitness evaluation of each individual of a sub-population is evaluated cooperatively as implemented in the initialisation stage. The procedure is repeated until the termination condition has been reached which is defined by the maximum number of fitness evaluations or a fitness value.

Algorithm 2. Cooperative Coevolution

```
Step 1:Employ neuron level decomposition and attain $k$ subcomponents
Step 2:Initialize and cooperatively evaluate each subcomponent implemented as a sub-population
foreach until termination do
    foreach each Sub-population do
        foreach $n$ Generations do
            Select and create new off-springs
            Cooperatively evaluate the new off-springs
            Update sub-population
        end
    end
end
```

3 Experiments and Results

This section presents the experimental evaluation of the proposed chaotic feature selection and reconstruction (CFSR) method for time series problems. We use cooperative neuro-evolution (CNE) as the designated learning algorithm for feedforward neural network (FNN).

3.1 Problem Description

The benchmark time series data employed are Mackey-Glass times series [14] and Lorenz time series [15], the two simulated time series while the real-world time series are the Sunspot time series [16], Laser time series [17] and Astrophysics time series [18]. Takens' embedding theorem [11] is applied to the selected features to reconstruct the data set. The values for the embedding dimension (D) and the time delay (T) has been set as follows. $D = 5$ and $T = 3$ for the Astrophysics and Sunspot time series. $D = 3$ and $T = 2$ for Lorenz and Mackey Glass time series. $D = 7$ and $T = 2$ for Laser time series.

These reconstructed vectors are then used to train the feedforward neural network. The prediction performance of the feedforward neural network is measured using the root mean squared error(RMSE) (Eq. 5) and the normalized mean squared error(NMSE) (Eq. 6)

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \quad (5) \qquad NMSE = \left(\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}\right) \quad (6)$$

where $y_i, \hat{y}_i$ and $\bar{y}_i$ are the observed data, predicted data and average of observed data, respectively. N is the length of the observed data. These results are also compared with related methods from the literature.

3.2 Experimental Design

We use a feedforward neural network with the sigmoid units in the hidden layer. In the output layer, sigmoid unit is employed for the Mackey Glass, Sunspot and Laser time series while a hyperbolic tangent unit is employed for the Lorenz and Astrophysics time series. A set of 50 independent experimental runs are executed for 3, 5 and 7 hidden neurons. Each sub-population in CNE is evolved a fixed number of generations in a round-robin fashion. This depth of search was set to 1 generation as it has shown to be suitable for neuro-evolution [10]. The G3-PCX algorithm was used to evolve all the sub-populations of CNE. A population size of 300 is used. We used 15000 as the maximum number of function evaluations for the termination condition for all the problems.

3.3 Results and Discussion

The results of 50 experimental runs with 95 % confidence interval for different number of hidden neurons are given in Table 1. We evaluate the results by comparing the different feature windows with the number of hidden neurons (H). The lowest values for the RMSE indicates the best performance.

In the Sunspot problem, the best performance was given by 5 hidden neurons on feature window size of 100. In this case, the proposed method reduced the original dataset by 42 % which has been the greatest reduction when compared to other feature windows, while achieving the best performance. The Laser and the Astrophysics problems achieved the best generalization performance. This was through the dataset generated on feature window of 50, with 7 and 5 hidden neurons, respectively. Hence, the proposed method reduced the original data set by 25 % for Laser problem, and 34 % for the Astrophysics problem.

The proposed method has been able to cope up with noise in the real world problems such as Sunspot, Astrophysics, and Laser. It can be observed from the results that large data sets get reduced greatly and also yields very comparable results. However, there is not a large reduction for smaller datasets of size 500. It is also observed that for the simulated time series, the best generalisation performance is consistently displayed by the same feature window that gives the best results. As for the real world time series, the results were not as consistent when we consider the generalization performance which could have been a result of the presence of noise.

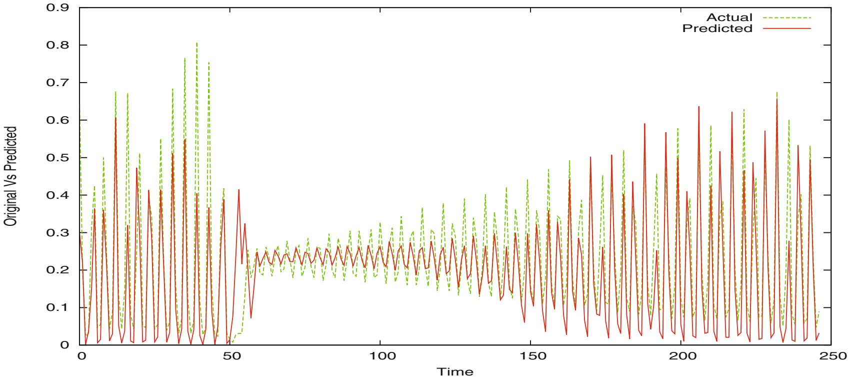

Fig. 1. Typical prediction performance of the proposed method on the test data set of Laser times series

In the Mackey-Glass and Lorenz problems, the best generalization performance was given for the reduced dataset achieved on the feature window of size 10. The proposed method reduced the Mackey-Glass data set by 35 % and the Lorenz dataset by 37 %. Figure 1 shows a typical prediction performance using the proposed data reduction method for the Laser times series on the test set.

Table 2 shows that the proposed method has been successful in reducing the training time for the featured training method when compared to the original dataset. It can be seen that the training time taken has been greatly reduced by the proposed method with the reduction in size of the original training dataset. The maximum reduction in time is of 68.39 % for the Sunspot time series data while the Astrophysics problem achieved a 61.69 % reduction, followed by Laser, Lorenz and Mackey Glass problems.

Table 1. Training and generalization performance (RMSE)

Problem	Feature window	H	Training	Generalization	Best
Sunspot	20	5	0.0243 ± 0.0005	0.0263 ± 0.0025	0.0141
	100	7	0.0108 ± 0.0005	0.0439 ± 0.0067	0.0152
Laser	50	7	0.0501 ± 0.0012	0.1450 ± 0.0068	0.0972
	100	7	0.0513 ± 0.0014	0.1555 ± 0.0043	0.1164
Astrophysics	50	5	0.0498 ± 0.0012	0.0656 ± 0.0032	0.0507
	100	5	0.0459 ± 0.0012	0.0665 ± 0.0038	0.0530
Mackey-Glass	10	7	0.0251 ± 0.0001	0.0087 ± 0.0002	0.0071
	100	5	0.0077 ± 0.0002	0.0130 ± 0.0012	0.0075
Lorenz	10	5	0.0462 ± 0.0059	0.0388 ± 0.0074	0.0103
	100	5	0.0405 ± 0.0070	0.1047 ± 0.0144	0.0133

Table 2. Training time and data reduction

Time series	Feature window	Reduced data set			Original data set			*T-Test*
		H	Best	Time (sec)	H	Best	Time (sec)	p value
Sunspot	100	7	0.0117	370.11	5	0.0110	1170.87	2.86E-06
Mackey Glass	10	7	0.0071	561.48	5	0.0031	746.78	3.28E-17
Laser	50	3	0.0972	387.99	5	0.0321	760.23	1.34E-42
Lorenz	10	5	0.0103	298.12	5	0.0037	575.99	5.70E-03
Astrophysics	50	5	0.0507	788.46	5	0.0507	2058.11	4.99E-06

Table 3. Comparison with the literature

Time series	Prediction method	RMSE	NMSE
Sunspot	Wavelet packet multilayer perceptron [22]		1.25E-01
	Co-evolutionary recurrent neural networks [10]	2.60E-02	3.62E-03
	Competitive co-evolutionary recurrent neural networks [23]	1.57E-02	1.31E-03
	Proposed CFSR	1.17E-02	8.15E-04
Lorenz	Recurrent neural networks [24]		1.85E-03
	Back-propagation and genetic algorithm with residual analysis [20]	2.96E-02	
	Co-evolutionary recurrent neural networks [10]	8.20E-03	1.28E-03
	Competitive co-evolutionary recurrent neural networks [23]	3.55E-03	2.41E-04
	Proposed CFSR	1.03E-02	5.58E-04
Mackey	Neural fuzzy network and meta-heuristics [19]	8.45E-03	
	Back-propagation and genetic algorithm with residual analysis [20]	1.30E-03	
	Co-evolutionary recurrent neural networks [10]	8.28E-03	4.77E-04
	Competitive co-evolutionary recurrent neural networks [23]	3.99E-03	1.11E-04
	Proposed CFSR	7.05E-03	2.93E-04
Laser	Multilayer Perceptron(MLP) [21]		1.72E-01
	Elman [21]		3.40E-02
	Non-linear Autoregressive model process with exogenous input (NARX) [21]		3.39E-02
	Proposed CFSR	9.72E-02	1.18E-01

Table 3 provides a comparison between the best results from Table 1 with related methods from literature. The RMSE and the NMSE for the best results is used for comparison. We note that the Astrophysics problem has not been used in literature. The proposed method has given better results when compared to related methods in literature such as evolutionary algorithms for training neural fuzzy networks [19] and co-evolutionary recurrent neural networks [10] for Mackey-Glass and Sunspot time series.

The proposed method performs better than back-propagation and genetic algorithm with residual analysis [20] for Lorenz time series. It also performs better than and multilayer-perceptron [21] for Laser time series. The reduced and reconstructed data set is able to eliminate irrelevant data, hence, reducing the prediction error and improving the overall efficiency of the neural network. The results also indicate that larger datasets are more favourable for the proposed method as seen with the real world problems that include Sunspot, Laser and Astrophysics time series.

In the literature, the prediction methods used the entire dataset without any feature selection. The goal for this paper was to achieve similar level of prediction performance with reduced dataset that is computationally less expensive for training. However, in some cases, the proposed method has achieved better prediction performance. This indicates that feature selection has been able to help further in generalisation performance of neural networks.

4 Conclusions and Future Work

We presented a chaotic feature selection and reconstruction method based on statistical analysis for time series prediction. It essentially implements data reduction by capturing most relevant features that are either noisy or chaotic in nature. The results show that the proposed method has been able to retain the prediction performance with a smaller dataset while reducing the training time. The results further show that the proposed method performs similar to the selected methods in the literature. Moreover, the proposed method has been able to reduce the size of the original dataset up to 42 % and the prediction time by up to 68 %.

In future work, it would be interesting to evaluate the feature selection method with other machine learning tools. The proposed method can also be extended to multi-variate time series and applied to problems that deal with very large time series datasets that include areas of astronomy and climate change.

References

1. Crone, S.F., Kourentzes, N.: Feature selection for time series prediction-a combined filter and wrapper approach for neural networks. Neurocomputing **73**(10), 1923–1936 (2010)
2. Chen, M., Mao, S., Liu, Y.: Big data: a survey. Mob. Netw. Appl. **19**(2), 171–209 (2014)
3. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artif. Intell. **97**(1), 273–324 (1997)
4. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
5. Yu, L., Liu, H.: Feature selection for high-dimensional data: a fast correlation-based filter solution. In: ICML, vol. 3, pp. 856–863 (2003)
6. Sandya, H.B., Hemanth Kumar, P., Patil, S.B.: Feature extraction, classification and forecasting of time series signal using fuzzy and garch techniques. In: National Conference on Challenges in Research and Technology in the Coming Decades (CRT 2013), IET, pp. 1–7 (2013)
7. Olszewski, R.T.: Generalized feature extraction for structural pattern recognition in time-series data. DTIC Document, Technical report (2001)
8. Cavalcante, E., Pereira, J., Alves, M.P., Maia, P., Moura, R., Batista, T., Delicato, F.C., Pires, P.F.: On the interplay of internet of things and cloud computing: a systematic mapping study. Comput. Commun. **8990**, 17–33 (2016)
9. Chen, C.P., Zhang, C.-Y.: Data-intensive applications, challenges, techniques and technologies: a survey on big data. Inf. Sci. **275**, 314–347 (2014)

10. Chandra, R., Zhang, M.: Cooperative coevolution of Elman recurrent neural networks for chaotic time series prediction. Neurocomputing **186**, 116–123 (2012)
11. Takens, F.: On the Numerical Determination of the Dimension of an Attractor. Springer, Heidelberg (1985)
12. Potter, M.A., Jong, K.A.: A cooperative coevolutionary approach to function optimization. In: Davidor, Y., Schwefel, H.-P., Männer, R. (eds.) PPSN 1994. LNCS, vol. 866, pp. 249–257. Springer, Heidelberg (1994). doi:10.1007/3-540-58484-6_269
13. Chandra, R., Frean, M., Zhang, M.: On the issue of separability for problem decomposition in cooperative neuro-evolution. Neurocomputing **87**, 33–40 (2012)
14. Mackey, M.C., Glass, L.: Oscillation and chaos in physiological control systems. Science **197**, 287–289 (1977)
15. Lorenz, E.: Deterministic non-periodic flows. J. Atmos. Sci. **20**, 267–285 (1963)
16. SILSO World Data Center: The International Sunspot Number (1834-2001), International Sunspot Number Monthly Bulletin and Online Catalogue, Royal Observatory of Belgium, Avenue Circulaire 3, 1180 Brussels, Belgium. http://www.sidc.be/silso/. Accessed 02 Feb 2015
17. Hübner, U., Abraham, N.B., Weiss, C.O.: Dimensions and entropies of chaotic intensity pulsations in a single-mode far-infrared NH3 laser. Phys. Rev. A **40**(11), 6354–6365 (1989)
18. The Santa Fe Time Series Competition Data. http://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html. Accessed 01 May 2015
19. Lin, C.-J., Chen, C.-H., Lin, C.-T.: A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **39**(1), 55–68 (2009)
20. Ardalani-Farsa, M., Zolfaghari, S.: Residual analysis and combination of embedding theorem and artificial intelligence in chaotic time series forecasting. Appl. Artif. Intell. **25**, 45–73 (2011)
21. Menezes, J.M.P., Barreto, G.A.: Long-term time series prediction with the NARX network: an empirical evaluation. Neurocomputing **71**(16), 3335–3343 (2008)
22. Teo, K.K., Wang, L., Lin, Z.: Wavelet packet multi-layer perceptron for chaotic time series prediction: effects of weight initialization. In: Alexandrov, V.N., Dongarra, J., Juliano, B.A., Renner, R.S., Tan, C.J.K. (eds.) ICCS-ComputSci 2001. LNCS, vol. 2074, pp. 310–317. Springer, Heidelberg (2001)
23. Chandra, R.: Competition and collaboration in cooperative coevolution of Elman recurrent neural networks for time-series prediction. IEEE Trans. Neural Netw. Learn. Syst. **26**(12), 3123–3136 (2015)
24. Mirikitani, D., Nikolaev, N.: Recursive bayesian recurrent neural networks for time-series modeling. IEEE Trans. Neural Netw. **21**(2), 262–274 (2010)