ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 419-424

Unsupervised Classifier for Monitoring and Diagnostic of Time Series

Stéphane Lecoeuche

Laboratoire I3D, Bâtiment P2, USTL, 59655 Villeneuve d'Ascq, France,
and EIPC, Campus de la Malassise, 62967 Longuenesse Cedex, France,
Phone: +33 321 388 510, Email: slecoeuche@eipc.fr

Abstract. It is assumed that complex systems are represented by parameters that evolve with time. Hence, it is possible to survey systems and to make diagnostics analyzing time series. The paper presents the development of a neural architecture. A membership degree of an input vector to a prototype is introduced along with the membership degree of the input to a class. The proposed unsupervised learning process makes possible the creation of new prototypes and new classes when necessary. The application to standard time series shows good results: 96% of the inputs are well classified using few prototypes.

Keywords: classification, auto-adaptation, monitoring, diagnostic, unsupervised learning, full covariance

1 Introduction

Industrial systems are most of the time quite difficult to monitor when the number of process data is large. So, failures are difficult to localize without using a model of the real system. To find such a model, many standard methods based on identification techniques could be applied [1], as well as neural based methods that are successfully used for identification of large scale systems [2] and of industrial components [3]. All these methods lead to a reduced set of representative parameters. For example, a neural based online identification technique (e.g. [4] [5]) leads to evolutions of the connection weights. So, the monitoring and the diagnostic of the functioning state of the complex system could be achieved by surveying the model parameters which have evolutions that are similar to time series. Hence, it has been chosen here to consider that the identification phase of a complex system is done, and that the modifications of the system are characterized by the evolutions of time series. So, the study focuses on the monitoring and diagnostic of such time series. Standard series are available on the internet [6] and are used in the present study.
In real cases, the representative parameters of the actual system would deviate, slowly or sharply; would shift, upwards or downwards; would oscillate. As the amplitudes of these evolutions are a priori not known, it is necessary to use dynamic architectures [7] and/or specific learning methodologies [8].

Only few architectures are able to adapt their architecture in accordance with the evolution of the input data, creating new prototypes and/or adapting existent prototypes when necessary, and simultaneously create new classes (new neurons on the output layer) or modify classes shape when necessary; i.e. when the newly created prototype is far from all already known prototypes. Among these architectures, it is possible to mention the Cluster Detection and Labeling Network (CDL) developed in 1998 by Eltoft et al. [9] and the improvements presented by Lurette et al. [10]. Here is presented a new rule for the iterative modification of the center and the covariance matrix of the prototypes that allows the adaptation of the classes with unknown data.

2 Description of the developed Neural Network

The concepts described here are illustrated by a classical three layers network. The input layer consists of as many neurons as components of the input vector called X_i. The hidden layer is totally connected to the input layer, and each neuron represents a prototype P_j of one class. So, the output of each neuron of the hidden layer is defined as the membership degree (eq.1) of a new example X_i to the corresponding prototype P_j.

$$\mu(P_j, X_i) = \exp(-\frac{1}{2\alpha_{P_j}}(d(P_j, X_i)^2)) \tag{1}$$

where

$d(P_j, X_i)^2 = (X_i - M_{P_j})^T \sum_{P_j}^{-1}(X_i - M_{P_j})$, Mahalanobis distance;

M_{P_j}, mean vector of the prototype;

$\sum_{P_j}^{-1}$, full covariance matrix [11] of the prototype;

and $\alpha_{P_j}^{k+1} = \max_{X_i \in P_j}\left(\frac{(X_i - M_{P_j}^{k+1})^T \Sigma_{P_j}^{-1}(X_i - M_{P_j}^{k+1})}{-2\ln(\mu_{\max})}\right)$, normalization factor.

The Mahalanobis distance has been chosen to give the prototypes an Hyper-Elliptical shape (see figure 2), for a better description of their neighborhood.

The output layer consists of as many neurons as detected classes. The output of each neuron of the output layer defines the membership degree (eq.2) of the example X_i to the corresponding class C_k.

$$\Psi(C_k, X_i) = \min(1, \sum_{P_j \in C_k} \mu(P_j, X_i)) \tag{2}$$

The main benefit of this neural network is that its structure is not a priori fixed or fixed after a unique learning phase like many classifiers. A particular learning principle has been developed in order to realize this structure adaptation, not only by adapting the connection weights but also by modifying the size of the hidden layer and the size of the output layer (figure 1).

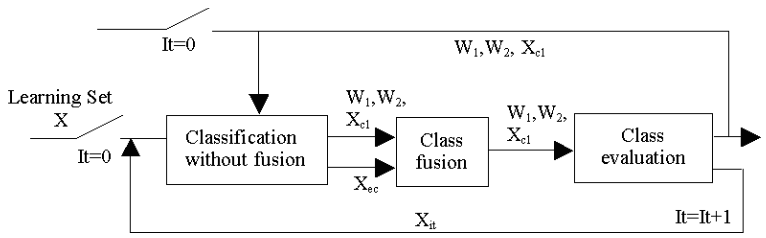
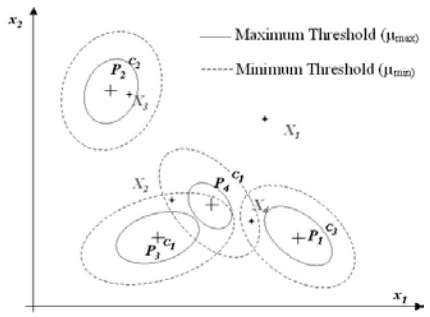


Figure 1: Learning phases.

During the first stage (classification without merging), for each new input vector, different cases could occur. The membership degree of X_i is calculated for each known prototype and compared with two thresholds. The table 1 summarizes the four cases illustrated by the figure 2.



X_1, X_2, X_3, X_4: input vectors
P_1, P_2, P_3, P_4 : known prototypes
C_1, C_2, C_3, : classes
P_k^j : prototype k belongs to Class j

Figure 2: Illustration of the membership of input vectors to prototypes.

Input vector	Membership degree to $P_1^{C_3}$	$P_2^{C_2}$	$P_3^{C_1}$	$P_4^{C_1}$	Then
X_1	ε	ε	ε	ε	New proto. P_5=X_1; new class C_4; $X_1 \in X_{cl}$
X_2	ε	ε	$> \mu_{min}$	$> \mu_{max}$	New proto. $P_6 = X_2$; belongs to C_1; $X_2 \in X_{cl}$
X_3	ε	$> \mu_{max}$	ε	ε	Adaptation of $P_2^{C_2}$; X_3 belongs to C_2; $X_3 \in X_{cl}$
X_4	$> \mu_{min}$	ε	ε	$> \mu_{max}$	New proto. P_7 =X_4; C_2 and C_3 merge; $X_4 \in X_{ec}$

Table 1: Different cases for the "classification without fusion" stage.

When necessary, the prototype P_j is adapted in the same way than in [7], but with the full covariance matrix Σ_{P_j}. This adaptation is performed in an iterative manner using the following relations (eq. 3 and 4).

$$M_{P_j}^{k+1} = \frac{k}{k+1} M_{P_j}^{k} + \frac{1}{k+1} X_i \tag{3}$$

$$\Sigma_{P_j}^{k+1} = \frac{k-1}{k} \Sigma_{P_j}^{k} + \frac{1}{(k+1)} (X_i - M_{P_j}^{k})^T (X_i - M_{P_j}^{k}) \tag{4}$$

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 419-424

This adaptation is illustrated in figure 3.

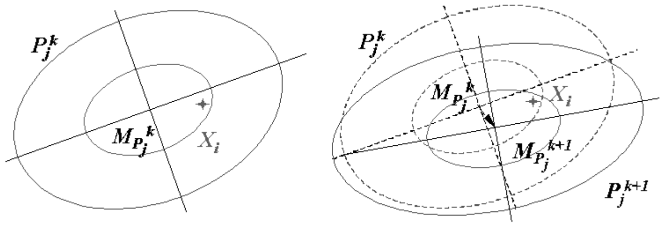


Figure 3: Illustration of the adaptation of the prototype.

3 Application to time series analysis

To validate the proposed development and to assure its abilities to detect and monitor the model of a real system, it has been tested on the "Synthetic Control Chart Time Series" database [6]. This database is composed of six different classes of control charts, described in figure 4 by six examples from each class.

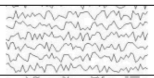
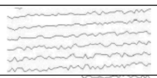
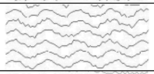
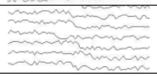
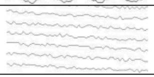
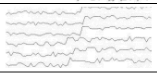
Normal (zoom)		Increasing trend	
Cyclic		Downward shift	
Decreasing trend		Upward shift	

Figure 4: Examples of analyzed times series.

This database has been chosen because each class can be associated to a typical kind of evolution of a real system, and therefore to the drift of its model. Each class is defined by 100 control charts and each one is defined by 60 samples.

3.1 List of discriminators

Before trying to classify time series, it is necessary to find discriminators. It has been chosen here to use:

- the standard deviation,
- the slope of the linear regression,
- the maximum difference between two consecutive coefficients of the local identification with ARX model[12] of the time series.

Although using the mean value could have been an easy way to discriminate the time series, it has been chosen not to take it into account. This corresponds to the fact that a long term evolution of the mean value is normal for some real-world systems.

3.2 Results

Figure 5 shows the result of the whole classification process using the following thresholds: $\mu_{min} = 0.3$, $\mu_{max} = 0.6$ and $\sigma = 0.08$.

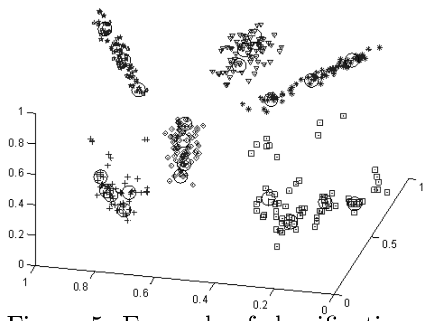


Figure 5: Example of classification

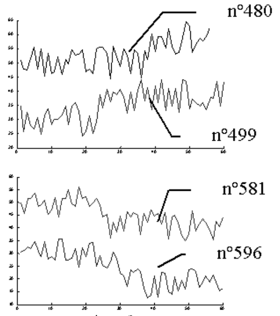


Figure 6: Ambiguous curves

To have an idea of the quality of the identification, it is necessary to give the percentage of well classified input vectors and the number of prototypes. In the presented case, 95.44% of the inputs vectors are in the right class, and 22 prototypes are obtained. All misclassified curves are "Shift" recognized as "Trend"; examples are given in figure 6. The figure 7 shows the benefits of our network in terms of number of prototypes, (17 with 2d projection). Using the "Gaussian RBF" SVM algorithm [13] would have led to more numerous prototypes (31) for good separation of the 6 classes (width of rbf =1.1).

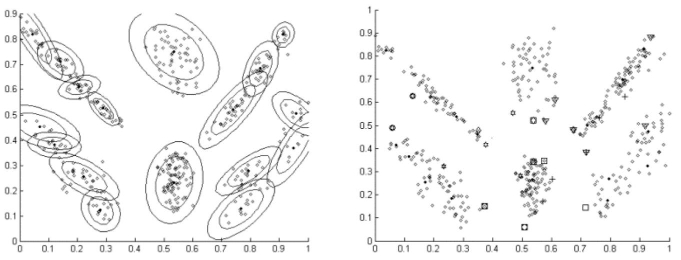


Figure 7: 2D illustrations of the prototypes for our method and SVM algorithm.

4 Conclusions

A new learning process has been presented. This process makes possible the creation of new prototypes and new classes when necessary. It has been shown that the number of prototypes is quite low for an efficient classifier. Its ability to adapt its structure (prototypes and classes) when new kind of data appears allows online classification; So, neither initial nor a priori representation data are needed.

Future studies will address improvements of the architecture such as initialization using the EM algorithm, merging of prototypes, and improvement of the class merging process. Other discriminators will also be taken into account.

References

[1] P. Borne, G. Dauphin-Tanguy, J. Richard, F. Rotella, and I. Zambettakis, *Modélisation et identification des processus, tome 2.* Editions Technip, Paris, 1992.

[2] D. T. Pham and S. J. Oh, "Identification of plant inverse dynamics using neural networks," *Artificial Intelligence in Engineering*, vol. 13, pp. 309–320, 1999.

[3] J. Teeter and M.-Y. Chow, "Application of functional link neural networks to HVAC thermal dynamic system identification," *IEEE Trans. on Industrial Electronics*, vol. 45, pp. 170–176, 1998.

[4] N. Sadegh, "A perceptron network for functional identification and control of nonlinear systems," *IEEE Trans. on Neural Networks*, vol. 4, pp. 982–988, 1993.

[5] M. Nørgaard, O. Ravn, N. K. Poulsen, and L. Hansen, *Neural networks for modeling and control of dynamics systems.* Springer-Verlag, 2000.

[6] D. T. Pham and A. B. Chan, "Control chart pattern recognition using a new type of self organizing neural networks," *Proc. Instn, Mech Engrs*, vol. 212, pp. 115–127, 1998.

[7] N. Zheng, Z. Zhang, G. Shi, and Y. Qiao, "Self-creating and adaptive learning of RBF networks," in *Proc. of the Intern. Joint Confer. on Neural Networks*, July 1999.

[8] A. Trunov and M. Polycarpou, "Automated fault diagnosis in nonlinear multivariable systems using a learning methodology," *IEEE Trans. on Neural Networks*, vol. 11, no. 1, pp. 91–101, 2000.

[9] T. Eltoft and R. deFigueiredo, "A new neural network for Cluster-Detection-and-Labeling," *IEEE Trans. on Neural Networks*, vol. 9, no. 1, pp. 1021–1035, 1998.

[10] C. Lurette and S. Lecoeuche, "Improvement of Cluster Detection and Labeling Neural Network by introducing elliptical basis functions," in *Proc. of the Intern. Confer. ICANN*, August 2001.

[11] M.W.Mak and S.Y.Kung, "Estimation of Elliptical Basis Function Parameters by the EM Algorithm with Application to Speaker Verification," *IEEE Trans. on Neural Networks*, vol. 11, no. 4, pp. 961–969, 2000.

[12] L.Ljung, *System Identification: Theory for the User.* Prentice Hall, 2 ed., 1999.

[13] S.Gunn, "Support Vector Machine for classification and regression," University of Southampton, 1998.