Detecting Changes of State in Heterogeneous Dynamic Processes with Time-dependent Models: A Soft-Computing Approach

Valdés, Julio

Proceedings of the IASTED International Joint Conference on Artificial Intelligence and Soft Computing (ASC'2004), 2004

National Research Council Canada
Institute for Information Technology

Conseil national de recherches Canada
Institut de technologie de l'information

NRC · CNRC

Detecting Changes of State in Heterogeneous Dynamic Processes with Time-dependent Models: A Soft-Computing Approach *

Valdés, J.
September 2004

* published in the Proceedings of the IASTED International Joint Conference on Artificial Intelligence and Soft Computing (ASC'2004). September 1-3, 2004. Marbella, Spain. ACTA Press, Anaheim, USA. pp 155-160. NRC 47389.

Canada

DETECTING CHANGES OF STATE IN HETEROGENEOUS DYNAMIC PROCESSES WITH TIME-DEPENDENT MODELS: A SOFT-COMPUTING APPROACH

Julio J. Valdés
Institute for Information Technology, National Research Council of Canada
1200 Montreal Road, Ottawa, ON K1A 0R6
Canada
julio.valdes@nrc-cnrc.gc.ca

ABSTRACT
This paper introduces a computational intelligence approach to the problem of detecting internal changes within dynamic processes described by heterogeneous, multivariate time series with imprecise data and missing values. A data mining process oriented to model discovery using a combination of neuro-fuzzy neural networks and genetic algorithms, is combined with the estimation of probability distributions and error functions associated with the set of best discovered models. The analysis of this information allows the identification of changes in the internal structure of the process, associated with the alternation of steady and transient states – zones with abnormal behavior – instability and other situations. This approach is rather general, and its potential is revealed by experiments with simulated and real world data.

KEY WORDS
neuro-fuzzy networks, evolutionary algorithms, probability distributions, heterogeneous multivariate time series, model discovery.

1. Introduction

Processing time-dependent heterogeneous information is a continuously growing need, especially in industrial and natural processes, which are usually characterized by variables of many different kinds. These heterogeneous variables are represented by magnitudes corresponding to different measurement scales (nominal, ordinal, interval and ratio), and also by more complex types such as images, signals, written reports, diagrams, etc. Some variables are obtained by measuring or recording instruments, whereas others are purely subjective, determined by human experts. As a consequence, they have different kinds of heterogeneity and uncertainty (imprecision, vagueness) associated with them.
With the developments in sensor, sampling and data gathering technologies, the databases collecting such data are growing at an enormous rate. In addition, the complexity of the physical observations, electronic or mechanical deficiencies, human errors and other factors, make these databases *incomplete*. Therefore, they usually contain large quantities of *missing values*.
In complex or poorly known processes, knowledge discovery oriented to unveil the underlying structure of the physical process is crucial, especially for revealing patterns and time dependencies, detecting abnormal behavior, instabilities, changes of state, deriving prediction criteria, and constructing forecasting procedures.
This paper discusses a hybrid approach based on a combination of *soft-computing* techniques for model discovery and model-change detection in multivariate time processes with different kind of variables, missing data and uncertainty. The models are represented by hybrid neural networks using *heterogeneous neurons,* which accept mixed, fuzzy and missing data, as input. The method finds sets of non-linear models having bounded prediction accuracy over a target signal, and characterizes the overall time dependencies between the heterogeneous time series as probability distributions over their sets of time lags. The main steps are: *i)*a search in the space of dependency models, and *ii)* a study of the probability distributions and their changes in a selected subset. These distributions are combined with the prediction errors associated with the discovered models.

2. Heterogeneous Domains and Multivariate Time Series

A formal approach for describing heterogeneous information in general observational problems was given in [11], and for constructing neuron models in [7],[8]and [1]. Different *information sources* are associated with the attributes, relations and functions, and these sources are associated with the nature of what is observed (e.g. point measurements, signals, documents, images, etc). They are described by mathematical sets of the appropriate kind called *source sets* (Ψ_i), constructed according to the nature of the information source to represent (e.g. point measurements of continuous variables by subsets of the reals in the appropriate ranges, structural information by directed graphs, etc). Source sets also account for incomplete information.

A heterogeneous domain is a Cartesian product of a collection of source sets: $\hat{H} = \Psi_1 \times \cdots \times \Psi_n$, where $n > 0$ is the number of information sources to consider. For example, consider a domain where objects are described by attributes like continuous crisp quantities, discrete features, fuzzy features, time-series, images, and graphs. Individually, they can be represented as Cartesian products of subsets of real numbers($\hat{R}$), nominal ($\hat{M}$) or ordinal sets($\hat{O}$), fuzzy sets($\hat{F}$), set of images ($\hat{I}$) **,** set of time series ($\hat{S}$) and sets of graphs ($\hat{G}$)**,** respectively, all properly extended for accepting missing values. Thus, the heterogeneous, time dependent domain is

$$\hat{H}^n(t) = \hat{N}^{n_N}(t) \times \hat{O}^{n_O}(t) \times \hat{R}^{n_R}(t) \times \hat{F}^{n_F}(t) \times \hat{I}^{n_I}(t) \times \hat{G}^{n_G}(t) \times \hat{S}^{n_S}(t)$$

where n_N is the number of nominal sets, n_O of ordinal sets, n_R of real-valued sets , n_F of fuzzy sets , n_I of image-valued sets, n_S of time-series sets, and n_G of graph-valued sets, respectively. A multivariate, heterogeneous time series is shown in Fig-1.

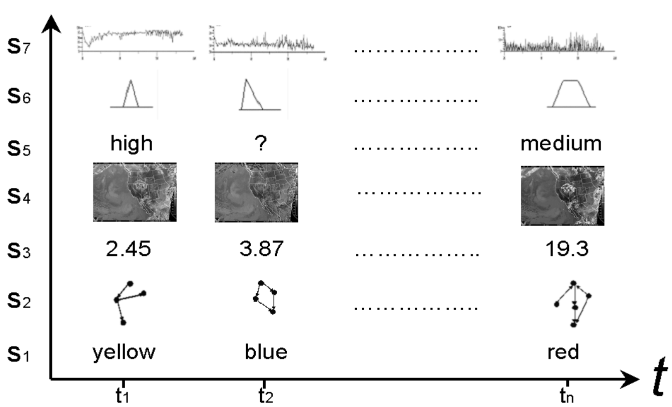

Fig. 1. An example of a heterogeneous, time-dependent multivariate process. Each row is a series of a different type: nominal, graph, ratio, image, ordinal, fuzzy, time-series. Attributes may have missing values (?). The sampling interval is assumed to be the same (and synchronized) for each of the individual series

3. Model Mining with Heterogeneous Neurons and Hybrid Neural Networks

The classical approaches in time series consider mostly univariate, homogeneous (real-valued), time series, without missing values [2],[5],[4]. Conventional multivariate approaches are complex and have difficulties in handling heterogeneity, imprecision and incompleteness. The purpose of model mining in heterogeneous, multivariate, time varying processes is to discover *dependency models*. A model expresses the relationship between values of a previously selected time series (the target), and a subset of the past values of the entire set of series. Different classes of functional models could be considered, in particular, a generalized non-linear auto-regressive (AR) model like the one given by Equation-1.

$$S_{\text{target}}(t) = \mathrm{F}\begin{pmatrix} S_1(t-\tau_{1,1}), S_1(t-\tau_{1,2}), \ldots, S_1(t-\tau_{1,p_1}), \\ S_2(t-\tau_{2,1}), S_2(t-\tau_{2,2}), \ldots, S_2(t-\tau_{2,p_2}), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ S_n(t-\tau_{n,1}), S_n(t-\tau_{n,2}), \ldots, S_n(t-\tau_{n,p_n}) \end{pmatrix} \quad (1)$$

Where n is the number of signals, $S_1, S_2, \ldots, S_n$ is the set of signals, **F**, is the unknown function, $p_1\ p_2, \ldots, p_n$ are the number of lags considered for signal, and τ_{ij} is the *j-th* time lag corresponding to signal *i*. For the same set of heterogeneous time series, a different model can be obtained for each of the series from the set, with different composition.

A hybrid soft-computing algorithm for approaching this kind of problems using genetic algorithms and heterogeneous neural networks has been given elsewhere [9]. This approach requires the simultaneous determination of: *(i)* the number of required lags for each series, *(ii)* the particular lags within each series carrying the dependency information, and *(iii)* the prediction function. A requirement on function **F** is to minimize a suitable prediction error, usually the root mean squared error (RMSE). The procedure is based on: *(a)* exploration of a subset of the model space with a genetic algorithm, and *(b)* use of a similarity-based neuro-fuzzy system representation for the unknown prediction function. As mentioned, statistical or other classical approaches either have difficulties handling these kinds of situations or cannot handle them at all. The size of the model space is immense and grows exponentially as the value of the maximal lag included in the model increases (considering only 10 time series and the first 20 time lags, the search space contains about 10^{60} models). Within this approach, the prediction function **F** is represented by a hybrid neural network with a hidden layer composed by heterogeneous neurons (h-neurons). A heterogeneous neuron is a general mapping $h : \hat{H} \times \hat{H} \rightarrow Y$, where $\hat{H}$ is a heterogeneous domain, and Y is an arbitrary set. If $x, w \in \hat{H}$, and $y \in Y$, then $y = h(x, w)$ A particular class of h-neurons is obtained when *Y* the real interval [0,1] and *h* is given by a composition of a similarity function [3], and an isotone automorphism $g:[0,1] \rightarrow [0,1]$ (usually a non-linear function). In this case the h-neuron is given by $h(x,w) = g \circ s(x,y)$ and called a similarity-based neuron or s-neuron (Fig-2).

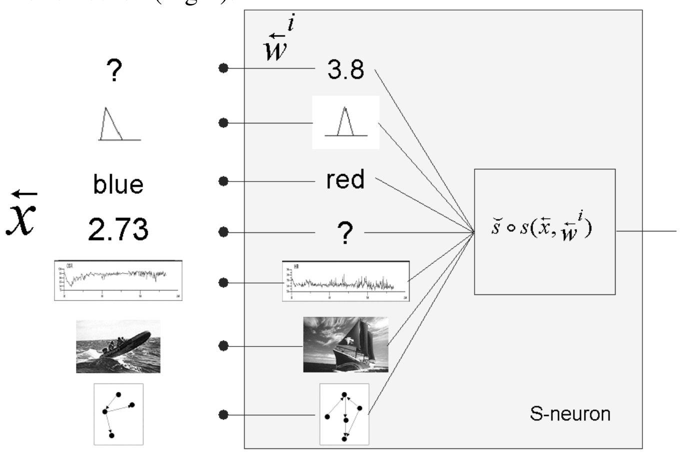

Fig. 2. A similarity-based heterogeneous neuron. Both the input and the neuron weights are objects from a heterogeneous domain (? is a missing value). The output is a similarity value.

This neuron model is flexible (heterogeneous data with missing values are its natural input, without the need of data type transformation or imputation of missing values), and it is robust. Networks using this neuron also have the general function approximation property [1]. The s-neuron can be coupled with classical neurons (aggregation function given by the scalar product, and activation given by the sigmoid or hyperbolic tangent), forming hybrid neural networks. The MVTSMM system [10],[12] is an implementation of this approach to model mining in heterogeneous time series as a parallel computing algorithm (Fig-3).

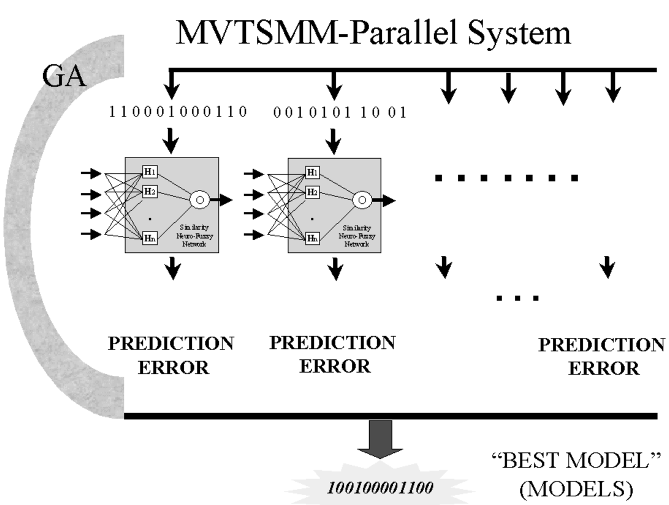

Fig. 3. Multivariate Time Series Model Miner System (MVTSMM). The arc is a parallel genetic algorithm evolving populations of similarity-based hybrid neural networks. The binary strings encode dependency patterns for the target signal. For each, a hybrid network is constructed and trained with a fast algorithm. The network represents the prediction function, and is applied to an independent test set. The best models (those with their RMSError under a preset threshold), are collected.

4.Models as a Function of Time

Consider a time interval $[t_1, t_2]$ centered at time t within the multivariate series. Let $\mathcal{M}$ be the space of all models (of the type given by Eq.1) associated with a set of *n* signals, and $\hat{\mathcal{M}} \subseteq \mathcal{M}$ the subset of models found by the algorithm under the corresponding set of parameters. Let $\hat{\mathcal{M}}_{\varepsilon} \subset \hat{\mathcal{M}}$ (ε is a positive real), be the set of all found models with errors upper-bounded by ε. If τ_{ij} is a time lag term as denoted in (Eq.1), then $P(\tau_{ij} \in \hat{\mathcal{M}}_{\varepsilon})(t)$ is the probability of this lag to be a part of an ε-good model, and in general, there will be a probability distribution $P(\tau \in \hat{\mathcal{M}}_{\varepsilon})(t)$ defined over the time lags τ for a fixed time interval $[t_1, t_2]$. Call it the *lag probability function* (l.p.f.), defined as $\mathcal{L}_{\varepsilon}(\tau, t) = P(\tau \in \hat{\mathcal{M}}_{\varepsilon})(t)$. The set of all l.p.f. for all t is the *lag probability spectrum.*

This distribution will depend on the nature of the process within that time interval. Therefore, if changes in the internal fabric of the process are taking place, like changes of state, abnormal behaviors or transients, the relationships between the variables will change as well, and so will their dependency patterns. On the other hand, if the set of parameters controlling the algorithm are kept constant within all time intervals considered, the ability of the algorithm to find good models will be an indication of internal changes within the process. The set of parameters determine both the search space as well as the search strategy. Therefore, finding good models in some time frames while failing to do so in others, indicates that *i)* the good models are out of the searchable subpace for those frames, *ii)* a longer search might be required, or *iii)* other problems exists. These effects will, undoubtedly, increase the prediction errors associated with the models found. If a process is internally stable, it should have a constant l.p.f. and constant rms-error function with time (in a statistical sense when considering empirical data).

The behavior of the l.p.f. at different times t_i, t_j can be evaluated by computing a Kuiper's test [6] with significance level α on the corresponding distributions. Clearly, other measures can be used.

Let us define an associated function for the test as $K_{\alpha}(\mathcal{L}_{\varepsilon}(\tau, t_1), \mathcal{L}_{\varepsilon}(\tau, t_2)) = 1$ if the two distributions are significantly different at the level α and 0 otherwise. Then, for a given time t_0, the function

$$\Delta(t_0) = \sup\{(t_0 - t) \mid (t < t_0) \,\&\, (K_\alpha(\mathscr{L}_\varepsilon(\tau,t), \mathscr{L}_\varepsilon(\tau,t_0)) = 1) \,\&\, (\forall t < \lambda < t_0)(K_\alpha(\mathscr{L}_\varepsilon(\tau,t), \mathscr{L}_\varepsilon(\tau,t_0)) = 0\}$$

gives the furthest time into the past of the process having a statistically similar lag probability function. The higher the value, the longer the process remained in the same state as the one corresponding to t_0. Small values of $\Delta(t_0)$ indicate rapid internal state changes in short times, possibly due to instability or transient regimes towards major state changes.

In an ideal stationary state, typical for a system in a fixed-stable state, the dependency patterns will be constant. Therefore, the set of lags of Equation (1) will be also constant, as well as the lag probability spectrum. An ideal model discovery process should find these lags with maximal probability within the successful models, whereas all of the other lags will appear with zero probability. For any of the predictor series, this means that a given set of lags will have high probability to be a part of good predictor models, and in principle, these probabilities will not change with time. However, if the system changes the state, a change in the dependency patterns it is to be expected, and therefore, also a change in the lag probability spectrum. This change will be more or less abrupt according to the length of the transient state.
Based on these considerations, a practical procedure for exploring the internal constitution of a process given by a heterogeneous, multivariate time series $\hat{H}^n(t)$ can be constructed by: *i)* specifying a time frame of a given length, *ii)* fixing a set of parameters for the model mining algorithm, *iii)* collecting the set of neuro-fuzzy networks representing the corresponding prediction functions, and *iv)* computing the lag probability spectrum and the mean prediction error for all time frames.

5. Examples

This section presents two examples of the application of the ideas discussed above. For simplicity, and without loss of generality, both examples are real-values and univariate: the first one is a simulation, and the other uses data from a real word domain.

A Simulation Experiment

A single time series of length 900 was constructed as a convex combination of two linear AR processes in the following way:

$$\begin{aligned} X(t) &= c_1 X(t-1) + c_2 X(t-2) + e(t), \\ Y(t) &= c_1 Y(t-2) + c_2\, Y(t-3) + e(t) \end{aligned}, \quad \text{where}$$

$c_1 = 0.4$, $c_2 = 0.5$, and $e(t) = N(0,0.001)$ is gaussian noise with mean=0 and variance=0.001. Clearly, the two processes have similar statistical properties, but differ in the time lags: (1,2) in the first, and (2,3) in the second, that is, just one single consecutive time lag. $Z(t) = \alpha(t)X(t) + (1-\alpha(t))Y(t)$, is the mixed process.
The convex combination coefficient was not constant, but a function of time, such that $\alpha(t) = 1$ for $t \le 300$, $\alpha(t) = 0$ for $t \ge 600$, and has a smooth transition between the two values for $t \in [300,600]$.
Then, the MVTSMM algorithm was applied using the following parameters: number of responsive neurons in the hidden layer=7, similarity derived from Euclidean distance (d) as $s = 1/(1+d)$, number of generations=30, population size=30, roulette-wheel selection, crossover probability=0.6, mutation probability=0.01, and elitism. Time frames of size 201 were set, exploring models up a maximum depth of 10 time lags. Within each frame, the first 101 values were used as training, and the remaining 100 as test. The 10-best discovered models were retained, and used for computing an empirical lag probability spectrum. Their mean RMSE was also obtained. In Figure 4 the spectrum is displayed as a gray-level image where for each time frame at time *t*, a black-white gray scale spans the [0,1] probability values for the given lag (along the vertical). For *t* under 300, the image contains only two bright horizontal strips, corresponding to lags 1 and 2, as expected ($Z(t) = X(t)$), whereas for *t* over 600, the bright strips are those of lags 2 and 3. also as expected ($Z(t) = Y(t)$). In the transition zone (*t* within 300 and 600), the strip of lag 1 fades whereas that of lag 3 gradually gets brighter, and that of lag 2 remains unchanged. The midpoint (450) where the process starts to be more *Y* than *X* can be clearly identified, with 150 time intervals of anticipation w.r.t. the new steady state. However, the RMSE shows no appreciable change, indicating the successful discovery of accurate models for all time frames. Therefore, the differences within the lag probability spectrum can not be explained by the presence of models of different prediction quality. The *Z* process itself does not provide clear distinctive features.

The simulation was purposely designed to make the detection. However, it shows that when the dependency models *are considered as random variables*, an in-depth model mining technique may unveil the existence of hidden internal changes within the process. Hidden or subtle changes can be detected, indicating that the system might be in a transient regime, possibly evolving towards a new state. The experiment shows that this approach has potential for anticipating forthcoming new states in a system.

Lake Ontario Water Level Fluctuations

The water levels of all of the Great Lakes have been observed for a long time due to its importance as a main hydrologic and climatologic factor. Lake Ontario was used as a real world example, and monthly averages over the period Jan-1900 to Dec 1990, (http://www.glerl.noaa.gov/data/now/wlevels/) were used in this second experiment as a real world example.

The experimental setting was almost the same as that of the previous experiment, with only a few exceptions (number of generations=100, number of population size=50, and maximum depth of 15 time lags).

The lag probability spectrum (Fig.5) clearly shows regions with different horizontal patterns, and therefore, zones with different model composition (Eq.1). The error function shows that there are zones where the best models found have poor performance as predictors (specially within the time range from 350 to 450). The boundaries defined by both the patterns of the lag spectrum and the behavior of the error function are in good agreement and they allow the segmentation of the process in terms of relatively stable, unstable, and transient states. These two different sources of information are related with different properties of the models, and they coincide in defining similar landmarks.

Conclusions

The computational intelligence approach introduced for the exploration of dynamic processes exhibits interesting properties from the point of view of detecting internal changes. Model changes and their associated properties seem to be good indicators of the alternation of steady and transient states, zones with abnormal behavior, instability and other situations, useful in characterizing and predicting the behavior of complex processes, as suggested by the results obtained with simulated and real world data. In particular, it seems that the algorithm is sensitive to subtle changes within the internal fabric of the dynamic process, which is a very promising feature from the point of view of predicting changes of state in critical domains with large anticipation. Nevertheless, these results are preliminary, and further experiments and studies are necessary.

Acknowledgements

This research was performed within the scope of the Adam project of the Integrated Reasoning Group, National Research Council of Canada (IIT-NRC). The author would like to thank Graeme Bonham-Carter from the Geologic Survey of Canada, as well as Robert Orchard, and Alan Barton, from the National Research Council of Canada (IIT-NRC).

References

[1] Belanche, Ll.: *Heterogeneous neural networks: Theory and applications*. PhD Thesis, Department of Languages and Informatic Systems, Polytechnic University of Catalonia, Barcelona, Spain,July 2000.

[2] Box, G., Jenkins, G.: *Time Series Analysis, Forecasting and Control*. Prentice Hall, 592 p, 1976.

[3] Chandon, J.L., and Pinson, S.: . Masson, Paris, 254 p, 1981.

[4] Masters,T.: *Neural, Novel & Hybrid Algorithms for Time Series Prediction*. John Wiley & Sons, 544 p, 1995.

[5] Pole,A., West M., Harrison J.: *Applied Bayesian Forecasting and Time Series Analysis*. CRC Press, 432p., 1994.

[6] Press, W.H, Teukolsky, S.A., Vetterling, W.T. Flannery, B.P: *Numerical Recipes in C*. Cambridge Univ. Press , 994 pp, 1992.

[7] Valdés, J.J, García R.: A model for heterogeneous neurons and its use in configuring neural networks for classification problems. In *Proceedings. of the International. Work-Conference on Artificial and Natural Neural Networks*, IWANN'97. Biological and Artificial Computation: From Neuroscience to Technology, (Mira, Moreno-Díaz, Cabestany eds.), *Lecture Notes in Computer Science* 1240, Springer-Verlag, pp 237-246, 1997.

[8] Valdés J.J, Belanche, L, Alquézar, R.: Fuzzy Heterogeneous Neurons for Imprecise Classification Problems. *International Journal of Intelligent Systems*, 15(3), 2000.

[9] Valdés, J.J.: Time Series Models Discovery with Similarity-Based Neuro-Fuzzy Networks and Evolutionary Algorithms. Proc. *IEEE World Congress on Computational Intelligence WCCI'2002*. IEEE, Hawaii, USA, May 12-17, 0-7803-7278-6/02, 2002.

[10] Valdés, J.J., Mateescu G: Time Series Models Discovery with Similarity-Based Neuro-Fuzzy Networks and Genetic Algorithms: A Parallel Implementation. *Third. Int. Conf. on Rough Sets and Current Trends in Computing RSCTC* 2002. Malvern, PA, USA, Oct 14-17. Alpigini, Peters, Skowron, Zhong (Eds.) *Lecture Notes in Computer Science (Lecture Notes in ArtificialIntelligence Series) LNCS 2475*, . Springer-Verlag , pp. 279-288, 2002.

[11] Valdés, J.J.: Similarity-based heterogeneous neurons in the context of general observational models: *NeuralNetwork World*, v.12, no. 5, p. 499-508, 2002.

[12] Valdés, J.J.,Barton, A, Paul, R.: Behavior of Similarity-Based Neuro-Fuzzy Networks and Evolutionary Algorithms in Time Series Model Mining. *Proc. : 9th Int. Conf. On Neural Information Processing ICONIP'02*. Nov. 18-22, 2002. Orchid Country Club, Singapore.

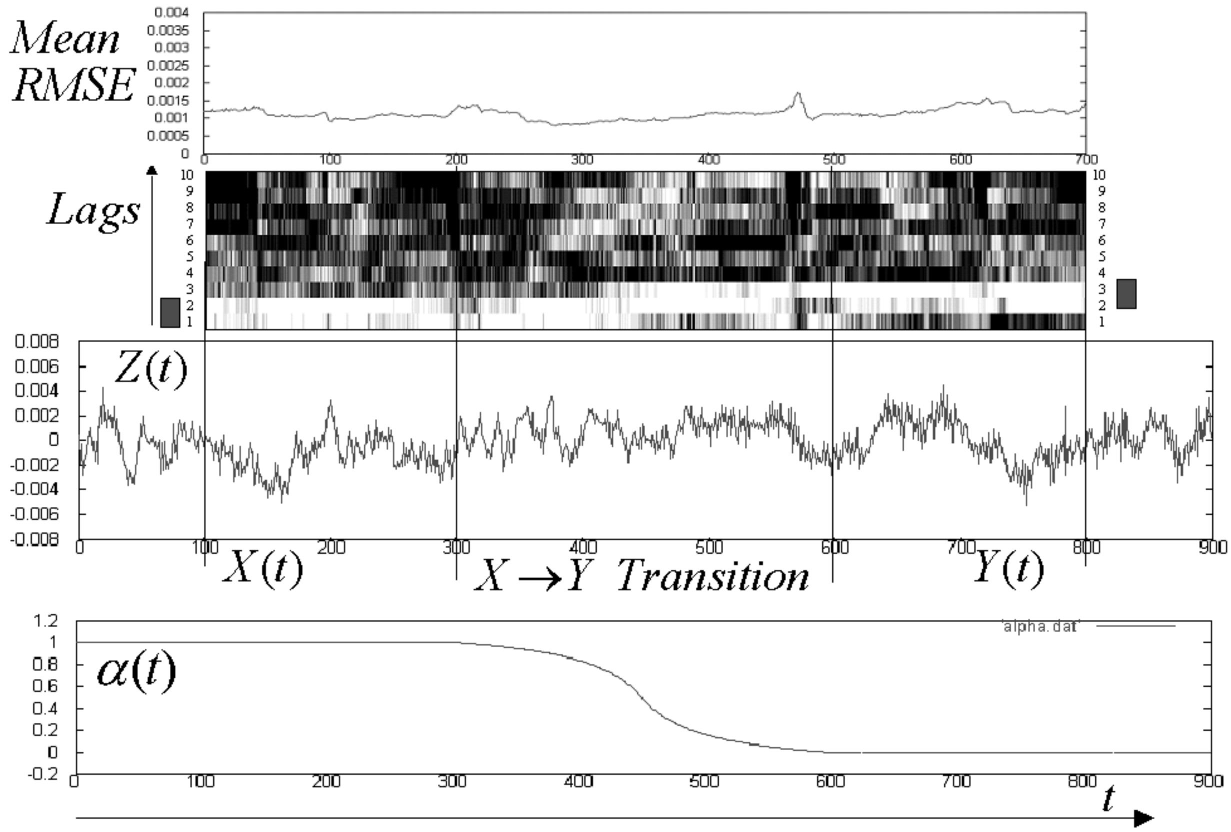

Fig. 4. *Z(t)* is a linear AR model computed as a convex combination ($\alpha(t)$)of two linear AR. The image is the lag probability spectrum. The convex combination coefficient $\alpha(t)$, and the mean RMSError for all models found, are shown as well.

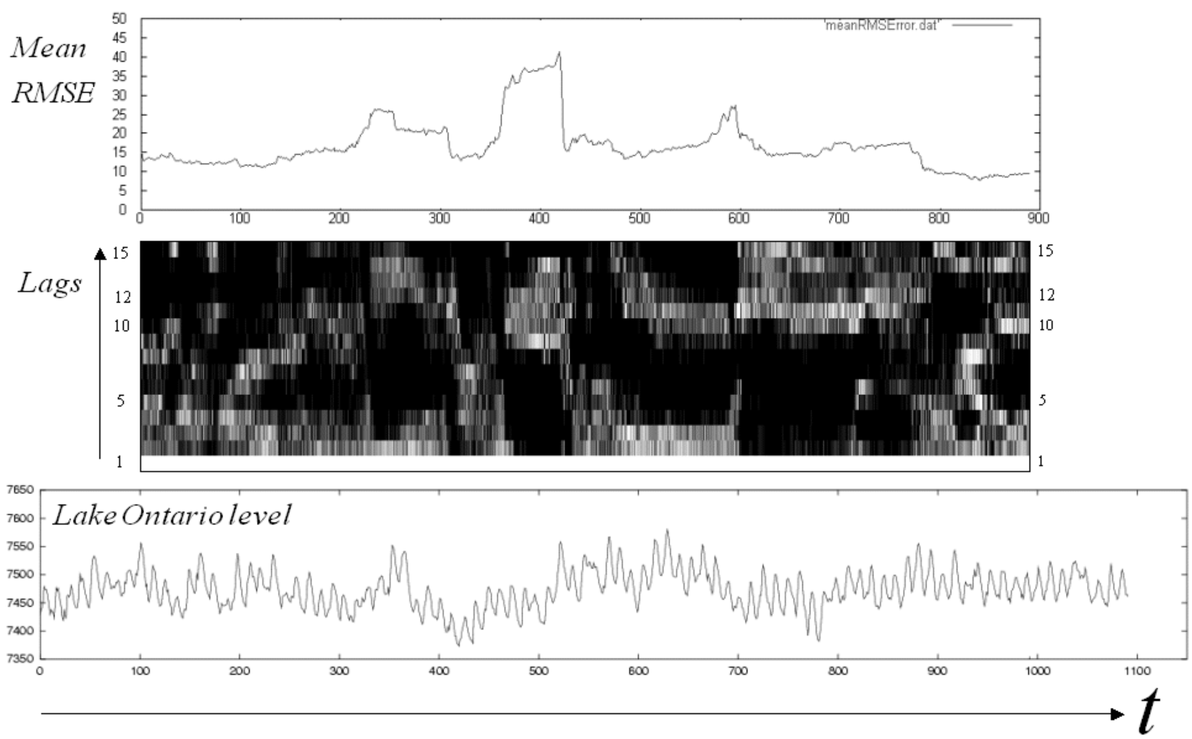

Fig. 5. Level fluctuations of Lake Ontario along with the lag probability spectrum and mean RMSError function.