TIME WARPING TECHNIQUES IN CLUSTERING TIME SERIES

Serge Parshutin and Galina Kuleshova

Department of Modelling and Simulation, Institute of Information Technology, Riga Technical University
1 Kalku Str., LV-1658 Riga, Latvia
Phone: +371 6708530, fax: +371 67089678, e-mail: serge.parshutin@rtu.lv, Galina.Kulesova@cs.rtu.lv.

The problem of obtaining an accurate forecast is becoming more and more significant as the production possibilities and technologies evolve. The accuracy of a forecast mainly depends on the dataset used for forecasting as well as on the methods employed. This study is devoted to the time series analysis. A set of known methods and techniques are used for analysing the time series; one of them is the Kohonen self-organising maps. The Time Warping techniques enable SOM to cluster time series of different duration, which is highly significant in product life cycle analysis and phase switching tasks. The main goal of the research is to perform a set of experiments aimed at comparing the efficiency of several Time Warping techniques and seeing how the chosen topology of neurons in the Self-organising map influences the final forecasting result.

Keywords: Self-organising maps, Time warping techniques, Clustering time series

1. Introduction

This paper is devoted to research on the time series analysis. The focus of the research is on monitoring and predicting the switching moment of the product life cycle phase. The problem of obtaining an accurate forecast is becoming more and more significant as production possibilities and technologies evolve. In our prior works [4, 5] we analysed the influence of the chosen data format on the precision of the system and the possibility of clustering time series of different duration. This study continues our previous works and concentrates on comparing the efficiency of classical Dynamic Time Warping and the Derivative Dynamic Time Warping algorithms, as well as on an analysis of the influence of the chosen topology of neurons in a self-organising map on the final forecasting result.

The paper is organised as follows. Section 2 describes the methods used together with source references; Section 3 presents experimental results and provides their analysis. Section 4 concludes the paper and outlines directions for further research.

2. Methods used

As was mentioned in the introduction, this research considers a neural system based on multiple self-organising maps and giving a possibility of clustering time series of different duration together with predicting the product life cycle phase switching period. The algorithm the classical Kohonen map uses is fully described in [3]. One of the characteristic features of the classical Kohonen map is that the input data vectors must be equal in length. Due to the problem described in the Introduction, the SOM has to be updated to be able to process data vectors of different length; one of the possible solutions is discussed in this section.

2.1. Modifications of Kohonen map

As was mentioned above, the classical Kohonen map can operate with data of equal length [3, 4]. To gain the possibility of clustering time series of different duration, some modifications in the classical Kohonen map are made. The classical Euclidean distance (1) calculating algorithm was substituted with one of the time warping techniques. The current study refers to the classical Dynamic Time Warping (DTW) algorithm [1, 2, and 4] and to the Derivative Dynamic Time Warping (DDTW) algorithm suggested in [2].

$$d_E(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^{n} (\xi_i + \eta_i)^2}\ , \tag{1}$$

The main difference between DTW and DDTW is that in DTW the normalized values are used to calculate the distance between the data vector and the weight vector, whereas DDTW brings an additional pre-processing step to the process - derivatives are calculated and then used in distance calculation. Expression (2) shows an approach to calculating the derivatives described in [2] that is used in the experimental part of the current study.

$$D[x] = \frac{(x_i - x_{i-1}) + ((x_{i+1} - x_{i-1})/2)}{2}, \quad 1 < i < m\,, \tag{2}$$

As can be seen from (2), the length of vector x will be shortened by two values; this will also influence the accuracy of the system as that kind of data transformation yields a loss of data [5]. The two values may not be a sufficient loss for a large time series with hundreds of values, but is quite sufficient for the problem the research is focused on, as the maximal length of the data is only 24 periods.

In other respects, DTW and DDTW are similar. The main steps of the Time Warping algorithm can be described as follows:

1. First, distance matrix $n \times m$ is built, where n and m are the lengths of the compared data vectors. Each cell of the distance matrix holds distance d_{ij} between the corresponding points of the vectors compared.
2. Starting from cell [1,1] and finishing in cell [n,m] by obeying the pre-defined rules for movements in the distance matrix, the warping path is defined. The total distance between the compared vectors will be the sum of local distances d_{ij} in the cells the warping path takes.

2.2. Main algorithm

The system itself consists of a set of self-organising maps. Each SOM can proceed with a number of vectors of different length. Figure 2.1 precisely displays the structure of the system.

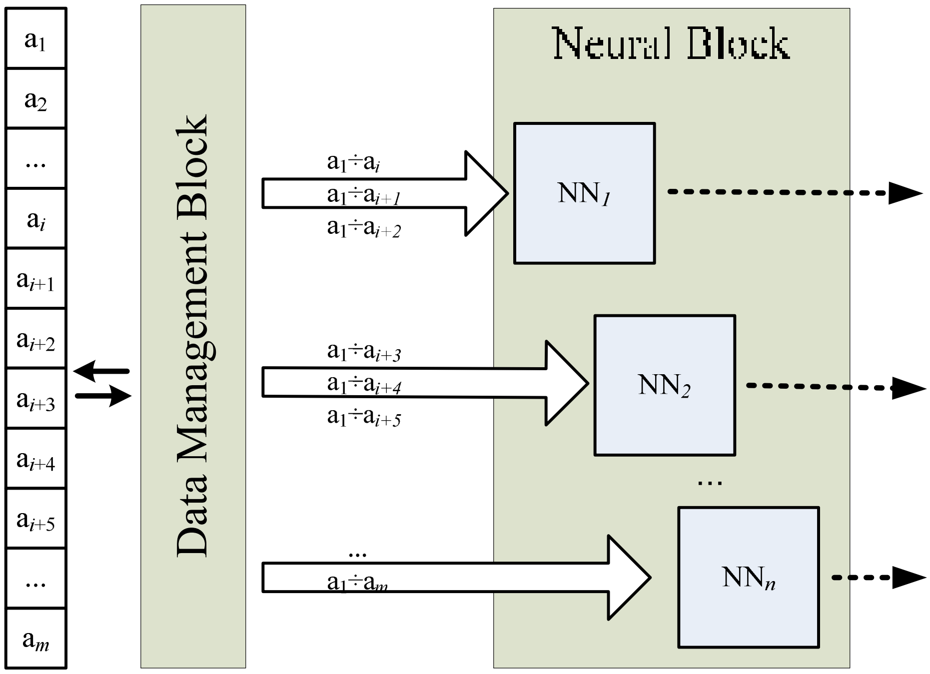

Figure 2.1. Structure of the system

Summing all the above, to solve the task described in Section 1, the following algorithm is presented:

1. Data pre-processing
   1.1. The chosen format of the data directly influences the effectiveness and precision of the system [5]. Due to that, a proper data format must be chosen to represent the data.
   1.2. Real data values should be normalized. While working with real data, most often the real bounds of a parameter are unknown, that is why the *Z-score* normalization is recommended as it uses the average value and the standard deviation to normalize data (3).

$$a_i' = \frac{a_i - \overline{a}}{\sigma_a}, \tag{3}$$

   The *Z-score* normalization that uses the mean absolute deviation (4) may also be used.

$$a_i' = \frac{a_i - \overline{a}}{s_a}, \tag{4}$$

2. Definition of the number of Self-organising maps and the values for parameters of the SOM
   2.1. By using the information available on a dataset: the length of the records in the dataset and the represented switching periods, the number of self-organising maps that will process the supplied dataset has to be defined. This step is necessary because during dynamic transformation of data, the information the data carry changes. For data with length of hundreds of values (medical data, series of sensor data, etc.) the distance in 5-10 points may not be so critical. But for demand data connected with the task described (Section 1), and having around 15 points in the time series, the difference in one point may already be critical. That is why, while analysing the length distribution of the data, it is necessary to define the subsets of records according to the

length and meaning of the data. The number of defined subsets may be used as the number of self-organising maps in the system, where each SOM processes the records of the defined length interval.

2.2. When part 2.1 is finished, the values of the parameters of each map must be set – map parameters, learning parameter, neighbourhood, etc. [3]. It is recommended to use the median of the length distribution of time series in a subset while setting the number of weights for neurons in the SOM. It will allow lessening data deformation during the SOM organisation process.

3. Learning and testing the system

3.1. In the course of system learning it is necessary to simulate the online data flowing into the system. So the realisation of the following steps is presumed:

3.1.1. Let's assume that the minimal length of a data vector a system can handle is four values, then when a time series with eight values with a life cycle phase (say Introduction-to-Maturity) switching in period 7 comes, it is necessary to start with sending the first four values of a time series, then the first five values and so on till eight, each time giving to the system the information that the life cycle phase switching period is 7. By simulating the online data flowing, the system learns to recognise the life cycle phase switching in a remote period, by having only the first few values.

3.2. Forecasting the switching. For a new record *r* of length *k* the closest cluster *c* is found in self-organising map *s* that holds data vector length *k*. During the SOM'*s* organisation process, different records with different switching periods appeared in cluster *c*. The statistics on which switching periods appeared in cluster *c* in the SOM *s* is used to make a decision on forecasting the switching period for the new record *r*. The most frequently observed switching is chosen as a forecast.

3.3. Solving a task like the one defined in Section 1, it is important to measure not only the numerical precision of a system, but also the logical precision. To measure the numerical precision, a *Mean Absolute Error (MAE)* can be used (5).

$$MAE = \frac{\sum_{i=1}^{h} |A_i - F_i|}{h}, \tag{5}$$

where h – a total number of the records in the test dataset; A – a period with a life cycle phase switching; F – a period with life cycle phase switching given by the system. For calculating a logical precision of a system, this approach is proposed: an error in the logic of a system is an event that is caused by one of the following two conditions: "There was a life cycle phase switching, but the system says that there was not" and "There was no life cycle phase switching, but the system says that there was". The number of errors is counted and divided by h, giving the logical precision of a system for the current test dataset.

3. Experimental part

This section shows an example of a practical realisation of the algorithms described above and provides an analysis of the experimental results obtained. The system is built using the *Microsoft Excel 2003* programming environment.

3.1. Data and description of experiments

The data used to organise and test the system is supplied within the ECLIPS project (see the Acknowledgements) and contains the demand during the introduction phase plus one period of the maturity phase of selected products. For each product the "Introduction to Maturity" switching period is supplied. The supplied dataset contains 199 time series with a minimal switching period equal to three and maximal – in the 23rd period. An example of the dataset is given in Figure 3.1. The time series data were normalized using the *Z-score* normalization approach.

No.	Switching	P1	P2	P3	P4	P5
1	11	-3.017	-2.403	-1.625	-1.009	-0.467
2	12	-1.989	-2.380	-1.717	-1.394	-1.327
3	7	-4.137	-1.820	-0.573	-0.102	0.162
4	5	-3.735	-2.357	-0.859	0.460	0.179
5	8	-1.718	-1.671	-1.769	-1.762	-1.575

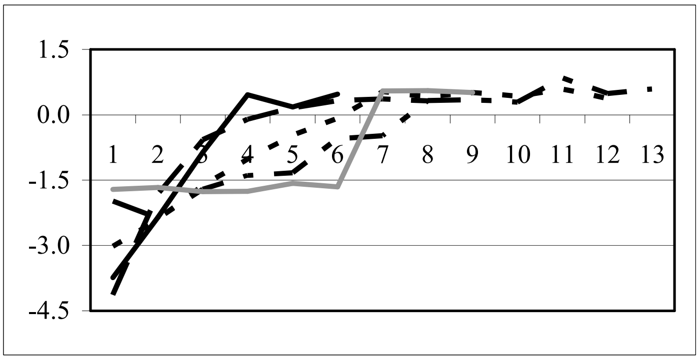

Figure 3.1. Example of normalized data (the first five periods in the table)

The main purpose of the experiments performed was to compare the Time Warping Techniques, as well as to analyse the impact of the topology of neurons in the Self-organising map onto the total efficiency of the system. In the present study five different topologies of the SOMs were compared. The size of the SOMs and the number of neurons are displayed in Table 1.

Table 1. Topology of neurons and size of self-organising maps

SOM1		SOM2		SOM3		SOM4		SOM5	
Neurons	Size	Neurons	Size	Neurons	Size	Neurons	Size	Neurons	Size
21	1x21	22	2x11	21	3x7	24	4x6	25	5x5

The learning parameters displayed in Table 2, were applied to each SOM described in Table 1.

Table 2. Self-organising map parameters

Learning parameter			**SIGMA for Gaussian neighbourhood**		
Start	End	Function	Start	End	Function
0.9	0.01	Exponential	0.5	0.01	Exponential

In total, 10 experimental runs of the system were performed. For each self-organising map mentioned algorithms DTW and DDTW were used by turns. The system was organising its SOMs during 100 learning cycles. A 10-fold cross-validation was used to test the system. The experimental results obtained are given and analysed in Subsection 3.2.

3.2. Results

This subsection discusses the experimental results obtained in performing a set of the experiments mentioned. Table 3 contains values for the mean absolute and logical errors of the system. The Online part of Table 3 shows the mean absolute and logical errors found while simulating the online data flowing (see point 3.1.1 of the main algorithm). The Offline part of Table 3 depicts errors found without simulating the Online data flowing into the system during the testing process.

Table 3. Mean absolute errors and logical errors

	Error type	**SOM1**	**SOM2**	**SOM3**	**SOM4**	**SOM5**
Online	MAE, DTW	1.8225	2.1818	2.1645	1.9833	1.7204
	MAE, DDTW	2.7515	3.1383	2.8194	2.7156	2.7349
	Logical, DTW	9.81%	10.42%	10.17%	9.82%	10.20%
	Logical, DDTW	14.39%	15.51%	14.73%	12.58%	13.53%
Offline	MAE, DTW	1.6261	2.3355	2.2405	2.1758	1.5705
	MAE, DDTW	2.3684	2.9471	2.5008	1.9947	2.3824
	Logical, DTW	45.68%	59.18%	55.68%	53.71%	54.24%
	Logical, DDTW	50.68%	58.82%	51.26%	41.74%	49.16%

The data in Table 3 are graphically represented in Figures 3.2 and 3.3. As can be seen from the figures, the precision of the system changes depending on the topology of neurons in the Self-organising maps in the neural block, which makes it possible to conclude that the topology of neurons in the SOM does influence the total performance of the system.

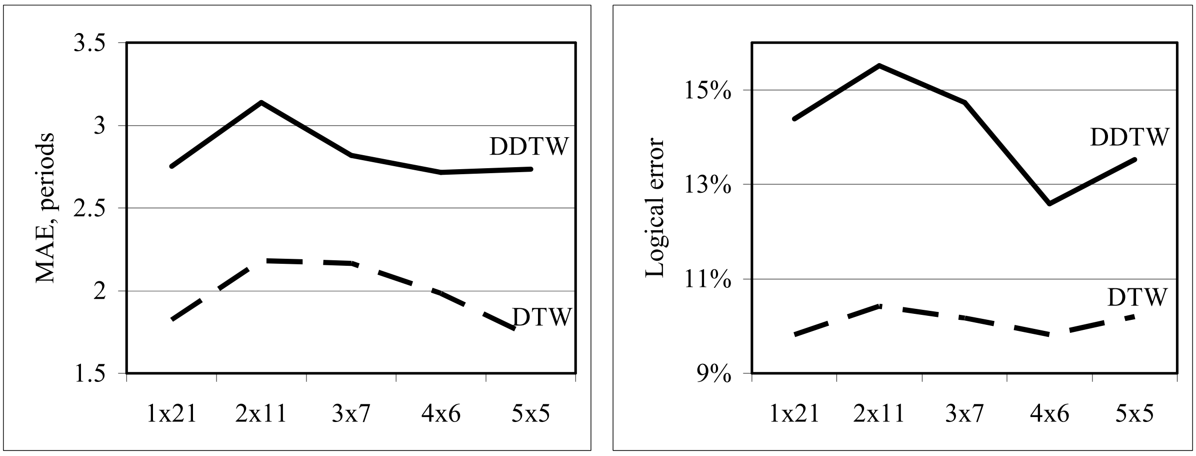

Figure 3.2. Online MAE and logical error

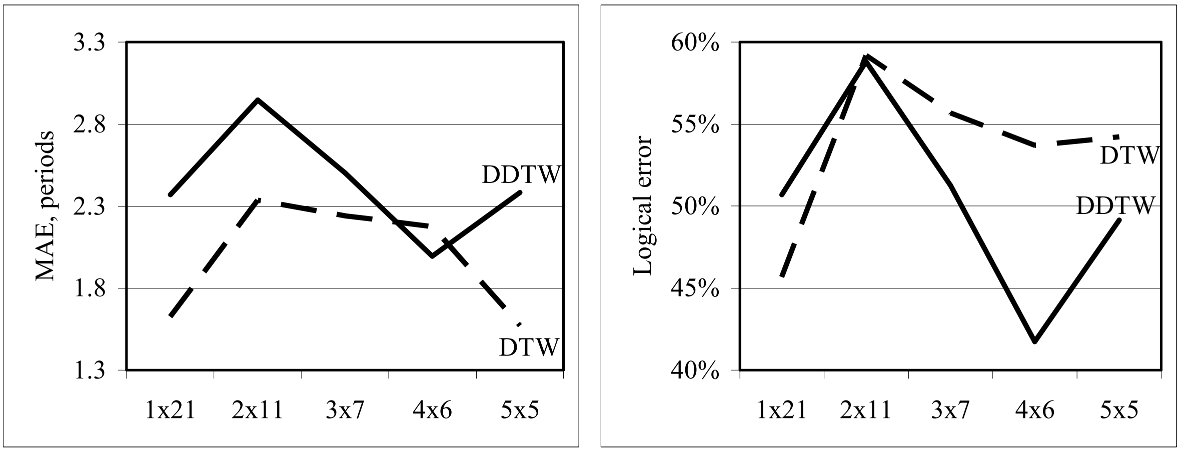

Figure 3.3. Offline MAE and logical error

While analysing the obtained experimental results, it is also possible to conclude that in the conditions defined, the use of the DTW algorithm makes it possible to lessen the mean absolute error, whereas the DDTW technique improves the logic of the system while taking a decision about the switching without using the online data flowing simulation.

For a more accurate comparison of the Time Warping techniques mentioned, the t-distribution based method for comparing the performance of the two algorithms described in [6] was applied. The parameters for t-distribution were chosen as follows: $\alpha = 0.05, k = 10$. The results obtained are summarized in Table 4 (Online results) and Table 5 (Offline results).

Table 4. Confidence intervals for online results

	SOM	+	-	Result		SOM	+	-	Result
MAE	1x21	-0.86858	-0.9895	DTW	**Logical error**	1x21	-0.04534	-0.04619	DTW
	2x11	-0.86567	-1.04734	DTW		2x11	-0.05066	-0.05105	DTW
	3x7	-0.50694	-0.80277	DTW		3x7	-0.04551	-0.04577	DTW
	4x6	-0.68618	-0.77852	DTW		4x6	-0.02725	-0.02807	DTW
	5x5	-0.94347	-1.08555	DTW		5x5	-0.03306	-0.03348	DTW

Table 5. Confidence intervals for offline results

	SOM	+	-	Result		SOM	+	-	Result
MAE	1x21	-0.67832	-0.80642	DTW	**Logical error**	1x21	-0.04573	-0.05427	DTW
	2x11	-0.50998	-0.71318	DTW		2x11	0.01032	-0.00295	No one
	3x7	0.00273	-0.52326	No one		3x7	0.05791	0.03052	DDTW
	4x6	0.38183	-0.01972	No one		4x6	0.12569	0.11379	DDTW
	5x5	-0.72942	-0.89427	DTW		5x5	0.05485	0.04673	DDTW

The negative confidence interval speaks in favour of DTW and positive confidence interval brings the DDTW in the prior place. If the confidence interval contains the 0 value, then it is not possible to say with certain confidence, which of the compared techniques is preferable.

Analysing the results shown in Tables 4 and 5, it is possible to conclude that DTW technique gives better results in all positions in the mean absolute error and in the logical error while testing the system in the online mode – with simulation of the online data flowing into the system (see point 3.1.1 of the main algorithm). In the offline mode the DTW returned better results for the mean absolute error in three of five cases; whereas in two of five cases it is not possible to define the preferable time warping technique with a certain confidence. But for the logical error in testing the system in the offline mode the DDTW appeared preferable in three out of five cases, once it was the DTW and once there was no leader between the compared time warping techniques.

4. Conclusions

The experimental results prove that the chosen topology of neurons in the self-organising map as well as the Time Warping technique have an impact on the accuracy of the SOM-based system. The results presented also provide an answer to the questions: “Which topology of neurons in the SOM is better to use?” and “Which Time Warping technique is preferable?” The answer is that there is no one best solution for all possible tasks. The result that will be achieved using a certain topology of neurons in the SOM is highly dependent on the data available, the chosen pre-processing methods and other factors. Every time a different task is solved, the chosen solution may be similar to those used before but will still be different. As regards the discussed and compared Time Warping techniques – DWT and DDTW, we may conclude that for the focus of our research the DDTW lessens the logical error in the Offline testing mode, but loses to the DTW in all other points.

The target of future research is improving the performance of the system by tuning the decision making algorithms used in the system and introducing new possible techniques aimed at improving the logic and precision of the system.

5. Acknowledgements

The present research was partly supported by the ECLIPS project of the European Commission "Extended collaborative integrated life cycle supply chain planning system".

6. References

1. Chu S., Keogh E., Hart D. and Pazzani M. Iterative Deepening Dynamic Time Warping. Second SIAM International Conference on Data Mining, 2002. Link: http://www.cs.ucr.edu/~eamonn/sdm-02.pdf, last viewed – April, 2008.
2. Keogh E. and Pazzani M. Derivative Dynamic Time Warping. First SIAM International Conference on Data Mining, 2001. Link: http://www.cs.ucr.edu/~eamonn/sdm01.pdf, last viewed – April, 2008.
3. Kohonen T. Self-Organising Maps. Springer, 3rd edition, 2001. ISBN 3-540-67921-9, 501 p.
4. Parshutin S. Clustering Time Series of Different Length Using Self-Organising Maps. Scientific Proceedings of Riga Technical University, Computer Science, October 2007. ISSN 1407-7493, 104-110 pp.
5. Parshutin S. Clustering Time Series Using Self-Organising Maps. (Russian Edition). Proceedings of IMSCAI-IV Conference. May 2007. ISBN 978-5-9221-0842-3, 465-472 pp.
6. Tan P.-N., Steinbach M. and Kumar V. Introduction to Data Mining. Pearson Education, 2006. ISBN 0-321-32136-7, 769 p.