International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:7, No:10, 2013

Adaptive Dynamic Time Warping for Variable Structure Pattern Recognition

S. V. Yendiyarov

Abstract**—**Pattern discovery from time series is of fundamental importance. Particularly, when information about the structure of a pattern is not complete, an algorithm to discover specific patterns or shapes automatically from the time series data is necessary. The dynamic time warping is a technique that allows local flexibility in aligning time series. Because of this, it is widely used in many fields such as science, medicine, industry, finance and others. However, a major problem of the dynamic time warping is that it is not able to work with structural changes of a pattern. This problem arises when the structure is influenced by noise, which is a common thing in practice for almost every application. This paper addresses this problem by means of developing a novel technique called adaptive dynamic time warping.

Keywords**—**Pattern recognition, optimal control, quadratic programming, dynamic programming, dynamic time warping, sintering control.

I. Introduction

IN recent years, there has been an increasing interest in developing different algorithms for time series pattern recognition. Most of them are based on dynamic time warping (DTW) [1]. However, a major problem of DTW is inability to work with structural changes of a pattern being searched. Many studies in DTW have been carried out [2]-[13]. However, far too little attention has been paid to the problem of variable structure pattern recognition.

The problem of pattern structure variation arises when the structure is influenced by noise, which is a common thing in practice for almost every application. The standard DTW approach uses only information about fixed pattern points and not structural elements between these points. Throughout this paper, the term 'structural element' will refer to linear or nonlinear part, which connects two adjacent fixed points in a pattern and the term 'fixed point' will refer to a point of a pattern (see Fig. 1 for better understanding).

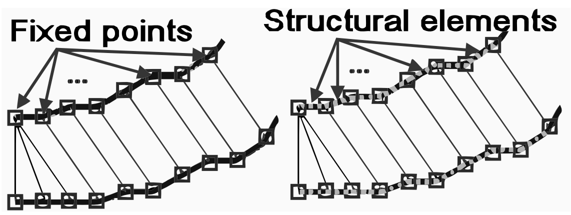

Fig. 1 On the left illustration for the term 'fixed point'; On the right illustration for the term 'structural element' (dotted lines)

S. V. Yendiyarov is with the Ural state mining university, Russia, Yekaterinburg (e-mail: endeyarov@olympus.ru).

In practice, structural elements form possible trajectories of variation of pattern structure. The DTW algorithm cannot use these elements because there are an infinite number of possible solutions to the initial problem of pattern matching, so that dynamic programming approach cannot be used (because one can select infinitely many points on every structural element, which forms a pattern).

This paper will focus on the problem of variable structure pattern recognition in time series data by means of developing a novel algorithm, which will be called adaptive dynamic time warping (ADTW).

The paper has been divided into three parts. The first part describes ADTW as an optimal control problem. The second section of this paper solves a quadratic programming problem, which is used to align points to structural elements of a pattern. Finally, in the third section, some conclusions will be drawn.

II. Adaptive Dynamic Time Warping as an Optimal Control Problem

Review The problem of pattern matching can be formulated in terms of an optimal control problem. More accurately, from a set of possible control actions U find such an action U^*, which will change a state of a system S from its initial state $\bar{\xi}_0$, that belongs to some set $\tilde{S}_0$, to some final state $\bar{\xi}_G \in \tilde{S}_G$ at the same time minimizing a criterion $W(\bar{\xi}_0, U)$.

A change of state of a system can be accomplished by means of control actions $\bar{u}_1, \bar{u}_2, ..., \bar{u}_G$:

$$U = (\bar{u}_1, \bar{u}_2, ..., \bar{u}_G) \tag{1}$$

where: $\bar{u}_j$ – a control action at step k, which changes a state of a system from $\bar{\xi}_{k-1}$ to $\bar{\xi}_k$. A criterion W, which should be minimized, can be calculated as follows:

$$W = \sum_{i=1}^{G} Z_i(\bar{\xi}_{i-1}, \bar{u}_i) \tag{2}$$

where: $Z_i(\bar{\xi}_{i-1}, \bar{u}_i)$ – a loss, which depends on the previous state $\bar{\xi}_{i-1}$ of a system and a selected control action $\bar{u}_i$, at step i.

Thus one has to select a control action $\bar{u}_k$ in such a way that it will minimize a criterion W. This statement can be written analytically:

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:7, No:10, 2013

$$W_k^*\left(\overline{\xi}_{k-1}\right)=\min_{\overline{u}_k}\left\{Z_k(\overline{\xi}_{k-1},\overline{u}_k)+W_{k+1}^*\left(\overline{\xi}_k\right)\right\} \qquad (3)$$

This equation is due to dynamic programming, which was originally used by Richard Bellman to describe the process of solving problems where one needs to find the best decisions one after another. Equation (1) assumes that a state $\overline{\xi}_{k-1}$ has been chosen already.

In this notation, a solution to the problem can be considered as minimization of the previous equation:

$$W_k^*\left(\overline{\xi}_0\right)=\min_U W\left(\overline{\xi}_0,U\right) \qquad (4)$$

A solution of the given problem is a sequence of some optimal control actions U^*:

$$U^*=(\overline{u}_1^{(*)},\overline{u}_2^{(*)},...,\overline{u}_G^{(*)}) \qquad (5)$$

Taking into account that these control actions U^* should belong to a given set of possible control actions $\tilde{\Theta}$:

$$\tilde{\Theta}=\left\{\tilde{\Theta}_1,\tilde{\Theta}_2,...,\tilde{\Theta}_G\right\},\forall\overline{u}_j^{(*)}\in\tilde{\Theta}_j \qquad (6)$$

Elements of $\tilde{\Theta}$ are determined by linear functions $w_j(X)$ with restricted domains, which are set by intervals $X_i\le X\le X_j$:

$$\tilde{\Theta}_j\in\left\{\tilde{\Omega}_{X_i}^{X_j}\subseteq w_j(X)\right\} \qquad (7)$$

where: $\tilde{\Omega}_{X_i}^{X_j}$ – it denotes a set of values, which is determined by linear functions $w_j(X)$ with restricted domains $X_i\le X\le X_j$.

Except the existing constraints, which are placed on the possible control actions, constraints on possible transitions are also placed. These constraints allow to avoid impossible transitions between different states.

For this reason, let assume that elements of a state $\overline{\xi}_j$ can be classified in several classes $A,B,C,...,W$:

$$э_k^{(z)}\in\overline{\xi}_j, z\in\left\{A,B,C,...,W\right\} \qquad (8)$$

where: $э_k^{(z)}$ – k'th element, which belongs to some class z, of a state $\overline{\xi}_j$.

Constraints on the possible transitions can be written in the following way:

$$\forall\, э_k^{(z)}\in\overline{\xi}_j \rightarrow э_q^{(r)}\in\overline{\xi}_i \qquad (9)$$

This means that for all elements $э_k^{(z)}$ (of some state $\overline{\xi}_j$), which belong to some class z, it is possible to move to an element $э_q^{(r)}$ (of some state $\overline{\xi}_i$), which belongs to some class r.

This formal approach can now be easily adopted in order to solve the task of time series data pattern matching. Recognition of a pattern should be realized by means of multi-scale approximation of a pattern. This task can be considered as fitting a piece-wise linear function to time series data at different scales.

As stated before, the possible control actions, at a given interval, should belong to some set, which is restricted by some linear function $w_j(z)$. Thus, the set has an infinite number of elements. Therefore, at the initial stage, the principle called 'expected losses' will be used. In order to do so, the linear functions $w_j(z)$ are replaced by their mean values $\overline{w}_j$ at the given interval:

$$\overline{w}_j=\left(\frac{1}{z_j-z_i}\int_{z_i}^{z_j}w_j(z)dz\right) \qquad (10)$$

This transformation of the initial pattern is illustrated in Fig. 2.

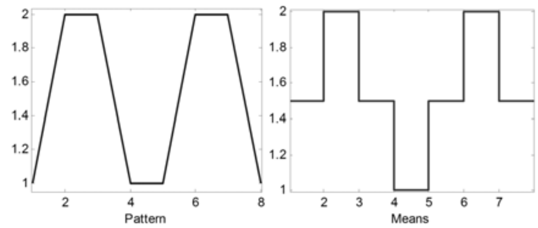

Fig. 2 Replacing linear functions $w_j(z)$ (on the left) by their means $\overline{w}_j$ (on the right)

After this transformation of the initial pattern, we can form a set of classes, which then can be used to classify elements $э_k^{(q)}$ of some state $\overline{\xi}_j$:

$$э_k^{(q)}\in\overline{\xi}_j, q\in\left\{U_1,U_2,...U_m,F_1,F_2,...,F_z\right\} \qquad (11)$$

where: U_j – a class, which relates an element $э_k^{(U_j)}$ to some interval j, which is characterized by $\overline{w}_j$, F_j – a class, which relates an element $э_k^{(F_j)}$ to some fixed point j of a pattern. Hence, in order to solve the task of pattern recognition it is enough to use only two classes F,U.

A set of possible control actions for the task of pattern recognition, at some step k, can be written as follows:

$$\overline{u}_k = \left\{ \overline{u}_k^{\left(\ni_1^{(U)}\right)}, ..., \overline{u}_k^{\left(\ni_m^{(U)}\right)}, ..., \overline{u}_k^{\left(\ni_1^{(F)}\right)}, ..., \overline{u}_k^{\left(\ni_z^{(F)}\right)} \right\} \quad (12)$$

where: $\overline{u}_k^{\left(\ni_j^{(U)}\right)}$ – it denotes a transition from some element of a state $\overline{\xi}_{k-1}$ to some j'th element of a state $\overline{\xi}_k$, which belongs to a class U.

A function $Z_i(\overline{\xi}_{i-1}, \overline{u}_i)$, which is used to calculate optimal approximation of a pattern to some time series data can be obtained from the expression:

$$Z_k\left(\overline{\xi}_{k-1}, \overline{u}_k\right) = \Delta E_j^{k-1} + P\left(\ni_j^{(k)}, \overline{u}_k^{\left(\ni_r^{(q)}\right)}\right) + R\left(\ni_j^{(k)}\right) \quad (13)$$

where: ΔE_j^{k-1} – expected losses, which we will incur when we select some element $\ni_j^{(k)}$ of a state $\overline{\xi}_{k-1}$, $P\left(\ni_j^{(z)}, \overline{u}_k^{\left(\ni_r^{(q)}\right)}\right)$ – a penalty function, which places constraints on possible transitions between states. More precisely, it places constraints on possible transitions from state $\overline{\xi}_{k-1}$ to $\overline{\xi}_k$. Meanwhile $\ni_j^{(z)}$ is an element of a state $\overline{\xi}_{k-1}$ and $\overline{u}_k^{\left(\ni_r^{(q)}\right)}$ is a control action, which determines an element $\ni_r^{(q)} \in \overline{\xi}_k$. $R\left(\ni_j^{(k)}\right)$ – a weighting function, which determines a gain or a prize for transitioning from a state $\ni_j^{(k)} \in \overline{\xi}_{k-1}$.

Expected losses, which we will incur when we select some element $\ni_j^{(k)}$ of a state $\overline{\xi}_{k-1}$ form a matrix of expected losses ΔE:

$$\Delta E = \begin{bmatrix} & U_1 & U_2 & ... & U_m & F_1 & F_2 & ... & F_z \\ \overline{\xi}_1 & \Delta E_1^1 & \Delta E_2^1 & ... & \Delta E_m^1 & \Delta E_1^1 & \Delta E_2^1 & ... & \Delta E_z^1 \\ \overline{\xi}_2 & \Delta E_1^2 & \Delta E_2^2 & ... & \Delta E_m^2 & \Delta E_1^2 & \Delta E_2^2 & ... & \Delta E_z^2 \\ ... & ... & & & & & & & \\ \overline{\xi}_G & \Delta E_1^G & \Delta E_2^G & ... & \Delta E_m^G & \Delta E_1^G & \Delta E_2^G & ... & \Delta E_z^G \end{bmatrix} \quad (14)$$

The elements ΔE_i^j of the matrix ΔE can be calculated by means of the following expression:

$$\Delta E_i^j = \left| \overline{\xi}_j - \overline{\lambda}_i \right|, \overline{\lambda} \in \{U_1, U_2, ...U_m, F_1, F_2, ..., F_z\} \quad (15)$$

where: U_j – a value of $\overline{w}_j$, F_j – a value of a fixed point of a pattern, $\overline{\xi}_j$ – a j'th value of time series data to be approximated by our initial pattern.

The weighting function $R\left(\ni_j^{(k)}\right)$, which determines a prize for transitioning from a state $\ni_j^{(k)} \in \overline{\xi}_{k-1}$:

$$R\left(\ni_j^{(k)}\right) = \begin{cases} k \in F_i, R = -G \cdot \Delta e^{MAX} \\ k \in U_i, R = 0 \end{cases} \quad (16)$$

where G – the length of time series data, Δe^{MAX} – the maximum expected loss ΔE_i^j.

The weighting function guarantees that all fixed points $F_1, F_2, ..., F_z$ of a pattern will be included in a solution of the problem. Thus, it preserves the inherent structure of a pattern.

The penalty function $P\left(\ni_j^{(z)}, \overline{u}_k^{\left(\ni_r^{(q)}\right)}\right)$ is used to place constraints on possible transitions between states and can be written as follows:

$$P\left(\ni_j^{(z)}, \overline{u}_k^{\left(\ni_r^{(q)}\right)}\right) = \begin{cases} \left(z \in U_l \cup j = r\right), P = 0 \\ \left(z \in U_l \cup j \neq r\right), P = +\infty \\ \left(z \in F_l \cup j = r-1\right), P = 0 \\ \left(z \in F_l \cup j \neq r-1\right), P = +\infty \end{cases} \quad (17)$$

In Fig. 3 a decision tree is shown, in order to clarify the meaning of $P\left(\ni_j^{(z)}, \overline{u}_k^{\left(\ni_r^{(q)}\right)}\right)$.

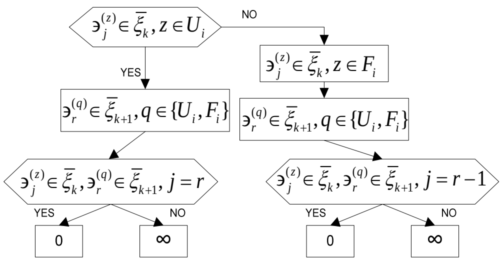

Fig. 3 Decision tree, which shows a behavior of the penalty function $P\left(\ni_j^{(z)}, \overline{u}_k^{\left(\ni_r^{(q)}\right)}\right)$

The penalty function $P\left(\ni_j^{(z)}, \overline{u}_k^{\left(\ni_r^{(q)}\right)}\right)$ allows to exclude some possible approximations, which do not obey certain predefined behavior.

III. Aligning Data Points to Structural Elements of a Pattern

In the previous section, we described a novel algorithm ADTW, which can be used to find patterns with varying structure in time series data. Despite the fact that ADTW can work with patterns with varying structure, we aware that ADTW has one limitation, which we will resolve in the present section. If we get back to (10) we will see that our linear functions $w_j(z)$ were replaced by their mean values. It actually means that we cannot precisely align our pattern and time series data. This problem is addressed in this section.

First of all, our pattern should be approximated by linear functions (i.e. each structural element of a pattern should be represented by a linear function). This representation is useful, especially in our case:

$$\bar{\bar{\Omega}}(x)=\begin{cases} x\in[1,2), \bar{\bar{\Omega}}(x)=\omega_1 x+\varepsilon_1 \\ \ldots \\ x\in[j,j+1), \bar{\bar{\Omega}}(x)=\omega_j x+\varepsilon_{j+1} \\ \ldots \\ x\in[n-1,n], \bar{\bar{\Omega}}(x)=\omega_n x+\varepsilon_n \end{cases} \quad (18)$$

where $\bar{\bar{\Omega}}(x)=\omega_j x+\varepsilon_{j+1}$ represents our linear functions $w_j(z)$, which place constraints on the possible pattern structure.

In the case, when some interval $x\in(\eta_1,\eta_2)$ contains only one point the alignment can be readily obtained. The optimal value of x^*_{opt} should be selected taking into account the following constraint:

$$\eta_1 < x_j < \eta_2 \quad (19)$$

In this case, the following equation should be solved:

$$\begin{aligned} &\tilde{Y}_j^{DATA}-(Ax_j^*+B)=0 \\ &x_j^*=\left(\frac{\tilde{Y}_j^{DATA}-B}{A}\right) \end{aligned} \quad (20)$$

where: A,B – coefficients of a linear model on the interval (η_1,η_2), x_j^* – the optimal value for the present interval.

The optimal value x_j^* can be calculated as follows:

$$x_j^*=\begin{cases} x_j^*\in(\eta_1,\eta_2), x_j^* \\ x_j^*<\eta_1, \eta_1+\xi \\ x_j^*>\eta_2, \eta_2-\xi \end{cases} \quad (21)$$

where: ξ is some small quantity (i.e. $\xi=0.0001$).

In the case, when some interval $x\in(\eta_1,\eta_2)$ contains more than one point i.e.:

$$X=\{x_1,\ldots,x_n\}, x_j\in(\eta_1,\eta_2) \quad (22)$$

In this case, (21) cannot be used to align a pattern and time series data. To solve this problem we propose the following quadratic programming problem:

$$F(\beta)=\frac{1}{2}X^T\mathbb{Q}X+\mathbb{R}^T X\to\min_{x_j} \quad (23)$$

Taking into account constraints:

$$\begin{aligned} &\mathbb{Z}X\le b \\ &\eta_1+\xi<x_j<\eta_2-\xi \end{aligned}$$

Our problem should be transformed in order to place it in the form of (23). It can be stated as follows:

$$F(\beta)=\sum_{j=1}^{d}\left(\left(Ax_j+B\right)-y_j\right)^2\to\min_{x_j} \quad (24)$$

We opened the brackets in (24) and discarded constants:

$$\begin{aligned} F(\beta)&=\sum_{j=1}^{d}\left(\left(Ax_j+B\right)-y_j\right)^2= \\ &=\sum_{j=1}^{d}\left(A^2x_j^2+2ABx_j-2Ax_jy_j-2By_j+y_j^2+B^2\right)= \\ &=\sum_{j=1}^{d}\left(A^2x_j^2+x_j\left(2AB-2Ay_j\right)\right)=\sum_{j=1}^{d}\left(Cx_j^2+D_jx_j\right) \end{aligned} \quad (25)$$

where: C – a constant, which equals A^2, D_j – a value, which depends on y_j and it can be calculated as $2AB-2Ay_j$.

Now we can obtain matrices $\mathbb{Q},\mathbb{R},\mathbb{Z}$:

$$\begin{aligned} &\mathbb{Q}_{d\,x\,d}=\begin{bmatrix} 2C & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 0 \\ \ldots & 0 & 2C & \ldots \\ 0 & 0 & \ldots & 2C \end{bmatrix}, \\ &\mathbb{Z}_{d\,x\,d}=\begin{bmatrix} 1 & -1 & 0 & \ldots \\ 0 & 1 & -1 & \ldots \\ \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 1 & -1 \\ 0 & 0 & \ldots & 0 \end{bmatrix} \\ &\mathbb{R}^T_{1xd}=[D_1,D_2,\ldots,D_d], b^T_{1xd}=[-\xi,-\xi,\ldots,0] \end{aligned} \quad (26)$$

A solution to this problem will be a vector of the optimal values of X^T on the interval (η_1,η_2).

The proposed model allows precisely aligning our pattern and time series data.

IV. Conclusion

In this paper the development of a novel algorithm, which is called adaptive dynamic time warping, was developed in order to increase recognition abilities of the standard DTW algorithm.

For this reason, we have introduced two important terms: ‘fixed point’ and ‘structural element’. The standard DTW algorithm allows only to use fixed points of a pattern, which seriously restricts its recognition abilities.

The proposed approach first approximates structural elements of a pattern by constants and then uses dynamic programming optimization procedure. After that the quadratic programming problem was formulated and solved in order to accurately align time series data points to the structural elements of a pattern.

The process of pattern matching was described in terms of an optimal control of some system, which should be moved from some initial state $\bar{\xi}_{k-1}$ to some final state $\bar{\xi}_k$, while minimizing some criterion W. In our case, the system represents a pattern being searched. Meanwhile control actions represent possible changes of the pattern structure.

In order to control matching procedure two functions were introduced: the penalty function and the weighting function. The first one is used to place constraints on possible transitions between states. The weighting function guarantees that all fixed points of a pattern will be included in a solution of the problem. Thus, it preserves the inherent structure of a pattern.

The proposed approach permits different penalty as well as weighting functions to be developed. This fact makes our approach very flexible.

This article is a direct continuation of the article [14], because we used proposed technique in our expert system to recognize temperature profiles of suction boxes, in order to control speed of the sintering car.

References

[1] D. Berndt, J. Clifford.: Using Dynamic Time Warping to Find Patterns in Time Series. In KDD-94: AAAI Workshop on Knowledge Discovery in Databases, Seattle, Washington, July 1994, pp. 359-370.

[2] K. Hiri-o-tappa, S. Narupiti, S. Pattara-atikom.: A novel approach of dynamic time warping for short-term traffic congestion prediction, Transportation Research Board 90th Annual Meeting, January 23-27, Washington, D.C., 2011, pp. 1-15.

[3] E. Keogh and C. Ratanamahatana.: Exact indexing of dynamic time warping, Knowl. Inf. Syst. 7(3), 2005, pp. 358-386.

[4] Y. Sakurai.: FTW: fast similarity search under the time warping distance, PODS, 2005, pp. 326-337.

[5] A. Fu, E. Keogh, L. Lau, C. Ratanamahatana, R. Wong.: Scaling and time warping in time series querying, VLDB, 17, 4, pp. 899-921, 2008.

[6] N. Gillian, R. Knapp, S. O'Modhrain.: Recognition of multivariate temporal musical gestures using n-dimensional dynamic time warping. Proc of the 11th Int'l conference on New Interfaces for Musical Expression, 2011.

[7] E. Keogh, L. Wei, X. Xi, M. Vlachos, S.H. Lee, P. Protopapas.: Supporting exact indexing of arbitrarily rotated shapes and periodic time series under Euclidean and warping distance measures. VLDB J. 18, 3, pp. 611-630, 2009.

[8] A. Mueen, E. Keogh.: Online discovery and maintenance of time series motifs. KDD, pp. 1089-1098, 2010.

[9] L. Ye, E. Keogh.: Time series shapelets: a new primitive for data mining. KDD, pp. 947-956, 2009.

[10] Y. Zhang, J. Glass.: An inner-product lower-bound estimate for dynamic time warping. ICASSP, pp. 5660-5663, 2011.

[11] E. Keogh.: Exact indexing of dynamic time warping.In VLDB, pp. 406–417, 2002.

[12] E. Keogh, M. Pazzani.: Scaling up dynamic time warping for data mining applications, KDD 2000, pp. 285-289, 2000.

[13] B. Yi.: Efficient Retrieval of Similar Time Sequences Under Time Warping, ICDE 1998, pp. 201-208, 1998.

[14] Yendiyarov, S., Zobnin B., Petrushenko S. Expert system for sintering process control based on the information about solid-fuel flow composition // Proceedings of World Academy of Science, Engineering and Technology, France, Issue 68, August 2012, pp. 861-868

S. V. Yendiyarov received his PhD degree in information processing and analysis from the Ural state mining university. His research interests are optimal control, signal processing, condition monitoring and agent-oriented systems.