HUNGARIAN JOURNAL
OF INDUSTRIAL CHEMISTRY
VESZPRÉM
Vol. 36(1-2) pp. 101-106 (2008)

PCA DRIVEN SIMILARTY FOR SEGMENTED UNIVARIATE TIME SERIES

Z. BANKÓ, J. ABONYI[1]✉

[1]University of Pannonia, Department of Process Engineering, P.O. Box 158. Veszprém H-8200, HUNGARY
✉E-mail: abonyij@fmt.uni-pannon.hu

Selection of the proper similarity measure is the cornerstone of all time series data mining task. In the recent years, many similarity measures have been introduced for the needs of chemical process engineering. These measures have been guided by data reduction methods due to the large amount of data. This data reduction can be done explicitly (by segmentation) as well as implicitly (by utilizing the latent variable space). Usually, the original multivariate data is projected into a single dimension with Principal Component Analysis (PCA) and segmentation is executed. However, the similarity measures which have been used to compare univariate, segmented representations of the original processes do not consider that the main information carried by the univariate representations is the correlation of the original variables. This paper introduces a PCA inspired similarity measure for these univariate segments. Finally, it is shown that the presented method can be seen as the logical extension of the Correlation Based Dynamic Time Warping (CBDTW) to univariate time series.

Keywords: segmentation, Dynamic Time Warping, Principal Component Analysis, Piecewise Linear Approximation

Introduction

A time series is a sequence of values measured as a function of time. These kinds of data are widely used in the field of chemical process engineering, namely for process control, fault detection and diagnosis of process transitions. The increasing popularity of knowledge discovery and data mining tasks for discrete data has indicated the growing need for similarly efficient methods for time series data. These tasks share a common requirement: a similarity measure has to be defined between the elements of a given database. Moreover, the results of the data mining methods from simple clustering (partitioning the data into coherent but not predefined subsets) and classification (placing the data into predefined, labelled groups) to complex decision-making systems used for process control are highly dependent on the applied similarity measure.

Related work

The similarity of multivariate time series can be approached from two different perspectives. The first way is the application of metrics based warping measures such as Dynamic Time Warping (DTW) and Longest Common SubSequence (LCSS). These techniques are perfectly suitable for univariate tasks like speech recognition, where the analyzed process is represented by one variable only. In most cases, these methods can be easily generalized for the needs of the multivariate time series where the process depends on two or more variables. However, their application for correlated multivariate time series is often not as effective as it is expected.

The direct comparison of the variables used by these approaches ignores the hidden process, i.e. the correlation between the process variables and this hidden process carries the real information in most process control task [1]. Hence, Principal Component Analysis (PCA) based similarity measures are used to overcome this problem. Krzanowski [2] defined the PCA similarity factor to measure the similarity between different data by comparing the hyperplanes (the dimensionality reduced latent variable spaces):

$$s_{PCA}(X_n, Y_n) = \frac{tr(U_{X_n,p}{}^T U_{Y_n,p} U_{Y_n,p}{}^T U_{X_n,p})}{p}$$

Where:
X_n – the first n-variable multivariate time series
Y_n – the second n-variable multivariate time series
$U_{Xn,p}$ and $U_{Yn,p}$ – the matrices of eigenvectors which belong to the most important $p \leq n$ eigenvalues of covariance matrices of X_n and Y_n, i.e. the two hyperplanes

The similarity factor has a geometrical explanation, because it measures the similarity between the two hyperplanes by computing the squared cosine values between all the combinations of the first p principal components from X_n and Y_n:

$$s_{PCA}(X_n, Y_n) = \frac{1}{p} \sum_{i=1}^{p} \sum_{j=1}^{p} \cos^2 \Theta_{i,j}$$

Where:
$\Theta_{i,j}$ – the angle between the ith principal component of X_n and the jth principal component of Y_n

The main advantage of Krzanowski's similarity factor is its optimal variable reduction property (from variance point of view) which makes it ideal for tasks with numerous variables. PCA similarity factor has also gained in popularity because of its outstanding feature which makes possible to recognize the direction of the changes in the distance between the variables, i.e. the rotation of the hyperplanes.

On the other hand, PCA similarity factor weights all principal components equally, hence it may not capture well enough the degree of similarity between the sequences when only a few principal components explain most of the variance. Thus, it was natural to define a modified PCA similarity factor that weights each principal component by its explained variance. M. C. Johannesmeyer [3] defined this modified PCA similarity factor by weighting each principal component with its eigenvalue:

$$s^{\lambda}_{PCA}(X_n, Y_n) = \frac{\sum_{i=1}^{p}\sum_{j=1}^{p}(\lambda_i^{X_n}\lambda_j^{Y_n})\cos^2\Theta_{i,j}}{\sum_{i=1}^{p}(\lambda_i^{X_n}\lambda_i^{Y_n})}$$

Where:
λ_i^{Xn} and λ_i^{Yn} – the corresponding eigenvalues of the ith and jth principal component of X_n and Y_n

This principle was developed by K. Yang [4] who presented the logical extension of PCA similarity factor and S^{λ}_{PCA} called Eros (Extended Frobenius Norm):

$$s_{Eros}(X_n, Y_n) = \sum_{i=1}^{n} w_i \left| U_{X_n}(i) U^T_{Y_n}(i) \right| = \sum_{i=1}^{n} w_i \left| \cos\Theta_i \right|$$

Where:
Θ_i – the angle between the two corresponding principal components
w_i – the weighting vector based on the eigenvalues of the sequences in the data set

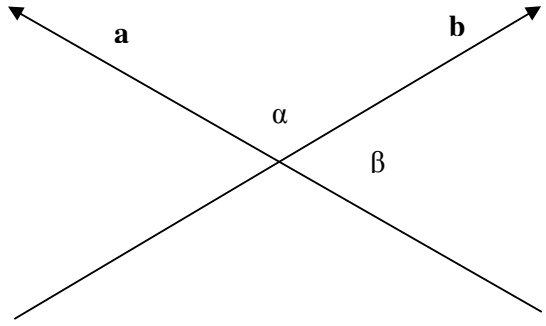

Figure 1: Two corresponding principal components

Generally speaking, the Eros measures the similarity of two multivariate time series by comparing the angle between the corresponding principal components and using the aggregated eigenvalues as weights, hence it takes into account the variance of each principal component. It has to be noted that Eros always computes the acute angle between the two principal components (eigenvectors). Therefore, as it is illustrated in Figure 1, when the angle (α) between the two corresponding eigenvectors is not acute, the absolute value of it is taken and the similarity between the two corresponding eigenvectors is computed by using the acute angle (β). More specifically, the inner product of two normal vectors, $\boldsymbol{a}$ and $\boldsymbol{b}$ in Figure 1, yields $cos(\alpha)$, while $cos(\beta) = cos(\pi-\alpha) = -cos(\alpha)$ is needed. Therefore, the absolute value of cosine of the angle between the eigenvectors is taken, so that $\cos(\alpha)$ is computed when $\alpha \leq \pi/2$, while $-cos(\alpha)$ is computed when $\alpha > \pi/2$.

The previously mentioned methods greatly improved the simple PCA similarity factor both in speed and in accuracy; however, they have not dealt with the biggest problem of every PCA related technique, i.e. PCA considers the time series as a whole but does not take into account the alterations of the correlation structure. This alternation affects the hyperplanes, therefore segmentation is required in most real-life applications to create homogeneous segments from correlation structure point of view. However, the segmentation raises another problem: *Although in many real-life applications a lot of variables must be simultaneously tracked and monitored, most of the segmentation algorithms are used for the analysis of only one time-variant variable* [5]. Hence, dimensionality reduction techniques are used, most likely PCA, to project the multivariate time series into the one dimensional space where any suitable univariate segmentation method such as Piecewise Aggregate Approximation (PAA) or Piecewise Linear Approximation (PLA) is executed. Finally, the segments are compared with a suitable similarity measure.

Finding this suitable similarity measure is a difficult task. Besides the obvious Euclidean distance other, more advanced measures can be used such as DTW. It proved its adaptability and superiority over other similarity measures in wide range of time series applications from speech recognition [6] to fingerprint verification [7] and as final proof of this Yanikoglu won the Signature Verification Conference with a DTW based algorithm in 2004 [8].

Keogh and Pazzani presented modifications on DTW algorithm to handle PAA [9] and PLA [10] representations of univariate time series. Although these new algorithms provide noticeably better results than Euclidean distance and speeds up the computationally expensive general DTW algorithm, they do not take the drift of the segments into account.

Thus, the segmentation has to be framed in another way. Instead of compressing the multivariate data to a univariate time series to apply PLA or PAA, the segmentation can be done in the multivariate space while the correlation is still considered. The authors introduced [11] two homogeneity measures as cost function for segmentation which are corresponding to the two typical applications of PCA models. The Q reconstruction error can be used to segment the time

series according to the direct change of the correlation among the variables, while the Hotelling's T^2 statistics can be utilized to segment the time series based on the drift of the center of the operating region.

Based on these new segmentation methods a novel similarity measure, called Correlation Based Dynamic Time Warping was created [12]. It was proven that it outperforms all of the previously introduced correlation based similarity measures when the multivariate time series are highly correlated. This paper introduces the univariate version of CBDTW which can be used for PCA projected univariate time series.

The rest of the paper is organized as follows. Section Background details the theoretical background of the proposed similarity measure, i.e. segmentation and DTW. In the next section the problems of the current DTW algorithms used for segmented representations are pointed out and the new similarity measure is presented. It is also discussed how it was derived from CBDTW. Section 4 conducts a detailed empirical comparison of the introduced PCA based method with currently used measures on verification databases widely used by the time series data mining community. Finally, validation is performed by clustering of temperature data from a sophisticated model of an industrial catalytic fixed bed tube reactor.

Background

An n-variable, m-element time series, $X_n = [x_1,x_2,\ldots,x_n]$, is an m-by-n element matrix, where $x_i = [x_i(1),x_i(2),\ldots,x_i(n)]^T$ is the ith variable and $x_i(j)$ its jth element. $X_n(j) = [x_1(j),x_2(j),\ldots,x_n(j)]$ is the jth sample of X_n. The similarity between X_n and Y_n is denoted by $s(X_n, Y_n)$, where $0 \le s(X_n, Y_n) \le 1$, $s(X_n, Y_n) = s(Y_n, X_n)$, and $s(X_n, X_n) = 1$. Obviously, the similarity is nothing more than a real number between zero and one which expresses the tightness of connection between the processes behind the time series. The closer the number is to one, the processes are treated more similar. In practice, the term distance or dissimilarity (d) is used instead of similarity. The value of the distance is given by a number ranged from zero to one. This can be associated with similarity: $d(X_n, Y_n) = 1 - s(X_n, Y_n)$.

Segmentation

The ith segment of X_n is a set of consecutive time points, $S_i(a,b) = [X_n(a), X_n(a+1),\ldots, X_n(b)]$. The c-segmentation of time series X_n is a partition of X_n to c non-overlapping segments, $S^c_{Xn} = [S_1(1,a),S_2(a+1,b),\ldots,S_c(k+1,m)]$. In other words, a c-segmentation splits X_n to c disjoint time intervals, where $1 \le a$ and $k \le m$.

The simplest but yet powerful segmentation technique for univariate time series is PAA. In this case, to reduce the m-length data from N, the time series are simply divided into N similar sized frames and each frame is represented by its mean value. Assuming that N is a factor of m, we get:

$$\underline{x}_1(i) = \frac{N}{m} \sum_{j=\frac{m}{N}(i-1)+1}^{\frac{m}{N}i} x_1(j)$$

Besides filtering noise, PAA can compensate phase shifts of time axis and difference sampling rates of time series and the distance between two PAA representations can be chosen almost freely; however, it cannot handle "locally elastic" shifts of the time axis and it is not enough tight representation for most time series as it is shown in Figure 2.

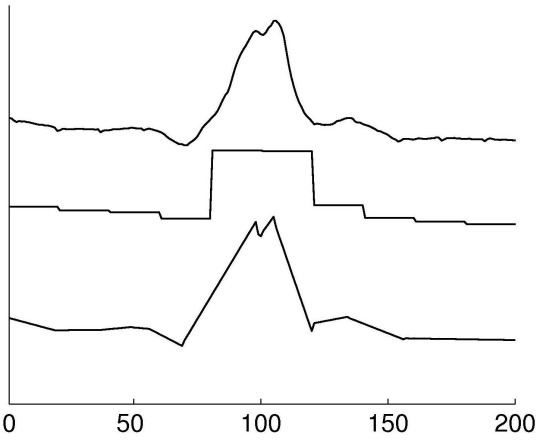

Figure 2: The original signal (top) and its PAA (middle) and PLA representation (bottom) using 10 segments

To correct these faults, more sophisticated methods are applied such as PLA which, however, arise the „segmentation problem", i.e. how to segment a times series quickly and how to represent each segment tight enough.

Segmentation can be framed in several ways [13], but its main goal is always the same: finding homogenous segments by the definition of a cost function, $cost(S_i(a,b))$. This function can be any arbitrary function which projects from the space of the time series to the space of the non-negative real numbers. Usually, $cost(S_i(a,b))$ is based on the distances between the actual values of the time series and the values given by a simple function f (constant or linear function, a polynomial of a higher but limited degree) fitted to the data of each segment:

$$\text{cost}(s_i(a,b)) = \frac{1}{b-a+1} \sum_{l=a}^{b} (d(X_n(l), f(X_n(l)))$$

Thus, the segmentation algorithms simultaneously determine the parameters of the models and the borders of the segments by minimizing the sum of the costs of the individual segments:

$$\text{cost}(s^c_{X_N}) = \min\left(\sum_{i=1}^{c} (\text{cost}(S_i(a,b))) \right)$$

This segmentation cost of a time series can be minimized by dynamic programming, which is

computationally intractable for many real datasets. Consequently, heuristic optimization techniques such as greedy top-down or bottom-up techniques are frequently used to find good but suboptimal *c*-segmentations:

- Bottom-Up: Every element of X_n is handled as a segment. The costs of the adjacent elements are calculated and two elements with the minimum cost are merged. The merging cost calculation of adjacent elements and the merging are continued until some goal is reached.
- Top-Down: The whole X_n is handled as a segment. The costs of every possible split are calculated and the one with the minimum cost is executed. The splitting cost calculation and splitting is continued recursively until some goal is reached.
- Sliding Window: The first segment is started with the first element of X_n. This segment is grown until its cost exceeds a predefined value. The next segment is started at the next element. The process is repeated until the whole time series is segmented.

All of these segmentation methods have their own specific advantages and drawbacks. Accordingly, the sliding window method is not able to divide up a sequence into predefined number of segments but this is the fastest method. The applied method depends on the given task. These heuristic optimization techniques were examined in detail through the application of Piecewise Linear Approximation [13]. It can be said if real-time (on-line) segmentation is not required, the best result will be reached by Bottom-Up segmentation.

While PAA represents an equisized segment with only one value, PLA replaces the original data with not equally sized segments of straight lines, i.e. a PLA segment of x_1 is a 4-taple:

$$\underline{x}_1(i) = \left[x_1(xl)_i, x_1(xr)_i, x_1(yl)_i, x_1(yl)_i\right]$$

Where:

$x_1(xl)_i$ and $x_1(xr)_i$ – left and right time coordinates of the *i*th segment of x_1

$x_1(yl)_i$ and $x_1(yr)_i$ – the values of x_1 in $x_1(xl)_i$ and $x_1(xr)_i$

Finding the optimal PLA a of time series and a suitable distance for this representation is a difficult task, that usually depends on the application. However, precison of the mostly used, traditional Euclidean distance (or any other L_p norm) can be significantly increased with the application of DTW.

Dynamic Time Warping

The traditional comparision approaches are rarely precise enough for the most applications. This is caused by the brittleness of the conventional similarity measures such as Euclidean distance. They are unable to handle the distortions in time axis, so these distortions almost randomly affect the distance between time series. The solution for this problem is the application of DTW which can "warp" the original time series (nonlinearly dilate or compress their time axes) to be similar in shape to the query series as much as possible.

To align two univariate sequences (x_1 and y_1) with DTW, firstly a grid *D* have to be defined with size of the two time series (m_x x m_y). Each cell of this matrix represents the actual distance between the appropriate indices of the two time series. In this step, any application-dependent distance like L_1 and L_∞ norms can be used but generally Euclidean distance is suggested because it allows of the efficient indexing of DTW. Considering this we have:

$$D(i, j) = \sqrt[2]{(x_1(i) - y_1(j))^2}$$

Using grid *D*, many arbitrary mappings – called warping paths – can be created between x_1 and y_1. However, the construction of a warping path $[p(1),p(2),\ldots,p(l)]$ has to be restricted with the following constraints:

- Boundary condition: The path has to start in $D(1,1)$ and end in $D(m_x,m_y)$.
- Monotonicity: The path has to be monotonous, i.e. always heading from $D(1,1)$ to $D(m_x,m_y)$. If $p(k) = D(i,j)$ and $p(k+1) = D(i',j')$ then $i'-i \geq 0$ and $j'-j \geq 0$.
- Continuity: The path has to be continuous. If $p(k) = D(i,j)$ and $p(k+1) = D(i',j')$ then $i'-i \leq 1$ and $j'-j \leq 1$.

To find the optimal warping path (the DTW distance of the two time series), every warping path has an assigned cost which is the sum of values of the affected cells divided by normalization constant *K*:

$$d_{DTW}(x_1, y_1) = \min\left(\frac{\sum_{i=1}^{l} p(i)}{K}\right)$$

The value of *K* depends on the application and in most cases this is the length of the path, but it can also be omitted. More information about the method of defining *K* and its significance can be found in Reference [14]. Note that the Euclidean distance is a special case of DTW, i.e. we choose the path that is located on the diagonal of grid *D* and $K = 1$.

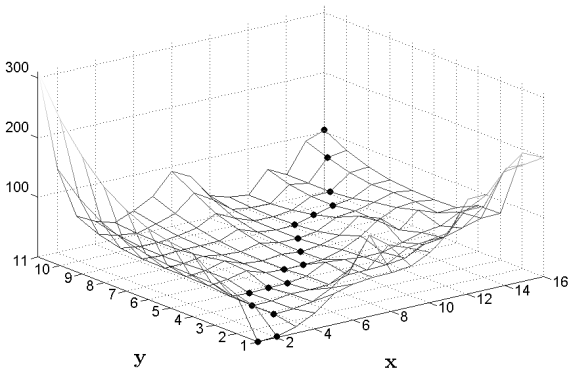

Figure 3: Cumulative distance matrix **D** and the optimal warping path on it

Obviously, the number of paths exponentially grows by the size of time series. Fortunately, the optimal path can be found in $O(m_x m_y)$ time with the help of dynamic programming using cumulative distance matrix **D**. A cell of the cumulative matrix contains the sum of the appropriate cell value in matrix *D* and the minimum of the three cells from where the cell can be reached:

$$\mathrm{D}(i,j) = \min\begin{bmatrix} \mathrm{D}(i-1,j) + D(i,j) \\ \mathrm{D}(i,j-1) + D(i,j) \\ \mathrm{D}(i-1,j-1) + D(i,j) \end{bmatrix}$$

The DTW distance between the two time series can be found in $\mathbf{D}(m_x,m_y)$.

Warping the representations

DTW can be applied easily on the PAA representations of univariate time series. Each segment is represented by its mean, thus the DTW algorithm is the same as for the original time series but the computational time is lowered to $O((m/N)^2)$.

Warping the PLA representation is much more difficult. Vullings et al. [15] were the first to use PLA based DTW on ECG data while Keogh and Pazzani [10] gave a generalized distance between PLA segments which can be used to fill matrix *D*:

$$d(\underline{x}_1(i), \underline{y}_1(j)) = \left(\frac{x(yl)_i + x(yr)_i}{2} - \frac{y(yl)_i + y(yr)_i}{2} \right)^2$$

As it can be seen this distance arises from PAA: it compares the means of the corresponding segments but it utilizes the tighter PLA representation. However, this tighter representation, i.e. the different length of the segments, introduces a new problem. The normalization constant *K* usually based on the length of the warping path but PLA, contrary to PAA, does not generate equidistant segments. Thus, Keogh and Pazzani suggested to recursively sum an additional variable on the warping path which stores the lengths of the visited segments.

PCA driven similarity of PLA representation

As it was mentioned before many problems arise when one would like to use data maning alogrithms on highly correlated mutivariate time series data. The high amount of data requires some reduction techniques. In chemical process engineering, usually PCA is used to create univariate time series from the multivariate data because PCA preserves the correlation of the original time series which is definitely an important factor to consider.

The generated univariate time series make it possible to use traditional segmentation techniques which are often required for two reasons: first, the highly appreciated DTW algorithm computational extensive thus further data compression is advised and second, process transitions and frauds can be revealed by segmentation.

Considering the previously mentioned, PAA is not suitable for our purposes because the segments are equidistant and they are represented by their means, hence PAA representation lost the direction of the latent variable in each segment. PLA generates a much tighter representation and it can preserve the direction information. In addition, it can be seen as the one dimensional version of PCA based segmentation presented in [12,13]. The authors used the Q reconstruction error, i.e. the goal of the segmentation was to minimize the sum of the squared Euclidean distances between the original and the reconstructed variables in each segment. PLA does the same, it minimizes the squared Euclidean distance between the original data point and their reconstructed pair (the closest point on the PLA segment). This reconstruction error is shown in case of PLA segmentation in Figure 4.

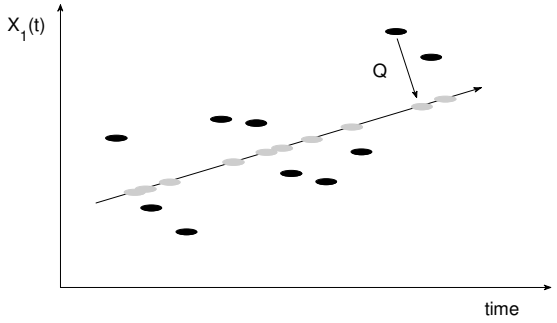

Figure 4: Time series x1 (black dots) and its reconstruction using the PLA representation (gray dots)

Unfortunately, the currently used DTW based measures do not take into account that the PLA created straight line is the latent variable of each segment while the mean of each segment are used. In [12], it is suggested to use any of the PCA based similarity measures to compare the hyperplanes of two segments when the segmentation was driven by PCA. This leads to represent each PLA segment with its angle of slope:

$$\underline{x}_1(i) = \mathrm{atan}\left(\frac{x(yr)_i - x(yl)_i}{x(xr)_i - x(xl)_i} \right)$$

Please note, none of the segments can be perpendicular to the time axis, hence no further restrictions are required. Now, any of the above presented PCA based similarity measures can be used to compare the segments based on the slope information. The interested reader might notice that the only difference between the reviewed PCA based similarity measures how they weights the angles between the hyperplanes. In the one dimensional space all of them can be reduced to the same equation:

$$d(\underline{x}_1(i), \underline{y}_1(i)) = \left| \cos\left(\mathrm{atan}\left(\frac{\underline{x}_1(i) - \underline{y}_1(i)}{1 + \underline{x}_1(i)\underline{y}_1(i)} \right) \right) \right|$$

Where:

$|cos(\ldots)|$ – can be replaced by $\cos^2(\ldots)$

This would be a perfect measure for hyperplanes of a PCA projection, where the hyperplanes are the coordinates of the projected space, thus their signs are not only meaningless but considering them would destroy the precision. However, the PLA representation of a segment of a univariate time series (i.e., its latent variable) is also represents the original time series in the same space, thus the sign information cannot be neglected. For example, if one compares $\underline{x}_1(i)=57$ (~89°) and $\underline{y}_1(i)=-57$ (~91°) as hyperplanes they are almost the same (only 2° is the difference); however, they are completely different from the time series point of view (growing and decreasing trends). So, the basis distance of DTW should be a function which can measure the difference between the signed slopes. For this paper, the simple squared Euclidean distance was chosen:

$$d(\underline{x}_1(i), \underline{y}_1(j)) = (\underline{x}_1(i) - \underline{y}_1(j))^2$$

The last thing to handle is the *K* constant. If it can be treated that preprocessing steps are properly executed (i.e. data is filtered and it is proved that no additional noise is added to it), the role of the *K* constant is changed. The proper preprocessing ensures that all of the segments are detected due to the underlying process and they are not detected due to the noise; moreover, the minimum number of elements of a segment can also be defined. Thus, an additional warping is achieved if the weights of the segments are not considered and the length of the warping path can be used as the value of *K* or it can be omitted as it is done in this paper.

Validation

According to [16], the presented similarity measure was compared against other methods using the free and widely used datasetes and classification algorithm of the UCR Time Series Classification/Clustering Homepage [17]. The datasets were kindly provided by Mr. Keogh on March 7, 2007. Please note, for easier reconstruction of the results, the algorihms were not executed on the four biggest dataset: Face (all), Two Patterns, Wafer, Yoga.

In this test, the suggested 1-NN classification algorithm was executed on the databases and the error rate (proportion of the faulty classified time series) of each database for every measure was recorded in Table 1 and Table 2. As a reference, the non-segmented time series was also compared with Euclidean distance and DTW, their results can be found in the third and fourth columns.

The first four colored columns contain the results of the currently used similarity measures. The time series were segmented with PAA and PLA to 30 pieces using Bottom-Up technique and Euclidean or the reviewed DTW distance was applied. Please note, this means that the mean of the segments were used to compare two segments irrespectively of the applied segmentation method.

Table 1: Results of the UCR time series classification algorithm on the first eight datasets

Name	Number of classes	Time Series Length	Euclidean Distance	No Warping Window DTW	PAA 30 Segments Euclidean	PAA 30 Segments DTW	PLA 30 Segments Euclidean	PLA 30 Segments DTW	Angle 30 Segments Euclidean	Angle 30 Segments DTW
50Words	50	270	0.37	0.31	0.36	0.36	0.35	0.40	0.46	0.37
Adiac	37	176	0.39	0.40	0.41	0.43	0.76	0.66	0.71	0.64
Beef	5	470	0.47	0.50	0.50	0.53	0.53	0.50	0.67	0.63
CBF	3	128	0.15	0.00	0.07	0.02	0.13	0.04	0.64	0.59
Coffee	2	286	0.25	0.18	0.25	0.25	0.18	0.29	0.32	0.11
ECG	2	96	0.12	0.23	0.13	0.21	0.20	0.20	0.23	0.19
Face (four)	4	350	0.22	0.17	0.17	0.23	0.39	0.30	0.72	0.39
Fish	7	463	0.22	0.17	0.21	0.22	0.51	0.38	0.46	0.15

Table 1: Results of the UCR time series classification algorithm on the second eight datasets

Name	Number of classes	Time Series Length	Euclidean Distance	No Warping Window DTW	PAA 30 Segments Euclidean	PAA 30 Segments DTW	PLA 30 Segments Euclidean	PLA 30 Segments DTW	Angle 30 Segments Euclidean	Angle 30 Segments DTW
Gun-Point	2	150	0.09	0.09	0.09	0.07	0.07	0.07	0.05	0.05
Lightning-2	2	637	0.25	0.13	0.25	0.26	0.31	0.20	0.39	0.44
Lightning-7	7	319	0.43	0.27	0.34	0.30	0.47	0.44	0.78	0.67
OliveOil	4	570	0.13	0.13	0.13	0.13	0.33	0.30	0.40	0.37
Osu Leaf	6	427	0.48	0.41	0.47	0.42	0.52	0.44	0.70	0.29
Swedish Leaf	15	128	0.21	0.21	0.21	0.21	0.44	0.28	0.51	0.22
Synthetic Control	6	60	0.12	0.01	0.01	0.04	0.01	0.01	0.67	0.67
Trace	4	275	0.24	0.00	0.29	0.10	0.19	0.03	0.17	0.01

'Angle' denotes the presented similarity measure. The time series were segmented to 30 pieces again but now the slope of each segment was considered. The provided representations were also compared with Euclidean distance and DTW. The best results are marked with dark grey for each database.

As it can be seen the newly presented method kept up with the expectations. However, some notes have to be made on the results. One can realize that the reduction obtained by the segmentation was only about one order of magnitude which ensures the tight representation even with PAA and 30 segments is too much for most databases when PLA is used. Moreover, all of the databases require different number of segments to get the best results.

The aim of this test was to show that there is really make sense to use the slope of a segment instead of its mean when the segment is provided by PLA. Obviously, the best PAA and PLA representation can be found for all distance measure but the idea behind the creation of such a test is to provide a unified test environment for all measures and to prevent the researcher from "over optimization".

Validation on an industrial fixed bed tube reactor

The proposed PCA driven similaity measure has also been applied for clustering of temperature data from a sophisticated model of an industrial catalytic fixed bed tube reactor.

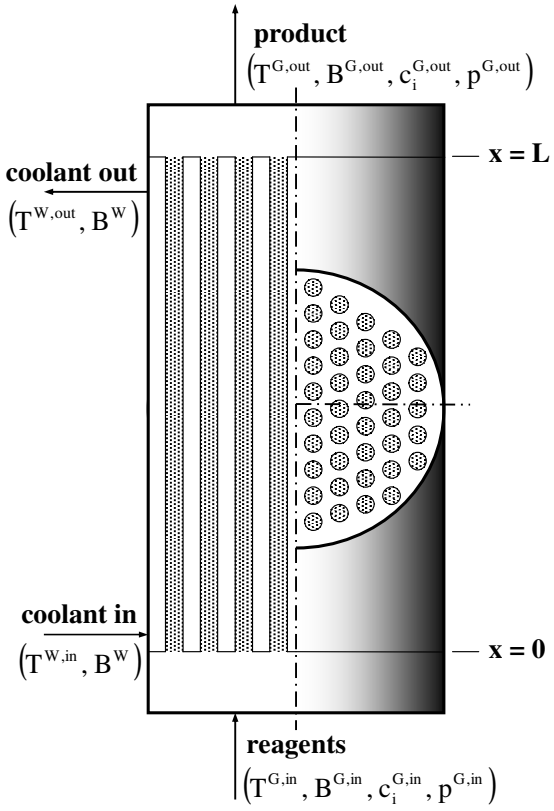

Figure 5: The simplified scheme of the studied reactor

The studied vertically built up reactor contains a great number of tubes with catalyst. Highly exothermic reaction occurs as the reactants rising up the tubes pass the fixed bed of catalyst particles and the heat generated by the reaction escapes through the tube walls into the cooling water. Due to this highly exothermic reaction which takes place in the catalyst bed the reactor very sensitive for the development of reactor runaway.

Reactor runaway means a sudden and considerable change in the process variables. The development of runaway is in very close relationship with the stability of reactor/model. Runaway has two main important aspects. In one hand runaway forecast has a safety aspect, since it is important for avoiding the damage the constructional material or in the worst case scenario the explosion of reactor. On the other hand, runaway has a technology aspect, since the forecast of the runaway can be used for avoiding the development of hot spots in catalytic bed. The selection of operation conditions is important to avoid the development of reactor runaway and to increase the lifetime of catalyst at same time. The worked out mathematical model of the studied reactor has been presented in Reference [5]. The model has been implemented in MATLAB and solved with a low order Runge-Kutta method.

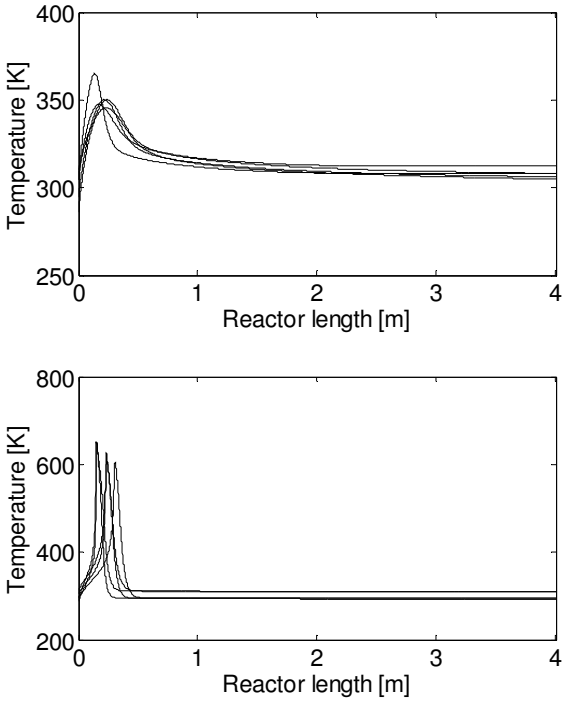

Figure 6: Normal (upper graph) and runaway (lower graph) profiles of the studied reactor

The obtained simulator was applied to calculate two kinds of profiles. The inlet conditions was set to provide five profiles which correspond to the normal working conditions and other five profiles which describe the development of reactor runaway. The presented similarity measure has been used to classify these profiles. The result of clustering can be seen in the dendogram of Figure 7.

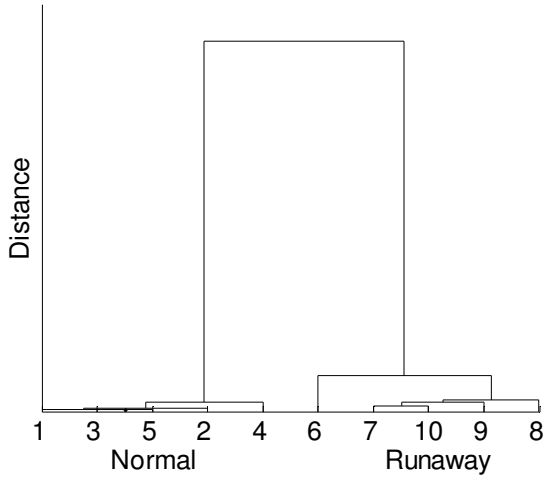

Figure 7: Clustering result of the five Normal (number 1-5) and five Runaway (number 6-10) profiles

Conclusion

We presented a new similarity measure for segmented univariate time series by considering the PLA representation of a segment as the latent variable. The proposed similarity was validated on many real world dataset used by data mining community and on the temperature profiles generated by a sophisticated model of an industrial reactor. Both evaluations show that worth to consider representing the segments by their slopes instead of their means and using this feature for the comparison. However, there is no decided difference between the two methods, thus we intend to combine these approaches in the future.

Acknowledgement

János Abonyi is grateful for the support of the Bolyai Research Fellowship of the Hungarian Academy of Sciences. The financial support of the TÁMOP-4.2.2-08/1/2008-0018 project is gratefully acknowledged.

REFERENCES

1. T. KOURTI: Application of Latent Variable Methods to Process Control and Multivariate Statistical Process Control in Industry, International Journal of Adaptive Control and Signal Processing 19 (2005), p. 213-246
2. W. KRZANOWSKI: Between-groups comparison of principal components, Journal of the American Statistical Society 74 (1979), p. 703-707
3. M. C. JOHANNESMEYER: Abnormal Situation Analysis Using Pattern Recognition Techniques and Historical Data, M.Sc. Thesis, University of California, Santa Barbara, CA (1999)
4. K. YANG, C. SHAHABI: A PCA-based Similarity Measure for Multivariate Time Series, Proceedings of the 2nd ACM International Workshop on Multimedia Databases (2004), p. 65–74
5. S. KIVIKUNNAS: Overview of Process Trend Analysis Methods and Applications, ERUDIT Workshop on Applications in Pulp and Paper Industry (1998), ERUDIT
6. H. SAKOE, S. CHIBA.: Dynamic Programming Algorithm Optimization for Spoken Word Recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing (1978)
7. Z. M. K. VAJNA: A Fingerprint Verification System Based on Triangular Matching and Dynamic Time Warping, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000), p. 1266-1276
8. A. KHOLMATOV, B. A. YANIKOGLU: Biometric Authentication Using Online Signatures, ISCIS (2004)
9. E. J. KEOGH, M. J. PAZZANI: Scaling up Dynamic Time Warping for Datamining Applications, Knowledge Discovery and Data Mining (2000)
10. E. J. KEOGH, M. PAZZANI: Scaling up Dynamic Time Warping to Massive Datasets, 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases (1999) vol. 1704
11. J. ABONYI, B. FEIL, S. NÉMETH, P. ÁRVA: Principal Component Analysis Based Time Series Segmentation, IEEE International Conference on Computational Cybernetics (2005)
12. Z. BANKÓ, J. ABONYI: Correlation Based Dynamic Time Warping of Multivariate Time Series, Computational Statistics & Data Analysis (In press)
13. E. J. KEOGH, S. CHU, D. HART, M. J. PAZZANI: An Online Algorithm for Segmenting Time Series, ICDM (2001)
14. H. SAKOE, S. CHIBA: Dynamic Programming Algorithm Optimization for Spoken Word Recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing (1978)
15. H. J. L. M. VULLINGS, M. H. G. VERHAEGEN, H. B. VERBRUGGEN: ECG Segmentation Using Time-Warping, Proceedings of the Second International Symposium on Advances in Intelligent Data Analysis, Reasoning about Data (1997)
16. E. J. KEOGH, S. KASETTY: On the Need for Time Series Data Mining Benchmarks: a Survey and Empirical Demonstration, Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2002)
17. E. J. KEOGH, X. XI, L. WEI, C. A. RATANAMAHATANA: The UCR Time Series Classification/Clustering Homepage, http://www.cs.ucr.edu/~eamonn/time_series_data/, Riverside CA. University of California - Computer Science and Engineering Department (2006)
18. T. VARGA, F. SZEIFERT, J. RÉTI, J. ABONYI: Analysis of the runaway in an industrial heterocatalytic reactor, Computer-Aided Chemical Engineering 24, (2007), p 751-756