A weighted distance for short time series

Agostino Tarsitano

Abstract Time-varying data have often been analyzed by methods that do not explicitly take into account time related changes. To overcome this problem, we segmented a time series in many components, each of which yields a separate unit of information that can act on a stand-alone basis for the computation of the distance function. As a consequence, the comparison of two time series becomes a comparison of two sets of matrices. To simplify the comparison, the procedure Distatis can be very helpful because it yields a compromise distance matrix that can be analyzed by using the usual techniques of cluster analysis.

1 Introduction

Data consisting of a sequence of observations collected over time are common and often unavoidable in many studies. In some cases, the sequences are fairly short (< 20 time points) because the process of gathering data and delivering them to users takes long time and/or is expensive. Time-varying data have often been analyzed by methods that either do not explicitly take into account time related changes or that are designed for long time series. To overcome this problem, a reasonable approach is to extract both slowly and quickly varying features thus providing a comprehensive view on the various aspects of the evolutive behavior of a time series. To this end, we developed a decomposition technique for identifying periodic patterns in relatively short time series. Our method is based on the observation that the trajectory of any time series can be segmented in many submovements according to position, velocity, acceleration, jerk, and higher derivatives each of which

Agostino Tarsitano
Dipartimento di Economia e Statistica, Università della Calabria
Via Pietro Bucci, cubo 1C, Rende (CS) - Italy
Tel.: +39-0984-492465, Fax:+39-0984-492421
e-mail: agotar@unical.it

provides a separate piece of information that can act as a stand-alone basis for the computation of the distance function. As a consequence, the comparison of two time series can be considered a comparison of two sets of matrices. To simplify the comparison, the procedure Distatis (see [1]) can be very helpful because it yields a compromise distance matrix that considers both the differences between the values and temporal information; moreover, it works irrespective of different baselines and scaling factors. The final distance matrix can be analyzed by using the usual technique of cluster analysis. In Section 2, we will propose a distance function that takes into account both the observed values and the proximity between the temporal behaviors of the time series. In Section 3, the procedure is assessed by a Monte Carlo simulation study. Conclusions and future research are then presented in Section 4.

2 A new weighted metric for short time series

The simplest distance function between two time series can be expressed by

$$M_k(\mathbf{X}_i, \mathbf{X}_j)^2 = \sum_{t=k+1}^{n} (x_{i,t} - x_{j,t})^2 \tag{1}$$

where k has to be given. Formula (1) is easy to compute and interpret, but it ignores the temporal structure of the values. In this sense, [2, 3] added to $M_0(\mathbf{X}, \mathbf{Y})$ the distance between the first differences (velocities) and the distance between the second differences (accelerations). This idea can be further extended if we incorporate knowledge about jerks (third differences) and higher differences

$$V_a(\mathbf{X}_i, \mathbf{X}_j) = \sum_{k=1}^{q} w_q \left\{ \frac{M_k}{M_{max}^{(k)}} \right\} \qquad \text{for} \quad i, j = 1, 2, \cdots, p \tag{2}$$

where $\mathbf{M}_k = M_k \left[\mathbf{X}_i^{(k)}, \mathbf{X}_j^{(k)} \right]$ for $k = 0, 1, \cdots, q$ is the matrix of the distances (1) for the original values ($k = 0$), first differences ($k = 1$) of two consecutive time points, second difference ($k = 2$), and so on until $k = q = \lceil (n-1)/2 \rceil$ (a bound is necessary to compute differences on enough data). The weights $\mathbf{w}$ are non negative and sum to one. $M_{max}^{(k)}$ is the distance of the furthest pair of sequences; thus, V_q has values between zero and one. The M_k's are, by construction, symmetric and non negative. The Distatis algorithm transforms the partial distance matrices into cross-product matrices, compute eingenvalues and eigenvectors of each one and then aggregate them to obtain a compromise matrix. Let $\mathbf{u}$ be an q x 1 vector of 1' and $\mathbf{I}$ the identity matrix of order q. Furthermore, suppose that $\mathbf{B}$ is the matrix of (1) for the p time series. A normalized cross-product matrix for $\mathbf{B}$ is determined as $\mathbf{T}_k = \lambda_{1,k}^{-1} \mathbf{S}_k$ where $\lambda_{1,k} > 0$ is the largest eigenvalue of $\mathbf{S}_k = \mathbf{CBC}^t$ where $\mathbf{C} = \mathbf{I} - q^{-1}\mathbf{u}\mathbf{u}^t$ is a centering matrix. The cross-product matrices are first vectorized $\tau_k = vec(\mathbf{T}_k), k = 1, 2, \cdots, q$ and then collected in the form of a p^2 x 1 matrix $\mathbf{Z} = [\tau_1, \tau_2, \cdots, \tau_k]$.

The central step is to create the aggregate matrix $\mathbf{A} = \left(\mathbf{N}^{-0.5}\mathbf{Z}\right)^t \left(\mathbf{Z}\mathbf{N}^{-0.5}\right)$ where $\mathbf{N}$ is a diagonal matrix whose elements are the diagonal elements of $\left(\mathbf{Z}^t\mathbf{Z}\right)$ and the generic element $a_{i,j}$ is the vectorial correlation coefficient between the matrix of changes derived form series i and the matrix of changes derived from series j ($a_{i,j} > 0$ because $\mathbf{B}$ is Euclidean). The final step of Distatis is the eigen-decomposition of $\mathbf{A}$ that gives $\mathbf{A} = \mathbf{P}\Lambda\mathbf{P}^t$ with $\mathbf{P}^t\mathbf{P} = \mathbf{I}$. Since $\mathbf{A}$ is positive then one of its eigenvalues, say μ_1, is positive and greater than or equal to all other eigenvalues and there a positive eigenvector $\mathbf{p}$ corresponding to μ_1. The compromise matrix is given by

$$\mathbf{T} = \sum_{k=1}^{q} w_k \mathbf{T}_k \quad \text{where} \quad \mathbf{w} = \left(\frac{1}{\mathbf{u}^t\mathbf{p}}\right)\mathbf{p} \tag{3}$$

The related compromise distance matrix is $\mathbf{D} = \Psi\mathbf{u}^t + \mathbf{u}\Psi^t - 2\mathbf{T}$ with $\Psi = diag\,(\mathbf{T})$. This matrix is the basis of clustering and classification methods based on distance matrix such as hierarchical methods or partitioning around medoids.

3 Evaluation on artificial data

We have evaluated the performance of the proposed technique by considering artificially generated data. In particular, our data sets were created using a random time series generator that produces $p = 48$ sequences belonging to a predefined number $k \in K = \{4,8,12\}$ clusters $C_1, C_2, \cdots, C_k$ of length $n \in N = \{11,17,23\}$. Each cluster includes a number of $48/k$ sequences given by the following expressions

$$\begin{aligned}
1,2 &: \mathbf{f}\sin\left[\pi\left(2\mathbf{t} + \delta 0.5\right)\right] + \mathbf{e} & 3,4 &: \mathbf{f}\sin\left[\pi\left(2\mathbf{t} + \delta/2 - 0.25\right)\right] + \mathbf{e} \\
5,6 &: \mathbf{f}\sin\left[\pi\left(4\mathbf{t} + \delta 0.5\right)\right] + \mathbf{e} & 7,8 &: \mathbf{f}\sin\left(2\pi\mathbf{t} - \mathbf{t} + \delta 0.5\right) + \mathbf{e} \\
9,10 &: \mathbf{f}\sin\left(4\pi\mathbf{t} - \mathbf{t} + \delta 0.5\ \right) + \mathbf{e} & 11,12 &: \mathbf{f}\sin\left[\pi\left(4\mathbf{t} + \delta/2 - 0.25\right)\right] + \mathbf{e}
\end{aligned}$$

where $\delta = 0$ for odd formulae otherwise $\delta = 1$; $t \in [-0.4, 0.4, by\ 0.8/(n-1)]$; $\mathbf{f} = (\phi_1\mathbf{t} + \phi_2)$. The parameter ϕ_1 is obtained randomly from $U(0.1, 0.5)$, ϕ_2 randomly from $U(1.1, 1.5)$, and $\mathbf{e}$ is a vector of random errors form $N(0, 1.25)$. All the quantities ϕ_1, ϕ_2, and $\mathbf{e}$ varies across time series.

To compare the stability of the algorithm outlined in Section (2), we repeated the data generation 1000 times for each k and for each n. The Classification Error Rate (CER) has been used as cluster validation measure of the PAM algorithm available in *R*. The numerical summaries are given in Table 1. The third column reports the number of times the Euclidean distance did not yield the better outcome. The other columns provide the average and the standard deviation of CER for various values of q in (3). A low average implies that clusters determined by the algorithm are in strict agreement with clusters from the original data. A low standard deviation indicates stability across the simulations. The following facts emerge from Table 1. The mean CER and the standard deviation of CER decrease as the length of the sequences increases.

Table 1 Summary of CER from 1000 simulations

k	n	ENB		Euc.	$q=1$	2	3	4	5	6	7	8	9	10
4	11	93	μ	0.08	0.20	0.18	0.15	0.13						
	11	93	σ	0.05	0.08	0.08	0.07	0.07						
	17	42	μ	0.01	0.11	0.11	0.08	0.06	0.05	0.03	0.03			
	17	42	σ	0.02	0.07	0.07	0.05	0.04	0.03	0.03	0.03			
	23	24	μ	0.00	0.05	0.08	0.05	0.03	0.02	0.02	0.01	0.01	0.01	0.01
	23	24	σ	0.01	0.05	0.06	0.04	0.03	0.02	0.02	0.02	0.02	0.01	0.01
8	11	202	μ	0.26	0.32	0.32	0.30	0.29						
	11	202	σ	0.04	0.05	0.05	0.05	0.05						
	17	207	μ	0.22	0.29	0.30	0.29	0.27	0.25	0.25	0.24			
	17	207	σ	0.03	0.05	0.05	0.05	0.05	0.05	0.04	0.04			
	23	164	μ	0.22	0.28	0.30	0.29	0.27	0.26	0.25	0.24	0.23	0.23	0.23
	23	164	σ	0.03	0.05	0.05	0.05	0.05	0.05	0.04	0.04	0.03	0.03	0.03
12	11	310	μ	0.33	0.38	0.37	0.35	0.34						
	11	310	σ	0.05	0.05	0.05	0.05	0.05						
	17	207	μ	0.27	0.35	0.35	0.33	0.32	0.30	0.29	0.29			
	17	207	σ	0.04	0.05	0.05	0.04	0.04	0.04	0.04	0.04			
	23	153	μ	0.25	0.34	0.34	0.33	0.32	0.31	0.30	0.29	0.28	0.28	0.27
	23	153	σ	0.04	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.03

This is in line with the idea that longer sequences allow progressively more detailed description of the underlying dynamics. The mean CER increases and the standard deviation of CER increases as the number of clusters increases. Such a result should not be considered as a surprising one; in fact, more groups imply higher probability of misclassification. The Euclidean distance becomes increasingly inadequate as the number of clusters increases and/or the length of the sequences decreases.

4 Conclusion

The Euclidean distance, although theoretically inappropriate, performs better than expected even with time series data. A slight improvement can sometimes be achieved by adding as many terms as possible in (2) (compatibly with the available computational resources). Further research is necessary to establish the appropriate number of differences which is advantageous and to find out the cases in which additional terms would determine a better classification.

References

1. Abdi, H. and O'Toole, A. J. and Valentin, D. and Edelman, B: DISTATIS: The Analysis of Multiple Distance Matrices. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 42–47, San Diego, CA, USA, (2005).
2. D'Urso, P. and Vichi, M.: Dissimilarities between trajectories of a three-way longitudinal data set. In A. Rizzi, M. Vichi and H.-H. Bock (Eds.) *Advances in data science and classification*, 585–592, Springer, Berlin, (1998).
3. D'Urso, P. : Dissimilarity measures for time trajectories, *Journal of the Italian Statistical Society*, **9**,53–83, (2000).