SIMILARITY MEASURE FOR HETEROGENEOUS MULTIVARIATE TIME-SERIES

Florence Duchêne[1], *Catherine Garbay*[1], *and Vincent Rialle*[1,2]

[1]Laboratory TIMC-IMAG, Faculté de médecine de Grenoble
38706 La Tronche, France (Europe)
phone: +33 4 56 52 00 71, fax: +33 4 56 52 00 22, email: Florence.Duchene@imag.fr
[2]Department of Medical Informatics (SIIM)
Michallon hospital, Grenoble, France (Europe)

ABSTRACT

Defining the similarity of objects is crucial in any data analysis and decision-making process. For those which effectively deal with moving objects, the main issue becomes the comparison of trajectories, also referred to as time-series. Moreover, complex applications may require an object to be a multidimensional vector of heterogeneous parameters. In that paper, we propose a similarity measure for heterogeneous multivariate time-series using a non-metric distance based on the *Longest Common Subsequence (LCSS)*. The proposed definition allows for imprecise matches, outliers, stretching and global translating of the sequences in time. We demonstrate the relevance of our approach in the context of identifying similar behaviors of a person at home.

1. INTRODUCTION

Measuring the similarity and dissimilarity between objects is crucial in any data analysis and decision-making process. Furthermore, many data analysis processes effectively deal with moving objects and need to compute the similarity between trajectories, also referred to as time-series.

In this work, we investigate the problem of defining a similarity measure of heterogeneous multivariate time-series. When dealing with complex issues involving the analysis of data recorded from several types of sensors or information sources, like most monitoring purposes, an object may be a multidimensional vector of heterogeneous parameters. One application is the monitoring of the health status of a person at home. The aim is to support the caregivers by providing information about unusual trends in the person's behavior. The decision-making process must then be able to recognize similar behaviors through the variation of quantitative or qualitative parameters monitored at home. Therefore, the similarity model should allow for heterogeneous components defining an object, as well as for imprecise matches, outliers, stretching and global translating of the sequences in time.

The rest of the paper is organized as follows. In section 2 we present related works. In section 3 we formalize the similarity measure. Section 4 provides the experimental validation of the proposed approach in the context of home health telecare. Finally, section 5 conludes the paper.

2. RELATED WORKS

The simplest approach typically used to define a similarity function is based on the Euclidian distance, or some extensions to support various transformations such as scaling or shifting. Chui *et al.* [3] have used it successfully for extracting one-dimensional time-series motifs in some specific cases. However, this model cannot deal with outliers and is very sensitive to small distorsions in the time axis.

Another approach is to use the *Dynamic Time Warping (DTW)* distance which allows stretching in time and comparing time-series of different lengths [7, 8]. However, a great amount of outliers still results in very large distances, even though the difference may be found in only a few points.

Non-metric techniques have then been introduced and efficiently used to better deal with noisy data [1, 4, 9]. The idea is to capture the intuitive notion that "two sequences should be considered similar if they have enough non-overlapping time-ordered pairs of subsequences that are similar" [1]. This refers to finding the *Longest Common Subsequence (LCSS)* between two time-series. This approach allows for outliers, different scaling factors, and baselines.

However, the above works mainly deal with low dimensional (from one to three dimensional) time-series and do not address the issue of heterogeneous components (quantitative or qualitative) describing a moving object. Our objective is then to extend the *LCSS* approach to heterogeneous multivariate time-series. For the purpose of evaluation, we compare the performance of *LCSS* to the use of *DTW* distances.

3. SIMILARITY MODEL

3.1 Guidelines

Comparing heterogeneous multivariate time-series, and especially time-series representative of human behaviors, we need a similarity model that can address the following issues:

- **Multivariate time-series.** Relevance for comparing moving objects described by several parameters.
- **Heterogeneous components.** Coherence of the similarity model for qualitative and quantitative parameters.
- **Imprecise matches.** Strong presence of noise, especially when considering human behaviors.
- **Outliers.** Might be introduced due to anomaly in the sensor or attributed to human failure or disruption.
- **Translation in time.** Similar behaviors may occur at any time.
- **Streching in time.** Different lengths allowed: dealing with human behaviors, a same activity does not always last the same duration.
- **Efficiency.** Efficient computation of the similarity.

The similarity model relevant to cope with these challenges is based on the *LCSS*. Indeed, dealing with noisy data have proven to be better handled using non-metric, based on the *LCSS*, than metric distances. Part of the variability in the values might however be removed by filtering the raw data, allowing to compute more accurate similarity measures from the pre-processed sequences. However, some other noise like large sequences of outliers cannot be handled *a priori* by any pre-processing. Because previous works about comparing time-series only concern quantitative data, another crucial point is to define a coherent model for expressing distances including both qualitative and quantitative parameters.

Computing distances between trajectories includes evaluating the distance between points. This distance is either integrated in the whole distance formula between trajectories in the case of a metric distance, or used to decide whether two points are similar using a threshold in the case of a non-metric distance. The next sections describe first the distance between points according to the type of parameter, and then its integration in computing the distance between trajectories.

3.2 Distance between points

We would like to allow the description of an object using several parameters of the following possible types:

- Quantitative
- Ordered qualitative
- Unordered qualitative

The simplest way of insuring the coherence of the similarity measure is to make the distances between two values range from 0 to 1 for each type of parameters. Let a and b be two values of a given parameter, and $d(a,b)$ the distance between these two values. In case of a qualitative parameter, let v be the number of variates, the possible values being then the integers from 1 to v. According to the parameter's type, $d(a,b)$ is defined as follows:

$$d(a,b) = |a-b|, \tag{1}$$

$$d(a,b) = \frac{|a-b|}{v-1}, \tag{2}$$

$$d(a,b) = min(|a-b|, 1). \tag{3}$$

The equations (2) and (3) are used respectively for ordered and unordered qualitative parameters. In the case (1) of quantitative parameters, getting a distance between 0 and 1 requires a step of normalization so that the possible values range from 0 to 1. We use a min-max normalization, where the minimum and maximum bounds are defined from experts or using statistical analysis of training sets. All values are then restricted to these bounds, lower and upper values being interpreted as noisy or eroneous. Let X_{min} and X_{max} be respectively the minimum and maximum bounds for the values x of a given parameter X. We define the normalized value $norm(x)$ of x as follows:

$$norm(x) = \frac{max\left(0, min\left(x, X_{max}\right) - X_{min}\right)}{X_{max} - X_{min}}$$

3.3 Distance between trajectories

The similarity function between trajectories is based on the *LCSS*, already used by Vlachos *et al.* [9] in the context of multidimensional (generally two or three dimensional) time-series of quantitative data. The overall idea is to count the number of couple of points from two sequences A and B that matches according to a pre-defined matching threshold ε, and when going through the temporal sequences (see Fig. 1). One point can never be associated twice to a point of the other sequence, so that the maximum number of associations is the minimum length of the two sequences. Another constant δ controls how far in time we can go in order to match points from one trajectory to the other one (see Fig. 2).

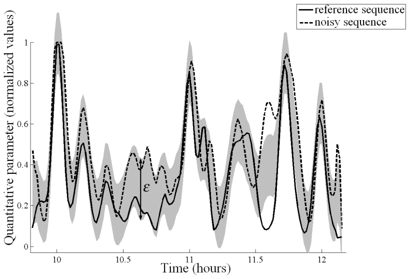

Figure 1: The notion of the *LCSS* matching within a region of ε. Comparing the trajectories point to point along the time axis, the pairs both within the gray region can be matched.

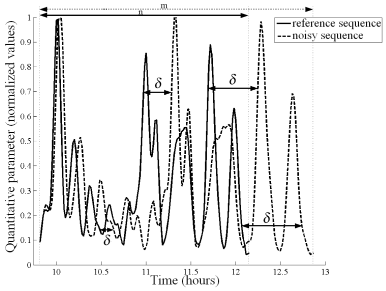

Figure 2: The notion of the *LCSS* matching within a region of δ. The points of two trajectories can be matched if the time interval is under the maximum authorized value for δ.

We assume objects are points moving in a p-dimensional space $(x_1, \ldots, x_p)$. Let $A = ((a_{x_1,1}, \ldots, a_{x_p,1}), \ldots, (a_{x_1,n}, \ldots, a_{x_p,n}))$ and $B = ((b_{x_1,1}, \ldots, b_{x_p,1}), \ldots, (b_{x_1,m}, \ldots, b_{x_p,m}))$ be the two trajectories of moving objects with size n and m respectively. For a trajectory A, let $Head(A)$ be the sequence: $Head(A) = ((a_{x_1,1}, \ldots, a_{x_p,1}), \ldots, (a_{x_1,n-1}, \ldots, a_{x_p,n-1}))$. Given an integer δ and a real number $0 < \varepsilon < 1$, the similarity function $LCSS_{\delta,\varepsilon}(A,B)$ is defined as follows [9]:

$$\begin{cases} 0 \quad \text{if } A \text{ or } B \text{ is empty}, \\ 1 + LCSS_{\delta,\varepsilon}(Head(A), Head(B)), \\ \qquad \text{if } \left|a_{x_k,n} - b_{x_k,m}\right| < \varepsilon, \forall 1 \le k \le p, \text{ and } |n-m| \le \delta, \\ max\left(LCSS_{\delta,\varepsilon}(Head(A), B), LCSS_{\delta,\varepsilon}(A, Head(B))\right) \\ \qquad \text{otherwise}. \end{cases} \tag{4}$$

The number of matching is normalized by the minimum length of the two trajectories, so that the similarity measure range from 0 to 1. Therefore the function $D_{\delta,\varepsilon}(A,B)$ between the two trajectories A and B is defined as follows [9]:

$$D_{\delta,\varepsilon}(A,B) = 1 - \frac{LCSS_{\delta,\varepsilon}(A,B)}{min(n,m)}.$$

$D_{\delta,\varepsilon}(A,B)$ verifies the properties of a distance.

An additional time constraint is however required to better deal with the case where every point of the shortest sequence match a point of the longest one, with no overlapping, and in time-ordered. A typical sample is the shortest sequence corresponding exactly to the beginning of the longest one. According to the previous definition, the similarity is then equal to 1, whatever the length of the longest sequence. Let N and M be the size of the sequences A and B respectively at the first step of the recurrent algorithm (4). To prevent from that kind of improper high similarity, we define an additional time constraint for the similarity of two points $(a_{x_1,n},\dots,a_{x_p,n})$ and $(b_{x_1,m},\dots,b_{x_p,m})$, as follows:

$$|n-m| \leq \delta \text{ and } |N-n-M+m| \leq \delta. \qquad (5)$$

Furthermore, we need to extend the similarity constraints to the case of qualitative parameters. The idea is to consider that two values of a qualitative parameter are similar only if they are equal. Therefore, we have defined a relevant ε value according to the parameter's type, as follows:

- Quantitative $0 < \varepsilon < 1$,
- Ordered qualitative $\varepsilon = \frac{1}{v-1}$,
- Unordered qualitative $\varepsilon = 1$.

The constraint on values for the similarity is then described using $d\left(a_{x_k,n}, b_{x_k,m}\right)$ (cf. 3.2) as follows:

$$d\left(a_{x_k,n}, b_{x_k,m}\right) < \varepsilon,\ \forall 1 \leq k \leq p.$$

3.4 Similarity computation

To compute the distance between trajectories, we have to run a *LCSS* computation. Most algorithms for finding the *LCSS* have their natural predecessors in either Hunt and Szymanski [6], or Hirschberg [5]. We use a variant of [5] proposed by Apostolico [2]. The running time is also improved by examining only the pairs of points verifying the time constraints described in (5). Then, if δ is small, the algorithm is very efficient. Another way of speeding-up the computation is to reduce the sampling rate. We show in the next section the influence of pre-processing the data.

4. EXPERIMENTAL RESULTS

4.1 Experimental process

The approach defined for computing the distance between trajectories is validated in the context of home health telecare, and especially under the strong presence of noise. A moving object corresponds to the evolution of various parameters representative of the a person's health status. We consider the following heterogeneous parameters that can be defined from a provision of sensors installed in the home:

- **Moves** (room occupied). Qualitative, unordered.
- **Postures**. Qualitative, ordered according to the effort required by the posture (lying down, sitting, and standing);
- **Activity levels**. Quantitative, in an arbitrary unit (representative of the body acceleration);
- **Mean heart rate**. Quantitative, in beat per minute.

Our objective is to identify a sort of profile of a person — their usual behaviors — from the global trends of these parameters, and then to detect any deviation from this profile. The distance between time-series is then expected to generate low values between sequences corresponding to the realisation of a same activity in same conditions, and higher values otherwise. Two experimental sets consist of:

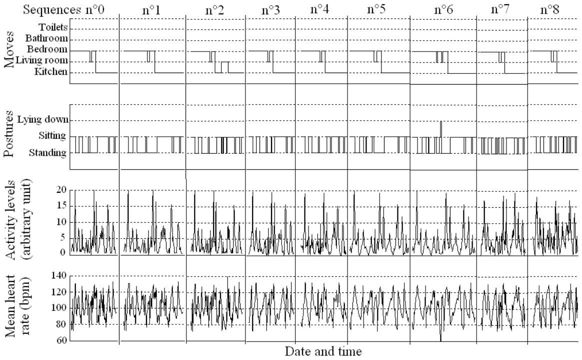

Figure 3: Sequences close to sequence 0 (class 0).

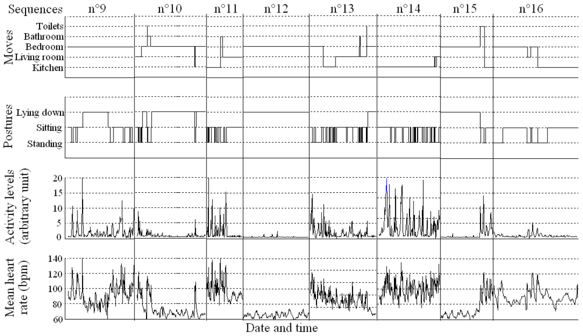

Figure 4: Sequences far from sequence 0 (class 1).

(1) Sequences 0 to 8 representative of a given activity — getting ready in the morning (see Fig. 3), generated from a reference sequence (sequence 0) by adding noise of three types: streching in time, variability in values, interruptions (consecutive outliers).

(2) Sequences 9 to 16 representative of other activities like sleeping, having a meal, having a quiet activity (see Fig. 4), including one sequence (sequence 16) corresponding to the reference activity (same moves) but carried out in bad conditions (slowness). This abnormal behavior may be detected if sequence 16 is not considered as representative of sequence 0.

The experimental process aims at classifying these sequences using a threshold on the distance to sequence 0. An appropriate distance may be able to properly discriminate the sequences: 1 to 8 associated to class 0, and 9 to 16 to class 1. We use both *DTW* and *LCSS* distances for comparison (see [7] for a clear review of *DTW* principle), and in each case the distances are computed from both raw and pre-processed data — that is sampling rate reduction to speed-up the computation, and filtering to remove some noise. Preliminary experimentations were required to define relevant values for the *LCSS* parameters (ε,δ) in the context of our application.

4.2 Discussion about the results

The classification results are presented on Fig. 5. As a general comment, we notice that *DTW* distances are really lower than *LCSS* ones, due (1) to different orders of computation — 1 for *LCSS* and 2 for *DTW*, and (2) to possible multiple associations of any point using *DTW*, so that the distance may

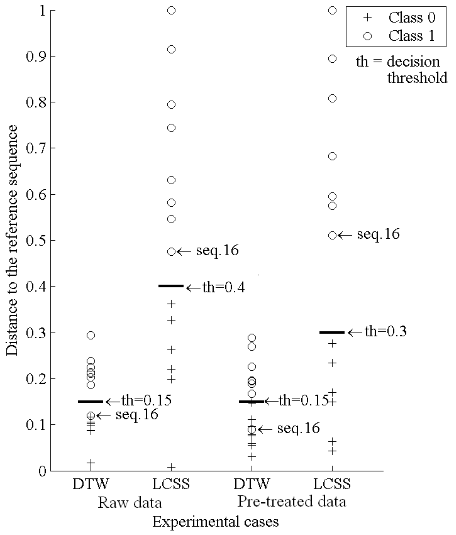

Figure 5: Distances between sequence 0 and the other experimental sequences, from either raw or pre-processed data, and using *DTW* and *LCSS* distances. Classes 0 and 1 correspond to the expected classification.

remain quite low.

The superiority of *LCSS* over *DTW* is pointed out by the results matching the expected classification only in the case of using *LCSS*. Using *DTW* distance fails in properly classifying sequence 16. The behavior of both *LCSS* — δ set with no restriction in time for associating points, as it is using *DTW* — and *DTW* when comparing sequences 0 and 16 is illustrated on Fig. 6. *DTW* allows for multiple associations, and all points must be matched, based on a minimum distance criterion. Then, because the sequences of moves and postures are very close, the poor number of points corresponding to low activity levels and mean heart rate in sequence 0 are associated to the large number of such points in sequence 16, and reciprocally for high values of activity levels and mean heart rate. That results in a low number of pairs corresponding to large distances, so that the distance between the two sequences remains low. The strength of *LCSS* is to base the similarity of points on a threshold criterion, allowing outliers, and excluding overlapping pairs. A higher *LCSS* distance is even obtained for sequence 16 by restricting the value of δ.

We also notice that the two classes are better separated when computing the distances from the pre-processed data. Filtering the sequences indeed results in removing at least part of the variability in the values.

5. CONCLUSION

In that paper, we have proposed a similarity measure for heterogeneous multivariate time-series using a non-metric distance based on *LCSS*. We have demonstrated the efficiency of our approach from an experimental set of sequences in the context of home health telecare. At a larger scale, its use to extract similar behaviors of a person at home along the days confirms the relevance of that measure. The generality of the method should make it suitable for other applications, with possibly assigning different weights to the parameters.

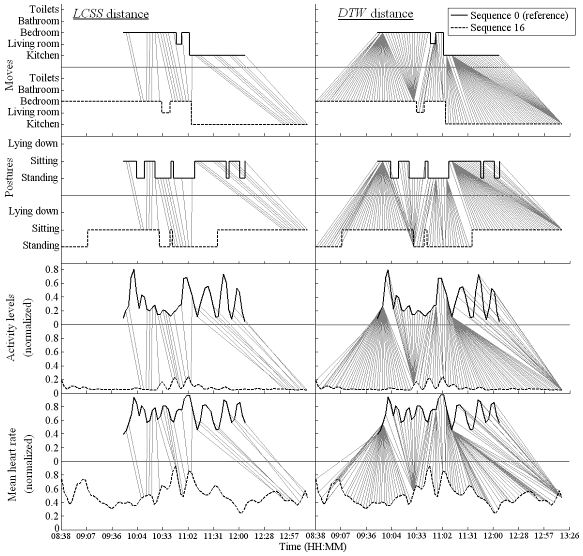

Figure 6: Pairs of points considered as similar when computing *LCSS* and *DTW* distances.

REFERENCES

[1] R. Agrawal, K. Sawhney, K. Shim, “Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases,” in *Proc. of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Zurich, Switzerland, September 1995, pp. 490–501.

[2] A. Apostolico, “String editing and longest common subsequence,” *G. Rozenberg and A. Salomaa, editors, Handbook of Formal Languages*, vol. 2, pp. 361–398, Berlin, 1997. Springer Verlag.

[3] B. Chiu, E. Keogh, and S. Lonardi, “Probabilistic Discovery of Time Series Motifs,” in *Proc. of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, August 2003, pp. 493–498.

[4] G. Das, D. Gunopulos, and H. Mannila, “Finding similar time series,” *Principles of Data Mining and Knowledge Discovery*, vol. 19, pp. 88–100, 1997.

[5] D. S. Hirschberg, “Algorithms for the longest common subsequence problem,” *JACM*, vol. 24 (4), pp. 664–675, 1977.

[6] J. W. Hunt & T. G. Szymanski, “A fast algorithm for computing longest common subsequences,” *CACM*, vol. 20 (5), pp. 350–353, 1977.

[7] E. Keogh, M. Pazzani, “Scaling up Dynamic Time Warping for Datamining Applications,” in *Proc. of the 21st Int. Conf. on Very Large Databases*, Boston, MA, 2000, pp. 285–289.

[8] J. B. Kruskall & M. Liberman, “The symmetric time warping algorithm: From continuous to discrete,” *Time Warps, String Edits and Macromolecules*, Addison-Wesley, 1983.

[9] M. Vlachos, G. Kollios, ans G. Gunopulos, “Discovering Similar Multidimensional Trajectories,” in *Proc. of the 18th ICDE*, San Jose, CA, 2002, pp. 673–684.