A Competitive Measure to Assess the Similarity Between Two Time Series

Joan Serrà and Josep Lluis Arcos

IIIA-CSIC, Artificial Intelligence Research Institute, Spanish National Research Council, Bellaterra, Barcelona, Spain.
{jserra,arcos}@iiia.csic.es

Abstract. Time series are ubiquitous, and a measure to assess their similarity is a core part of many systems, including case-based reasoning systems. Although several proposals have been made, still the more robust and reliable time series similarity measures are the classical ones, introduced long time ago. In this paper we propose a new approach to time series similarity based on the costs of iteratively jumping (or moving) between the sample values of two time series. We show that this approach can be very competitive when compared against the aforementioned classical measures. In fact, extensive experiments show that it can be statistically significantly superior for a number of data sources. Since the approach is also computationally simple, we foresee its application as an alternative off-the-shelf tool to be used in many case-based reasoning systems dealing with time series.

1 Introduction

Data in the form of time series pervades almost any scientific domain [9, 11]. Observations that unfold over time usually represent valuable information subject to be analyzed, classified, predicted, or interpreted [4, 8, 10]. Real-world examples include financial data (e.g. stock market fluctuations), medical data (e.g. electrocardiograms), computer data (e.g. log sequences), or motion data (e.g. geolocation of moving objects). Dealing with time series represents a challenge for these and many other scientific domains.

Dealing with time series has also been a challenge for the case-based reasoning (CBR) community. Apart from the two workshops on time series prediction held in the 2003 and 2004 International Conferences on CBR [6, 7], several CBR systems have coped with cases involving time series or sequential information. Xiong and Funk [22] presented a CBR system managing symbolic time series from a medical domain. Their approach was based on the identification of key sub-sequences and on the transformation of the original time series into a more compact representation. In a preliminary work [3], the authors demonstrated the value of incorporating knowledge discovery techniques to CBR and, in particular, the value of the technique they used to extract significant sub-sequences, which allowed them to automatically discover non-trivial regularities. Montani et al. [13] used the discrete Fourier transform (DFT) in their CBR system to reduce the comparison of (entire) time series to just the first Fourier coefficients, and also to implement indexing structures for the time series. Other CBR systems reduce the dimensionality by transforming the time series using temporal abstractions [1] or hierarchical symbol abstractions [21]. A further interesting approach dealing with time series is the CASEP2 system [23], which was proposed as a hybrid system that, combining CBR and artificial neural networks, performed time series classification in an efficient way. Also related to time series is the Ceaseless CBR model introduced by Martín and Plaza [12], which processes a continuous data stream holding several problem descriptions.

A core issue when dealing with time series is determining their pairwise similarity, i.e. the degree to which a given time series resembles another one. In fact, a time series dissimilarity (or similarity) measure is central to many mining, retrieval, classification, and clustering tasks [4, 10]. However, deriving a measure that correctly reflects time series dissimilarities is not straightforward. Apart from dealing with a high dimensionality (time series can be roughly considered as multi-dimensional data), the calculation of such measures needs to be fast, robust, and efficient. Moreover, there is the need for generic dissimilarity measures, so that they can be readily applied to any data set, being this application the final goal or just an initial approach to a given task.

With years, several time series dissimilarity measures have been proposed. However, it seems that the most common measures, proposed long time ago, turn out to be the most competitive ones [10, 20]. Wang et al. [20] perform an extensive comparison of classification accuracies for 13 different time series dissimilarity measures across 38 contrasting data sources (we also refer the interested reader to [20] for pointers to the original references proposing or using such measures in the context of mining time series data). After reporting the results, one of the main conclusions of the study is that, despite of the new proposals, the Euclidean and dynamic time warping (DTW) [14, 15] dissimilarity measures are extremely difficult to beat, remaining two of the most robust, simple, generic, and efficient measures.

In this paper we propose a new time series dissimilarity measure based on minimum jump costs (MJCs). The main idea behind this measure is that it reflects the cumulative cost of iteratively 'jumping' from one time series to the other, starting at the beginning of a time series until the end of any of them is reached, and without going backwards. As it will be shown by extensive and rigorous experiments, MJC clearly outperforms the Euclidean distance. Moreover, we will see that MJC can statistically significantly outperform DTW for a number of data sets. This, jointly with the computationally simple operations behind MJC, makes it a good candidate measure to be incorporated to any standard toolkit for time series similarity, retrieval, or classification and, by extension, to any case-based reasoning system dealing with time series.

The remaining of the paper is organized as follows. We first present some scientific background by outlining the calculation of the Euclidean and DTW dissimilarity measures (Sec. 2). The description of the MJC dissimilarity measure comes next (Sec. 3). We then explain our evaluation methodology and present the obtained results (Secs. 4 and 5, respectively). A conclusion section ends the paper (Sec. 6).

2 Scientific Background

Across years, several dissimilarity measures have been proposed, the most simple ones being variants of the $\mathrm{L_p}$ norm,

$$d_{\mathrm{L_p}}(\mathbf{x},\mathbf{y}) = \sqrt[p]{\sum_{i=1}^{M}(x_i - y_i)^p}, \tag{1}$$

where p is a positive integer, M is the length of the time series, and x_i and y_i are the i-th element of time series $\mathbf{x}$ and $\mathbf{y}$, respectively. Usually $p = 2$, yielding the Euclidean distance, one of the first generic dissimilarity measures proposed for time series [2]. In case $\mathbf{x}$ and $\mathbf{y}$ were not of the same length, one can always re-sample one to the length of the other, an approach that works well for many data sources [10]. The Euclidean distance is one of the most used and efficient time series dissimilarity measures. Indeed, its accuracy may be very difficult to beat in some scenarios, specially when the length of the time series increases [20]. Nonetheless, we believe that such affirmation needs to be carefully assessed with extensive experiments and under broader conditions, considering different distance-exploiting algorithms.

Another classical option for computing the dissimilarity between two time series is dynamic time warping (DTW) [14, 15]. DTW belongs to the group of so-called *elastic* dissimilarity measures [10, 20], and works by optimally aligning (or 'warping') the time series in the temporal dimension so that the accumulated cost of this alignment is minimal. In its most basic form, this cost can be obtained by dynamic programming, recursively applying

$$D_{i,j} = \delta(x_i, y_j) + \min\{D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1}\} \tag{2}$$

for $i = 1, \ldots, M$ and $j = 1, \ldots, N$, being M and N the lengths of time series $\mathbf{x}$ and $\mathbf{y}$, respectively. Except for the first cell, which is initialized to $D_{0,0} = 0$, the matrix D is initialized to $D_{i,j} = \infty$ for $i = 0, 1, \ldots, M$ and $j = 0, 1, \ldots, N$. In the case one deals with uni-dimensional time series, the sample (or local) dissimilarity function $\delta()$ is typically taken to be the square of the difference between x_i and y_j, i.e. $\delta(x_i, y_j) = (x_i - y_j)^2$. In the case we deal with multidimensional time series or we have some domain-specific knowledge, the sample dissimilarity function $\delta()$ must be chosen appropriately, although many times the Euclidean distance is used.

The final dissimilarity measure between time series $\mathbf{x}$ and $\mathbf{y}$ typically corresponds to the total accumulated cost $d_{\mathrm{DTW}}(\mathbf{x},\mathbf{y}) = D_{M,N}$. A normalization of $d_{\mathrm{DTW}}(\mathbf{x},\mathbf{y})$ can be performed on the basis of the alignment of the two time series, which is found by backtracking from $D_{M,N}$ to $D_{0,0}$ [14]. However, in preliminary analysis we found the normalized variant to be equivalent, or sensibly less accurate, than the unnormalized one.

Several constrains can be applied in the computation of D. A common operation is to introduce a window parameter w [15], such that the recursive formula of Eq. 2 is only applied for $i = 1, \ldots, M$ and

$$j = \max\{1, i' - w\}, \ldots, \min\{N, i' + w\}, \tag{3}$$

where i' is progressively adjusted for dealing with different time series lengths, i.e. $i' = \lfloor iN/M \rceil$, using $\lfloor\ \rceil$ as the round-to-the-nearest-integer operator. Notice that if $w = 0$ and $N = M$, $D_{M,N}$ will correspond to the squared Euclidean distance. Notice furthermore that when $w = N$ we are using the unconstrained version of DTW.

The introduction of constrains, and specially of the window parameter w, generally carries some advantages [10, 14, 20]. For instance, they prevent from 'pathological alignments' (which typically go beyond the main diagonal of D) and, therefore, they usually provide better dissimilarity estimates. In addition, DTW constrains allow for reduced computational costs, since only a percentage of the cells in D needs to be examined.

DTW stands as the main benchmark against to which new dissimilarity measures need to be compared with [20]. Very few measures have been proposed that systematically outperform DTW for a number of different data sources. However, these measures are usually more complex than DTW, sometimes requiring extensive parameter tuning of one or more parameters. Additionally, no careful, rigorous, and extensive evaluation of the accuracy of these measures had been initially done, and further studies fail to assess the statistical significance of their improvement [20]. In this paper we pay special attention to all these aspects in order to formally assess the benefits of the measure we propose.

3 Minimum Jump Costs Dissimilarity

We now detail the calculation of the minimum jump costs (MJC) dissimilarity measure. The main idea behind MJC is that, if a given time series $\mathbf{x}$ resembles $\mathbf{y}$, the cost of iteratively 'jumping' between their samples should be small. In other words, if $\mathbf{x}$ and $\mathbf{y}$ are similar, we could only draw short lines between them when placed on the same time axis. Intuitively, these jumps (or lines) should be iteratively done from the beginning of the time series until we reach an end, otherwise we would be discarding some possibly relevant parts of the time series. Similarly, if we kept jumping (or drawing lines) both forward and backwards, we could be iterating an infinite number of times. Thus we force to jump (or draw) in the forward direction only. Finally, since we want a single number reflecting the global dissimilarity between $\mathbf{x}$ and $\mathbf{y}$, the most straightforward solution is to add the costs of performing a jump (or the lengths of the lines). A more formal definition follows.

Let $\mathbf{x} = x_1, \ldots x_M$ and $\mathbf{y} = y_1, \ldots y_N$ be two time series of potentially different lengths M and N, respectively. We define the minimum jump costs (MJC) dissimilarity measure d_{XY} as the cumulative minimal cost for iteratively jumping from one time series to the other, i.e.

$$d_{\mathrm{XY}} = \sum_i c_{\min}^{(i)}, \tag{4}$$

where $c_{\min}^{(i)}$ is the cost of the i-th jump, which should be minimal. Supposing that for the i-th jump we are at time step t_x of time series $\mathbf{x}$ and that we previously visited time step $t_y - 1$ of $\mathbf{y}$,

$$c_{\min}^{(i)} = \min\left\{ c_{t_x}^{t_y}, c_{t_x}^{t_y+1}, c_{t_x}^{t_y+2}, \ldots \right\}, \tag{5}$$

where $c_{t_x}^{t_y+\Delta}$ is the cost of jumping from x_{t_x} to $y_{t_y+\Delta}$ and $\Delta = 0, 1, 2, \ldots$ is an integer time step increment such that $t_y + \Delta \le N$. Notice that we can only go forward, i.e. we cannot visit time series samples before t_x or t_y. After a jump is made, t_x and t_y are updated accordingly, i.e. t_x becomes $t_x + 1$ and t_y becomes $t_y + \Delta + 1$. This way we enforce that no time step position is repeated and that the iterative algorithm does not go backwards. As mentioned, the formulation of Eq. 5 corresponds to a jump from time series $\mathbf{x}$ to $\mathbf{y}$. In case we want to jump from $\mathbf{y}$ to $\mathbf{x}$, only t_x and t_y need to be swapped in Eq. 5. We start the iterations at $t_x = 1$, considering $t_y = 1$, and jump between $\mathbf{x}$ and $\mathbf{y}$ until an end of a time series is reached, i.e. until $t_x = M$ or $t_y = N$.

To define a jump cost $c_{t_x}^{t_y+\Delta}$ we consider the temporal and the magnitude dimensions of the time series. Therefore we define

$$c_{t_x}^{t_y+\Delta} = (\phi\Delta)^2 + \delta(x_{t_x}, y_{t_y+\Delta}), \tag{6}$$

where ϕ represents the cost of advancing in time and $\delta()$ is the magnitude dissimilarity function, which we take to be $\delta(x_{t_x}, y_{t_y+\Delta}) = (x_{t_x} - y_{t_y+\Delta})^2$, equivalently to what we do with DTW (Eq. 2). We set ϕ proportional to the standard deviation σ expected for the time series,

$$\phi = \beta \frac{4\sigma}{\min\{M, N\}}, \tag{7}$$

and introduce the parameter $\beta \in [0, \infty)$, $\beta \in \mathbb{R}$, which controls how difficult is to advance in time. A value of $\beta = 0$ implies no cost ($\phi = 0$), whereas values of $\beta \to \infty$ imply that only samples at time stamp t_y will be considered ($\Delta = 0$, see Eq. 6). This latter case makes d_{XY} equal to the squared Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$.

Finally, notice that d_{XY} is asymmetric. Depending whether we start at x_1 or y_1 we will obtain different values. To obtain a symmetrized dissimilarity measure we use

$$d_{\mathrm{MJC}}(\mathbf{x}, \mathbf{y}) = \min\left\{ d_{\mathrm{XY}}, d_{\mathrm{YX}} \right\}, \tag{8}$$

where d_{XY} and d_{YX} are the cumulative MJCs obtained by starting at x_1 and y_1, respectively. Measures d_{XY}, d_{YX}, and by extension $d_{\mathrm{MJC}}(\mathbf{x}, \mathbf{y})$ can be considered as elastic measures [20].

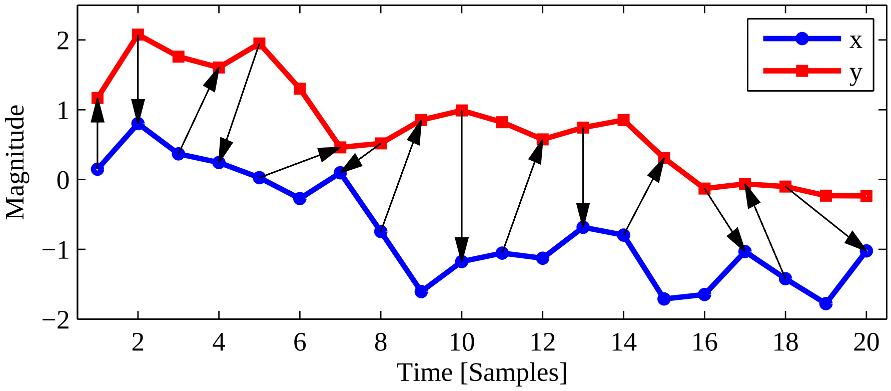

Fig. 1. Example of the recursive jumps performed between time series $\mathbf{x}$ and $\mathbf{y}$. The algorithm starts with time series $\mathbf{x}$ at $t_x = 1$ (x_1) and ends when $t_x = M$ or $t_y = N$ (x_{20} in the example).

Fig. 1 helps explaining the calculation of d_{XY}. Suppose that we are at sample x_5 and that we previously jumped from y_5 to x_4 (hence the values of $t_x = 5$ and $t_y = 6$). We now want to jump to time series $\mathbf{y}$ again. In addition, we want the cost of the jump to be minimal. Therefore, we evaluate Eq. 6 for all possible $t_y + \Delta$, i.e. for $\Delta = 0, 1, \ldots, 14$ (from time steps 6 to 20). With that we obtain that the best jump option is y_7 ($\Delta = 1$). After the jump we update t_x to 6 and t_y to 8, the next time steps that will be considered in the following iteration.

Algorithms 1 and 2 provide the implementation details for the whole dissimilarity calculation. Notice that we do not need to compute all possible costs, thanks to the introduction of the monotonically increasing term $\phi\Delta$ which, furthermore, can be precomputed. Notice also that since d_{XY} is cumulative, an early abandoning strategy can be additionally implemented to speed-up computations [10]. This way, if only the first nearest neighbor of a time series was required, we would only accumulate costs until we reached the smallest $d_{\text{XY}}^{\text{best}}$ found so far, exiting the process before its end since the current cumulative cost d_{XY} could not be smaller than $d_{\text{XY}}^{\text{best}}$. See [10] for more details about this procedure.

4 Evaluation Methodology

The efficacy of a time series dissimilarity measure is commonly evaluated by the classification accuracy it achieves [10, 20]. For that, the error ratio of a distance-based classifier is calculated for a given labeled data set, understanding the error ratio as the number of wrongly classified items divided by the total number of tested items. The standard choice for the classifier is the one nearest neighbor (1NN) classifier. Following [20], we can enumerate several advantages of using

Algorithm 1 dXY($\mathbf{x}$,$\mathbf{y}$)

```
Input: Time series $\mathbf{x} = x_1, \ldots x_M$ and $\mathbf{y} = y_1, \ldots y_N$
Output: Cumulative MJC dissimilarity measure $d_{\mathrm{XY}}$
$t_x, t_y \leftarrow 1$
$d_{\mathrm{XY}} \leftarrow 0$
while $t_x \leq M$ and $t_y \leq N$ do
    $d_{\mathrm{XY}} \leftarrow d_{\mathrm{XY}}$+cmin($\mathbf{x}$,$t_x$,$\mathbf{y}$,$t_y$)
    if $t_x > M$ or $t_y > N$ then
        break
    end if
    $d_{\mathrm{XY}} \leftarrow d_{\mathrm{XY}}$+cmin($\mathbf{y}$,$t_y$,$\mathbf{x}$,$t_x$)
end while
return $d_{\mathrm{XY}}$
```

Algorithm 2 cmin($\mathbf{x}$,t_x,$\mathbf{y}$,t_y)

```
Input: Time series $\mathbf{x} = x_1, \ldots x_M$ and $\mathbf{y} = y_1, \ldots y_N$; time indices $t_x$ and $t_y$
Output: Minimum jump cost $c_{\min}$; updated $t_x$ and $t_y$
$c_{\min} \leftarrow \infty$
$\Delta, \Delta_{\min} \leftarrow 0$
while $t_y + \Delta \leq N$ do
    $c \leftarrow (\phi\Delta)^2$
    if $c \geq c_{\min}$ then
        if $t_y + \Delta > t_x$ then
            break
        end if
    else
        $c \leftarrow c + (x_{t_x} - y_{t_y+\Delta})^2$
        if $c < c_{\min}$ then
            $c_{\min} \leftarrow c$
            $\Delta_{\min} \leftarrow \Delta$
        end if
    end if
    $\Delta \leftarrow \Delta + 1$
end while
$t_x \leftarrow t_x + 1$
$t_y \leftarrow t_y + \Delta_{\min} + 1$
return $c_{\min}$
```

this approach. First, the error of the 1NN classifier critically depends on the dissimilarity measure used. Second, the 1NN classifier is parameter-free and easy to implement. Third, there are theoretical results relating the error of a 1NN classifier to errors obtained with other classification schemes. Fourth, some works suggest that the best results for time series classification come from simple nearest neighbor methods. We refer to [20] and references therein for more details about these aspects.

To asses a classifier's error, out-of-sample validation needs to be done. In our experiments we follow a two-fold cross-validation scheme [16] with balanced data sets (same number of items per class). We repeat the validation 10 times and report average error ratios. To assess the statistical significance of the difference between two error ratios we employ the Friedman's test [5], a non-parametric two-way analysis of variance that deals with dependent samples. We use $p < 0.05$ and apply the Bonferroni adjustment to compensate for multiple experiments [16]. Therefore, using k folds, r repetitions, and s data sets, the actual p^*-value corresponds to $p^* < 1 - \sqrt[krs]{1-p}$. Hence, with our setting, $p^* < 7.124 \cdot 10^{-5}$.

We perform experiments with 36 different time series data sets from the UCR time series repository [11]. This is the world's biggest time series repository, and some authors estimate that it makes up to more than 90% of all publicly-available, labeled data sets [20]. It comprises synthetic, as well as real-world data sets, and also includes one-dimensional time series extracted from two-dimensional shapes [11]. The 36 data sets considered here practically correspond to the totality of the UCR repository. Only 4 data sets were discarded prior to and independently from the present work. Within the 36 data sets, the number of classes ranges from 2 to 50, the number of time series per data set ranges from 56 to 9,236, and time series lengths go from 24 to 1,882 samples (a total of 728,611,296 samples from 51,888 time series have been processed). For further details on these data sets we refer to the cited references.

Before performing the experiments, all time series from all data sets were Z-normalized so that they had zero mean and unit variance. Furthermore, in the training phase of our cross-validation we performed an in-sample optimization of the measures' parameters. This optimization step consisted of a grid search within a suitable range of parameter values. For DTW we used 30 linearly-spaced integer values of $w \in [0, 0.25N]$ plus $w = N$ (the unconstrained DTW variant). For MJC we used 30 linearly-spaced real values of $\beta \in [0, 25]$ plus $\beta = 10^{10}$ (in practice corresponding to the squared Euclidean distance variant of $\beta \to \infty$). After the grid search, the parameter value yielding to the best in-sample error ratio was kept for out-of-sample testing.

5 Results

A full account of the error ratios obtained with the Euclidean distance, DTW, and MJC for the 36 data sets is provided in Table 1. A baseline consisting of using a random dissimilarity measure is also reported. For that we draw a random

Table 1. Average error ratios for the 36 data sets used in the paper. The * symbol indicates that the dissimilarity measure is statistically significantly superior to all others (see text). Best results are highlighted in bold, independently of their statistical significance.

Data set	Random	Euclidean	DTW	MJC
50words	0.980	0.528	**0.361**	**0.361**
Adiac	0.972	0.374	0.378	**0.365**
Beef	0.802	0.487	0.495	**0.458**
CBF	0.670	0.018	**0.001**	**0.001**
ChlorineConcentration	0.667	0.116	**0.111**	0.115
CincECGTorso	0.750	0.003	**0.000***	0.003
Coffee	0.498	**0.017**	0.036	0.054
DiatomSizeReduction	0.749	0.014	**0.009**	0.010
ECG200	0.496	**0.126**	0.133	0.138
ECGFiveDays	0.501	0.012	0.012	**0.001***
FaceFour	0.748	0.155	0.085	**0.041**
FacesUCR	0.929	0.171	0.069	**0.044***
Fish	0.851	0.194	0.196	**0.119***
GunPoint	0.505	0.079	0.027	**0.013**
Haptics	0.796	0.615	0.584	**0.569**
InlineSkate	0.856	0.558	0.521	**0.432***
ItalyPowerDemand	0.498	**0.035**	**0.035**	0.039
Lighting2	0.508	0.322	**0.189***	0.284
Lighting7	0.850	0.428	**0.260**	0.345
MALLAT	0.873	0.021	**0.017**	0.018
MedicalImages	0.902	0.350	**0.276**	0.325
Motes	0.499	0.090	0.068	**0.039***
OliveOil	0.744	0.129	**0.125**	0.147
OSULeaf	0.840	0.434	0.395	**0.296***
SonyAIBORobotSurface	0.501	0.024	0.023	**0.019**
SonyAIBORobotSurfaceII	0.507	0.027	0.026	**0.019**
StarLightCurves	0.666	0.126	**0.117**	0.120
SwedishLeaf	0.933	0.220	0.157	**0.124***
Symbols	0.833	0.038	**0.020**	0.021
SyntheticControl	0.834	0.099	**0.010***	0.035
Trace	0.748	0.210	**0.001***	0.055
Two-Patterns	0.749	0.030	**0.000***	0.001
TwoLeadECG	0.498	0.007	**0.001**	0.003
Wafer	0.504	**0.004**	**0.004**	0.005
WordsSynonyms	0.960	0.535	0.379	**0.355**
Yoga	0.496	0.082	0.071	**0.061***

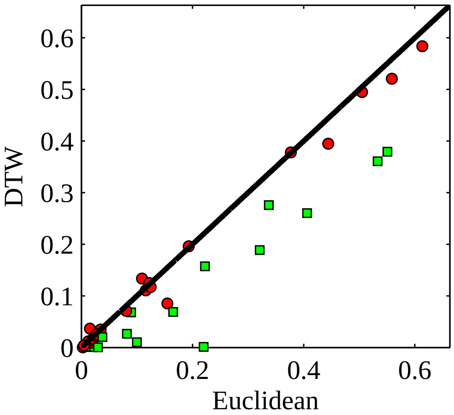

Fig. 2. Pairwise comparison between Euclidean and DTW dissimilarity measures. Values in the lower-right triangular part indicate better results for DTW (better results for Euclidean distance would be scattered in the upper-left triangular part). Green squares indicate statistically significant differences in error ratios (non-significant ratios are denoted with red dots).

number from the uniform distribution between 0 and 1 and return this number as the actual dissimilarity between two time series. The rest of the procedure is the same as for the other dissimilarity measures tested.

First we compare the error ratios of the Euclidean and DTW dissimilarity measures (Fig. 2). We observe that, with the considered data, DTW is usually superior to the Euclidean distance. In 15 of the 36 data sets the error ratios obtained for DTW are statistically significantly below the ones obtained for the Euclidean distance. Notice though that for a few data sets the Euclidean distance is slightly but not statistically significantly superior to DTW. This is due to the fact that the optimization step fails to learn a better w parameter value, which for these specific cases would have been $w = 0$.

We now turn our attention to the proposed measure based on MJCs (Fig. 3). When comparing it with the Euclidean distance (Fig. 3 left) we find that MJC is usually superior. In fact, we have an equivalent situation as we had when comparing DTW and the Euclidean distance. In 17 of the 36 data sets the error ratios obtained for MJC are statistically significantly below the ones obtained for the Euclidean distance. Again, for the very same reason outlined before, the Euclidean distance is slightly but not statistically significantly superior to MJC in a few data sets.

The interesting comparison though is between DTW and MJC (Table 1 and Fig. 3 right). At a first sight, their error ratios look very similar. DTW's error ratio is lower than MJC's in 16 of the 36 data sets and MJC's is lower than DTW's in also 16 of the 36 data sets. However, if we just focus on statistically significant results, MJC outperforms DTW in 8 of the 36 cases while DTW only

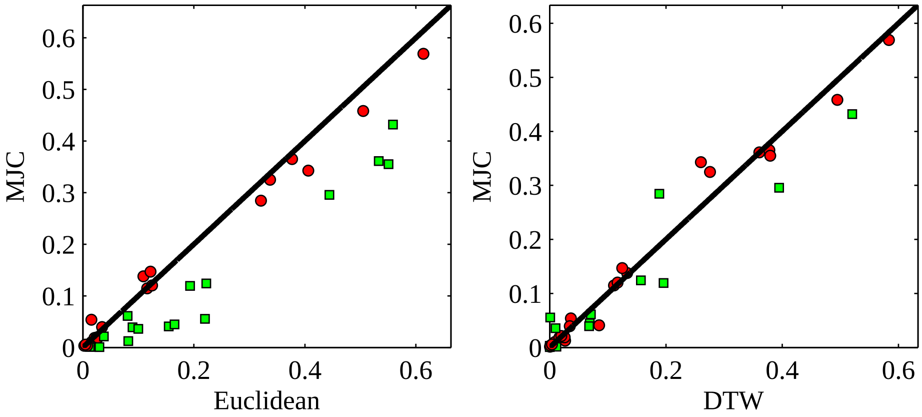

Fig. 3. Pairwise comparison between Euclidean and MJC dissimilarities (left) and between DTW and MJC (right).

outperforms MJC in 5 of the 36 cases. This points towards a slight superiority of MJC with respect to DTW.

From the error ratios reported with the considered data sets we see that the Euclidean distance is never statistically significantly superior to DTW nor to MJC (which is a clear consequence of the fact that both DTW and MJC incorporate the Euclidean distance as a special case of their parameters value). This reduces the comparison to DTW and MJC. Therefore, summarizing, we see that MJC outperforms DTW in 8 of the 36 data sets (≈22%), that DTW outperforms MJC in 5 of the 36 data sets (≈14%), and that for the remaining 23 data sets (≈64%) the error ratios are comparable within statistical significance. The fact that MJC outperforms DTW for roughly 22% of the considered data sets highlights the potential of the former and has clear implications for researchers and practitioners dealing with new data, as such new data set could potentially be one of the data sets where MJC statistically significantly outperforms the classical DTW.

6 Conclusion and Discussion

We have presented a new approach to assess dissimilarities between time series based on minimum jump costs (MJC). Beyond the novelty of the concept, we have shown that it is computationally easy to implement (just a few lines of code) and that further efficiency issues can be deployed. More importantly, we have shown that the MJC dissimilarity measure is very competitive. Under rigorous and extensive experiments we find that, in many situations, it can statistically significantly outperform dynamic time warping, a dissimilarity measure which is regarded as very difficult to beat. All these facts encourage the incorporation of the MJC dissimilarity measure to the standard off-the-shelf toolkit for retrieving and classifying time series data.

Intuitively, it seems clear that by deriving a data-specific dissimilarity measure targeted to a particular problem one would always outperform generic measures such as the Euclidean, DTW, or MJC. However, this does not preclude considering these generic measures as part of an initial approach or assessment. Furthermore, it could also well be the case that, for such a specific data set, the derived, data-specific measure was comparable to or even less competitive than the three measures considered here. In these cases, the usage of a potentially complex data-specific dissimilarity measure could be difficult to justify.

We also notice that all considered time series dissimilarity measures are 'global' dissimilarity measures, i.e. they match whole time series. These are the big majority of time series dissimilarity measures. However, measures considering 'local' or subsequence matches and their variants do also exist (see e.g. [18, 19]). Given a data set that needs of such local matches, a common operation to still use global dissimilarity measures is to partition the whole time series into multiple subsequences, either by exploiting some previous knowledge of the data or simply by a brute-force moving window strategy. This partitioning increases the number of comparisons between (sub)series and sometimes implies a further operation to merge the result of such comparisons (e.g. by taking the mean or the maximum similarity found [18, 19]).

In this contribution we do not specifically treat the case of multidimensional time series. Indeed, all the considered data sets from the UCR time series repository are uni-dimensional. Nonetheless, one should notice that the multidimensional case can be easily handled. One option could be to consider each component or dimension as a single time series, calculate its dissimilarity, and finally aggregate all such dissimilarities to form a global measure (potentially weighting individual dissimilarities). However, one should notice that the formulation of both DTW and MJC naturally incorporates the possibility to deal with multidimensional time series, since they both use a sample (local) dissimilarity function $\delta()$ (Eqs. 2 and 6), which may be problem-specific and adapted to the particular nature of the considered time series.

The number of CBR systems that deal with time series data is increasing in domains such as health care or industrial monitoring. Although an important issue is the selection of the appropriate sample dissimilarity function $\delta()$, the availability of powerful, general-purpose measures for comparing time series is required to speed-up the development of these CBR systems. In this research, MJC has been evaluated in 36 data sets with time series of lengths ranging from 24 to 1,882 samples. Although for some of the considered data sets the time series are relatively long, they generally model a unique complex pattern. Thus, we believe that the identification of key sub-sequences such as in [17, 22] or the dimensionality reduction applied in [13] would not improve classification performance in these data sets. Contrastingly, generic dissimilarity measures such as the ones considered here can be very useful in data sets where, after recurrent patterns have been identified, the resulting sub-sequences still need to be compared.

Acknowledgments

This research has been partially funded by 2009-SGR-1434 from Generalitat de Catalunya, TIN2009-13692-C03-01 from the Spanish Government, and EU Feder funds. JS also acknowledges JAEDOC069/2010 from Consejo Superior de Investigaciones Científicas.

References

1. Bottrighi, A., Leonardi, G., Montani, S., Portinale, L., Terenziani, P.: Intelligent data interpretation and case base exploration through temporal abstractions. In: Bichindaritz, I., Montani, S. (eds.) Case-Based Reasoning. Research and Development, Lecture Notes in Computer Science, vol. 6176, pp. 36–50. Springer Berlin / Heidelberg (2010)
2. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases. In: Proc. of the ACM SIGMOD Int. Conf. on Management of Data. pp. 419–429 (1994)
3. Funk, P., Xiong, N.: Case-based reasoning and knowledge discovery in medical applications with time series. Computational Intelligence 22(3/4), 238–253 (2006)
4. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann, Waltham, USA (2005)
5. Hollander, M., Wolfe, D.A.: Nonparametric statistical methods. Wiley, New York, USA, 2nd edn. (1999)
6. Kanawati, R., Malek, M., Salotti, S. (eds.): Proc. of the 1st Workshop on Applying CBR to Time Series Prediction (ICCBR-2003). Dept. of Computer and Information Science, Northwestern University of Science and Technology (2003)
7. Kanawati, R., Salotti, S. (eds.): Proc. of the 2nd Workshop on Applying CBR to Time Series Prediction (ECCBR-2004). Dept. of Sistemas Informaticos y Programación, Universidad Complutense de Madrid (2004)
8. Kantz, H., Schreiber, T.: Nonlinear time series analysis. Cambridge University Press, Cambridge, UK (2004)
9. Keogh, E.: Machine learning in time series databases (and everything is a time series!). Tutorial at the AAAI Int. Conf. on Artificial Intelligence (2011)
10. Keogh, E., Kasetty, S.: On the need for time series data mining benchmarks: a survey and empirical demonstration. Data Mining and Knowledge Discovery 7(4), 349–371 (2003)
11. Keogh, E., Zhu, Q., Hu, B., Hao, Y., Xi, X., Wei, L., Ratanamahatana, C.A.: The UCR time series classification/clustering homepage (2011), `http://www.cs.ucr.edu/%7eeamonn/time_series_data`
12. Martín, F.J., Plaza, E.: Ceaseless case-based reasoning. In: Funk, P., González Calero, P.A. (eds.) European Conference on Case-Based Reasoning (ECCBR), pp. 287–301. No. 3155 in Lecture Notes in Artificial Intelligence, Springer-Verlag (2004)
13. Montani, S., Portinale, L., Leonardi, G., Bellazzi, R., Bellazzi, R.: Case-based retrieval to support the treatment of end stage renal failure patients. Artificial Intelligence in Medicine 37(1), 31–42 (2006)

14. Rabiner, L.R., Juang, B.: Fundamentals of speech recognition. Prentice-Hall, Upper Saddle River, USA (1993)
15. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. on Acoustics, Speech, and Language Processing 26(1), 43–50 (1978)
16. Salzberg, S.L.: On comparing classifiers: pitfalls to avoid and a recommended approach. Data Mining and Knowledge Discovery 1(3), 317–328 (1997)
17. Serrà, J., Müller, M., Grosche, P., Arcos, J.L.: Unsupervised detection of music boundaries by time series structure features. In: Proc. of the AAAI Int. Conf. on Artificial Intelligence. p. In press (2012)
18. Serrà, J., Serra, X., Andrzejak, R.G.: Cross recurrence quantification for cover song identification. New Journal of Physics 11(9), 093017 (2009)
19. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. Journal of Molecular Biology 147, 195–197 (1981)
20. Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., Keogh, E.: Experimental comparison of representation methods and distance measures for time series data. Data Mining and Knowledge Discovery In press (2012), `http://dx.doi.org/10.1007/s10618-012-0250-5`
21. Xia, B.B.: Similarity search in time series data sets. MSc thesis, Simon Fraser University, Burnaby, Canada (1997)
22. Xiong, N., Funk, P.: Concise case indexing of time series in health care by means of key sequence discovery. Applied Intelligence 28(3), 247–260 (2008)
23. Zehraoui, F., Kanawati, R., Salotti, S.: CASEP2: hybrid case-based reasoning system for sequence processing. In: Funk, P., González Calero, P.A. (eds.) European Conference on Case-Based Reasoning (ECCBR), pp. 449–463. No. 3155 in Lecture Notes in Artificial Intelligence, Springer-Verlag (2004)