Generalization of the L_p norm for time series and its application to Self-Organizing Maps

John A. Lee and **Michel Verleysen**
Machine Learning Group, DICE, Université catholique de Louvain
Louvain-la-Neuve, Belgium
{lee, verleysen}@dice.ucl.ac.be

Abstract - *Time series are often encoded in vectors and analyzed using standard vectorial tools (distances, inner products, etc.). Most of them neglect the temporal structure of time series. This paper proposes a generalization of the L_p norm that takes the temporal structure into account. This norm remains computationally simple and keeps useful properties, like e.g. differentiability, which allow integrating the new norm into Self-Organizing Maps to analyze sets of time series. Experiments on artificially generated data show the advantages and specificities of the proposed norm.*

Key words - Self-Organizing Maps, metric, distance, time series

1 Introduction

Self-Organizing Maps [3] (SOMs) are powerful data analysis and visualization tools. They usually project data on a two-dimensional predefined map, allowing strong nonlinearities in the projection. The operations performed in the SOM algorithm and its variants depend on a metric: the metric is used in the choice of the 'winning' prototype or BMU (Best-Matching Unit, i.e. the closest prototype to an input datum), and in the adaptation rule of the prototypes. Usually, the traditional Euclidean distance is used in SOMs. In some cases however, the Euclidean distance might prove not appropriate to the specific nature of data. This might be the case when processing time series, i.e. sets of consecutive values in time; regressors built using a fixed-size sliding window on a time series are an example of such data [7]. In this case, measuring the Euclidan distances between two regressors does not take the temporal structure into account: one could switch values in time without any effect on the distance. Clearly, some information contained in the temporal structure is lost. This papers presents a simple measure between time series taking that structure into account and its integration into a SOM. Other approaches can be found in [1, 5, 2].
After a background section (Section 2) about distances and norms aimed at reminding some concepts and defining the notations, Section 3 presents the proposed norm and the corresponding distance. Next, Section 4 embeds this metric into the SOM algorithm. Section 5

Michel Verleysen is a Senior Research Associate of the Belgian National Fund for Scientific Research (F.N.R.S.).

presents results obtained on artificially-generated time series. The increased robustness of the proposed metric is put forward, by comparison with the Euclidean one, when the series is polluted by noise. Finally, conclusions are drawn in Section 6.

2 Vectors, norms and distances

Working directly on the vectors usually does not take the order of the coordinates into account. As a consequence, the temporal aspect of sequences and time series is lost. For example, if a D-dimensional vector $\mathbf{x}$ is written according to $\mathbf{x} = [x_1, \ldots, x_i, \ldots, x_D]^T$, then the L_p norm (Minkowski norm) is defined as:

$$L_p(\mathbf{x}) = \|\mathbf{x}\|_p = \left(\sum_{i=1}^{D} |x_i|^p \right)^{\frac{1}{p}} \quad . \tag{1}$$

As the sum is commutative, the order of the coordinates x_i is meaningless.
Starting from L_p, the Minkowski distance is defined as $d_p(\mathbf{x}, \mathbf{y}) = L_p(\mathbf{x} - \mathbf{y})$. The distance d_p is a metric for $p = 1, 2, 3, \ldots, \infty$, as it respects the two following axioms: (i) *Non-degeneracy* ($d_p(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$) and (ii) *Triangle inequality* ($d_p(\mathbf{x}, \mathbf{y}) \leq d_p(\mathbf{z}, \mathbf{x}) + d_p(\mathbf{z}, \mathbf{y})$). *Positivity* ($0 \leq d_p(\mathbf{x}, \mathbf{y})$) and *Symmetry* ($d_p(\mathbf{x}, \mathbf{y}) = d_p(\mathbf{y}, \mathbf{x})$) are properties of a metric that can be derived from the two previous axioms.
A special case is $p = 2$, leading to the Euclidean norm and distance. The Euclidean norm and distance are by far the most commonly used ones, because they correspond to the intuitive notions of length and distance in our three-dimensional world. Another important aspect of the Euclidean distance is that it is very practical when involved in objective functions expressed as a Sum of Squares Error (SSE). Typically, within the framework of vector quantization, for a set of vectors $\mathcal{X} = \{\mathbf{x}^j | 1 \leq j \leq N\}$, a SSE is written as

$$\text{SSE} = \frac{1}{2} \sum_{j=1}^{N} f(d_2^2(\mathbf{x}^j, \mathbf{y}^*)) \quad , \tag{2}$$

where $f : \mathbb{R} \to \mathbb{R}$ is a positive and monotonic function. For example, in the K-means algorithm, the vector quantization error is defined as in Eq. 2, with f being the identity and $\mathbf{y}^*$ being the closest prototype from the current data vector $\mathbf{x}^j$. Using a SSE that involves squared Euclidean distance is useful because its derivative is linear; optimizing the SSE is then possible with standard optimization tools like (stochastic) gradient ascent/descent. Differentiating $d_2^2(\mathbf{x}, \mathbf{y})$ with respect to coordinate x_i of $\mathbf{x}$ results in

$$\frac{\partial d_2^2(\mathbf{x}, \mathbf{y})}{\partial x_i} = 2\Delta_i \quad , \tag{3}$$

where $\Delta_i = (x_i - y_i)$. Therefore,

$$\frac{\partial \text{SSE}}{\partial y_i^*} = \frac{1}{2} f'(d_2^2(\mathbf{x}^j, \mathbf{y}^*)) \frac{\partial d_2^2(\mathbf{x}^j, \mathbf{y}^{k(j)})}{\partial y_i^{k(j)}} = f'(d_2^2(\mathbf{x}^j, \mathbf{y}^*))(x_i^j - y_i^*) \quad , \tag{4}$$

where f' is the derivative of f with respect to its argument.
As already mentioned, the Euclidean distance or any other distance derived from the Minkowski norm is insensible to the temporal structure of sequences and time series. In the next section, a generalization of the L_p norm to time-dependent vectors is proposed.

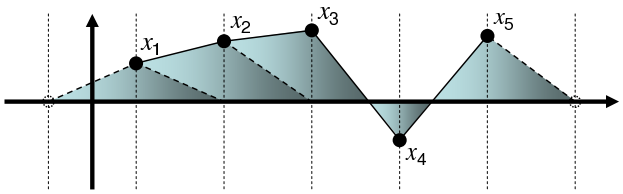

Figure 1: Illustration of the L_p^{TS} norm. The norm involves the areas of the triangles located on the left and right sides of each coordinate.

3 A generalization of the L_p norm to sequential vectors

With respect to Eq. 1, taking the temporal structure of sequences into account is achieved by involving the previous and next values of x_i in the i-th term of the sum, instead of x_i alone. Assuming that the sampling period τ is constant, the proposed norm is

$$L_p^{\mathrm{TS}} = \left(\sum_{i=1}^{D} (A_{i-1} + A_{i+1})^p \right)^{\frac{1}{p}} \quad , \tag{5}$$

where

$$A_{i-1} = \begin{cases} \frac{\tau}{2}|x_i| & \text{if } 0 \leq x_i x_{i-1} \\ \frac{\tau}{2} \frac{x_i^2}{|x_i|+|x_{i-1}|} & \text{if } 0 > x_i x_{i-1} \end{cases} \quad \text{and } A_{i+1} = \begin{cases} \frac{\tau}{2}|x_i| & \text{if } 0 \leq x_i x_{i+1} \\ \frac{\tau}{2} \frac{x_i^2}{|x_i|+|x_{i+1}|} & \text{if } 0 > x_i x_{i+1} \end{cases} \tag{6}$$

are respectively the areas of triangles on the left and right sides of x_i as shown in Fig. 1. Just as for L_p, the value of p is assumed to be a positive integer. At the left and right ends of the sequence, x_0 and x_D are assumed to be equal to zero. Assuming further that $\tau = 1$, it comes out that $L_p^{\mathrm{TS}} = L_p$ if the coordinates x_i are either all positive or all negative and $L_p^{\mathrm{TS}} \leq L_p$ otherwise. The computational cost of evaluating L_p^{TS} remains comparable to the one of L_p: the implementation requires only a few additional operations.
Defining $\delta_p(\mathbf{x}, \mathbf{y}) = L_p^{\mathrm{TS}}(\mathbf{x} - \mathbf{y})$, it can easily be shown that δ_p is a metric: it satifies the non-degeneracy and triangle inequality axioms, and thus possesses the same positivity and symmetry properties as the Minkowski distance. From the point of view of Functional Data Analysis [4, 6], time series are discretized functions of time. The distance between functions $x(t)$ and $y(t)$ can be measured by

$$\mathcal{D}_p(x, y) = \left(\int_t (x(t) - y(t))^p dt \right)^{\frac{1}{p}} \quad . \tag{7}$$

It can easily be seen that both d_p and δ_p are discrete approximations of $\mathcal{D}_p$. However, δ_p approximates $\mathcal{D}_p$ slightly better than d_p and this difference may be important when $\mathcal{D}_p(x, y)$ is small or when the sampling quality is poor (long sampling period and/or noisy observations). In the specific case where $p = 2$, simple developments show that

$$\frac{\partial \delta_2^2(\mathbf{x}, \mathbf{y})}{\partial x_i} = \frac{\tau^2}{2}(2 - u_{i-1} - u_{i+1})(v_{i-1} + v_{i+1})\Delta_i \quad , \tag{8}$$

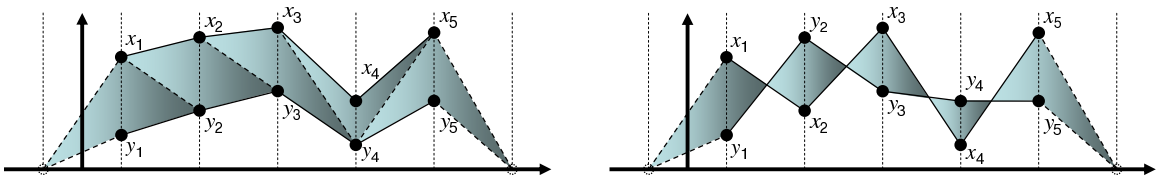

Figure 2: Illustration of the distance $\delta_p(\mathbf{x}, \mathbf{y})$ derived from the L_p^{TS} norm. Points are placed in such a way that the absolute differences $|\Delta_i| = |x_i - y_i|$ are equal on the left and right plot. This means that the Euclidean distance remains constant for both configurations. On the other hand, the proposed metric gives a shorter distance in the second configuration (areas are smaller).

where

$$u_{i-1} = \begin{cases} 0 & \\ \left(\frac{\Delta_{i-1}}{|\Delta_i|+|\Delta_{i-1}|}\right)^2 & \end{cases} \quad , \quad v_{i-1} = \begin{cases} 1 & \text{if } 0 \leq \Delta_i \Delta_{i-1} \\ \frac{|\Delta_i|}{|\Delta_i|+|\Delta_{i-1}|} & \text{if } 0 > \Delta_i \Delta_{i-1} \end{cases} \quad , \tag{9}$$

$$u_{i+1} = \begin{cases} 0 & \\ \left(\frac{\Delta_{i+1}}{|\Delta_i|+|\Delta_{i+1}|}\right)^2 & \end{cases} \quad , \quad v_{i+1} = \begin{cases} 1 & \text{if } 0 \leq \Delta_i \Delta_{i+1} \\ \frac{|\Delta_i|}{|\Delta_i|+|\Delta_{i+1}|} & \text{if } 0 > \Delta_i \Delta_{i+1} \end{cases} \quad . \tag{10}$$

As for the norm, it can easily be shown that $\delta_p(\mathbf{x}, \mathbf{y}) \leq d_p(\mathbf{x}, \mathbf{y})$, with equality reached if and only if the Δ_i are either all positive or all negative. On the other hand, neglecting border effects (or taking $D \to +\infty$), $\delta_p(\mathbf{x}, \mathbf{y}) = d_p(\mathbf{x}, \mathbf{y})/2$ when the signs of consecutive Δ_i alternate. Figure 2 shows that this property of the distance $\delta_p(\mathbf{x}, \mathbf{y})$ allows distinguishing fine differences in the trend of the time series $\mathbf{x}$ and $\mathbf{y}$, rather than taking additive noise into account. Two pairs $\{\mathbf{x}, \mathbf{y}\}$ of time series are shown on the left and right sides of Fig. 2 respectively. Both pairs are characterized by the same $|\Delta_i|$ values; this means that the distance $d_p(\mathbf{x}, \mathbf{y})$ is the same in both exemples. On the other hand, the value of $\delta_p(\mathbf{x}, \mathbf{y})$ is about half of the one of $d_p(\mathbf{x}, \mathbf{y})$. The use of the $\delta_p(\mathbf{x}, \mathbf{y})$ reaches the goal of characterizing the temporal aspect of vectors. Indeed, on the left of the figure, the series may be considered as significantly different, because of the systematic (here approximately constant) differences between their respective coordinates. On the right of the figure however, differences are not systematic and probably result much more from additive noise than from an expected or observed trend.
The L_p^{TS} metric proposed in this section can be used in any data analysis tool. The next section shows how it can be embedded into the conventional SOM algorithm.

4 Integrating the proposed metric into a SOM

From the point of view of vector quantization, a SOM may be seen as a generalization of competitive learning algorithms. For example, Eq. 2 with function f equal to the identity is the objective function of K-means. Usually, competitive learning algorithms implement the optimization of the SSE criterion by stochastic gradient descent. This leads to a simple update rule applied sequentially to each datum: the closest prototype is moved in the direction of the datum. This update rule is implemented through a partial derivative of the same form as in Eq. 4; the update of each coordinate y_i^* of the BMU $\mathbf{y}^*$ is given by

$$y_i^* \leftarrow y_i^* + \alpha \frac{\partial \mathrm{SSE}}{\partial y_i^*} \quad , \tag{11}$$

where α is a learning rate taking usually decreasing values between 0 and 1. This rule is called *Winner Takes All* because it updates a single prototype at the presentation of a datum. In a SOM, the update rule is generalized in order to update all prototypes (*Winner Takes Most*). For this purpose, the prototypes are given not only coordinates $\mathbf{y}^k$ in the D-dimensional data space but also fixed coordinates $\mathbf{z}^k$ on a (usually) two-dimensional topological map. Most often, the prototypes are evenly spaced on a rectangular or hexagonal grid. Taking this additional information into account, the update rule becomes

$$y_i^k \leftarrow y_i^k + \alpha h\left(\frac{d_2(\mathbf{z}^*, \mathbf{z}^k)}{\lambda}\right)\frac{\partial \mathrm{SSE}}{\partial y_i^k} \quad , \tag{12}$$

where k runs over all prototypes, $*$ is the index of the BMU and h is the so-called neighborhood function. This function decreases as the distance between the updated prototype and the BMU increases on the topological map; the parameter λ of the neighborhood function h decreases during the iterations of the algorithm.
In the update rule of the SOM, the proposed metric can be integrated into the partial derivative of the SSE, similarly to Eq. 4 and using the partial derivative from Eq. 8. As a consequence, the strength of the update is not isotropic anymore, as it was the case when using the Euclidean distance. Noisy patterns around a prototype have little influence on it; on the other hand, patterns that have a different trend or shape keep their full influence. To that extent, the proposed metric makes the SOM more robust with respect to additive noise in the case of time series.
Note that the distance $\delta_p(\mathbf{x}, \mathbf{y})$ could be used to characterize curves in any data analysis tool; however, it takes more importance in SOMs. Indeed in this algorithm, the most important distances are the ones used between each datum and its closest prototype. By construction, such distances are small. Figure 2 intuitively justifies that the distance $\delta_p(\mathbf{x}, \mathbf{y})$ differs from $d_p(\mathbf{x}, \mathbf{y})$ when time series $\mathbf{x}$ and $\mathbf{y}$ are very similar; in SOMs, the distance $\delta_p(\mathbf{x}, \mathbf{y})$ will thus seldom degenerate to $d_p(\mathbf{x}, \mathbf{y})$, confirming the potential interest of the former when data have a temporal structure.

5 Experiments

5.1 Material and method

A data set $\mathcal{X}$ of $N = 5000$ times series (or sequential vectors) is artificially generated with $D = 10$. Each time series $\mathbf{x}^j$ can be written as $x_i^j = a_j + b_j t_i$, where $t_i = (2i-1-D)/(2D)$. The offset a_j and slope b_j are randomly drawn from uniform distributions, ranging respectively from -1 to $+1$ and from -2 to $+2$. Gaussian noise with standard deviation σ between 0 and 1 by steps of 0.1 is then added to each coordinate x_i^j of each vector $\mathbf{x}^j$, in order to generate a second data set $\mathcal{Y}$. The first data set $\mathcal{X}$ is called the *reference* one and the second the *noisy* one. Figure 3 shows a small subset of the obtained time series. As these time series depend only on two parameters, they lie on a manifold whose intrinsic dimension is two in their ten-dimensional embedding space. Hence a rectangular two-dimensional SOM should be sufficient to represent the underlying manifold. A two-dimensional SOM including 75 prototypes (5-by-15 grid) evenly located on a hexagonal grid has been used. In the ten-dimensional space, prototype coordinates are initialized with the help of a PCA of the data set $\mathcal{Y}$.

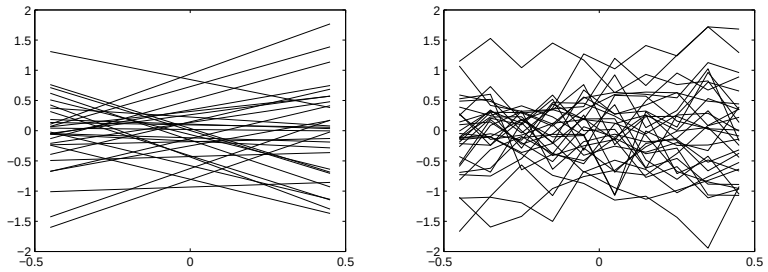

Figure 3: Thirty time series drawn from the data sets $\mathcal{X}$ and $\mathcal{Y}$, respectively without and with noise (standard deviation equal to 0.3).

The SOM algorithm is then run only on the noisy data set $\mathcal{Y}$. The whole $\mathcal{Y}$ is swept 60 times in random order during the SOM learning phase (60 epochs). The learning rate α varies from 0.5 to 0.05 and the neighborhood width λ varies from 30 to 0, both in a hyperbolic way. Experiments are carried out both with the Euclidean distance d_2 and the proposed distance δ_2. actually, according to the discussion about noise in the previous section, one expects that experiences realized with noisy vectors and using the distance δ_2 will give results that are closer from results on non-noisy vectors than experiences realized with the distance d_2. In order to measure these differences, the following quantities are measured for SOMs run with both distances:

1. The BMU-equality rate: the BMU with respect to either d_2 or δ_2 is computed for each vector $\mathbf{y}^j$ in $\mathcal{Y}$ and compared to the BMU of the corresponding reference vector $\mathbf{x}^j$. The rate is the proportion of pairs $\{\mathbf{x}_j, \mathbf{y}_j\}$ for which the BMU is the same.

2. The BMU-distance MSE: as for the previous criterion, BMUs are computed for all $\mathbf{x}_j$ and corresponding $\mathbf{y}_j$. Then for each j the Euclidean distance between both BMU is measured in the topological map space (in the hexagonal grid of the SOM). These distance are squared and averaged in order to obtain a MSE.

3. The residue MSE: as data $\mathbf{x}^j$ are generated using a linear model, a SOM that is run on $\mathcal{X}$ should yield prototypes with linearly dependent coordinates. However in these experiments, the SOM is run on the set $\mathcal{Y}$. The sensitivity to the noise (characterizing the differences between $\mathcal{X}$ and $\mathcal{Y}$) can then be measured by the deviation of the prototype shapes from a straight line. In order to measure this deviation, a linear model is fitted to the coordinates of each prototype and the residues are computed. They are then squared and averaged in order to obtain a MSE.

Each of these three criterions measures in some way the sensitivity to the noise level σ. The differences between the measures obtained with the Euclidean distance d_2 and the proposed distance δ_2 will thus reveal if it is less sensitive to noise than d_2, and thus more able to catch the time structure in the vectors.

5.2 Results and discussion

Figure 4 shows the prototypes of the SOM run on the $\mathcal{Y}$ set, using the distance d_2 (left) and the Euclidean one (right). Prototypes that are obtained with δ_2 are clearly smoother; they result

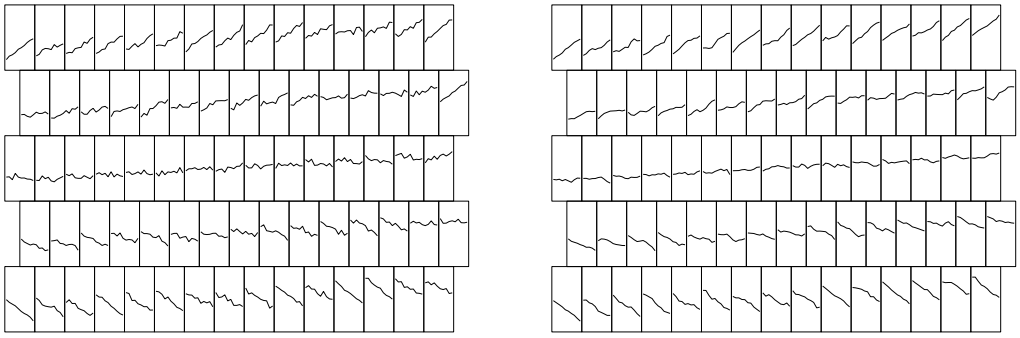

Figure 4: Representation of the 2D grid space of the SOMs; for each prototype, the corresponding time series (or D-dimensional coordinates) is drawn. Results for a SOM run on the data set described in Section 5.1 (standard deviation of noise is 0.3). On the left the SOM is run with the Euclidean distance d_2. On the right it is run with the proposed distance δ_2. As can be seen, the SOM running with δ_2 yields similar results, except that D-dimensional coordinates of prototypes are less noisy than with the Euclidean distance.

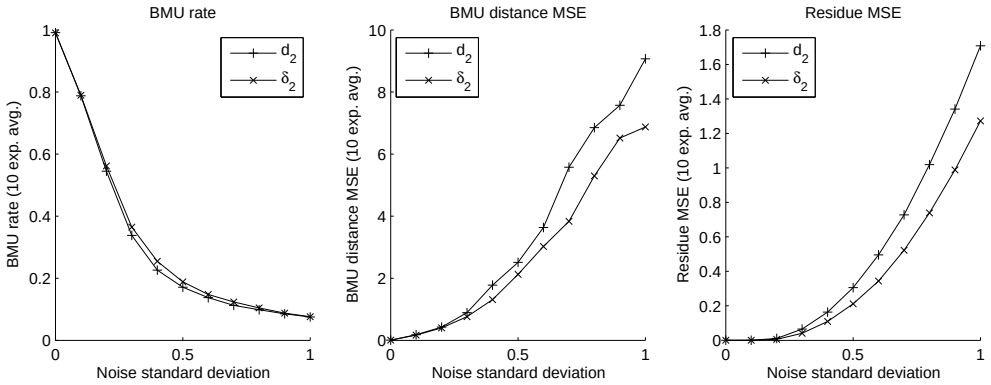

Figure 5: BMU-equality rate, BMU-distance MSE, and residue MSE, w.r.t. the level of noise.

from the quantization of the time-dependent vectors and are not too influenced by the added noise. High-frequency components of the white noise are filtered. On the contrary, prototypes on the left are more influenced by noise. Figure 5 shows values of the three measures detailed in the previous section. The left plot shows the BMU-equaliy rate that decreases with the level of noise, as expected. The use of the distance δ_2 shows a slightly improved insensitivity to noise. This difference is more obviously noticed in the BMU-distance MSE criterion (middle plot) that measures the mismatch distance on the map instead of the match rate. The right plot illustrates the ability of the SOM to learn the temporal dependencies in the vectors versus the added noise; of course the first is expected, while the second should be avoided as much as possible. In the first case, the prototype representations will be close to straight lines, whereas in the second case they will be polluted by noise. The right part of Fig. 4 clearly shows a greater ability of the distance δ_2 to neglect the high-frequency components of the noise and thus to capture the low-frequency trends.

Contrarily to other techniques, the use of δ_2 does not require any data preprocessing. More-

over, the proposed metric is model-free and parameter-free. For example, in Functional Data Analysis, distances between functions (or time series) are sometimes computed as distances between vectors of regression parameters. This forces the user to design a particular regression model. The use of filters leads to the same problem.

6 Conclusion

This paper presents a simple distance measure able to exploit the temporal structure of sequences or time series. Unlike the Euclidean or Minkowski distances, it takes into account the numbering of the vector coordinates, making it possible to extract additional information from the vectors, which is usually lost in the processing. The proposed distance is integrated into Self-Organizing Maps, and tested on a set of artificially generated time series. It is shown by algorithmic considerations and by simulations that the use of the proposed distance both takes into account the temporal structure of data and reduces the influence of additive noise. Functional Data Analysis (FDA) methods [4, 6] also take the temporal structure of data into account, by fitting splines or other basis functions to data; the principle is applicable to SOMs too [5]. Compared to FDA, the proposed distance measure allows working with low-dimensional vectors (a 10-dimensional example is illustrated in this paper), is simple, computationally efficient, non-parametric and does not force any smoothing.
Future work aims at using the proposed metric on short time series of medical measurements (typically the effect of drugs on living organisms). Another application will be the analysis of regressors built from a single time series using sliding windows.

References

[1] G. J. Chappell and J. G. Taylor. The temporal Kohonen map. *Neural Networks*, 6:441–445, 1993.

[2] B. Hammer and T. Villmann. Classification using non-standard metrics. In M. Verleysen, editor, *Proceedings of ESANN 2005, 12th European Symposium on Artificial Neural Networks*, pages 303–316. D-side public., Bruges (Belgium), April 2005.

[3] T. Kohonen. *Self-Organizing Maps*. Springer, Heidelberg, 2nd edition, 1995.

[4] J. Ramsay and B. Silverman. *Functional Data Analysis*. Springer Series in Statistics. Springer (Berlin), 1997.

[5] F. Rossi, B. Conan-Guez, and A. El Golli. Clustering functional data with the SOM algorithm. In M. Verleysen, editor, *Proceedings of ESANN 2004, 11th European Symposium on Artificial Neural Networks*, pages 305–312. D-side public., Bruges (Belgium), April 2004.

[6] F. Rossi, N. Delannay, B. Conan-Guez, and M. Verleysen. Representation of functional data in neural networks. *Neurocomputing*, 64:183–210, March 2005.

[7] G. Simon, A. Lendasse, M. Cottrell, J.-C. Fort, and M. Verleysen. Double quantization of the regressor space for long-term time series prediction: method and proof of stability. *Neural Networks*, 17:1169–1181, 2004.