Neurocomputing 72 (2009) 1444–1454

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Nonlinear data projection on non-Euclidean manifolds with controlled trade-off between trustworthiness and continuity

V. Onclinx [a,*], V. Wertz [a], M. Verleysen [b]

[a] Machine Learning Group, CESAME, Université catholique de Louvain, Av. Georges Lemaitre, 4, B-1348 Louvain-la-Neuve, Belgium
[b] Machine Learning Group, DICE, Université catholique de Louvain, Place du Levant, 3, B-1348 Louvain-la-Neuve, Belgium

ARTICLE INFO

Available online 10 January 2009

Keywords:
Nonlinear dimensionality reduction
Distance-based data projection method
Trade-off between trustworthiness and continuity
Optimization on manifolds

ABSTRACT

This paper presents a framework for nonlinear dimensionality reduction methods aimed at projecting data on a non-Euclidean manifold, when their structure is too complex to be embedded in an Euclidean space. The methodology proposes an optimization procedure on manifolds to minimize a pairwise distance criterion that implements a control of the trade-off between trustworthiness and continuity, two criteria that, respectively, represent the risks of flattening and tearing the projection. The methodology is presented as general as possible and is illustrated in the specific case of the sphere.

1. Introduction

Measuring and collecting large numbers of data and features become nowadays increasingly easy. For this reason system identification, machine learning, data mining and other industrial and research domains have more and more to deal with high-dimensional data. Before a quantitative analysis of such data, acquiring prior knowledge is of primary importance to guide the choice between models and data analysis methods. Extracting relevant and useful information may be performed by data projection, or dimensionality reduction methods [9,13,18,23,26,29,38]; data projection aims at visualizing high-dimensional data in a lower dimensional space, for example a two- or three-dimensional representation. Note that data projection must be understood here as covering both linear and nonlinear methods. By removing or reducing redundancies and noise, data projection makes easier the observation of proximity relationships between data, specificities in the data distribution such as clusters, etc.

Data projection methods try to minimize the loss of information between the original data and the projected ones. The loss of information can, however, be defined in various ways, including the ability of the method to preserve distances between data [15,16,26,29,32–34], and/or to preserve the topology or the neighbourhood relationships [2,4,13,17,39]. Methods based on these families of criteria are named distance-based and topology-based dimensionality reduction methods, respectively. CCA and CDA [15,16] for example aim at minimizing a suitable pairwise distance criterion, while Isotop [17] and LLE [12,28,31] try to preserve the neighbourhoods, i.e. the local topology.

Most of these methods have been developed to project data on a low-dimensional Euclidean manifold, i.e. $\mathbb{R}^2$ or $\mathbb{R}^3$ in most cases. Nevertheless, when the data structure is too complex, restricting the projection space to an Euclidean manifold constrains the method and does not make use of a possible specific nature of the data.

To circumvent this problem, first ideas are sketched in manifold learning methods where original data are assumed to be close to an unknown manifold, i.e. a topological space which can be Euclidean only locally. Some manifold learning methods motivate the use of graph distances to avoid some shortcomings due to Euclidean distances; [7,8,16,33,34] relate the graph distance to the geodesic distance.

Other methods directly use the structure of the manifold to improve the projection results. Using geometric arguments, these methods map the data onto a tangent space of the manifold which is therefore Euclidean [6,21,44]. However, they make the strong hypothesis that the manifold does not intercept itself: loops in the data set are not allowed. Having for example in mind a cylindrical or spherical manifold, it is easy to see that its projection cannot avoid flattening or tearing effects, possibly yielding a poor global projection quality.

Topology-based dimensionality reduction methods such as self-organizing maps release this hypothesis by projecting data on manifolds; actually, in this context, the projection on manifolds with loops, such as spheres, cylinders and tori, is widely used [11,20,22,27,30,41,42]. In addition to allowing a better projection of distribution with loops, projecting on spheres or tori reduces border effects that make interpretation of the resulting visualization more difficult. These possible advantages

* Corresponding author.
E-mail addresses: victor.onclinx@uclouvain.be (V. Onclinx), vincent.wertz@uclouvain.be (V. Wertz), michel.verleysen@uclouvain.be (M. Verleysen).

0925-2312/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.12.018

are, however, not exploited in distance-based dimensionality reduction methods.

This paper presents a methodology for distance-based dimensionality reduction on manifolds, even with loops. It offers the already mentioned advantages of increasing the flexibility to adapt the projection to the intrinsic nature of the data, and of reducing the border effects if a compact manifold is chosen.

When data are projected on non-Euclidean manifolds, visualization can become more difficult (though some non-Euclidean manifolds are easy to visualize, like the sphere, the cylinder, etc.). The choice between projecting on Euclidean or non-Euclidean manifolds is thus both data- and application-dependent. This paper does not address this question, which is left for future research. Note, however, that the same question is important in the context of topology-based projection methods (such as the self-organizing map) too, and is nowadays mostly unanswered: when is it better to use a rectangular grid, a cylindrical one, a torus, etc. Only the expertise of the analyzer and some possible information from the application is usually used to guide this choice.

This paper first presents the methodology in the general case where there is no restriction on the manifold, except some regularity hypotheses; then, as an example, we will show how we can take advantage of this framework to develop a projection method on the surface of a sphere. Using a similar derivation, one could develop methods to project on other manifolds.

A limitation of most traditional distance-based dimensionality reduction methods is that they optimize a criterion that is designed either to minimize the tearing effects or the flattening ones. In this paper, a flexible criterion is adopted, in which the user can define *a priori* (or by visualization feedback) the compromise between these effects. It is shown that this flexibility can be used to achieve a specific compromise between the trustworthiness and the continuity of the projection, as detailed in [36–38,40].

The sequel of this paper is organized as follows. Section 2 introduces the pairwise distance criterion and its relation to quality criteria such as the trustworthiness and the continuity. Section 3 then presents an appropriate optimization procedure of this criterion, resulting in a projection on manifolds. To take into account the manifold topology and its curvature, a line-search approach is adopted, making use of the theory of the optimization on manifolds [1]. The optimization procedure is first described in a generic way, and then detailed in the specific case of the projection on a sphere. Finally, first results are presented in Section 4, including comparisons between the projection on a sphere and the projection on the $\mathbb{R}^2$ Euclidean space. The ideas of this paper were first sketched in [25].

2. Pairwise distance criterion

Distance-based dimensionality reduction methods attempt to make the distances between data in the projection space as close as possible to the corresponding distances in the original space. Making all corresponding distances equal is of course impossible: the degrees of freedom are the locations of the data in the projection space, while the constraints are the equality of the distances between all pairs of data. In a standard situation (many data, low-dimensional projection space), there will be much more constraints than degrees of freedom. Furthermore, since projection methods aim at visualizing the data distribution, the projection space is often $\mathbb{R}^2$ (sometimes $\mathbb{R}^3$). In this context, the dimension of the projection space is usually smaller than the intrinsic dimension of the original data such that loss of information cannot be avoided.

It is thus necessary to define an error (or objective) criterion which will be minimized to achieve the projection results. As it is illustrated in Section 2.1 through the simple example of a cylindrical distribution projected on the two-dimensional Euclidean space, some compromise between flattening and tearing is bound to happen. Section 2.2 defines a pair of measures, trustworthiness and continuity [36] to count the number of errors linked to flattening and tearing. These errors, based on ranks, are discrete and thus difficult to optimize directly. Alternative continuous measures are then presented in Section 2.3, and combined in a single criterion including a user-controlled trade-off parameter. These error criteria are independent from a possible constraint on the manifold shape. Optimizing the criterion under manifold constraints will be addressed in Section 3.

2.1. Flattening and tearing: an illustrative example

Data projection methods have to deal with a trade-off between the risks of flattening and tearing the data distribution. This compromise is presented by an illustrative example. Let us assume that data lie close to a cylinder embedded in the three-dimensional Euclidean space. In order to project the data on the $\mathbb{R}^2$ Euclidean space, one option is to rip the manifold along a generating line. After ripping, the cylinder can easily be unfolded on the $\mathbb{R}^2$ Euclidean space as shown in Fig. 1. In this simple example, all pairs of data that are close in the $\mathbb{R}^2$ projection space are also close in the original space. The projection is said to be trustworthy, as what is seen (proximity relationships in the projection space) can be trusted. However, because the cylinder has been torn, the projection is not continuous: close input data (in the original space) do not necessarily lead to close output data (in the projection space).

Another option to project the cylinder on $\mathbb{R}^2$ is to flatten it, as illustrated in Fig. 2. In this case, two opposite generating lines are projected one on another. In this case, all pairs of data that are close in the original space remain close in the projection space; the projection is said to be continuous. However, it is not

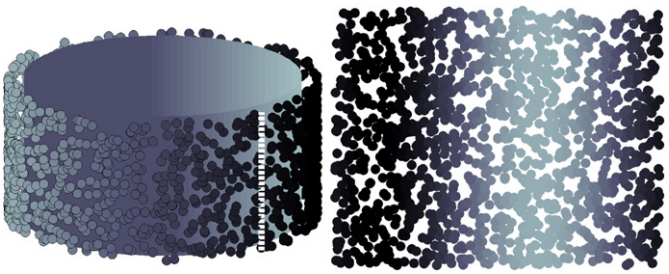

Fig. 1. The cylinder is torn when projected on $\mathbb{R}^2$; the resulting projection is trustworthy but not continuous.

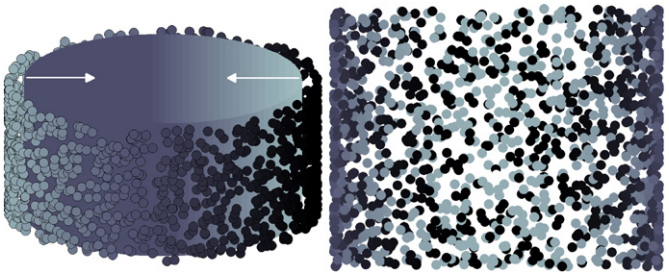

Fig. 2. The cylinder is flattened when projected on $\mathbb{R}^2$; the resulting projection is continuous but not trustworthy.

trustworthy anymore: data that are seen close (in the projection space) may come from opposite parts of the original distribution.

There is no answer to the question whether it is better to have a trustworthy or a continuous projection. This depends on the point of view of the user and on the objective of the projection in the application context. For this reason, it is proposed in Section 2.3 to optimize a user-controlled compromise between both criteria.

2.2. Trustworthiness and continuity quality measures

The two types of difficulties encountered when projecting a manifold with loops on an Euclidean space naturally lead to two quality measures that count points that are close in one space but not in the other space [36]. Let us consider a data set $\{\mathbf{x}_i, 1 \leqslant i \leqslant N\}$ in the original high-dimensional space, and the corresponding data set $\{\mathbf{y}_i, 1 \leqslant i \leqslant N\}$ in the projection space.

Measuring if the projection is trustworthy consists in first selecting the k closest points to $\mathbf{y}_i$ in the projection space, or in other words the points in the k-neighbourhood of $\mathbf{y}_i$. Then, the corresponding points in the original space are identified. If they are all in the k-neighbourhood of $\mathbf{x}_i$, then the projection is fully trustworthy. The set of those points that are not in the k-neighbourhood of $\mathbf{x}_i$ while their corresponding points are in the k-neighbourhood of $\mathbf{y}_i$ is denoted $U_k(\mathbf{x}_i)$. Let $r(\mathbf{x}_i, \mathbf{x}_j)$ be the rank distance of point $\mathbf{x}_j$ with respect to $\mathbf{x}_i$ defined by: $r(\mathbf{x}_i, \mathbf{x}_j) = s$ if $\mathbf{x}_j$ is the sth-nearest neighbour of $\mathbf{x}_i$; if the point $\mathbf{x}_j$ belongs to the set $U_k(\mathbf{x}_i)$, the rank $r(\mathbf{x}_i, \mathbf{x}_j)$ is thus larger than k. Averaging these ranks over all points in $U_k(\mathbf{x}_i)$ and for all $\mathbf{x}_i$ leads to the trustworthiness measure [36]:

$$T_k = 1 - A_k \sum_{i=1}^{N} \sum_{\mathbf{x}_j \in U_k(\mathbf{x}_i)} (r(\mathbf{x}_i, \mathbf{x}_j) - k), \tag{1}$$

where $A_k = 2/Nk(2N - 3k - 1)$ is a coefficient that ensures the bounds $0 \leqslant T_k \leqslant 1$.

Conversely, the continuity considers the set $V_k(\mathbf{y}_i)$ of projected points that are not in the k-neighbourhood of $\mathbf{y}_i$ while the corresponding points are in the k-neighbourhood of $\mathbf{x}_i$ in the original space. The rank distance in the projected space is denoted by $\hat{r}(\mathbf{y}_i, \mathbf{y}_j)$; the continuity measure is then defined by [36]

$$C_k = 1 - A_k \sum_{i=1}^{N} \sum_{\mathbf{y}_j \in V_k(\mathbf{y}_i)} (\hat{r}(\mathbf{y}_i, \mathbf{y}_j) - k). \tag{2}$$

According to these definitions, the projection is trustworthy or continuous if the corresponding quality measure is close to 1. Again, except in specific circumstances such as the original data lying on an Euclidean manifold, no projection method can achieve both perfect trustworthiness and continuity. Comparisons between methods should therefore always keep this trade-off in mind.

Note that other criteria have been defined in the literature to measure the number of points that are close in one space but not in the other one. The mean relative rank errors [18] (MRREs) are among them. They differ from the trustworthiness and continuity by the weightings applied to the ranks of the points in the U_k and V_k neighbourhoods. Recently, a unifying framework for rank-based error criteria in dimensionality reduction methods has been defined in [19].

2.3. Flattening and tearing errors

Once a compromise between trustworthiness and continuity has been set, optimizing these criteria (or a mixture of them) with respect to the locations $\mathbf{y}_i$ in the low-dimensional space would define the projection. However, the criteria are discrete and their optimization is therefore difficult.

To circumvent this difficulty, most distance-based dimensionality reduction methods do not optimize ranks, but some weighted differences between the distances D_{ij} between $\mathbf{x}_i$ and $\mathbf{x}_j$ in the original space and the corresponding distances δ_{ij} in the projected space.

Let us first define the following unweighted criterion that corresponds to the criterion of the classical multi-dimensional scaling [35,43]:

$$f = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} (D_{ij} - \delta_{ij})^2. \tag{3}$$

The problem with this criterion is that errors between large D_{ij} and δ_{ij} distances will dominate. The projection will therefore be more influenced by pairs of points that are far one from the other, rather than by pairs of close points. This is against the intuition: pairs of neighbouring points should dominate, as in the trustworthiness and continuity measures.

The solution is then to give more weight to pairs of close points. Dividing each term by the distance D_{ij} in the original space favours the continuity of the projection: a pair with small D_{ij} and large δ_{ij} will largely contribute to the error function [29]. The tearing error is then defined as

$$T_{err} \equiv \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \frac{(D_{ij} - \delta_{ij})^2}{D_{ij}}. \tag{4}$$

Conversely, by weighting each term with the corresponding distance δ_{ij} in the projected space, the trustworthiness of the projection is favoured [14]. Actually, if two projected data are close, i.e. δ_{ij} is small, whereas the corresponding points are not close in the original space, the flattening error increases:

$$F_{err} \equiv \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \frac{(D_{ij} - \delta_{ij})^2}{\delta_{ij}}. \tag{5}$$

Finally, as already argued, the trade-off between the flattening and tearing errors should be controlled by the user [37,39,40]. This leads to the final cost function to be optimized:

$$f \equiv \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \left(\lambda \frac{(D_{ij} - \delta_{ij})^2}{D_{ij}} + (1 - \lambda) \frac{(D_{ij} - \delta_{ij})^2}{\delta_{ij}} \right), \tag{6}$$

where $\lambda \in [0, 1]$ is a user-defined parameter controlling the trade-off.

3. Projection on manifolds

This section explains how to minimize the cost function (6) presented above. Because the projected points have to lie on a manifold embedded in an Euclidean space, three main options can be considered. First, one can simply consider using classical constrained optimization techniques [5,24] and introducing Lagrange multipliers to deal with the constraints that each of the projected points has to satisfy to belong to the desired manifold. Note, however, that in this case, the optimization problem has as many constraints as the number of points which means in our problem N constraints, and hence N additional optimization variables (the Lagrange multipliers).

A second option would be to parameterize the optimization problem using coordinates which directly express that the projection belongs to the manifold. A parametric representation of the manifold is, however, not always easily available (although it is the case for the sphere that we will use later); moreover, such a representation may have singularities which can result in

numerical problems. As an example, the *north pole* of the sphere is defined in the spherical coordinate space by the set $\{(\phi,\theta)|\theta=\pi/2, 0\leqslant\phi\leqslant 2\pi\}$. Furthermore, the implementation of the optimization procedure has to deal with the non-Euclidean topology of the projection manifold (the loops in data); in the case of the sphere, the points corresponding to the extreme values of the azimuth angle (0 and 2π) must be identical. These topology considerations are difficult to deal with when computing the gradients of the optimization procedures.

The third option, which is followed here, is to apply the theory of optimization on manifolds [1]. In this context, the problem is reformulated as an unconstrained optimization problem in a constrained search space; the geometric constraints express that projected data must lie on a manifold at each step of the optimization procedure but without other restriction. Because the manifold is embedded in a larger dimensional Euclidean space, the optimization on manifolds algorithm has a lower complexity than classical constrained optimization algorithms on the embedding space. This alternative has thus good numerical properties and convergence results [1].

In Section 3.1, we will describe how the optimization problem under consideration can be solved by making use of this theory in the general case where no specific manifold is considered. Section 3.2 will then deal with the specific optimization procedure to project data on the surface of a sphere.

3.1. General algorithm for projection on manifolds

To search for a minimum of a cost function f, the classical line-search algorithm (negative gradient search) is adapted. Actually, optimizing on a manifold does not allow movement along a straight line as it is the case in classical optimization procedures. However, the curves of the manifold can favourably replace the straight search directions since they include the curvature of the manifold and the specificities of the manifold topology.

Section 3.1.1 introduces the principal concepts of differential geometry while Section 3.1.2 briefly presents the theory of optimization on manifolds. Then Section 3.1.3 develops the methodology to minimize the cost function (6) on a manifold.

3.1.1. Differential geometry concepts

For the description of the algorithm, concepts from differential geometry are used: a d-dimensional manifold $\mathcal{N}$ is informally defined as a set in which the neighbourhood of each point is homeomorphic to a subset of the $\mathbb{R}^d$ Euclidean space.

Therefore, the product of two manifolds is also a manifold. In other words, let $\mathcal{N}_1$ and $\mathcal{N}_2$ be two manifolds with respective dimensions d_1 and d_2; the Cartesian product of these two manifolds $\mathcal{M}\equiv\mathcal{N}_1\times\mathcal{N}_2$ is the (d_1+d_2)-dimensional manifold of all pairs (y_1,y_2) such that the resulting manifold is defined by $\mathcal{M}=\{(y_1,y_2)|y_1\in\mathcal{N}_1, y_2\in\mathcal{N}_2\}$.

Moreover, we consider here differentiable manifolds embedded in Euclidean spaces so that the tangent space of the manifolds can be evaluated for each point $\mathbf{y}_i\in\mathcal{N}$. Assuming that the geometric constraints of the manifold $\mathcal{N}$ are defined by a function $\mathbf{F}\in\mathscr{C}^1(\mathcal{N})$:

$$\mathcal{N}\equiv\{\mathbf{y}_i|\mathbf{F}(\mathbf{y}_i)=0\}, \tag{7}$$

its tangent space $\mathscr{T}_{\mathbf{y}_i}\mathcal{N}$ evaluated at point $\mathbf{y}_i$ is defined by

$$\mathscr{T}_{\mathbf{y}_i}\mathcal{N}\equiv\{\mathbf{u}_i|\mathbf{u}_i^{\mathrm{T}}\nabla\mathbf{F}(\mathbf{y}_i)=0\}, \tag{8}$$

where $\nabla\mathbf{F}$ is the gradient of $\mathbf{F}$.

Note that it can be interesting to also consider a family of manifolds $\mathcal{N}_{\boldsymbol{v}}$ parameterized by a vector v. For example, in Section 3.2, the radius of the sphere is such a parameter.

In this paper, we use this concept to project N data on a manifold. Since each projected data $\mathbf{y}_i$ lies on the same manifold $\mathcal{N}_{\boldsymbol{v}}$, the set of constraints is expressed by

$$\begin{cases}\mathbf{y}_1\in\mathcal{N}_{\boldsymbol{v}},\\ \quad\vdots\\ \mathbf{y}_N\in\mathcal{N}_{\boldsymbol{v}}.\end{cases} \tag{9}$$

This projection will be performed by minimizing the cost function (6) with respect to all the N projected data $\mathbf{y}_i$. Hence, the cost function can be specified as

$$\begin{aligned}f:\mathcal{N}_v\times\cdots\times\mathcal{N}_v\to\mathbb{R}:(\mathbf{y}_1,\ldots,\mathbf{y}_N)&\mapsto f(\mathbf{y}_1,\ldots,\mathbf{y}_N)\\ &=\frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\left(\lambda\frac{(D_{ij}-\delta_{ij})^2}{D_{ij}}+(1-\lambda)\frac{(D_{ij}-\delta_{ij})^2}{\delta_{ij}}\right).\end{aligned} \tag{10}$$

To keep the notation short, the vector $\mathbf{y}=(\mathbf{y}_1,\ldots,\mathbf{y}_N)$ that gathers the projected data is defined on the manifold $\mathcal{M}_{\boldsymbol{v}}$ which is the Cartesian product of N manifolds $\mathcal{N}_{\boldsymbol{v}}$:

$$\mathcal{M}_{\boldsymbol{v}}\equiv\mathcal{N}_{\boldsymbol{v}}\times\mathcal{N}_{\boldsymbol{v}}\times\cdots\times\mathcal{N}_{\boldsymbol{v}}. \tag{11}$$

It follows that the cost function is defined on the manifold $\mathcal{M}_v$.

Next, the manifold $\mathcal{M}_{\boldsymbol{v}}$ is reformulated using the function $\vec{\mathbf{F}}\in\mathscr{C}^1(\mathcal{M}_{\boldsymbol{v}})$ that expresses the geometric constraints which the vector $\mathbf{y}$ has to satisfy

$$\mathcal{M}_{\boldsymbol{v}}\equiv\{(\mathbf{y}_1,\ldots,\mathbf{y}_N,\boldsymbol{v})|\vec{\mathbf{F}}(\mathbf{y}_1,\ldots,\mathbf{y}_N,\boldsymbol{v})=0\}. \tag{12}$$

By differentiating $\vec{\mathbf{F}}$, the tangent space of the manifold $\mathcal{M}_{\boldsymbol{v}}$ at point $(\mathbf{y}_1,\ldots,\mathbf{y}_N,\boldsymbol{v})$ is defined as the set of vectors $\mathbf{u}=(\mathbf{u}_1,\ldots,\mathbf{u}_N,\mathbf{u}_v)$ satisfying

$$\mathscr{T}_{\mathbf{y}}\mathcal{M}_{\boldsymbol{v}}\equiv\{(\mathbf{u}_1,\ldots,\mathbf{u}_N,\mathbf{u}_v)|(\mathbf{u}_1,\ldots,\mathbf{u}_N,\mathbf{u}_v)^{\mathrm{T}}J\vec{\mathbf{F}}(\mathbf{y}_1,\ldots,\mathbf{y}_N,\boldsymbol{v})=0\}, \tag{13}$$

where $J\vec{\mathbf{F}}(\mathbf{y}_1,\ldots,\mathbf{y}_N,\boldsymbol{v})$ is the Jacobian of $\vec{\mathbf{F}}$ evaluated at point $(\mathbf{y}_1,\ldots,\mathbf{y}_N,\boldsymbol{v})$.

3.1.2. Optimization on manifolds

This section is dedicated to the theory of optimization on manifolds. The section first presents the standard line-search algorithm which is aimed at finding a point z^* that minimizes a cost function f without any additional constraint. Conceptually, this algorithm translates a point $z(t)$ in a gradient-related search direction to find a minimum.

Second, the proposed optimization procedure that deals with the manifold constraints is briefly described to show how the standard line-search algorithm has to be adapted. The algorithm requires mainly two adaptations. One has to define a search direction that takes into account the manifold constraints and the translation of the current point on the manifold has to be performed.

The proposed optimization algorithm is an adaptation of the standard line-search direction algorithm. Assuming that the standard algorithm has to find a minimum $z^*\in\mathbb{R}^d$ of a cost function $f:\mathbb{R}^d\to\mathbb{R}$, the classical algorithm translates a point $\mathbf{z}(t)$ along a descent direction $\boldsymbol{\eta}_t$ until a minimum is found; the algorithm is thus iterative. Briefly described, assuming that the classical line-search algorithm has successfully reached the k first iterations, the location $\mathbf{z}(k)$ is updated by searching in a gradient-related direction $\boldsymbol{\eta}_k$ with a step size t_k that ensures a sufficient decrease of the cost function; this standard algorithm follows the following scheme:

$$\mathbf{z}(k+1)=\mathbf{z}(k)+t_k\boldsymbol{\eta}_k. \tag{14}$$

The theory of optimization on manifolds [1] adapts the standard line-search scheme (14) to minimize a cost function $f:\mathcal{N}\to\mathbb{R}$ on a manifold $\mathcal{N}$. Note that we just consider here the minimization

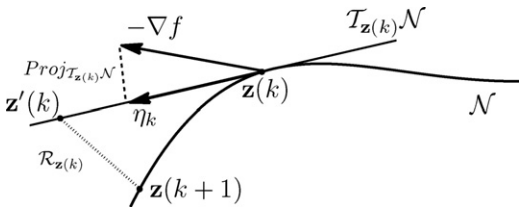

Fig. 3. Optimization iteration.

of a cost function on a manifold, not the projection on a manifold; to avoid confusion, the notation $\mathbf{z}(k)$ is thus used. Two main tasks will be described in the following of the section:

- An adequate search-direction η_k has to be defined to consider the characteristics of the manifold.
- The current location $\mathbf{z}(k)$ has to be translated with respect to the direction η_k while staying on the manifold.

The algorithm described below briefly introduces the main steps which will be detailed in the sequel of the paper.

Optimizing on a manifold has to consider the geometric constraints due to the manifold. As shown in Fig. 3 that illustrates a single iteration of the adapted line-search algorithm, the search direction $\boldsymbol{\eta}_k$ has to stay close to the manifold, hence on the tangent space $\mathcal{T}_{\mathbf{z}(k)}\mathcal{N}$. This direction is achieved by orthogonally projecting the descent direction $-\nabla f$ on the tangent space:

$$\eta_k = Proj_{\mathcal{T}_{\mathbf{z}(k)}\mathcal{N}}(-\nabla f). \tag{15}$$

Secondly, the algorithm translates the location $\mathbf{z}(k)$ in the gradient-related direction η_k with a step size t_k. Let $\mathbf{z}'(k) = \mathbf{z}(k) + t_k\eta_k$ be the new location; this one is in the affine subspace $\mathbf{z}(k) + \mathcal{T}_{\mathbf{z}(k)}\mathcal{N}$. Next, $\mathbf{z}'(k)$ is "retracted" on the manifold $\mathcal{N}$ to satisfy the manifold constraint; the retraction function $\mathcal{R}_{\mathbf{z}(k)}$: $\mathcal{T}_{\mathbf{z}(k)}\mathcal{N} \to \mathcal{N}$ is a deterministic projection that smoothly maps vectors of the tangent space to the manifold. The new iterate is thus defined by

$$\mathbf{z}(k+1) = \mathcal{R}_{\mathbf{z}(k)}(t_k\eta_k). \tag{16}$$

This general scheme describing the principle of the optimization on manifolds will be detailed (tangent space and retraction function) in the next section in the specific case of the projection on manifolds (optimization of Eq. (6)). It is shown in [1,3] that optimizing a function on a differentiable manifold by such a gradient-based technique converges, provided that the search direction is gradient-related and that the step size is chosen; for example, by the Armijo rule as it will be explained later.

3.1.3. Projection on manifolds

This section implements the above described general procedure for optimizing a function on a manifold, in the specific case of the projection (Eq. (6)). Let us denote

$$\mathbf{y}(k) = (\mathbf{y}_1(k), \ldots \mathbf{y}_N(k)), \tag{17}$$

where vector $\mathbf{y}_i(k)$ is the projection of the ith data on the manifold at iteration k. It is thus necessary to define the search direction η_k, the retraction function $\mathcal{R}_{\mathbf{y}(k)}$, and the step size t_k.

Search direction: To project on a manifold $\mathcal{M}_v$, we first have to define a search direction η_k. The search direction is determined by evaluating the gradient $\nabla f(\mathbf{y}_1(k), \ldots, \mathbf{y}_N(k), \boldsymbol{v}(k))$ of the cost function f with respect to the variables $\mathbf{y}_i$ and with respect to the manifold parameter vector $\boldsymbol{v}$ if appropriate.

The aim of the gradient is to evaluate locally the decrease of the cost function f. However, because of the geometric constraints, this direction may point far away from the manifold $\mathcal{M}_{\boldsymbol{v}(k)}$. In order to include the curvature of the manifold, the gradient is projected on the tangent space $\mathcal{T}_{\mathbf{y}(k)}\mathcal{M}_{\boldsymbol{v}(k)}$ related to the location $\mathbf{y}(k)$. The gradient-related search direction η_k is thus defined by

$$\eta_k = Proj_{\mathcal{T}_{\mathbf{y}(k)}\mathcal{M}_{v(k)}}(-\nabla f). \tag{18}$$

Translation of a point and retraction function: Vector $\mathbf{y}(k)$ can thus be translated in the direction η_k. However, translating the point in a straight direction is not allowed since the point will go far away from the manifold. To stay on the manifold, $\mathbf{y}(k)$ has to move along a curve $\gamma_{\mathbf{y}(k)}(t)$ which is tangent to the direction η_k at $\mathbf{y}(k)$.

The construction of such a curve can be achieved by a retraction function $\mathcal{R}_{\mathbf{y}(k)} : \mathcal{T}_{\mathbf{y}(k)}\mathcal{M}_{v(k)} \mapsto \mathcal{M}_v$ which is a deterministic smooth mapping from the tangent space $\mathcal{T}_{\mathbf{y}(k)}\mathcal{M}_{v(k)}$ to the manifold $\mathcal{M}_v$, as illustrated in Fig. 4. This retraction function satisfies the two following conditions.

1. Denoting $0_{\mathbf{y}(k)}$ the zero element of the tangent space $\mathcal{T}_{\mathbf{y}(k)}\mathcal{M}_{v(k)}$, the retraction function has to map this point $0_{\mathbf{y}(k)}$ on $\mathbf{y}(k)$:
$$\mathcal{R}_{\mathbf{y}(k)}(0_{\mathbf{y}(k)}) = \mathbf{y}(k). \tag{19}$$
2. Differentiating the retraction function satisfies
$$D\mathcal{R}_{\mathbf{y}(k)}(0_{\mathbf{y}(k)}) = id_{\mathcal{T}_{\mathbf{y}(k)}\mathcal{M}_{v(k)}}, \tag{20}$$
where $id_{\mathcal{T}_{\mathbf{y}(k)}\mathcal{M}_{v(k)}}$ denotes the identity mapping on $\mathcal{T}_{\mathbf{y}(k)}\mathcal{M}_{v(k)}$. The identity mapping satisfies the two following equations:
$$\forall \eta \in \mathcal{T}_{\mathbf{y}(k)}\mathcal{M}_{v(k)}, \quad id_{\mathcal{T}_{\mathbf{y}(k)}\mathcal{M}_{v(k)}}\eta = \eta, \tag{21}$$
$$(id_{\mathcal{T}_{\mathbf{y}(k)}\mathcal{M}_{v(k)}})^{\mathrm{T}}\mathcal{J}\vec{\mathbf{F}}(\mathbf{y}_1(k), \ldots, \mathbf{y}_N(k), v(k)) = \mathbf{0}. \tag{22}$$
See [1] for details on the retraction function and the identity mapping.

Hence, the curve $\gamma_{\mathbf{y}(k)}(t)$ can be defined by $\gamma_{\mathbf{y}(k)}(t) \equiv \mathcal{R}_{\mathbf{y}(k)}(t\eta_k)$. Actually, the first condition (19) ensures:

$$\gamma_{\mathbf{y}(k)}(0) = \mathcal{R}_{\mathbf{y}(k)}(0_k) = \mathbf{y}(k), \tag{23}$$

while the second condition (20) is related to the derivative condition:

$$\dot{\gamma}_{\mathbf{y}(k)}(0) = D\mathcal{R}_{\mathbf{y}(k)}(0_k)[\eta_k] = id_{\mathcal{T}_{\mathbf{y}(k)}\mathcal{M}_{v(k)}}\eta_k = \eta_k. \tag{24}$$

The numerical implementation of the curve $\gamma_{\mathbf{y}(k)}$ is not straightforward. The updating of $\mathbf{y}(k)$ and of $v(k)$ can be seen by first building a new vector $(\mathbf{y}'(k), v'(k))$:

$$\begin{pmatrix} \mathbf{y}'^{\mathrm{T}}(k) \\ v'^{\mathrm{T}}(k) \end{pmatrix} = \begin{pmatrix} \mathbf{y}^{\mathrm{T}}(k) \\ v^{\mathrm{T}}(k) \end{pmatrix} + t_k\eta_k. \tag{25}$$

Because $(\mathbf{y}'(k), v'(k))$ lies in the affine subspace $(\mathbf{y}(k), v(k))^{\mathrm{T}} + \mathcal{T}_{\mathbf{y}(k)}\mathcal{M}_{v(k)}$, it has to be retracted *on* the manifold to $(\mathbf{y}(k+1), v(k+1))$ by using the retraction function $\mathcal{R}_{\mathbf{y}(k)}$.

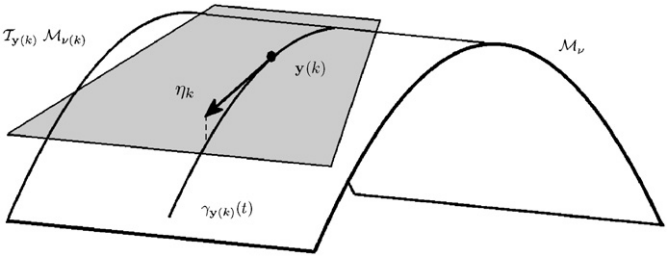

Fig. 4. Construction of the curve $\gamma_{\mathbf{y}(k)}(t)$ with the retraction function $\mathcal{R}_{\mathbf{y}(k)}$.

Step size: Finally, the step size t_k has to provide a sufficient decrease of the cost function. Indeed, if the step size is too small, the algorithm will converge slowly; conversely, if it is too large, the cost function could increase. The algorithm has thus to determine a suitable value for the step size. Let us denote $\alpha>0$ an upper bound on the step size ($t_k\leqslant\alpha$). To allow its adaptation during the optimization procedure, the step size t_k is implemented by the formula:

$$t_k=\beta^{m_k}\alpha, \tag{26}$$

where β is a fixed scalar belonging to [0,1] that shows how the step size can be decreased and m_k is a positive integer that changes during the optimization iterations to adapt the length of the step size.

To ensure a sufficient decrease of the cost function, m_k is the smallest integer such that the Armijo condition (27) is satisfied. The decrease of the cost function, $f(\mathbf{y}(k))-f(\mathbf{y}(k+1))$, is compared with the first order approximation of the decrease of the cost function. When the location $\mathbf{y}(k)$ is translated in the direction η_k with a step size $\alpha\beta^{m_k}$, the cost function is estimated by $f(\mathbf{y}(k+1))=f(\mathbf{y}(k))-\alpha\beta^{m_k}\|\eta_k\|^2$.

The Armijo condition is thus expressed by

$$f(\mathbf{y}(k))-f(\mathbf{y}(k+1))\geqslant\sigma\alpha\beta^{m_k}\|\eta_k f\|^2. \tag{27}$$

Here, the scalar σ belongs to [0,1] which is used to define a bound on the expected decrease of the cost function during a single iteration; typically σ is close to 0 ($\sigma=0.1$). See [1] for more details.

Algorithm: The algorithm to project data on a manifold is finally summarized in Algorithm 1. The different steps are repeated until convergence, i.e. until the norm of the search direction η_k is close to 0.

Algorithm 1. Projection on a manifold.

Input: Manifold $\mathscr{M}_{v(0)}$; cost function $f\in\mathscr{C}^1$; retraction $\mathscr{R}_{\mathbf{y}(0)}$ from $\mathscr{T}_{\mathbf{y}(0)}\mathscr{M}_{v(0)}$ to $\mathscr{M}_{v(0)}$; scalars $\alpha>0$, β, $\sigma\in[0,1]$; an initial value of the manifold parameter vector $\boldsymbol{v}(0)$; an initial iterate $\mathbf{y}(0)=(\mathbf{y}_1(0),\ldots,\mathbf{y}_N(0),v(0))$;
Output: The optimal projected data $\mathbf{y}=(\mathbf{y}_1,\ldots,\mathbf{y}_N)$ and the optimal manifold parameter vector $\boldsymbol{v}$;
repeat
Evaluate the gradient $\nabla f(\mathbf{y}_1(k),\ldots,\mathbf{y}_N(k),\boldsymbol{v}(k))$;
Evaluate the projection of ∇f on the tangent space:
$\eta_k=Proj_{\mathscr{T}_{\mathbf{y}(k)}\mathscr{M}_{v(k)}}(\nabla f)$;
Evaluate $\begin{pmatrix}\mathbf{y}^{\mathrm{T}}(k)\\ v^{\mathrm{T}}(k)\end{pmatrix}=\begin{pmatrix}\mathbf{y}^{\mathrm{T}}(k)\\ v^{\mathrm{T}}(k)\end{pmatrix}-\beta^{m_k}\alpha\eta_k$ and retract it with $\mathscr{R}_{\mathbf{y}(k)}$ on the manifold such that the Armijo condition is satisfied:
$f(\mathbf{y}(k))-f(\mathbf{y}(k+1))\geqslant\sigma\beta^{m_k}\alpha\|\eta_k\|^2$;
$k=k+1$;
until $\|\eta_k\|$ is close to 0

3.2. Projection on a sphere

In this section, the general projection methodology using the theory of optimization on manifolds is applied to the projection on the surface of a sphere. The section begins by defining the manifold and the tangent space. After these definitions, the analytic expressions of the gradient ∇f and of its projection on the tangent space η are given. The presentation of the retraction function $\mathscr{R}_{\mathbf{y}}$ ends this section.

Since the projection on the sphere is achieved by the minimization of the cost function (6) where only pairwise distances are required, the centre of the sphere is taken at the origin, without loss of generality. Therefore all projected vectors $(\mathbf{y}_i, 1\leqslant i\leqslant N)$ have the same norm and the geometric constraints of the sphere are expressed by the function:

$$\vec{\mathbf{F}}:\mathbb{R}^{3N}\times\mathbb{R}^+\to\mathbb{R}^N:(\mathbf{y}_1,\ldots,\mathbf{y}_N,R)\to\vec{\mathbf{F}}(\mathbf{y}_1,\ldots,\mathbf{y}_N,R)=\begin{pmatrix}\mathbf{y}_1^{\mathrm{T}}\mathbf{y}_1-R^2\\ \vdots\\ \mathbf{y}_N^{\mathrm{T}}\mathbf{y}_N-R^2\end{pmatrix} \tag{28}$$

that must be equal to $\mathbf{0}$. Since the optimal sphere radius cannot be determined *a priori*, it is considered as a parameter of the sphere which has to be optimized as well (this is the $\boldsymbol{v}$ parameter mentioned in the previous section). The manifold is then defined by the expression

$$\mathscr{M}_R\equiv\{(\mathbf{y}_1,\ldots,\mathbf{y}_N,R)\in S_R^3\times\cdots\times S_R^3\times\mathbb{R}^+|\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_i-R^2=0,1\leqslant i\leqslant N\} \tag{29}$$

which represents N geometric constraints. This is not exactly a sphere but the Cartesian product of N spheres having the same radius as it was discussed in the previous section (Eq. (12)).

Now, regarding Eq. (13), the tangent space related to a point $(\mathbf{y},R)=(\mathbf{y}_1,\ldots,\mathbf{y}_N,R)$ is obtained by differentiating the constraint function $\vec{\mathbf{F}}$:

$$T_{\mathbf{y}}\mathscr{M}_R\equiv\{(\mathbf{u}_1,\ldots,\mathbf{u}_N,u_R)\in\mathbb{R}^3\times\cdots\times\mathbb{R}|\mathbf{y}_i^{\mathrm{T}}\mathbf{u}_i-Ru_R=0,1\leqslant i\leqslant N\}. \tag{30}$$

The distance δ_{ij} in the projected space is obviously defined by the geodesic distance $\delta_{ij}=R\arccos(\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j/\|\mathbf{y}_i\|\|\mathbf{y}_j\|)$.

One powerful property of the optimization on manifolds assesses that if two functions are numerically equivalent on a manifold, no matter how they are analytically defined, their gradients remain equal after projection on the tangent space. In our case, if one prefers to define the distance as $\tilde{\delta}_{ij}=R\arccos(\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j/R^2)$, despite the partial derivatives are different, they will stay equivalent after their projection on the tangent space of the sphere. The choice of the first definition is motivated by the easiness of evaluating the projection on the tangent space and thus by improvements in the computation time. Actually, the projection of the gradient can be analytically defined when using the first definition of the distance; conversely, the second definition of the distance, $\tilde{\delta}_{ij}$, necessitates the use of an orthogonal projection algorithm to orthogonally project ∇f on the tangent space.

Now, the gradient of the cost function (6) is evaluated with respect to the locations on the sphere $\mathbf{y}_i$ and the radius R. To improve the readability of the following results, the notation κ_{ij} is introduced:

$$\kappa_{ij}=\frac{\partial f}{\partial\delta_{ij}}=\frac{-1}{N(N-1)}\left(2\lambda\frac{D_{ij}-\delta_{ij}}{D_{ij}}+(1-\lambda)\frac{D_{ij}^2-\delta_{ij}^2}{\delta_{ij}^2}\right). \tag{31}$$

The partial derivative of f with respect to R is given by

$$\frac{\partial f}{\partial R}=\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\frac{\partial f}{\partial\delta_{ij}}\frac{\partial\delta_{ij}}{\partial R}=\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\kappa_{ij}\arccos\frac{\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j}{\|\mathbf{y}_i\|\|\mathbf{y}_j\|}. \tag{32}$$

The partial derivative with respect to the location $\mathbf{y}_i$ is given by

$$\frac{\partial f}{\partial\mathbf{y}_i}=\sum_{j=1,j\neq i}^{N}\left(\frac{\partial f}{\partial\delta_{ij}}\frac{\partial\delta_{ij}}{\partial\mathbf{y}_i}+\frac{\partial f}{\partial\delta_{ji}}\frac{\partial\delta_{ji}}{\partial\mathbf{y}_i}\right)=2\sum_{j=1,j\neq i}^{N}\kappa_{ij}\frac{-1}{\sqrt{1-\left(\frac{\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j}{\|\mathbf{y}_i\|\|\mathbf{y}_j\|}\right)^2}}\left(\frac{\mathbf{y}_j}{\|\mathbf{y}_i\|\|\mathbf{y}_j\|}-\frac{\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j}{\|\mathbf{y}_i\|^2\|\mathbf{y}_j\|}\frac{\mathbf{y}_i}{\|\mathbf{y}_i\|}\right). \tag{33}$$

The last derivative is not defined when the locations $\mathbf{y}_i$ and $\mathbf{y}_j$ are antipodal. The user can assign a constant value to

$(\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j/\|\mathbf{y}_i\|\|\mathbf{y}_j\|)^2$ when this expression is close to 1. Furthermore, the derivative $\partial f/\partial \mathbf{y}_i$ is orthogonal to the location $\mathbf{y}_i$. This property helps in the evaluation of the projection on the tangent space.

Condition (30) that the gradient $-\nabla f$ has to satisfy to belong to the tangent space is not fulfilled here:

$$-\frac{\partial f}{\partial \mathbf{y}_i}^{\mathrm{T}}\mathbf{y}_i + R\frac{\partial f}{\partial R} = R\frac{\partial f}{\partial R} \neq 0, \quad \forall i = 1,\ldots,N \tag{34}$$

but the right-hand side $(R\partial f/\partial R)$ is the same for all the N conditions. Hence we need to project gradient $-\nabla f$ on the tangent space $\mathscr{T}_{\mathbf{y}}\mathscr{M}_R$ as follows:

$$\eta = -\nabla f - \begin{pmatrix} \frac{1}{R(N+1)}\frac{\partial f}{\partial R}\mathbf{y}_1 \\ \vdots \\ \frac{1}{R(N+1)}\frac{\partial f}{\partial R}\mathbf{y}_N \\ -\frac{N}{R(N+1)}\frac{\partial f}{\partial R}R \end{pmatrix} = -\begin{pmatrix} \frac{\partial f}{\partial \mathbf{y}_1} + \frac{1}{R(N+1)}\frac{\partial f}{\partial R}\mathbf{y}_1 \\ \vdots \\ \frac{\partial f}{\partial \mathbf{y}_N} + \frac{1}{R(N+1)}\frac{\partial f}{\partial R}\mathbf{y}_N \\ \frac{1}{(N+1)}\frac{\partial f}{\partial R} \end{pmatrix}. \tag{35}$$

The last step of the iteration is to look for a new location in the direction η along a curve $\gamma_{\mathbf{y}}(t)$ on the sphere tangent to this vector. Denoting $\eta = (\eta_1,\ldots,\eta_N,\eta_R)$ be a vector of the tangent space, the retraction function $\mathscr{R}_{\mathbf{y}}$ that satisfies conditions (19) and (20) performs the deterministic projection from the tangent space to the sphere:

$$\mathscr{R}_{\mathbf{y}} : \mathscr{T}_{\mathbf{y}}\mathscr{M}_R \to \mathscr{M} : \eta \to \mathscr{R}_{\mathbf{y}}(\eta_1,\ldots,\eta_N,\eta_R) = \begin{pmatrix} (R+\eta_R)\frac{\mathbf{y}_1+\eta_1}{\|\mathbf{y}_1+\eta_1\|} \\ \vdots \\ (R+\eta_R)\frac{\mathbf{y}_N+\eta_N}{\|\mathbf{y}_N+\eta_N\|} \\ R+\eta_R \end{pmatrix}. \tag{36}$$

4. Results

To evaluate the performance of the proposed methodology, experiments are performed on real data. The improvements of projecting on manifolds are assessed by comparing the projection on a sphere with the projection on the $\mathbb{R}^2$ Euclidean space by means of the trustworthiness (1) and of the continuity (2) quality measures.

In the first part, virtual face pictures are projected on a sphere while the second part shows results on pictures representing handwritten numbers 4.

4.1. Face pictures

A first experiment is performed on the widely known database of virtual face pictures [18,21,34,44]. This data set gathers 698 pictures of 64×64 pixels of the same face taken from different view angles and lighting. The dimension of the embedding original space is thus 4096 while the intrinsic dimension of the data distribution is only 3. Actually, the elevation and the azimuth angles of the camera and also the angle of lighting describe the pictures. These last three parameters are known and can be used to evaluate qualitatively the performances of the method although they are not used as input in the projection method. Samples of these pictures are presented in Fig. 5.

The original data are not directly introduced in the algorithm; the pairwise distances D_{ij} must be first estimated. As it is assumed that the original data are close to an unknown manifold embedded in the high-dimensional space, the pairwise *geodesic* distances are approximated by building a graph in the data distribution; the nodes of the graph represent the data and the edges implement a proximity relationship between them. For each original datum $\mathbf{x}_i$, we consider its k-neighbourhood that includes the k closest data: edges are then defined to join these points to $\mathbf{x}_i$.

Fig. 5. Sample of face database.

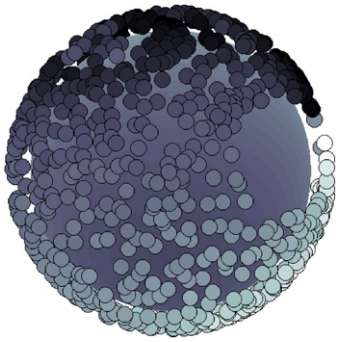

Fig. 6. Face data projected on the sphere with $\lambda = 0.6$.

Note that the size k of the neighbourhoods must ensure the connectivity of the resulting graph. The value of $k = 15$ has been chosen experimentally.

Each edge is then weighted by the corresponding Euclidean distance. The distance between any pair of points can then be evaluated by the shortest path in the graph, as proposed in [8,7,16,33,34] for example with the Dijkstra algorithm, yielding an approximation of the geodesic distances.

In order to assess the quality of the proposed method on this data set, the faces are projected both on $\mathbb{R}^2$ and on the sphere according to the same criterion (6); these two search spaces have the same intrinsic dimension. On the spherical manifold, the distance used is the geodesic distance that was defined in Section 3.2 ($\delta_{ij} \equiv R\,\mathrm{acos}(\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j/\|\mathbf{x}_i\|_2\|\mathbf{x}_i\|_2)$). Conversely, the geodesic distance in $\mathbb{R}^2$ is the Euclidean distance ($\delta_{ij} \equiv \|\mathbf{x}_i - \mathbf{x}_j\|_2$). When projecting on $\mathbb{R}^2$, criterion (6) is minimized by a standard line-search algorithm. Different experiments are performed with different values of the user-defined parameter λ.

The visualization of the projected data is represented on the sphere in Fig. 6 and in the spherical coordinate space in Fig. 7. The colour of the points varies with the azimuth angle of the camera; as can be seen on these figures, the evolution of the colour is generally smooth which confirms a good trustworthiness and continuity of the projection.

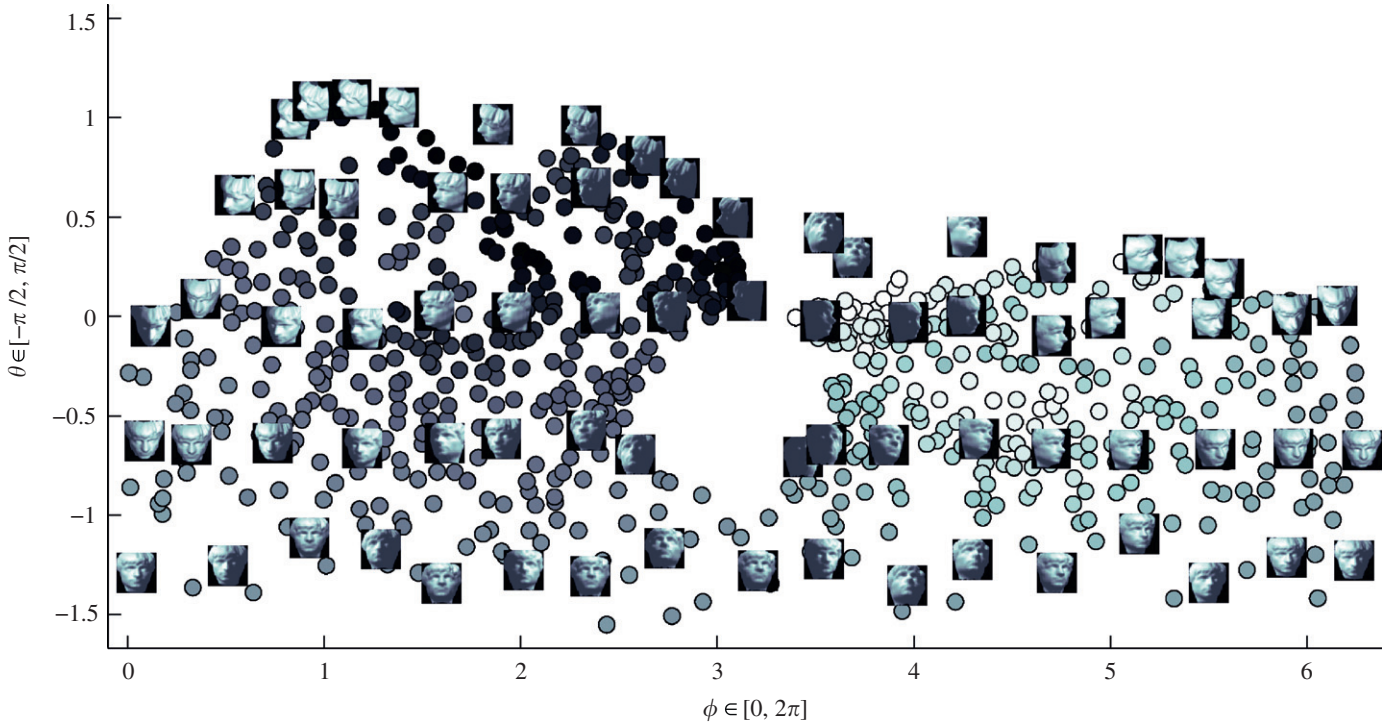

Fig. 7. Face data projected on the sphere for $\lambda = 0.6$, in the spherical coordinate space (ϕ, θ) where the colour varies with the azimuth angle of the camera.

In the spherical coordinate space, only a few pictures are represented as pictures to increase the readability of the result and to show the smoothness of the projection. Since the azimuth angle $\phi = 0$ corresponds to the azimuth angle $\phi = 2\pi$ in the spherical coordinate space, the data on the left of Fig. 7 are close to the right ones. Moreover, because of the singularities in the north and south poles, the upper data (close to $\theta = \pi/2$) are close from each other and so are the lowest ($\theta = -\pi/2$).

The distribution seems to intercept itself in the centre of Fig. 7 where dark points are close to lighter ones (see also the right side of the sphere represented in Fig. 6); this is probably due to the poor light. Indeed, the construction of the geodesic distances is such that these two pictures are close in the original space as they are both dark.

To assess objectively the quality of the projection, the comparison is performed by evaluating the trustworthiness and continuity quality measures, as defined in Eqs. (1) and (2); the next results show the preservation of the 15-neighbourhood. As previously discussed, the closer these measures are to 1, the most trustworthy and continuous the projection is. As shown in Fig. 8, the results of the projection on a sphere are closer to (1,1) than those corresponding to the projection on $\mathbb{R}^2$.

Furthermore, the cost function (6) is smaller when it is minimized on the sphere than when it is minimized in the $\mathbb{R}^2$ Euclidean space, as shown in Fig. 9.

Figs. 10 and 11 show, respectively, the resulting projection on the sphere and on the $\mathbb{R}^2$ Euclidean space. In these figures, the colour varies with the *elevation* angle of the camera. One can see that the colour varies more smoothly on the sphere than in the Euclidean space.

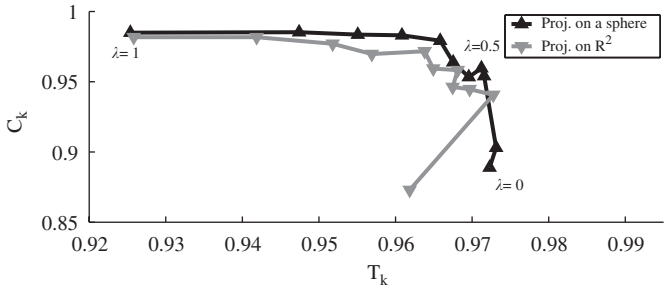

Fig. 8. Comparison between the projections of face data on $\mathbb{R}^2$ and on the sphere with the trustworthiness (T_k) and the continuity (C_k) quality measures.

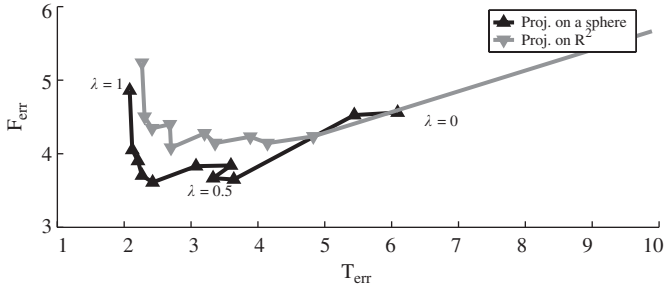

Fig. 9. Comparison between the projections of face data on $\mathbb{R}^2$ and on the sphere with the tearing (T_{err}) and the flattening errors (F_{err}).

4.2. Handwritten numbers

The second experiment is performed on the MNIST database [10,12]. The data set gathers pictures of 28×28 pixels of all the numbers (from 0 to 9) that are often used to test classification methods. Each class of handwritten numbers contains a training and a test set; we will focus here on the projection of the number 4 whose test set contains 982 pictures.

Again, the data are projected with different values of the user-defined parameter λ both on the $\mathbb{R}^2$ Euclidean space and on the sphere. For the present data set, the geodesic distance matrix in

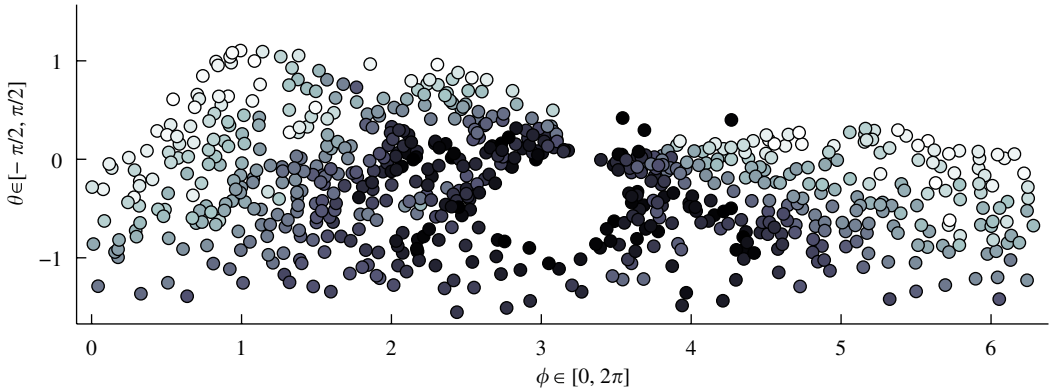

Fig. 10. Projection on the sphere of the face data where the colour varies with the elevation angle of the camera.

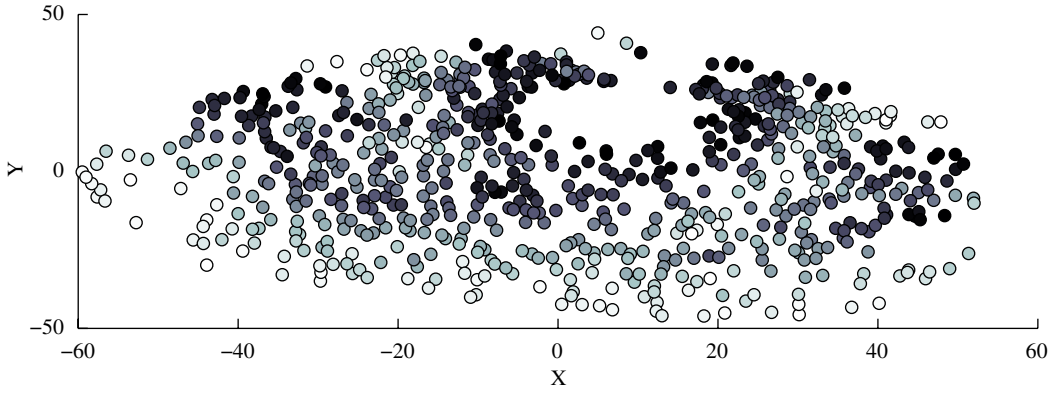

Fig. 11. Projection on the Euclidean space of the face data where the colour varies with the elevation angle of the camera.

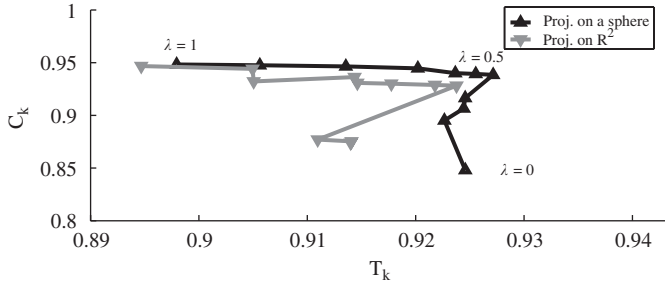

Fig. 12. Comparison between the projections of handwritten numbers 4 on $\mathbb{R}^2$ and on the sphere with the trustworthiness (T_k) and the continuity (C_k) quality measures.

the high-dimensional space is approximated by building the graph, as presented in the previous section, using the 20-closest neighbours.

The trustworthiness and continuity quality measures are then computed and plotted in Fig. 12; here, the parameter k of the trustworthiness and continuity quality measures is 20 (the quality measures evaluate the preservation of the 20-neighbourhoods). This graph illustrates the improvements of the projection which is more trustworthy and continuous on the sphere than on the Euclidean space.

The resulting projection on the sphere with $\lambda = 0.5$ is illustrated in Fig. 13.

The trustworthiness of the projection seems good as the largest handwritten numbers 4 are projected in the same region of the sphere, here in the middle of the figure. The italic numbers stay mainly on the right of the figure.

5. Conclusion

This paper describes a method to project data on non-Euclidean manifolds by minimizing a pairwise distance criterion. The data are assumed to be close to a low-dimensional manifold embedded in a high-dimensional space, as commonly assumed in nonlinear projection methods. However, loops in this manifold are allowed (such as in the cylinder and the sphere). Projecting a manifold with loops cannot be achieved without having to implement a compromise between trustworthiness and continuity. This is achieved by minimizing a pairwise distance criterion including a user-defined balance between flattening and tearing errors. Beside allowing the projection on any manifold, including those with loops, the method presented in this paper decreases border effects that appear when trying to project a non-Euclidean manifold on an Euclidean one.

The minimization of the objective criterion is performed by using the theory of optimization on manifolds. The methodology is illustrated through the example of the projection on the sphere; it can be easily extended to any other manifold. Results show the improvement of projecting on manifolds achieved by the method, both qualitatively (visual result) and quantitatively (flattening and tearing errors, and trustworthiness and continuity).

The possibility to project on non-Euclidean manifolds, as detailed in this paper, introduces new questions: When do we have to project on a non-Euclidean manifold, and on which one? As the main motivation behind the use of non-Euclidean manifolds is to better model loops in the original data, the detection of loops and their characterization seems the way to address these questions and will be the topics of further work. A related direction is to design measures of global topology that can

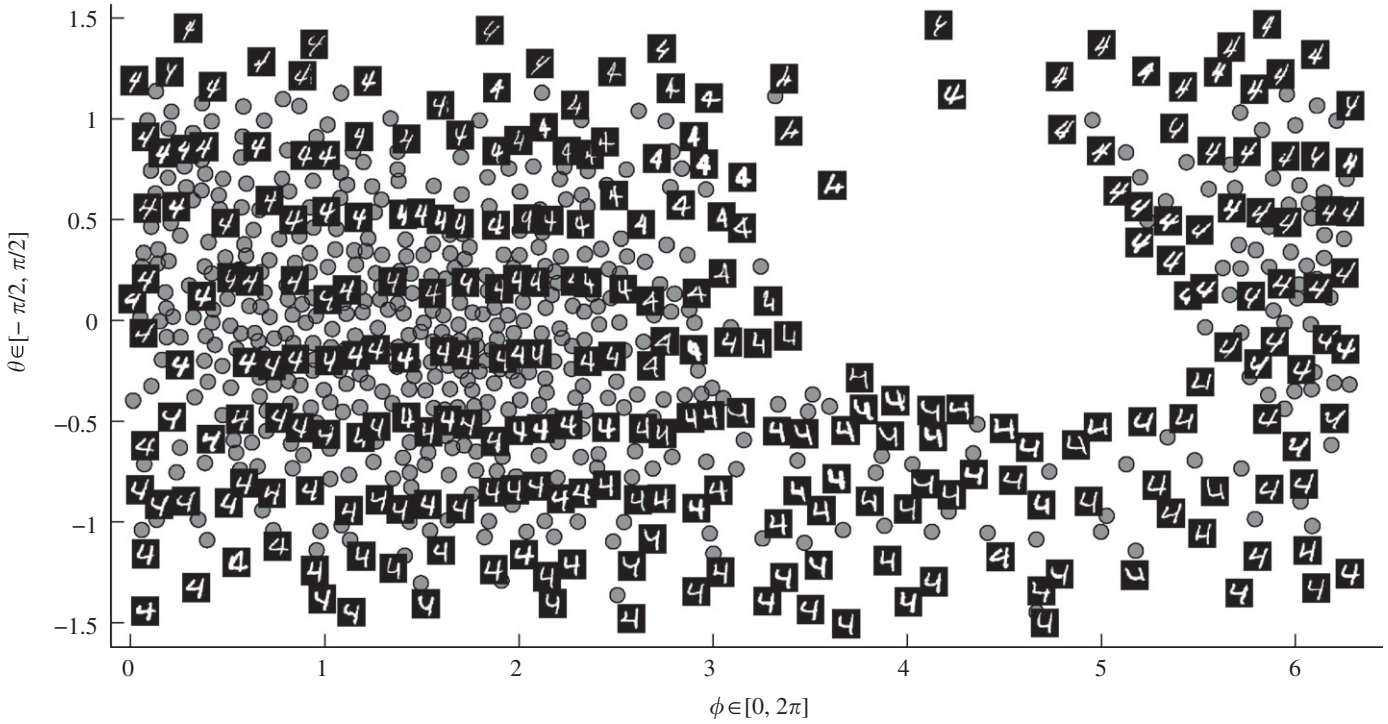

Fig. 13. Handwritten number 4 data projected on the sphere with $\lambda = 0.5$, in the spherical coordinate space (ϕ, θ).

objectively be compared between projections on various Euclidean and non-Euclidean manifolds.

Acknowledgements

V. Onclinx is funded by a grant from the Belgium F.R.I.A. Part of this work present research results of the Belgian Network DYSCO (Dynamical Systems, control, and Optimization) funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its author(s).

The authors thank Prof. Pierre-Antoine Absil for his suggestions on the theory of optimization on manifolds.

References

[1] P.-A. Absil, R. Mahony, R. Sepulchre, Optimization Algorithms on Matrix Manifolds, Princeton University Press, Princeton, NJ, January 2008.
[2] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation 15 (6) (2003) 1373–1396.
[3] D.P. Bertsekas, Nonlinear Programming, Athena Scientific, September 1999.
[4] C.M. Bishop, M. Svensén, C.K.I. Williams, GTM: the generative topographic mapping, Neural Computation 10 (1) (1998) 215–234.
[5] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, March 2004.
[6] A. Brun, C.-F. Westin, M. Herberthson, H. Knutsson, Fast manifold learning based on Riemannian normal coordinates, in: H. Kälviäinen, J. Parkkinen, A. Kaarna (Eds.), Proceedings of the 14th Scandinavian Conference on Image Analysis (SCIA'05), Joensuu, Finland, Lecture Notes in Computer Science, vol. 3540, Springer, Berlin, June 2005, pp. 920–929.
[7] P.A. Estevez, A.M. Chong, Geodesic nonlinear mapping using the neural gas network, in: International Joint Conference on Neural Networks, IJCNN '06, October 23–27 2006, pp. 3287–3294.
[8] P.A. Estevez, C.J. Figueroa, Online data visualization using the neural gas network, Neural Networks 19 (6) (2006) 923–934.
[9] D. Gering, Linear and nonlinear data dimensionality reduction, Ph.D. Thesis, MIT, Cambridge, MA, 2002.
[10] J. Goldberger, S. Roweis, Hierarchical clustering of a mixture model, in: L.K. Saul, Y. Weiss, L. Bottou (Eds.), Advances in Neural Information Processing Systems, vol. 17, MIT Press, Cambridge, January 5, 2004, pp. 505–512.
[11] S. Horata, T. Ikemura, T. Yukawa, Torus self-organizing map for genome informatics, in: Proceedings of WSOM'05, 5th Workshop on Self-organizing Maps, September 5–8, 2005, pp. 235–242.
[12] O. Kayo, Locally linear embedding algorithm, extensions and applications, Ph.D. Thesis, Universitatis Ouluensis, April 2006.
[13] T. Kohonen, Self-Organizing Maps, second ed., Springer, Berlin, 1995.
[14] J.B. Kruskal, Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, Psychometrika 29 (1964) 1–28.
[15] J.A. Lee, A. Lendasse, N. Donckers, M. Verleysen, A robust nonlinear projection method, in: ESANN 2000, European Symposium on Artificial Neural Networks, Bruges, Belgium, April 26–28, 2000, pp. 13–20. ESANN, D-Facto public.
[16] J.A. Lee, A. Lendasse, M. Verleysen, Nonlinear projection with curvilinear distances: isomap versus curvilinear distance analysis, Neurocomputing 57 (2003).
[17] J.A. Lee, M. Verleysen, Nonlinear projection with the isotop method, in: J.R. Dorronsoro (Ed.), Artificial Neural Networks, London, UK, Lecture Notes in Computer Science, vol. 2415, ICANN, Springer, Berlin, Augustus 2002, pp. 933–938.
[18] J.A. Lee, M. Verleysen, Nonlinear Dimensionality Reduction, Springer Science+Business Media, LLC, 2007.
[19] J.A. Lee, M. Verleysen, Rank-based quality assessment of nonlinear dimensionality reduction, in: ESANN 2008, European Symposium on Artificial Neural Networks, Bruges, Belgium, April 23–25, 2008, pp. 49–54. ESANN, d-side publi.
[20] J.X. Li, Visualization of high-dimensional data with relational perspective map, Information Visualization 3 (1) (2004) 49–59.
[21] T. Lin, H. Zha, Riemannian manifold learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 796–809.
[22] H. Nishio, Md. Altaf-Ul-Amin, K. Kurokawa, K. Minato, S. Kanaya, Spherical SOM with arbitrary number of neurons and measure of suitability, in: Proceedings of WSOM'05, 5th Workshop on Self-organizing Maps, September 5–8, 2005, pp. 323–330.
[23] M. Niskanen, O. Silvén, Comparison of dimensionality reduction methods for wood surface inspection, in: QCAV 2003, 6th International Conference on Quality Control by Artificial Vision, vol. 5132, SPIE, 2003, pp. 178–188.
[24] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, Berlin, August 1999.
[25] V. Onclinx, V. Wertz, M. Verleysen, Nonlinear data projection on a sphere with a controlled trade-off between trustworthiness and continuity, in: ESANN 2008, European Symposium on Artificial Neural Networks, Bruges, Belgium, April 23–25, 2008, pp. 43–48. ESANN, d-side publi.
[26] K. Pearson, Analysis of a complex statistical variables into principal components, Journal of Educational Psychology 24 (1933) 417–441.
[27] H. Ritter, Self-organizing maps on non-Euclidean spaces, in: S. Oja, E. Kaski (Eds.), Kohonen Maps, Elsevier, Amsterdam, 1999, pp. 97–108.
[28] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[29] J.W. Sammon, A nonlinear mapping algorithm for data structure analysis, IEEE Transactions on Computers CC-18 (5) (1969) 401–409.

[30] A. Sangole, G.K. Knopf, Visualization of randomly ordered numeric data sets using spherical self-organizing feature maps, Computers and Graphics 27 (6) (2003) 963–976.
[31] L.K. Saul, S.T. Roweis, Think globally, fit locally: unsupervised learning of low dimensional manifolds, Journal of Machine Learning Research 4 (2003) 119–155.
[32] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (5) (1998) 1299–1319.
[33] J.B. Tenenbaum, Mapping a manifold of perceptual observations, in: M. Jordan, M. Kearns, S. Solla (Eds.), Advances in Neural Information Processing Systems (NIPS 2007), vol. 10, 1998, pp. 682–688.
[34] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[35] W.S. Torgerson, Multidimensional scaling: theory and method, Psychometrika 17 (4) (1952) 401–419.
[36] J. Venna, S. Kaski, Neighborhood preservation in nonlinear projection methods: an experimental study, in: ICANN '01: Proceedings of the International Conference on Artificial Neural Networks, London, UK, Springer, Berlin, August 21–25, 2001, pp. 485–491.
[37] J. Venna, S. Kaski, Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity, in: Proceedings of WSOM'05, 5th Workshop on Self-organizing Maps, WSOM, September 5–8, 2005, pp. 695–702.
[38] J. Venna, S. Kaski, Comparison of visualization methods for an atlas of gene expression data sets, Information Visualization 6 (2) (2007) 139–154.
[39] J. Venna, S. Kaski, Nonlinear dimensionality reduction as information retrieval, in: Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS 2007), March 21–24, 2007, pp. 568–575.
[40] J. Venna, S. Kaski, Local multidimensional scaling, Neural Networks 19 (6) (2006) 889–899.
[41] Y. Wu, M. Takatsuka, Fast spherical self organizing map-use of indexed geodesic data structure, in: WSOM05, 2005, pp. 455–462.
[42] Y. Wu, M. Takatsuka, Spherical self-organizing map using efficient indexed geodesic data structure, Neural Networks 19 (6–7) (2006) 900–910.
[43] G. Young, A.S. Householder, Discussion of a set of points in terms of their mutual distances, Psychometrika 3 (1) (1938) 19–22.
[44] Z. Zhang, H. Zha, Principal manifolds and nonlinear dimensionality reduction via tangent space alignment, SIAM Journal on Scientific Computing 26 (1) (2005) 313–338.

Victor Onclinx was born in Aye, Belgium, in 1982. He got his master degree of applied mathematics in 2005 from Université catholique de Louvain, Belgium. He is working on his Ph.D. thesis since October 2006 in the Applied Mathematics Department of the Université catholique de Louvain, Belgium. Since 2007 he is working under a research fellow grant from the FRIA. He is also an active member of the Machine Learning Group at UCL (www.dice.ucl.ac.be/mlg). The main topics covered by his thesis is nonlinear data projection.

Vincent Wertz was born in Liège, Belgium, in 1955. He received his engineering degree in applied mathematics in 1978 and a Ph.D. in control engineering in 1982, both from the Université catholique de Louvain. He is now a professor at the Université catholique de Louvain, in Louvain-la-Neuve, Belgium, after having held permanent position with the NFSR. His main research interests are in the fields of identification, machine learning, predictive control, fuzzy control and industrial applications. He is author or co-author of two books and more than 175 journal and peer reviewed conference papers. Lately, he has also been involved in a major pedagogical reform of the first and second year teaching programme of the school of engineering.

Michel Verleysen was born in 1965 in Belgium. He received the M.S. and Ph.D. degrees in electrical engineering from the Université catholique de Louvain (Belgium) in 1987 and 1992, respectively. He was an invited professor at the Swiss E.P.F.L. (Ecole Polytechnique Fédérale de Lausanne, Switzerland) in 1992, at the Université d'Evry Val d'Essonne (France) in 2001, and at the Université ParisI-Panthéon-Sorbonne from 2002 to 2008, respectively. He is now a professor at the Université catholique de Louvain, and Honory Research Director of the Belgian F.N.R.S (National Fund for Scientific Research). He is editor-in-chief of the Neural Processing Letters journal, chairman of the annual ESANN conference (European Symposium on Artificial Neural Networks), associate editor of the IEEE Transactions on Neural Networks journal and member of the editorial board and program committee of several journals and conferences on neural networks and learning. He is author or co-author of more than 200 scientific papers in international journals and books of communications to conferences with reviewing committee. He is co-author of the scientific popularization book on artificial neural networks in the series "Que Sais-Je?," in French, and of the "Nonlinear Dimensionality Reduction" book published by Springer in 2007. His research interest includes machine learning, artificial neural networks, self-organization, time-series forecasting, nonlinear statistics, adaptive signal processing, and high-dimensional data analysis.