

Dimensionality reduction by rank preservation

Victor Onclinx, John A. Lee, Vincent Wertz, and Michel Verleysen

Abstract— Dimensionality reduction techniques aim at representing high-dimensional data in low-dimensional spaces. To be faithful and reliable, the representation is usually required to preserve proximity relationships. In practice, methods like multidimensional scaling try to fulfill this requirement by preserving pairwise distances in the low-dimensional representation. However, such a simplification does not easily allow for local scalings in the representation. It also makes these methods suboptimal with respect to recent quality criteria that are based on distance rankings. This paper addresses this issue by introducing a dimensionality reduction method that works with ranks. Appropriate hypotheses enable the minimization of a rank-based cost function. In particular, the scale indeterminacy that is inherent to ranks is circumvented by representing data on a space with a spherical topology.

I. INTRODUCTION

Dimensionality reduction [8], [12] aims at finding a low-dimensional representation of high-dimensionality data, e.g. for visualization purposes. The quality of the representation is usually evaluated by checking that proximity relationships are preserved. In this way, close data items are represented close to each other, whereas distant items remain far from each other.

In practice, the generic principle of proximity preservation can be instantiated in various ways. For instance, principal component analysis [18] tries to preserve pairwise scalar products. Multidimensional scaling [22] and closely related techniques [5], [10], [19] attempt to reproduce pairwise distances. The success of these methods mainly depends on how distance errors are weighted in the definition of the cost function. Giving more importance to small distances usually allows data to unfold [19], [25]. The choice of the metric might be important as well. Geodesic distances, for instance, better capture the intrinsic structure of data than Euclidean distances [6], [11], [21].

This paper describes a dimensionality reduction technique that tries to preserve rank information instead of distances. Ranks are obtained by sorting either each row or column in a matrix of pairwise distances. The idea is motivated by the fact that modern quality criteria for dimensionality reduction evaluate the preservation of ranks [13] or, equivalently, of K -ary neighborhoods [23]. These works tend to show that the principle of proximity preservation that underlies

dimensionality reduction is rendered more faithfully by ranks than distances. If the use of ranks raises no particular issue in a quality criterion, the situation becomes more complicated in the context of a cost function as sketched in [14]. For example, ranks are piecewise constant functions of distances and thus have a zero derivative almost everywhere. The presence of ranks in a cost function therefore requires specific optimization tools. In contrast to distance matrices, rank matrices are not necessarily symmetric. Ranks also introduce a scale indeterminacy: multiplying all distances with a constant factor leaves the ranks unchanged. The method suggested in this paper addresses these issues with ad hoc solutions. In particular, the scale indeterminacy is circumvented by representing the data on a sphere, instead of a Cartesian space. Several recent works have investigated the feasibility of representing data on a differentiable manifold [15], [17]; however, they do not consider the question of optimizing ranks instead of distances.

The remainder of this paper is organized as follows. Section II introduces a generic cost function for dimensionality reduction, which is first instantiated with distances. Next, Section II justifies the use of ranks and plugs them into the cost function. As the data are immersed in a spherical shell, Section III briefly presents some concepts of optimization on manifolds. Section IV describes in details the optimization technique that minimizes the proposed rank-based cost function. Section V presents some experimental results, including comparisons with state-of-the-art techniques.

II. COST FUNCTION

The definition of a cost function for dimensionality reduction should ideally be guided by or related to the quality criterion that one seeks to optimize. If the expression of the criterion is simple and has good properties (such as differentiability and no indeterminacy), then the cost function can obviously be the criterion itself. In dimensionality reduction however, recent quality criteria involve ranks of sorted distances. The usual approach is then to design the cost function as a surrogate of the criterion, with useful properties for its optimization. For instance, differentiability can be obtained by approximating rank preservation with distance preservation.

The study of quality criteria also tells us that dimensionality reduction can be subject to two types of errors. Intrusions [13] (or flattening errors [17]) occur when two distant groups of neighboring points are mixed together in the low-dimensional representation. In contrast, extrusions [13] (or tearing errors [17]) occur when a neighborhood is split into distant parts in the representation. For example, in the case of a spherical manifold, two planar representations

V.O. is with the ICTEAM Institute and the Machine Learning Group (MLG) of the Université catholique de Louvain, Av. G. Lemaitre, B-1348 Louvain-la-Neuve, Belgium (victor.onclinx@uclouvain.be); he is funded by a F.R.I.A. grant. J.A.L. is a Research Associate with the Belgian National Fund of Scientific Research (F.R.S.-FNRS); he is a member of the IREC Institute and MLG. V.W. and M.V. are with the ICTEAM and MLG.

Part of this paper presents 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 authors.

are possible. If the spherical shell is crushed, then intrusions dominate, whereas a representation made of two caps next to each other like an Earth map is mostly extrusive.

Within the framework of distance preservation, the balance between intrusions and extrusions can be expressed in the following way. First, let $\mathbf{X} = [\mathbf{x}_i]_{1 \leq i \leq N}$ and $\mathbf{Y} = [\mathbf{y}_i]_{1 \leq i \leq N}$ denote the data set and its low-dimensional representation, with $\mathbf{x}_i \in \mathbb{R}^n$, $\mathbf{y}_i \in \mathbb{R}^p$, and $p \leq n$. Pairwise distances between items \mathbf{x}_i and \mathbf{x}_j for $1 \leq i, j \leq N$ are given by $d_{ij} = d_{\mathbf{X}}(\mathbf{x}_i, \mathbf{x}_j)$ where $d_{\mathbf{X}}$ is the distance measure in the high-dimensional space. Conversely, $\delta_{ij} = \delta_{\mathbf{Y}}(\mathbf{y}_i, \mathbf{y}_j)$ denotes pairwise distances between items \mathbf{y}_i and \mathbf{y}_j with respect to the distance measure $\delta_{\mathbf{Y}}$ in the low-dimensional space. This allows us to write

$$E(\mathbf{Y}; \mathbf{X}) = \lambda \sum_{i,j=1}^N \frac{(d_{ij} - \delta_{ij})^2}{b + d_{ij}} + (1-\lambda) \sum_{i,j=1}^N \frac{(d_{ij} - \delta_{ij})^2}{b + \delta_{ij}}, \quad (1)$$

where parameter λ controls the tradeoff between intrusions and extrusions [24]. In (1), squared differences of distances are gathered in to differently weighted terms. The weights are inversely proportional to the distance in either the high-dimensional space (first term) or in the low-dimensional space (second term). Parameter b influences the decrease of the weights. Knowing that intrusions correspond to cases where $\delta_{ij} \ll d_{ij}$, their weight is light in the first term of (1). In contrast, a light weight in the second term is associated with cases where $\delta_{ij} \gg d_{ij}$, that is, with extrusions.

In addition to ensuring differentiability, the approximation of rank preservation with distance preservation also circumvents the scale indeterminacy that is inherent to ranks. On the other hand, it annihilates the main advantage of using ranks, which is their allowance for local scalings in the low-dimensional representation. This property is however highly desirable, even to obtain a planar representation of such a simple manifold as a half sphere. Concentrating points near the manifold center does not break local neighborhood relationships and facilitates flattening; rank preservation makes this concentration possible. In contrast, distance preservation entails too strong constraints.

The above arguments pledge for designing a cost function based on ranks instead of distances. For this purpose, one rewrites (1) as

$$E(\mathbf{Y}; \mathbf{X}) = \lambda \sum_{i,j=1}^N \frac{(r_{ij} - \rho_{ij})^2}{b + r_{ij}} + (1-\lambda) \sum_{i,j=1}^N \frac{(r_{ij} - \rho_{ij})^2}{b + \rho_{ij}}, \quad (2)$$

where r_{ij} denotes the ranks of \mathbf{x}_j with respect to \mathbf{x}_i . Formally, nonreflexive rank without ties are defined by

$$r_{ij} = |\{k \text{ s.t. } d_{ik} < d_{ij} \text{ or } (d_{ik} = d_{ij} \text{ and } k < j)\}|.$$

Similarly, we have in the low-dimensional space

$$\rho_{ij} = |\{k \text{ s.t. } \delta_{ik} < \delta_{ij} \text{ or } (\delta_{ik} = \delta_{ij} \text{ and } k < j)\}|.$$

Exactly as in (1), the two weighted terms allow either intrusions or extrusions and λ controls their respective importance.

III. OPTIMIZATION ON MANIFOLD

Due to the scale indeterminacy, the use of ranks in an iterative optimization technique such as gradient descent can lead to degenerate solutions. The successive updates of the estimate can cause either a concentration or a dispersion of all points in the low-dimensional representation when optimizing rank-based functions. From a force-directed placement [7] point of view, as used by RankVisu [14], the most likely degenerate solution depends on whether forces that move the points are repulsive or attractive. Several workarounds can address the specific case of a dispersion. At first glance, the easiest solution is to constrain the low-dimensional representation to lie inside a compact subset of the Cartesian space. In practice however, the addition of constraints makes the optimization more complicated. As an alternative solution, we suggest representing the data set on a spherical shell. This section briefly introduces the necessary concepts of optimization on manifolds, which are applicable to the sphere.

Classical optimization procedures [4], [16] are aimed at optimizing a cost function on a Euclidean space. If we consider variable $\mathbf{z} \in \mathbb{R}^n$ and minimization problem

$$\min_{\mathbf{z}} f(\mathbf{z}), \quad (3)$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}$, an iterative gradient-descent algorithm can be schematized as follows. At the $(k+1)$ th iteration, the optimization procedure updates previous estimate $\mathbf{z}(k)$ towards the minimum of cost function f . First, the algorithm updates the search direction $\boldsymbol{\eta}(k)$; in the case of the steepest descent method, this direction is given by $\boldsymbol{\eta}(k) = -\nabla f(\mathbf{z}(k))$. Secondly, the line-search algorithm moves the estimate along the search direction $\boldsymbol{\eta}(k)$. Update rule

$$\mathbf{z}(k+1) = \mathbf{z}(k) + \alpha_k \boldsymbol{\eta}(k)$$

yields the next estimate, with $\alpha_k \in \mathbb{R}$ being the step size. The latter has to ensure a sufficient decrease of the cost function [4], [16]. After the update, we have $f(\mathbf{z}(k+1)) < f(\mathbf{z}(k))$ and we proceed until a (local) minimum is reached.

In the case of cost function $f: \mathcal{M} \rightarrow \mathbb{R}$ whose domain is manifold \mathcal{M} , the theory of optimization on manifold [1] generalizes classical tools and procedures. For instance, the search direction $\boldsymbol{\eta}(k)$ and the iterative update rule are adapted in order to take the manifold constraint into account. The optimization problem in (3) is rewritten as

$$\min_{\mathbf{z} \in \mathcal{M}} f(\mathbf{z}),$$

where $\mathcal{M} \subseteq \mathbb{R}^n$ is a differentiable manifold, which is informally defined as a set that is locally homeomorphic to a subset of a Euclidean space. Manifold \mathcal{M} expresses the geometric constraints that item \mathbf{z} has to satisfy. In our case, manifold \mathcal{M} is a sphere and we could carry out the optimization with the equation of the sphere as a constraint. Classical constrained optimization [3], [4], [16] usually introduces Lagrange multipliers to solve such a problem. Since all N points of the low-dimensional representation

have to lie on the manifold, N constraints — and hence N Lagrange multipliers — are added. This approach obviously complicates the problem.

An unconstrained optimization using spherical coordinates suffers from annoying shortcomings as well. The singularities at both poles cause numerical problems. The poles are represented by segment lines in the spherical coordinate space: for example, the north pole is defined by

$$\{(\phi, \theta) | \theta = \frac{\pi}{2}, \phi \in [0, 2\pi]\}.$$

In contrast, optimization on a manifold generalizes the principle of a gradient descent by forcing the updated estimate $\mathbf{z}(k+1)$ to remain on manifold \mathcal{M} . Because of the manifold curvature, moving the estimate along line-search direction $\boldsymbol{\eta}(k)$ is not sufficient. Instead, the estimate should follow a geodesic of the manifold. The remainder of this section describes how one can adapt both the search direction and the update rule in order to stick to a manifold geodesic.

An adequate search direction $\boldsymbol{\eta}(k)$ is defined as follows. Since gradient $-\nabla f(\mathbf{z}(k))$ may not encompass the manifold constraint, we first project orthogonally on tangent space $\mathcal{T}_{\mathbf{z}(k)}\mathcal{M}$. Actually, denoting a geodesic curve γ such that $\gamma(0) = \mathbf{z}(k)$, its derivative $\gamma'(0)$ belongs to the tangent space $\mathcal{T}_{\mathbf{z}(k)}\mathcal{M}$. Forcing the search direction $\boldsymbol{\eta}(k)$ to belong to the tangent space allows the evaluation of a geodesic curve satisfying $\gamma'(0) = \boldsymbol{\eta}(k)$, hence the update of $\mathbf{z}(k)$ along this geodesic curve.

To translate $\mathbf{z}(k)$ along the geodesic curve γ , the algorithm first evaluates a new item $\mathbf{z}'(k) = \mathbf{z}(k) + \alpha_k \boldsymbol{\eta}(k)$. Then, to satisfy the manifold constraint, the latter is mapped, i.e. *retracted*, on the manifold. This task is achieved by a retraction function $\mathcal{R}_{\mathbf{z}(k)} : \mathcal{T}_{\mathbf{z}(k)}\mathcal{M} \rightarrow \mathcal{M}$ which is informally defined as a smooth mapping from the tangent space to the manifold; formal definition and properties can be found in [1]. For instance, the retraction function allows the definition of the geodesic curve:

$$\gamma : \mathbb{R} \rightarrow \mathcal{M}, t \mapsto \gamma(t) = \mathcal{R}_{\mathbf{z}(k)}(t\boldsymbol{\eta}(k)).$$

Moreover, the properties of the retraction function [1] ensure the following conditions

$$\begin{cases} \gamma(0) = \mathbf{z}(k) \\ \gamma'(0) = \boldsymbol{\eta}(k) \end{cases}.$$

Note that the step size α_k is set to ensure a sufficient decrease of the cost function. Figure 1 illustrates the procedure that is detailed and theoretically justified in [1].

IV. REDUCING THE DIMENSIONALITY WITH RANKS

The forthcoming paragraphs describe a practical way to minimize cost function $E(\mathbf{Y}; \mathbf{X})$ defined in (2) on the unit sphere $\mathcal{S} = \{\mathbf{y} \in \mathbb{R}^3 | \|\mathbf{y}\|_2 = 1\}$. For this purpose, the low-dimensional representation denoted by $\mathbf{Y}(k) = [\mathbf{y}_i(k)]_{1 \leq i \leq N}$ is updated in two steps. First, we determine a set of suitable search directions $\boldsymbol{\eta}_i(k)$; their calculation is detailed further below. Once the search directions are projected on their associated tangent space

$$\mathcal{T}_{\mathbf{y}_i(k)}\mathcal{S} = \{\boldsymbol{\eta} \in \mathbb{R}^3 | \boldsymbol{\eta}^T \mathbf{y}_i(k) = 0\},$$

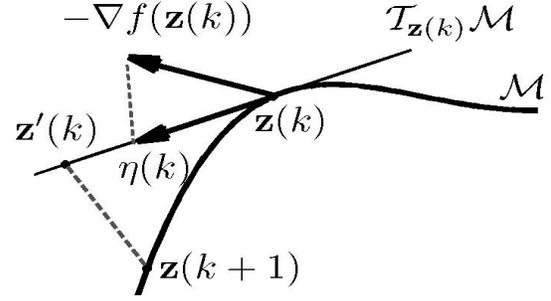


Fig. 1. Illustration of the $(k+1)^{th}$ iteration of the optimization on manifold procedure. Adapted from [17].

the first update can be written as

$$\mathbf{y}'_i(k) = \mathbf{y}_i(k) + \alpha_k \boldsymbol{\eta}_i(k),$$

where $\alpha_k \in \mathbb{R}$ is the step size. In the second step, vectors $\mathbf{y}'_i(k)$ are retracted on sphere \mathcal{S} in order to satisfy the geometric constraints. The retraction function

$$\mathcal{R}_{\mathbf{y}_i(k)} : \mathcal{T}_{\mathbf{y}_i(k)}\mathcal{S} \rightarrow \mathcal{S}$$

normalizes the vectors such that the new iterate is defined by

$$\begin{aligned} \mathbf{y}_i(k+1) &= \mathcal{R}_{\mathbf{y}_i(k)}(\alpha_k \boldsymbol{\eta}_i(k)) \\ &= \frac{\mathbf{y}_i(k) + \alpha_k \boldsymbol{\eta}_i(k)}{\|\mathbf{y}_i(k) + \alpha_k \boldsymbol{\eta}_i(k)\|}. \end{aligned}$$

The value of the step size α_k is adjusted to ensure that the cost function (2) decreases at each iteration.

In Section III, the search directions are computed by differentiating the cost function. However, ranks are piecewise constants and their derivatives are null almost everywhere. As a workaround, we use a specific model of the matrix of pairwise ranks given by $\mathbf{R} = [\rho_{ij}]_{1 \leq i, j \leq N}$. This model allows us to empirically relate distances to ranks and thus evaluate $\boldsymbol{\eta}_i(k)$ in an alternative way. For some given configuration of $\mathbf{Y}(k)$, Fig. 2 shows the observed relationship between δ_{ij} (the pairwise distances) and ρ_{ij} (the pairwise ranks). At

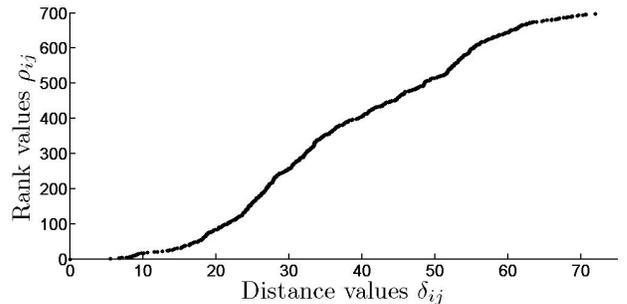


Fig. 2. Representation of the rank values ρ_{ij} with respect to the distance values δ_{ij} related to an embedded data \mathbf{y}_i ; $1 \leq j \leq N$.

each update of $\mathbf{Y}(k)$, this relationship can be interpolated by

differentiable functions $\psi_j : \mathcal{S} \rightarrow \mathbb{R} : \mathbf{y}(k) \mapsto \psi_j(\mathbf{y}(k))$ that satisfy the following conditions:

$$\psi_j(\mathbf{y}_i(k)) = \rho_{ij}(k), \forall (i, j). \quad (4)$$

To be consistent with the asymmetry of ranks, the interpolation functions ψ_j only depend on the value of $\mathbf{y}_i(k)$. The interpolation problem amounts to finding functions $\psi_j : \mathbb{R}^+ \rightarrow \mathbb{R} : \delta \mapsto \tilde{\psi}_j(\delta)$ such that

$$\tilde{\psi}_j(\delta_{ij}) = \rho_{ij}, \forall (i, j),$$

where the geodesic distance δ_{ij} between \mathbf{y}_i and \mathbf{y}_j is given by

$$\delta_{ij} = \arccos \left(\frac{\mathbf{y}_i^T \mathbf{y}_j}{\|\mathbf{y}_i\| \|\mathbf{y}_j\|} \right).$$

Using the cost function in (2) and the definition of $\tilde{\psi}_j$, the minimization problem is thus rewritten as

$$\min_{\mathbf{Y} | \mathbf{y}_i \in \mathcal{S}} E(\mathbf{Y}; \mathbf{X}), \quad (5)$$

where

$$E(\mathbf{Y}; \mathbf{X}) = \lambda \sum_{i,j=1}^N \frac{(r_{ij} - \tilde{\psi}_j(\delta_{ij}))^2}{b + r_{ij}} + (1 - \lambda) \sum_{i,j=1}^N \frac{(r_{ij} - \tilde{\psi}_j(\delta_{ij}))^2}{b + \tilde{\psi}_j(\delta_{ij})}.$$

Search direction $\boldsymbol{\eta}_i(k)$ is computed by differentiating the cost function in (5) with respect to \mathbf{y}_i . Because of the asymmetry of ranks, the derivative is given by

$$\boldsymbol{\eta}_i(k) = - \sum_{j=1}^N \frac{\partial E}{\partial \rho_{ij}} \frac{\partial \rho_{ij}}{\partial \mathbf{y}_i}(\mathbf{y}_i(k)), \quad (6)$$

where $\frac{\partial E}{\partial \rho_{ij}}$ is evaluated by differentiating (2):

$$\frac{\partial E}{\partial \rho_{ij}} = -2\lambda \frac{(r_{ij} - \rho_{ij})}{b + r_{ij}} - (1 - \lambda) \frac{(r_{ij} - \rho_{ij})(r_{ij} + \rho_{ij} + 2b)}{(b + \rho_{ij})^2}.$$

Since functions ψ_j and $\tilde{\psi}_j$ interpolate ranks ρ_{ij} , the derivative of the latter is expressed by

$$\begin{aligned} \frac{\partial \rho_{ij}}{\partial \mathbf{y}_i} &= \frac{\partial \psi_j}{\partial \mathbf{y}_i} \\ &= \frac{\partial \tilde{\psi}_j}{\partial \delta_{ij}} \frac{\partial \delta_{ij}}{\partial \mathbf{y}_i}. \end{aligned}$$

We can thus rewrite (6) as

$$\boldsymbol{\eta}_i(k) = - \sum_{j=1}^N \frac{\partial E}{\partial \rho_{ij}} \frac{\partial \rho_{ij}}{\partial \delta_{ij}} \frac{\partial \delta_{ij}}{\partial \mathbf{y}_i}(\mathbf{y}_i(k)), \quad (7)$$

where

$$\frac{\partial \delta_{ij}}{\partial \mathbf{y}_i} = \frac{- \left(\frac{\mathbf{y}_j}{\|\mathbf{y}_i\| \|\mathbf{y}_j\|} - \frac{\mathbf{y}_i^T \mathbf{y}_j \mathbf{y}_i}{\|\mathbf{y}_i\|^2 \|\mathbf{y}_j\|} \right)}{\sqrt{1 - \left(\frac{\mathbf{y}_i^T \mathbf{y}_j}{\|\mathbf{y}_i\| \|\mathbf{y}_j\|} \right)^2}}. \quad (8)$$

Equation (8) shows that $\frac{\partial \delta_{ij}}{\partial \mathbf{y}_i}$ is orthogonal to \mathbf{y}_i . Therefore this derivative and the search direction $\boldsymbol{\eta}_i(k)$ are on the tangent space $\mathcal{T}_{\mathbf{y}_i(k)} \mathcal{S}$.

Since $\frac{\partial \rho_{ij}}{\partial \delta_{ij}} = \frac{\partial \tilde{\psi}_j}{\partial \delta_{ij}}$, we can evaluate $\frac{\partial \rho_{ij}}{\partial \delta_{ij}}$ by differentiating the interpolation functions $\tilde{\psi}_j$. Nevertheless, results are computed by approximating $\frac{\partial \rho_{ij}}{\partial \delta_{ij}}$: the method derives the rank distribution from distances by a central difference algorithm.

As mentioned in the introduction, dimensionality reduction by rank preservation faces two big issues: the scale indeterminacy of distances with respect to their ranks and non-differentiability of ranks. The method we describe addresses the first one by immersing data in spherical shell; this avoids degenerate representations with outliers and far away scattered points. The non-differentiability of ranks is circumvented by interpolating the observed relationship between ranks and distances. In contrast, the methods described in the literature rely on less straightforward solutions. For instance, in [14], the authors tackle the scale indeterminacy by initializing their rank-preserving method with the result of a distance-preserving method (such as Isomap [21]). Next, they update the data representation by minimizing two cost functions, with the one depending on distances and the other on ranks. A large weight is given to the preservation of distances during the first iterations in order to control the scale of the representation and to prevent points to scatter around. To minimize the second cost function, the same method addresses the non-differentiability of ranks by using pairwise point permutations and force directed placement [7].

V. RESULTS

This section presents experiments on several data sets. The first part of this section is dedicated to the embedding of two toy example data sets. Results on the immersion of a hyperboloid on the unit sphere shows the improvements of preserving ranks rather than pairwise distances. The results are assessed both by visualization and quantitatively by evaluating quality criteria [13]. The second data set gathers items generated on a *frog skin* manifold.

In the second part of this section, experiments are performed on a real data set of pictures of a virtual face [12], [20], [26]. The results of the rank-based and the distance-based method [17] are compared with several state-of-the-art dimensionality reduction methods: Isomap [21], Geodesic Nonlinear Mapping [6], Curvilinear Distance Analysis [11], Stochastic neighborhood embedding [9], Local Tangent Space Alignment [26].

A. Toy example : the hyperboloid

This section presents the embedding results of a hyperboloid on a sphere. 1000 data are randomly generated on the hyperboloid $\mathcal{H} \equiv x^2 + y^2 = \frac{(1+z^2)^2}{2}$, where $z \in [-1.5, 1.5]$. Fig. 3 illustrates the *high-dimensional* data where the color varies with the azimuthal angle of the hyperboloid. [5], [6], [11], [21] approximate the pairwise geodesic distances to avoid shortcuts due to the Euclidean distance. The approximation is achieved by building a graph in the data distribution. Each node, i.e. each item, is joined to its 15 closest neighbors; the resulting edges are weighted

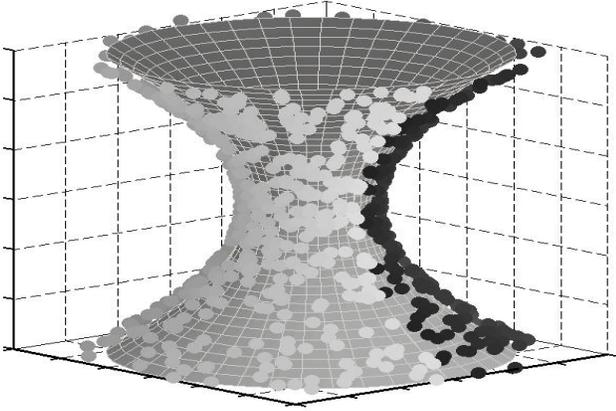


Fig. 3. Original data generated on the hyperboloid \mathcal{H} . The color varies with the azimuthal angle.

by the corresponding Euclidean distances. Then, the pairwise geodesic distances are approximated by computing the shortest path in the connected graph.

The hyperboloid is embedded on a sphere with the proposed rank-based method and the corresponding distance-based method defined by minimizing (1). The minimization procedure aimed at preserving pairwise distances on a sphere is detailed in [17]; this methods also determined the optimal value assigned to the radius of the sphere. Simulations are computed for different values of $\lambda \in [0, 1]$ and $b \in \mathbb{R}^+$. The values of those parameters are set experimentally to improve the quality criteria detailed in [13].

Fig. 4 presents the quality criteria. The two upper curves illustrate the percentage of the K -ary neighborhoods that are preserved. For a fixed value of the size of the neighborhood K , for example $K = 100$, the quality criterion counts data that are in the 100-ary neighborhoods on the hyperboloid and that remain in the corresponding 100-ary neighborhoods when they are embedded on the sphere.

The lower curves quantify the extrusion and intrusion behavior of those local neighborhoods. When the curve is positive, the low-dimensional representation is more intrusive than extrusive. Conversely, the embedding is more extrusive when the curve is negative. Intuitively an intrusive (resp. extrusive) immersion is related to a flattened (resp. torn) immersion.

In Fig. 4, local neighborhoods are better preserved by the rank-based dimensionality reduction than by the distance based method. Considering, for example the 100-ary neighborhoods, 80 percents of the original data remain in those neighborhoods after embedding them with the rank-based method. Conversely, only 65 percents of the 100-ary neighborhoods are correctly preserved by the distance-based dimensionality reduction method.

Fig. 5 (a) and Fig. 5 (b) illustrate the dimensionality reduction results when $\lambda = 0.6$ and $b = 10$ for the rank-based method. In those figures, the color varies with the azimuthal angle of the hyperboloid. Results of Fig. 5 (b) are

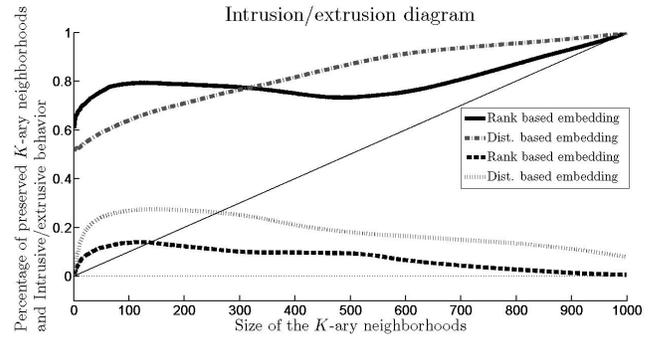


Fig. 4. Quality criteria for the rank-based method where $\lambda = 0.6$ and $b = 10$ and for the distance-based method when $\lambda = 0.4$ and $b = 0.1$

represented in the spherical coordinate space. Note that there are singularities in the north and south pole of the sphere. Moreover, because of the topology of the sphere, embedded data in the left of Fig. 5 (b) are close to the items in the right part of this figure.

The color varies smoothly, with respect to the azimuthal angle of the hyperboloid, which assesses the preservation of the topology. Those results illustrate the benefit of preserving ranks. Actually, the rank-based method enables the local contractions (for data close to the extremities of the hyperboloid) and dilatations (for data close to the center of the distribution, i.e. when $z \approx 0$) of distances.

Fig. 5 (c) and Fig. 5 (d) present the result for the distance-based method when $\lambda = 0.4$ and $b = 0.1$. The distance-based method flattens the hyperboloid since the color does not vary smoothly; this observation is confirmed by the intrusive behavior observed in Fig. 4.

Moreover data are immersed on a small part of the sphere. The resulting low-dimensional representation is close to a flattened immersion on the \mathbb{R}^2 Euclidean space; the topology, hence the loops, of the hyperboloid is not preserved.

B. A frog immersed on a sphere

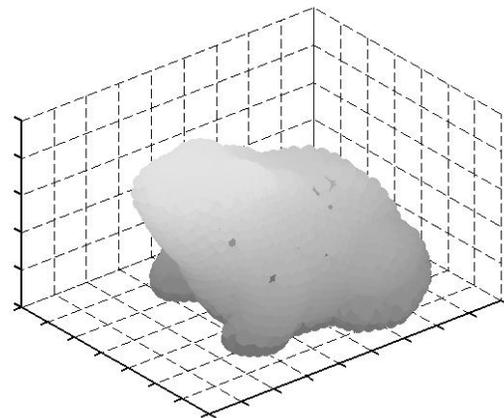


Fig. 6. Data generated on a frog manifold. The color varies with the height of the frog.

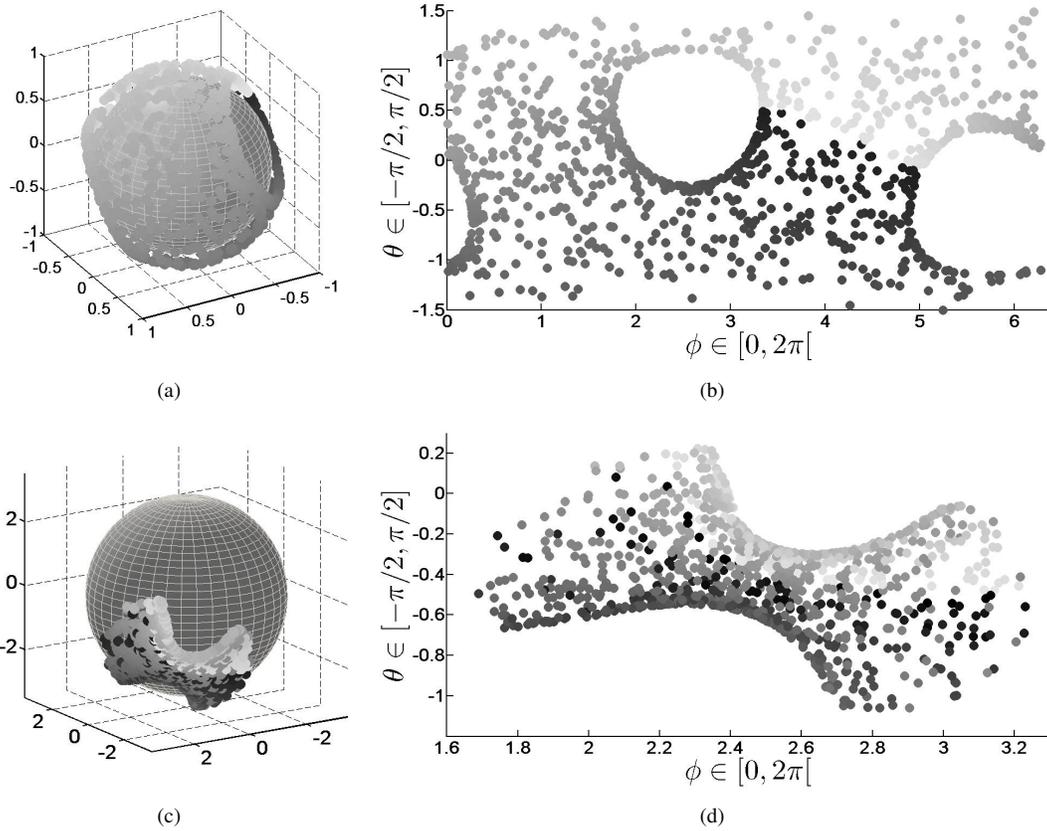


Fig. 5. Representation on the sphere (a) and in the spherical coordinate space (b) of the embedding of the hyperboloid by the rank-based dimensionality reduction method; Representation on the sphere (c) and in the spherical coordinate space (d) of the embedding of the hyperboloid by the distance-based dimensionality reduction method. The color varies with the azimuthal angle of the hyperboloid.

1000 data are generated on a close manifold that has the shape of a *frog* [2] which is homeomorphic to the sphere; the frog skin manifold is topologically equivalent to a spherical shell: the former can be smoothly deformed into the other. This manifold is illustrated in Fig. 6 where the color varies with the height of the frog. Before embedding the frog on the sphere, the geodesic distances are approximated. Simulations are performed both with the rank-based and the distance-based methods.

The results of the rank-based method are presented in Fig. 7 where the color varies smoothly with respect to the height of the frog. Since the homeomorphism between the sphere and the frog skin manifold is nearly an isometric isomorphism, the distance-based method performs as well as the rank-based method as illustrated in Fig. 8. More than 85 percent of the K -ary neighborhoods are preserved when $K > 6$.

C. Virtual face data set

This data set gathers 698 pictures of 64×64 pixels of a virtual face taken from different azimuthal and elevation angles and from different lightings [12], [21], [26]. While pictures are 4096-dimensional vectors, the intrinsic dimension of the picture manifold is only 3. The angle of the camera and of the lighting are known parameters; they are used to qualitatively

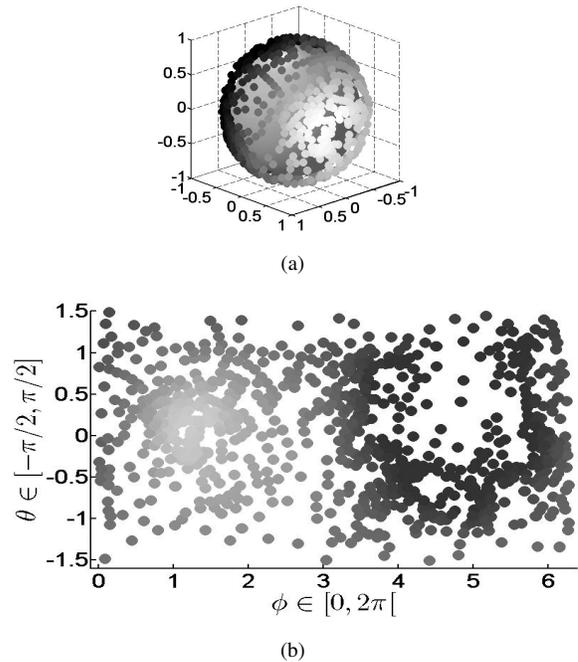


Fig. 7. Embedding of the frog skin manifold on the sphere by the rank-based dimensionality reduction method. Illustration of the results (a) on the sphere and (b) in the spherical coordinate space. The color varies with the height of the frog.

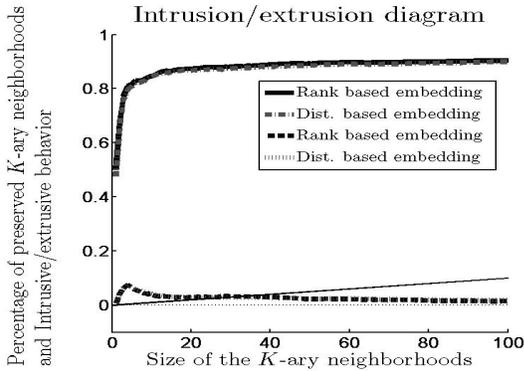


Fig. 8. Quality criteria for the embedding of the frog where $\lambda = 0.4$ and $b = 10$ for the rank-based method and $\lambda = 0.8$ and $b = 1$ for the distance-based method.

evaluate the performances of the method.

This experiment compares several state-of-the-art methods: Isomap [21], Geodesic nonlinear mapping (GNLM) [6], Stochastic neighborhood embedding (SNE) [9], Local Tangent Space Alignment (LTSA) [26] and Curvilinear Distance Analysis (CDA) [11]. Isomap, GNLM and CDA are respectively variants of the Multidimensional scaling (MDS) [22], the Sammon mapping [19] and the Curvilinear Component Analysis (CCA) [5]; those methods use the geodesic distance rather than the usual Euclidean distance. To improve the results performed by SNE, the method is adapted by using the approximation of the geodesic distances.

Fig. 9 (a) illustrates the quality criteria for the different dimensionality reduction methods. Only the percentage of preserved neighborhoods are represented to increase the readability of the figure. Despite CDA improves embedding results for small K -ary neighborhoods ($K \leq 15$), this method is not efficient for medium and large neighborhoods. One can observe that the rank-based method clearly outperforms the other ones, with respect to those quality criteria.

Fig. 9 (b) shows the embedding results for the rank-based dimensionality reduction method. In this figure, the color varies smoothly with respect to the azimuthal angle of the camera. In the center of Fig. 9 (b), dark items are close to light ones due to the poor light level; all those figures are dark.

VI. CONCLUSION

Dimensionality reduction methods try to preserve proximity relationships. Traditionally, the embedding is performed by minimizing a pairwise distance cost function while the quality is assessed by rank-based quality criteria. Those recent quality criteria motivate this work since the preservation of local neighborhoods is quantified by comparing rank matrices.

The dimensionality reduction method presented in this paper preserves rank relationships to render more faithfully the preservation of neighborhoods, hence the local topology of the data distribution. This method improves embedding results by allowing local scaling of distances. Nevertheless,

the rank-based method does not outperform distance-based methods on data sets that are homeomorphic to the low-dimensional space.

The embedding is defined by the minimization of a rank-based cost function that expresses the tradeoff between the risk of flattening and tearing the low-dimensional representation. This compromise is implemented and controlled by a user-defined parameter. Moreover, an additive parameter controls the decrease of the weight function, hence the importance given to small ranks.

Preserving ranks raises some issues: ranks are asymmetric, piecewise constant and they suffer of a scaling invariance. Data distributions are thus immersed on a compact manifold, the unit sphere, to avoid the spread of the data distribution.

Since ranks are piecewise constant, this distribution is interpolated with respect to the corresponding distances. Hypotheses on the ranks allow the differentiation of the interpolation functions. Moreover, those hypotheses and the theory of optimization on manifolds make possible the minimization of the cost function.

First results show the benefit of the proposed rank preservation algorithm. The improvements of the rank-based method are evaluated by comparing the results with a corresponding distance-based method. They are both assessed qualitatively by visualizing the embedding results and quantitatively by using quality criteria. Furthermore, the performances of the method are compared with several state-of-the-art dimensionality reduction methods.

The choice between preserving distances or ranks is left for further works. Nevertheless, since the preservation of ranks is motivated by its ability to scale locally distances, first answers could be found by quantifying those contractions and dilatations. Moreover, the improvements are assessed by comparing this method with the corresponding distance-based method. However, further works will try to determine appropriate cost functions for the rank preservation purpose.

REFERENCES

- [1] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press, January 2008.
- [2] P.-E. Bernard, J.-F. Remacle, R. Comblen, V. Legat, and K. Hillewaert, "High-order discontinuous galerkin schemes on general 2d manifolds applied to the shallow water equations," *Journal of Computational Physics*, vol. 228, no. 17, pp. 6514–6535, 2009.
- [3] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, September 1999.
- [4] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, March 2004.
- [5] P. Demartines and J. Hérault, "Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets," *IEEE transactions on neural networks*, vol. 8, no. 1, pp. 148–154, 1997.
- [6] P. Estevez and C. Figueroa, "Online data visualization using the neural gas network," *Neural Networks*, vol. 19, no. 6, pp. 923–934, 2006.
- [7] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Softw. Pract. Exper.*, vol. 21, no. 11, pp. 1129–1164, 1991.
- [8] D. Gering, "Linear and nonlinear data dimensionality reduction," Ph.D. dissertation, MIT, 2002.
- [9] G. Hinton and S. Roweis, "Stochastic neighbor embedding," in *Advances in Neural Information Processing Systems 15*. MIT Press, 2002, pp. 833–840.

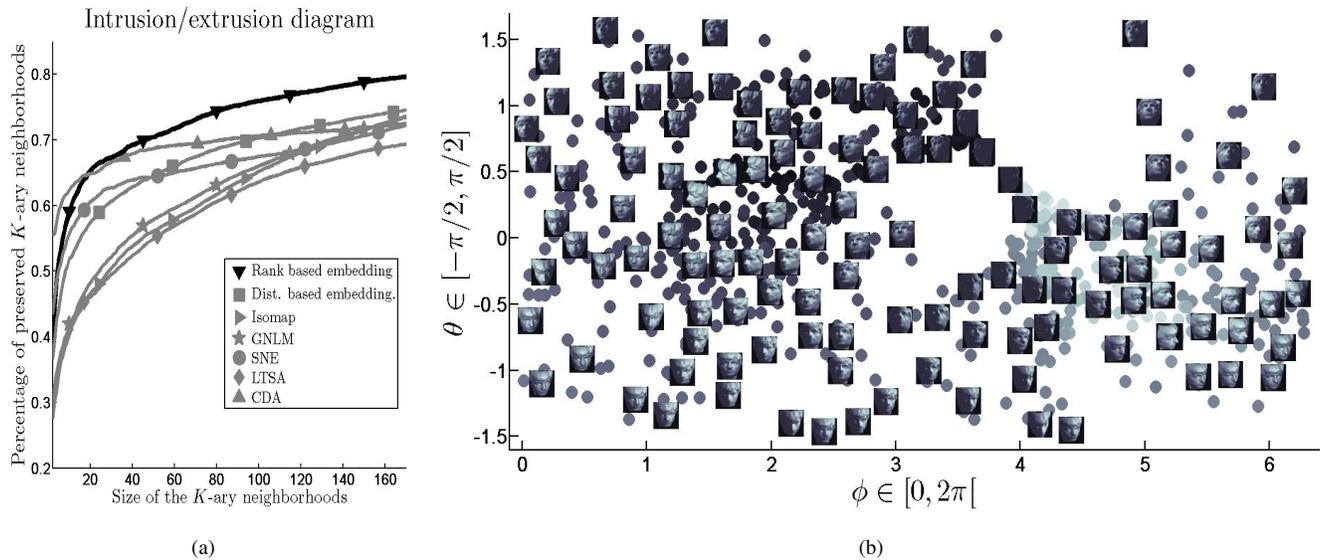


Fig. 9. (a) Quality criteria for the embedding of the virtual face data set; (b) Virtual data set embedded by the rank-based projection method where the color varies with the azimuthal angle of the camera.

- [10] J. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, pp. 1–28, 1964.
- [11] J. Lee, A. Lendasse, and M. Verleysen, "Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis," *Neurocomputing*, vol. 57, 2003.
- [12] J. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*. Springer Science+Business Media, LLC, 2007.
- [13] J. A. Lee and M. Verleysen, "Quality assessment of dimensionality reduction: Rank-based criteria," *Neurocomputing*, vol. 72, no. 7-9, pp. 1431–1443, 2009.
- [14] S. Lespinats, B. Fertil, P. Villemain, and J. Hérault, "Rankvisu: Mapping from the neighborhood network," *Neurocomputing*, vol. 72, no. 13-15, pp. 2964–2978, 2009.
- [15] J. Li, "Visualization of high-dimensional data with relational perspective map," *Information Visualization*, vol. 3, no. 1, pp. 49–59, 2004.
- [16] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, August 1999.
- [17] V. Onclinx, V. Wertz, and M. Verleysen, "Nonlinear data projection on non-euclidean manifolds with controlled trade-off between trustworthiness and continuity," *Neurocomputing*, vol. 72, no. 7-9, pp. 1444–1454, 2009, advances in Machine Learning and Computational Intelligence - 16th European Symposium on Artificial Neural Networks 2008.
- [18] K. Pearson, "Analysis of a complex statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, pp. 417–441, 1933.
- [19] J. Sammon, "A nonlinear mapping algorithm for data structure analysis," *IEEE Transactions on Computers*, vol. CC-18, no. 5, pp. 401–409, 1969.
- [20] J. Tenenbaum, "Mapping a manifold of perceptual observations," in *Advances in Neural Information Processing Systems (NIPS 2007)*, M. Jordan, M. Kearns, and S. Solla, Eds., vol. 10, 1998, pp. 682–688.
- [21] J. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, December 2000.
- [22] W. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, December 1952.
- [23] J. Venna and 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-Verlag, August 21-25 2001, pp. 485–491.
- [24] —, "Local multidimensional scaling," *Neural Networks*, vol. 19, no. 6, pp. 889–899, 2006.
- [25] K. Q. Weinberger and L. K. Saul, "An introduction to nonlinear dimensionality reduction by maximum variance unfolding," in *AAAI'06: proceedings of the 21st national conference on Artificial intelligence*. AAAI Press, July 16-20 2006, pp. 1683–1686.
- [26] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM J. Sci. Comput.*, vol. 26, no. 1, pp. 313–338, 2005.