Citation: Cite this paper as: **S. Fiori**, *Visualization of Riemannian-manifold-valued elements by multidimensional scaling*, Neurocomputing, Vol. 74, No. 6, pp. 983 – 992, February 2011

Visualization of Riemannian-Manifold-Valued Elements by Multidimensional Scaling

Simone Fiori

Dipartimento di Ingegneria Biomedica, Elettronica e Telecomunicazioni (DiBET)
Facoltà di Ingegneria, Università Politecnica delle Marche,
Via Brecce Bianche, Ancona I-60131, Italy
(eMail: s.fiori@univpm.it)

Abstract

The present contribution suggests to utilize a multidimensional scaling algorithm as a visualization tool for high-dimensional smoothly-constrained learnable-system's patterns that lie on Riemannian manifolds. Such visualization tool proves useful in machine learning whenever learning/adaptation algorithms insist on high-dimensional Riemannian parameter manifolds. In particular, the manuscript describes the cases of interest in the recent scientific literature that the parameter space is the set of special orthogonal matrices, the unit hypersphere and the manifold of symmetric-positive definite matrices. The paper also recalls the notion of multidimensional scaling and discusses its algorithmic implementation. Some numerical experiments performed on toy problems help the readers to get acquainted with the problem at hand, while experiments performed on independent-component-analysis data as well as averaging data show the usefulness of the proposed visualization tool.

Keywords: Riemannian parameter manifold, geodesic distance, multidimensional scaling, visualization tool.

1. Introduction

In machine learning, signal processing and intelligent data analysis, patterns to analyze and adaptive systems' parameters are oftentimes organized as vectors or matrices whose entries satisfy non-linear constraints.

For example, symmetric positive-definite matrices find a wide range of applications, such as in the analysis of deformation [43, 45], in automatic and intelligent control [6], in pattern recognition [41, 50], in speech emotion classification [51] as well as in the design and analysis of wireless cognitive dynamic systems [27]. Several applications involve orthogonal connection patterns with unitary determinant, such as invariant visual perception [44], modeling of DNA chains [28, 35], automatic object pose estimation [49], the study of plate tectonics [42], blind source separation and independent component analysis [32], curve subdivision in nonlinear spaces [43, 52], supervised feature construction methods in inductive transfer learning [39]. A number of adaptive signal/data processing algorithms learn parameter-vectors on the unit-hypersphere, as blind channel deconvolution algorithms [17, 19, 46], robust constrained beamforming algorithms [16] and data classification algorithms that work by linear discrimination based on non-Gaussianity discovery [38]. Moreover, Stiefel-manifold-based algorithms have become increasingly popular in the scientific literature, as optimal linear data compression, noise reduction and signal representation by principal/minor component analysis and principal/minor subspace decomposition algorithms [1, 14, 29, 37], algorithms for smart sensor arrays [5, 40], direction-of-arrival estimation algorithms [1, 34], linear programming algorithms [4] and techniques to perform factor analysis in psychometrics [13]. As another interesting case, the manifold of real symplectic matrices found applications that range from quantum computing [3, 24] to the control of beam systems in particle accelerators [11] and from computational ophthalmology [25, 26] to vibration analysis [47] and

control theory [10].

The above-mentioned cases may be framed as manifold-valued patterns analysis. A problem of practical impact is the visualization of such high-dimensional data. Manifold-valued data and parameters manifolds are notoriously difficult to visualize and present in a straightforward form (see, for example, [21, 53]). A possible solution to high-dimensional manifold-valued patterns visualization problem is provided by multidimensional scaling (MDS). Multidimensional scaling is a numerical technique whose goal is to find out a low-dimensional representation of high-dimensional abstract objects suitable for graphical representation. In particular, the aim of multidimensional scaling is to reduce the dimensionality of data while retaining their mutual proximity properties. As the purpose of multidimensional scaling is to compute a set of coordinate vectors (in $\mathbb{R}^2$ or $\mathbb{R}^3$, for visualization purpose) whose distribution reflects a given pattern of proximity, a key point in visualization is the possibility to compute distances among objects on a manifold. The present manuscript deals with Riemannian manifolds. Give a Riemannian manifold $\mathbb{X}$, it is possible to define a binary function $d : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ that measures the distance of two sufficiently close elements. Therefore, given a collection $\{x_k\}_k$ of high-dimensional patterns in $\mathbb{X}$, it is possible to compute their pairwise distances $d(x_i, x_j)$. The purpose of multidimensional scaling is to compute a set of vectors $z_k \in \mathbb{R}^p$ such that the distances $\|z_i - z_j\|$ are as close as possible to the distances $d(x_i, x_j)$ via a MDS-map $\mathbb{X} \to \mathbb{R}^p$. From an algorithmic point of view, therefore, multidimensional scaling inputs a proximity matrix with elements $d(x_i, x_j)$ and outputs a set of 2-dimensional or 3-dimensional vectors that retain (as closely as possible) the given pairwise proximity.

The proposed visualization tool is intended as a support for testing and evaluating machine learning and machine-learning-based signal processing algorithms insisting on Riemannian manifolds. The Section 2 of the present manuscript reviews metric notions of Riemannian manifolds and the theory of multidimensional scaling as well as details on its implementation. Sec-

tion 2 also discusses the content of the present manuscript in relation with the field of "manifold learning". Section 3 shows numerical examples about high-dimensional manifold-valued data visualization. Section 4 concludes the paper.

2. Riemannian manifolds and multidimensional scaling

The present section reviews some metric notions of smooth manifolds and recalls the way that multidimensional scaling works, its fundamental properties and details about its implementation on a computational platform. The mathematical instrument to cope with Riemannian manifolds is differential geometry. For a general reference on differential geometry, see [48]. Throughout the paper, the over-dot denotes the derivative $\frac{d}{dt}$ while the double over-dot denotes the derivative $\frac{d^2}{dt^2}$.

2.1. Riemannian manifolds

On a Euclidean space $\mathbb{E}$, the distance between two points $x, y \in \mathbb{E}$ may be measured as $d^{\mathrm{E}}(x,y) \stackrel{\mathrm{def}}{=} \|x - y\|$, where symbol $\|\cdot\|$ denotes the 2-norm. Such a distance is indeed the length of a straight line connecting endpoints x and y. Such a notion of distance may be extended to a generic Riemannian manifold via the notion of geodesic arcs (which generalize straight lines) and arc length on manifolds.

Let the dataspace of interest be denoted as $\mathbb{X}$. It is supposed to be a Riemannian manifold. Its tangent space at point $x \in \mathbb{X}$ is denoted as $T_x\mathbb{X}$. Given a point $x \in \mathbb{X}$ and any smooth curve $c : [-a\ a] \to \mathbb{X}$, with $a > 0$, such that $c(0) = x$, the tangent space $T_x\mathbb{X}$ is a linear space of dimension $\dim(\mathbb{X})$ generated by the tangent vectors $\dot{c}(0)$. A Riemannian manifold is equipped with a symmetric, positive-definite inner product. Given any pair of tangent vectors $v, w \in T_x\mathbb{X}$, their inner product is denoted by:

$$\langle v, w \rangle_x \in \mathbb{R}. \tag{1}$$

A inner product on $T_x\mathbb{X}$ turns the manifold $\mathbb{X}$ into a metric space. Let $c : [0\ 1] \to \mathbb{X}$ denote a smooth curve on the manifold $\mathbb{X}$. The length of the

curve c is given by:

$$\ell(c) \stackrel{\text{def}}{=} \int_0^1 \langle \dot{c}(t), \dot{c}(t) \rangle_{c(t)}^{\frac{1}{2}} dt. \tag{2}$$

The curve $g : [0\ 1] \to \mathbb{X}$ of minimal length is termed geodesic. Normal parametrization is assumed, namely, the quantity $\langle \dot{g}(t), \dot{g}(t) \rangle_{g(t)}$ keeps constant for every $t \in [0\ 1]$. Such minimal length, namely, the quantity $\ell(g)$, is termed *geodesic distance* between endpoints $g(0) \in \mathbb{X}$ and $g(1) \in \mathbb{X}$. The Riemannian distance between endpoints is denoted by $d(g(0), g(1)) \stackrel{\text{def}}{=} \ell(g)$. Because of normal parametrization, it holds $\ell(g) = \langle \dot{g}(0), \dot{g}(0) \rangle_{g(0)}^{\frac{1}{2}}$. (The above setting may be extended to include pseudo-Riemannian manifolds. Even more general metric spaces might be taken into account, although this exceeds the scope of the present paper.)

Given a manifold of interest, the geodesic distance between two points may be computed by two separate steps, namely, by computing the closed-form of geodesic arcs joining the two points and then by computing its length through equation (2). A way to compute the geodesics on a Riemannian manifold which is computationally profitable is via the variational problem:

$$\delta \int_0^1 \langle \dot{c}(t), \dot{c}(t) \rangle_{c(t)}^{\frac{1}{2}} dt = 0, \tag{3}$$

$$\text{with } \langle \dot{c}(t), \dot{c}(t) \rangle_{c(t)} \text{ constant over the geodesic arc.} \tag{4}$$

In the above equations, the symbol δ denotes variation. The variation refers to the change of the value of the integral when moving from the curve $c(t)$ to an infinitesimally-close curve. For a general reference on the calculus of variations, readers may see, e.g., [23]. Working out the variational problem (3)-(4) gives rise to a second-order differential equation of the form $\ddot{c} = -\Gamma_c(\dot{c}, \dot{c})$, where the function Γ is termed Christoffel form. When the resolvent differential equation is solved under the pair of initial conditions $g(0) = x \in \mathbb{X}$ and $\dot{g}(0) = v \in T_x\mathbb{R}$, the solution is denoted as $g_{x,v}$. In this case, the geodesic curve $g_{x,v}(t)$ emanates from the point x with initial tangent direction v.

The above setting can be made use of in the study of the manifolds of interest in the scientific literature such as the manifold of special orthogo-

nal matrices, the unit hypersphere and the manifold of symmetric-positive definite matrices.

The *special orthogonal group* of matrices is defined as:

$$\mathbb{SO}(q) = \left\{ x \in \mathbb{R}^{q \times q} | x^T x = e_q,\ \det(x) = 1 \right\}, \tag{5}$$

where symbol e_q denotes a $q \times q$ identity matrix, symbol ^{T} denotes ordinary transpose, while symbol det denotes determinant. The tangent spaces of the manifold $\mathbb{SO}(q)$ possess the structure $T_x\mathbb{SO}(q) = \{v \in \mathbb{R}^{q \times q} | v^T x + x^T v = 0\}$. The canonical inner product on the special orthogonal group is:

$$\langle u, v \rangle_x = \mathrm{tr}(u^T v), \tag{6}$$

with $u, v \in T_x\mathbb{SO}(q)$ and where the operator $\mathrm{tr}(\cdot)$ denotes matrix trace. The variational problem that affords the calculation of the geodesic arcs on the manifold $\mathbb{SO}(q)$ with respect to the metric (6) reads:

$$\delta \int_0^1 \mathrm{tr}(\dot{c}^T \dot{c}) dt = 0. \tag{7}$$

The calculus of variations applied to the above problem gives:

$$\int_0^1 \mathrm{tr}(\delta\dot{c}^T \dot{c} + \dot{c}^T \delta\dot{c}) dt = 2 \int_0^1 \mathrm{tr}\left(\dot{c}^T \frac{d\delta c}{dt} \right) dt. \tag{8}$$

Applying the method of integration by parts to the last integral and recalling that the variation δc vanishes at the borders of the geodesic arc leads to the equation:

$$\int_0^1 \mathrm{tr}(\ddot{c}^T \delta c) dt = 0 \text{ with } \delta c \in T_c\mathbb{SO}(q). \tag{9}$$

As the variation δc is arbitrary in the interior of the curve, the above integral vanishes if an only if $\ddot{c} = -\sigma c$ with $\sigma^T = \sigma \in \mathbb{R}^{q \times q}$. The arc c belongs entirely to the space $\mathbb{SO}(q)$, therefore it holds $c^T c = e_q$. Deriving such equation twice with respect to the parameter t gives $\dddot{c}^T c + 2\ddot{c}^T \dot{c} + c^T \ddot{c} = 0$. Recalling that $\dddot{c}^T c = -c^T \sigma c$, it is readily found that $\sigma = c(\ddot{c}^T \dot{c})c^T$. Hence, the geodesic equation in Christoffel form reads $\ddot{c} = -c(\dot{c}^T \dot{c})$. The geodesic curve associated

to the metric (6) that solves the geodesic equation in Christoffel form may be written in closed form as:

$$g_{x,v}(t) = x \exp(t x^T v). \tag{10}$$

In the above equation, operator exp denotes matrix exponential, which is defined by the series $\exp(x) = \sum_{k=0}^{\infty} x^k/k!$. Now, given two distinct points $x, y \in \mathbb{SO}(q)$, a geodesic arc $g_{x,v}(t)$ joining them must satisfy the condition $g_{x,v}(1) = x \exp(x^T v) = y$, hance, $v = x \log(x^T y)$. Consequently, the geodesic distance between points $x, y \in \mathbb{SO}(q)$ satisfies:

$$\begin{aligned} d^2(x,y) &= \mathrm{tr}(v^T v) = \mathrm{tr}(\log^T(x^T y) x^T x \log(x^T y)) \\ &= -\mathrm{tr}(\log^2(x^T y)). \end{aligned} \tag{11}$$

The operator log denotes matrix logarithm, which is defined by the series $\log(x) = -\sum_{k=1}^{\infty}(e_q - x)^k/k$.

The differentiable manifold termed *unit hypersphere* is defined as:

$$\mathbb{S}^{q-1} = \left\{ x \in \mathbb{R}^q | x^T x = 1 \right\}. \tag{12}$$

The tangent space at the point $x \in \mathbb{S}^{q-1}$ has structure $T_x\mathbb{S}^{q-1} = \{v \in \mathbb{R}^q | v^T x = 0\}$. The canonical inner product on the unit hypersphere is:

$$\langle u, v \rangle_x = u^T v, \ \forall u, v \in T_x\mathbb{S}^{q-1}, \tag{13}$$

The variational problem that affords the calculation of the geodesic arcs on the manifold $\mathbb{S}^{q-1}$ with respect to the metric (13) reads:

$$\delta \int_0^1 \dot{c}^T \dot{c} dt = 0. \tag{14}$$

It may be solved by reasoning as already done for the special orthogonal group of matrices. The associated geodesic may be given a closed-form expression, namely:

$$g_{x,v}(t) = x \cos(\|v\| t) + v \|v\|^{-1} \sin(\|v\| t), \tag{15}$$

which represents a great circle on the hypersphere. The geodesic distance associated to such geodesic arc-function may be calculated as shown before.

The manifold of *symmetric positive-definite matrices* is defined as:

$$\mathbb{S}^+(q) = \{x \in \mathbb{R}^{q\times q} | x^T - x = 0,\ x > 0\}. \tag{16}$$

The tangent spaces have thus the structure $T_x\mathbb{S}^+(q) = \{v \in \mathbb{R}^{q\times q} | v^T - v = 0\}$. The canonical metric adopted for the manifold of symmetric positive definite matrices is:

$$\langle u, v\rangle_x = \mathrm{tr}(ux^{-1}vx^{-1}). \tag{17}$$

In order to find the Christoffel equation for the geodesic arc, the following variational problem needs to be addressed:

$$\int_0^1 \delta \mathrm{tr}(\dot{c}c^{-1}\dot{c}c^{-1})dt = 0, \tag{18}$$

which is equivalent to:

$$\int_0^1 \mathrm{tr}\left(\frac{d}{dt}(\delta c)c^{-1}\dot{c}c^{-1} + \delta(c^{-1})\dot{c}c^{-1}\dot{c}\right)dt = 0,\ \delta c \in T_c\mathbb{S}^+(q). \tag{19}$$

From the identity $cc^{-1} = e_q$, it follows $\delta(c^{-1}) = -c^{-1}(\delta c)c^{-1}$. Substituting the expression of the variation $\delta(c^{-1})$ into the above equation and integrating the first term by parts, gives:

$$\int_0^1 \mathrm{tr}\left(\left(\frac{d}{dt}(c^{-1}\dot{c}c^{-1}) + c^{-1}\dot{c}c^{-1}\dot{c}c^{-1}\right)\delta c\right)dt = 0,\ \delta c \in T_c\mathbb{S}^+(q). \tag{20}$$

The expression in parentheses that multiplies the arbitrary symmetric variation δc is symmetric too, therefore the above integral vanishes if and only if:

$$\frac{d}{dt}(c^{-1}\dot{c}c^{-1}) + (c^{-1}\dot{c})^2c^{-1} = 0. \tag{21}$$

By computing the derivative with respect to the parameter t and by recalling that $\frac{d}{dt}c^{-1} = -c^{-1}\dot{c}c^{-1}$, the Christoffel equation $\ddot{c} = \dot{c}c^{-1}\dot{c}$ arises. The associated geodesic curve reads:

$$g_{x,v}(t) = x^{\frac{1}{2}}\exp(tx^{-\frac{1}{2}}vx^{-\frac{1}{2}})x^{\frac{1}{2}}. \tag{22}$$

The above expression requires the evaluation of square root of a symmetric positive definite matrix. By recalling that $x^{-1}\exp(v)x = \exp(x^{-1}vx)$, the above expression simplifies into $g_{x,v}(t) = x\exp(tx^{-1}v)$. The geodesic distance associated to the geodesic arc-function (22) may be calculated as explained before.

The table 1 summarizes the features of the manifolds of interest, namely, their dimension and the Riemannian distances computed on the basis of the definition (2). Except for the unit hyper-sphere, the dimension of the manifolds of interest grows quickly, as shown in the Figure 1.

Space	**Dimension**	**Inner product**	**Distance function**
$\mathbb{SO}(q)$	$\frac{1}{2}q(q-1)$	$\langle u,v\rangle_x = \mathrm{tr}(u^T v)$	$d(x,y) = \sqrt{-\mathrm{tr}(\log^2(x^T y))}$
$\mathbb{S}^q$	$q-1$	$\langle u,v\rangle_x = u^T v$	$d(x,y) = \arccos(x^T y)$
$\mathbb{S}^+(q)$	$\frac{1}{2}q(q+1)$	$\langle u,v\rangle_x = \mathrm{tr}(x^{-1}ux^{-1}v)$	$d(x,y) = \sqrt{\mathrm{tr}(\log^2(x^{-1}y))}$

Table 1: Summary of the features of manifolds of interest in applications.

2.2. Multidimensional Scaling (MDS)

One of the purposes of multidimensional scaling [8, 30, 31] is to provide a visual representation of the pattern of proximities among a set of high-dimensional objects. Let $\mathbb{X}$ be a high-dimensional Riemannian manifold with distance function $d(\cdot,\cdot)$ and let $\{x_k\}_k$ be a given collection of n elements of $\mathbb{X}$. The aim of multidimensional scaling is to compute a collection of n Euclidean coordinate-vectors $\{z_k\}_k$ ($z_k \in \mathbb{R}^p$, with $p = 2$ or $p = 3$) that replicates the pattern of proximities among the elements x_k. In this instance, multidimensional scaling finds a set of vectors in the two-dimensional or three-dimensional Euclidean space such that the matrix of Euclidean distances among them corresponds – as closely as possible – to some function of the objects' proximity matrix according to a criterion function termed *stress*.

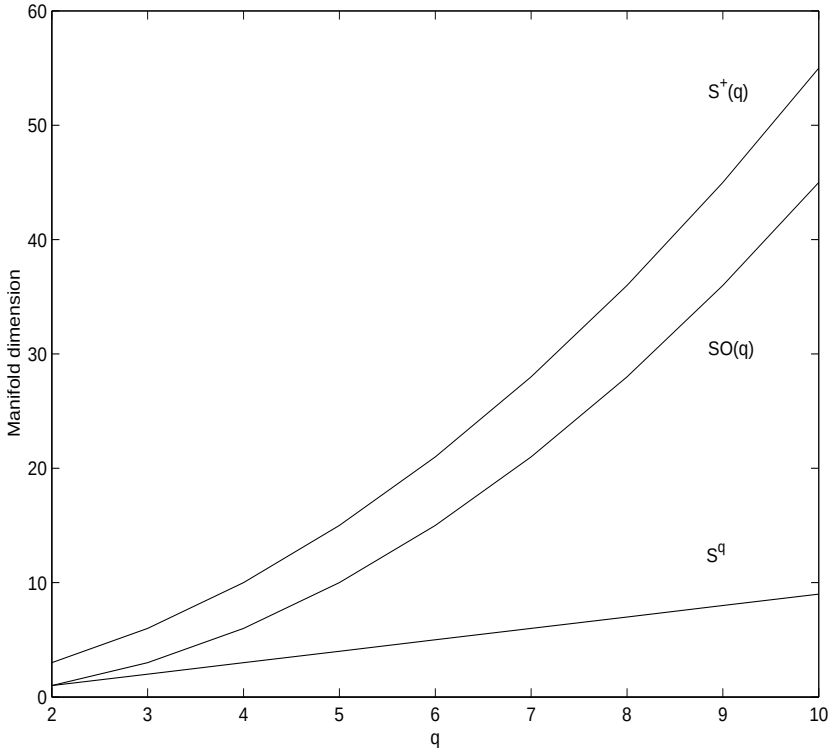

Figure 1: Dimension of the manifolds $\mathbb{SO}(q)$, $\mathbb{S}^q$ and $\mathbb{S}^+(q)$ versus the parameter q. The dimension of a manifold is the number of independent variables needed to parameterize any element of the manifold.

Known stress functions may be unified by the weighted stress function:

$$\Phi(\{z_k\}_k) \stackrel{\text{def}}{=} \sum_i \sum_{j<i} w_{i,j}(\|z_i - z_j\| - d(x_i, x_j))^2, \tag{23}$$

where the quantities $w_{i,j} = w_{j,i} > 0$ denote weights. The canonical stress function, termed *Kruskal stress function*, obtains by setting parameters $w_{i,j} = 1$. Another well-known stress function is the *Sammon stress function*, which obtains by setting $w_{i,j} = d^{-1}(x_i, x_j)$. The Sammon stress function emphasizes the relevance of distances that were originally small. The recent paper [12] introduces the *Dwyer-Koren-Marriott stress function*, that is obtained by the weighting scheme $w_{i,j} = d^{-2}(x_i, x_j)$.

The correspondence $x_k \leftrightarrow z_k$ induced by multidimensional scaling is not unique. In fact, if $\{z_k\}_k$ is a minimizer set of the stress function (23), for a given dimensionality reduction problem, $r \in \mathbb{R}^p$ is a constant displacement vector and $R \in \mathbb{SO}(p)$ is a p-dimensional rotation, also $\{Rz_k + r\}_k$ is a minimizer set.

Multidimensional scaling may be used as a proximity/similarity visualization tool for high-dimensional data as it computes two-dimensional or three-dimensional vectors $z_k \in \mathbb{R}^p$, corresponding to the original elements $x_k \in \mathbb{X}$, that capture the fundamental information about mutual distances. The axes corresponding to the coordinates of the vectors z_k, referred to as 'fictitious coordinates', do not possess any physical meaning, in general. All that matters in an MDS map are the proximity properties. On a MDS-based visualization corresponding to a non-zero stress, the computed coordinate-vectors are imperfect representations of the original data: The greater the stress, the greater the distortion.

A key observation in the implementation of multidimensional scaling is that, rather than optimizing the actual stress function (23), it is more computationally convenient to optimize a quadratic majorization of it [9]. The following details on optimization of a majorizing function are adapted from reference [12]. Define the matrix $A \in \mathbb{R}^{n\times n}$ as follows:

$$A_{i,j} = \begin{cases} -w_{i,j} & \text{for } i \neq j, \\ \sum_{s\neq i} w_{i,s} & \text{for } i = j. \end{cases} \tag{24}$$

Moreover, given a collection of n vectors $\{u_k\}_k$ ($u_k \in \mathbb{R}^p$), define the matrix $C(\{u_k\}_k)$ as:

$$C_{i,j}(\{u_k\}_k) = \begin{cases} -\frac{w_{i,j}d(x_i,x_j)}{\|u_i-u_j\|} & \text{for } i \neq j, \\ -\sum_{s\neq i} C_{i,s}(\{u_k\}_k) & \text{for } i = j. \end{cases} \tag{25}$$

To ease the notation, define matrix $Z \in \mathbb{R}^{n\times p}$, whose rows coincide with the n coordinate-vectors z_k^T, and matrix $U \in \mathbb{R}^{n\times p}$, whose rows coincide with the n coordinate-vectors u_k^T. The stress function (23) is bounded from above by

the quadratic form $\Psi(Z;U)$ defined as:

$$\Psi(Z;U) \stackrel{\text{def}}{=} \sum_i \sum_{j<i} w_{i,j} d^2(x_i, x_j) + \sum_a (Z_a^T A Z_a - 2 Z_a^T C(U) U_a), \tag{26}$$

where Z_a denotes the a^{th} column of the matrix Z and U_a denotes the a^{th} column of the matrix U $(a = 1, \ldots, p)$. The stress function (23) and the new function (26) relate by $\Phi(Z) \leq \Psi(Z;U)$, with equality holding when $Z = U$. In order to iteratively solve the optimization problem of minimizing the criterion (23), the following scheme may be put into effect:

0. Set initial guess $U \in \mathbb{R}^{n \times p}$,
1. Update $U \leftarrow \arg \min_{V \in \mathbb{R}^{n \times p}} \Psi(V;U)$,
2. Repeat from 1 until done,
3. $Z \leftarrow U$.

It is immediate to verify that the optimization problem of minimizing the criterion (26) is equivalent to p independent optimization problems, one for each axis. Consequently, the optimization problem (26) may be reduced to the solution of p quadratic problems of the kind:

$$Z_a^T A Z_a - 2 Z_a^T C(U) U_a. \tag{27}$$

As the matrix A is positive semi-definite, the criterion function to optimize is convex. Moreover, note that the matrix A is constant across the p optimization problems. In order to iteratively solve the convex quadratic optimization problem:

$$\min_{z \in \mathbb{R}^n} (z^T A z + z^T b), \ A \geq 0, \ b \in \mathbb{R}^n, \tag{28}$$

the following gradient steepest descent algorithm with line search [12] may

be put into effect[1]:

$$
\begin{aligned}
&0.\ \text{Set initial guess } z \in \mathbb{R}^n,\\
&1.\ \text{Set gradient } \gamma \ \leftarrow\ 2Az + b,\\
&2.\ \text{Set stepsize } s \ \leftarrow\ \frac{1}{2}\frac{\gamma^T\gamma}{\gamma^T A\gamma},\\
&3.\ \text{Update } z \ \leftarrow\ z - s\gamma,\\
&4.\ \text{Repeat from 1 until done.}
\end{aligned}
$$

As the matrix A is semi-definite, it is rank-deficient. As a consequence, the optimization problem (28) may not be solved in closed form and does not admit a unique solution.

2.3. Relationships with general dimensionality reduction problem

One of the problems with high-dimensional data is that, in many cases, not all the measured variables are important for understanding the underlying phenomena. While certain novel methods can construct models from high-dimensional data, it is still of interest in many applications to reduce the dimension of the original data prior to any modeling. An interesting example of such situation arises in computational biology [54]. Dimensionality reduction aims at determining a transformation of high-dimensional data into a representation of reduced dimensionality. Ideally, the reduced representation has a dimensionality that corresponds to the intrinsic dimensionality of the data, namely, the minimum number of parameters needed to account for the observed properties of the data. A recent comprehensive survey on dimensionality reduction is [33]. A well-known instance of dimensionality reduction is feature selection. In machine learning, pattern recognition and image processing, data may be of too large dimensionality to be processed as they are and may be redundant. A transformation of the data is sought for that produces a reduced representation in terms of *features* through a

[1]Calculations show that the optimal stepsize s given in [12] is incorrect of a factor 2.

procedure termed 'feature extraction'. The next step consists in choosing the extracted features that retain the relevant information from the data with respect to the desired task. As an example, an application of feature extraction/selection to machine vision in robotics is described in [15].

The dimensionality-reduction data-mapping may be *linear* as well as *non-linear*. Linear techniques include principal component analysis, independent component analysis, factor analysis and linear discriminant analysis. Non-linear techniques include multidimensional scaling, "isomap", neural auto-encoding, random projection and local linear embedding. Non-linear techniques such as the "isomap" and local linear embedding exploit the notion of *manifold learning*. Manifold learning algorithms take a finite data set and try to discover the structure of the manifold that the data belong to and then to reduce the dimensionality of the data by approximating the distances among points on the manifold. The problem addressed by the present paper is quite different. The starting point of the present paper is that the structure of the data-manifold is known and the distances on these manifolds may be calculated in closed form.

As mentioned, multidimensional scaling renders a distance matrix into a low-dimensional Euclidean space. The specialized literature illustrates applications where the target space of rendering, rather than being a Euclidean space, is a smooth manifold itself. In particular, *spherical multidimensional scaling* deals with the problem of rendering a matrix of distances onto a (low-dimensional) sphere, for example $\mathbb{S}^2$. Spherical multidimensional scaling has applications in texture mapping, image analysis as well as dimensionality reduction for finite dimensional distributions. A spherical dimensionality reduction method may have considerable impact in domains that represent data as histograms or distributions, such as in document processing and speech recognition [2]. An algorithmic framework for spherical multidimensional scaling has recently been described in [2]. Specific 2-dimensional and 3-dimensional visualization techniques on spheres based on clustering have been developed in bioinformatics [7, 36]. In the present context, spherical

multidimensional scaling is not attractive because the obtained visualization appears less appealing than the standard planar visualization on $\mathbb{R}^2$ or the spatial visualization on $\mathbb{R}^3$. A problem that might arise, for instance, is that on a static spherical visualization a hemisphere of the target space $\mathbb{S}^2$ is always hidden to the view.

3. Numerical Illustrative Examples

The present section aims at illustrating the behavior of the MDS-based manifold-valued elements visualization tool. In all the following examples, the stress function proposed in [12] is made use of.

The computational complexity of the algorithm is essentially the complexity of an unconstrained quadratic optimization problem of size n, where n denotes the number of samples. For a discussion on the computational complexity of the optimization algorithm of section 2.2, readers may refer to the source paper [12].

3.1. Expository cases-study

The following examples aim at illustrating the numerical features of the multidimensional scaling algorithm utilized as a visualization tool.

The first example concerns the famous experiment with city-distances: The distance pattern among 9 cities inputs the multidimensional scaling algorithm which computes the coordinates of 9 bi-dimensional points as shown in the Figure 2. The figure shows that the multidimensional-scaling optimization algorithm tends to minimize the stress which indeed is a measure of discrepancy between the pattern of proximity among cities ($9 \times 9 = 81$ distances shown on the upper-right panel) and the pattern of proximity of the coordinate vectors (81 distances shown on the lower-right panel).

The following two examples of MDS maps concern a distribution of random points in $\mathbb{R}^2$ visualized on $\mathbb{R}^2$. As the minimum distance problem (23) does not possess a unique solution, it is not reasonable to expect that a MDS

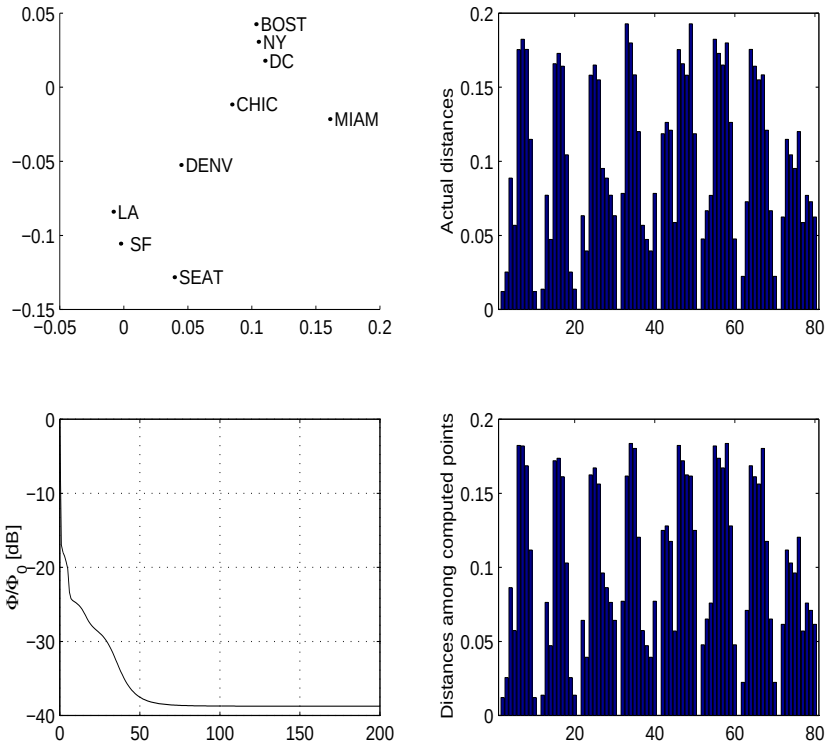

Figure 2: Example on city-distances. Upper-left panel: Locations of the computed coordinate-points (labeled for clarity). Lower-left panel: Stress function during iteration normalized to initial stress function (ratio expressed in decibels). Upper-right and lower-right panels: Pattern of distances among actual points and computed coordinate-points, respectively.

map $\mathbb{R}^2 \to \mathbb{R}^2$ coincides necessarily with the identity map. The set of coordinate points computed by the algorithm of section 2.2 depends on the chosen initial state. Figure 3 shows a result obtained with initial guess just slightly randomly shifted with respect to the actual points. In this case, the computed coordinate-points and the actual points coincide essentially. The same holds true for the 10×10 matrices of actual and computed distances. The same experiment was repeated with a random initial guess: Figure 4 shows that, although the set of coordinate-points replicates the distance pattern of the actual points, their locations are seemingly unrelated with those of

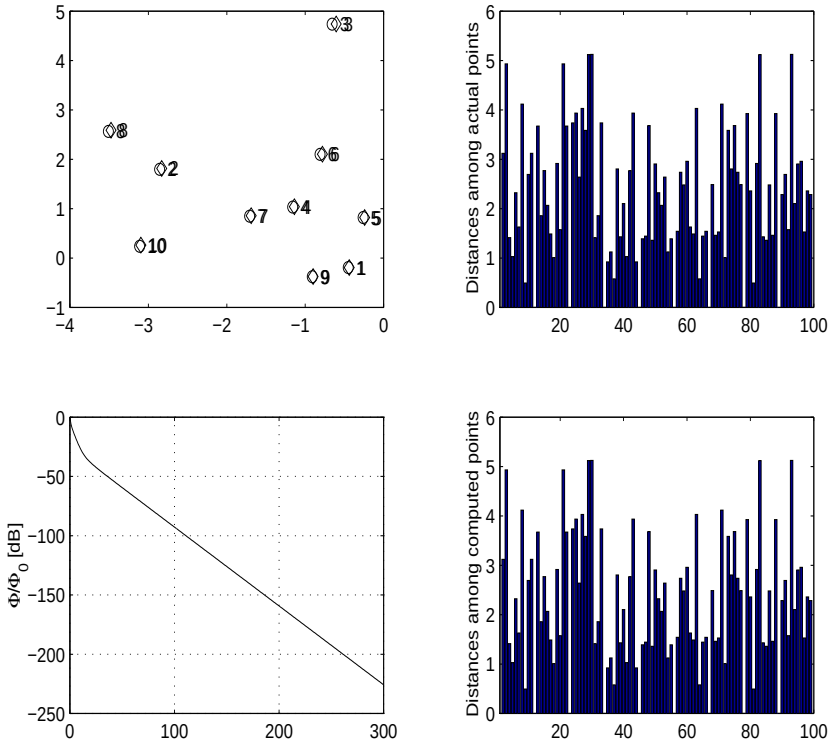

Figure 3: Illustrative case-study on finding a $\mathbb{R}^2 \to \mathbb{R}^2$ MDS map starting from a slightly randomly shifted initial guess. Upper-left panel: Locations of the actual points (diamond) and the computed coordinate-points (open circles). Note that the points are numbered for clarity. Lower-left panel: Stress function during iteration normalized to initial stress function (ratio expressed in decibels). Upper-right and lower-right panels: Pattern of distances among actual points and computed coordinate-points, respectively.

the actual points. However, points that are actually close to each other (for example, points marked as 3 and 9) keep close to each other in the computed MDS-based representation.

A further example concerns the 2-dimensional visualization of a distribution of points on a three-dimensional unit sphere. Figure 5 shows the actual points on a three-dimensional spherical surface, labeled for clarity. Figure 6 shows the result obtained by applying the multidimensional scaling algorithm of section 2.2. In this instance, the MDS map is of kind $\mathbb{S}^2 \to \mathbb{R}^2$. The points

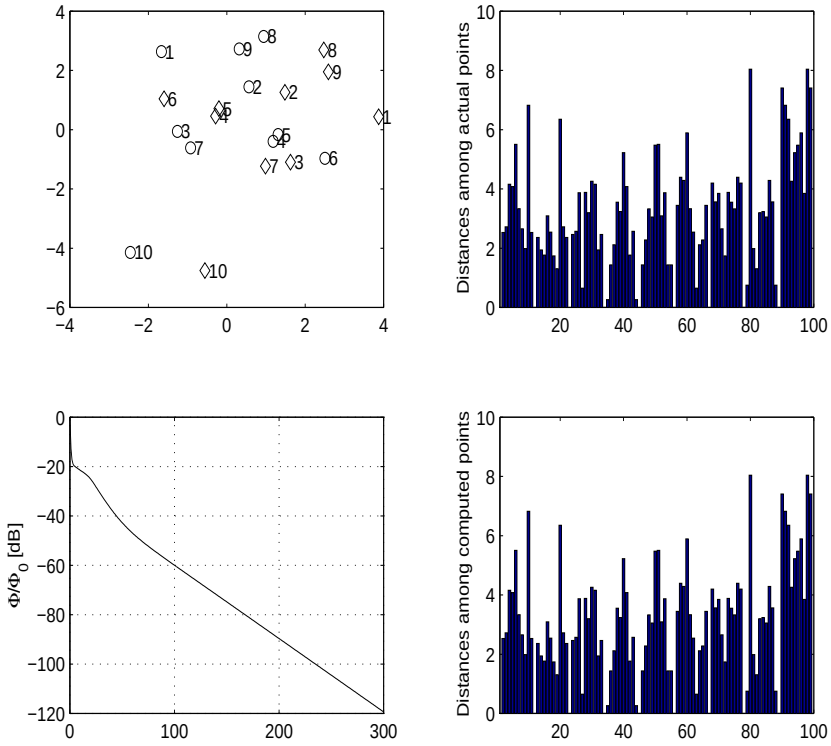

Figure 4: Illustrative case-study on finding a $\mathbb{R}^2 \to \mathbb{R}^2$ MDS map starting from a random initial guess. Upper-left panel: Locations of the actual points (diamond) and the computed coordinate-points (open circles). Note that the points are numbered for clarity. Lower-left panel: Stress function during iteration normalized to initial stress function (ratio expressed in decibels). Upper-right and lower-right panels: Pattern of distances among actual points and computed coordinate-points, respectively.

that are close to each other on the sphere (for example, points 16 and 19) are close to each other on the bi-dimensional plane too.

3.2. Illustrative applications

The following examples aim at illustrating the usage of multidimensional scaling as a visualization tool for adaptive algorithms which insist on Riemannian manifolds.

The MDS-based visualization tool may be profitably used to inspect the trajectory of the manifold-valued state of a machine-learning algorithm. An

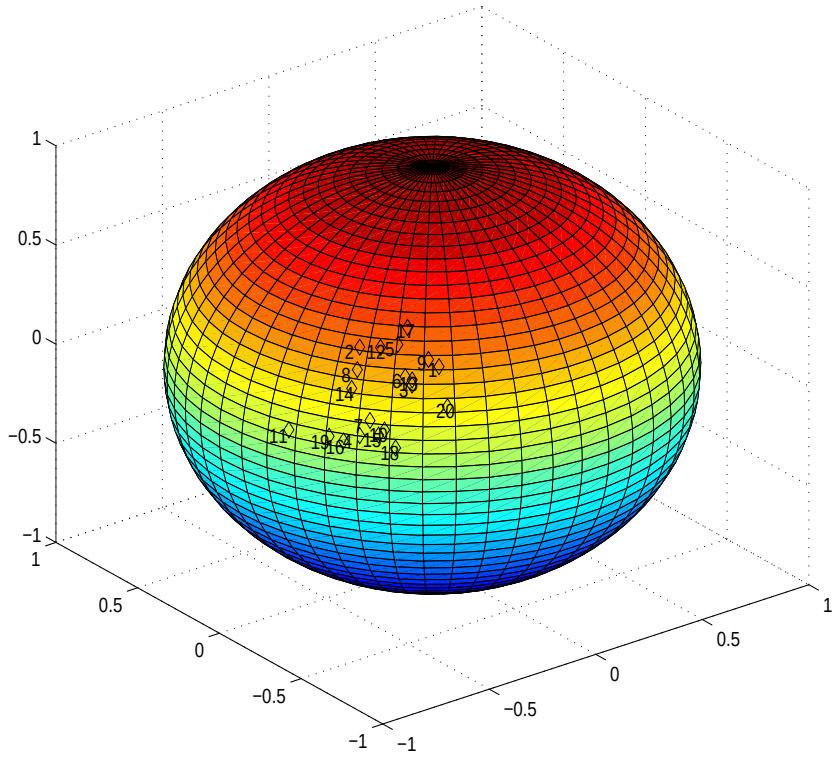

Figure 5: Illustrative case-study on finding a $\mathbb{S}^2 \to \mathbb{R}^2$ MDS map. Random distribution of points on the unit sphere.

exemplary application of MDS-based visualization tool is to inspect the learning trajectory of a independent component analysis algorithm. In independent component analysis, the observable stream may first get pre-whitened, so that learning a separation operator may be reduced to the space of multi-dimensional rotations $\mathbb{SO}(q)$, where q denotes the actual number of independent components. As a case-study, an instance of independent component analysis algorithm, namely a non-negative independent component analysis (NNICA) algorithm, is considered, with $q = 9$. According to Table 1, every 9×9 orthogonal matrix with unitary determinant may be parameterized with $9 \times 8/2$ independent parameters. It forms, therefore, a 36-dimensional space, whose elements may be visualized on a 2-dimensional or a 3-dimensional

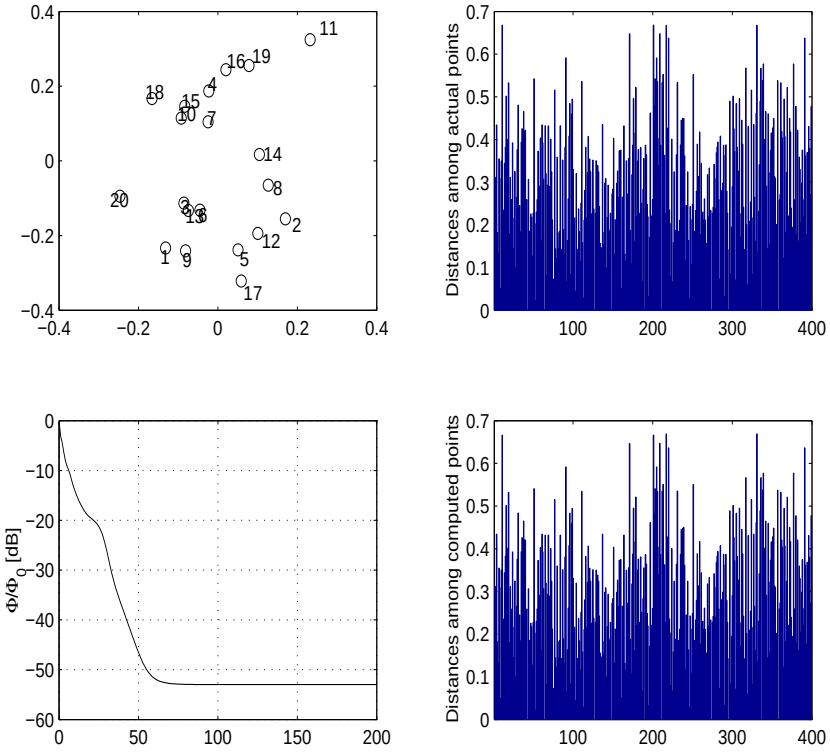

Figure 6: Illustrative case-study on finding a $\mathbb{S}^2 \to \mathbb{R}^2$ MDS map. Upper-left panel: Stress function during iteration normalized to initial stress function (ratio expressed in decibels). Lower-left panel: Distribution of computed points, labeled for clarity. Upper-right and lower-right panels: Pattern of distances among actual points and computed coordinate-points, respectively.

space by the help of a dimensionality reduction tool such as the multidimensional scaling. Figure 7 refers to the trajectory of a NNICA algorithm [18] which learns on the parameter-space $\mathbb{SO}(9)$. Every time-step of the algorithm generates a 9×9 orthogonal matrix with unitary determinant. The illustrated visualization by multidimensional scaling is based on a map $\mathbb{SO}(9) \to \mathbb{R}^3$. The NNICA algorithm run through 200 time-steps, down-sampled to 20 steps for visual tidiness (initial point labeled as '1', final point labeled as '20'). The succession of points (especially the accumulation of points corresponding to the final learning steps) clearly evidences a convergent learning trajectory.

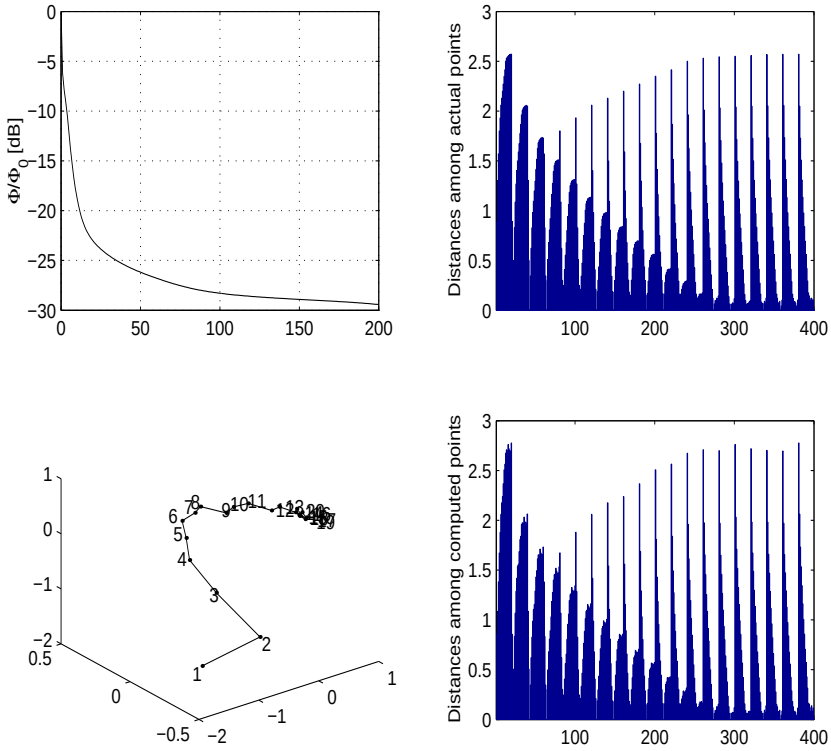

Figure 7: Illustrative application about finding a $\mathbb{SO}(9) \to \mathbb{R}^3$ MDS map. The trajectory refers to a NNICA learning algorithm. The sequence of points rather clearly reveals a trajectory on the three-dimensional visualization space.

A further example of application of visualization of high-dimensional data arises from the need of inspecting the behavior of an averaging algorithm on (matrix-type) Lie groups. Data-averaging over Lie groups has a number of important applications, such as getting rid of random fluctuations in measurements and computing an average connection pattern out of a set of learnt neural networks [20]. An interesting case-study is the inspection of an averaging algorithm over the group of symmetric positive-definite matrices $\mathbb{S}^+(q)$. By the notions of Fréchet mean and Fréchet variance [22], the manifold $\mathbb{S}^+(q)$ may be equipped with first-order and second-order statistical descriptors, of particular use, e.g., in medical neuroimaging. Informally, a Fréchet average matrix is defined as a matrix that lays as close as possible to all matrices to

average. Namely, it minimizes the sum of squared distances between itself and all matrices in the dataset. As a case-study, an instance of averaging over the group of symmetric positive-definite matrices is considered, with $q = 9$. According to Table 1, every 9×9 symmetric positive-definite matrix may be parameterized with $9 \times 10/2$ independent parameters. It forms, therefore, a 45-dimensional space, whose elements may be visualized on a 2-dimensional or a 3-dimensional space by the help of multidimensional scaling. Figure 8 refers to a data-constellation that represents a set of 50 symmetric positive-definite matrices to average. The illustrated visualization by multidimensional scaling is based on a map $\mathbb{S}^+(9) \to \mathbb{R}^2$. In the Figure 8, the

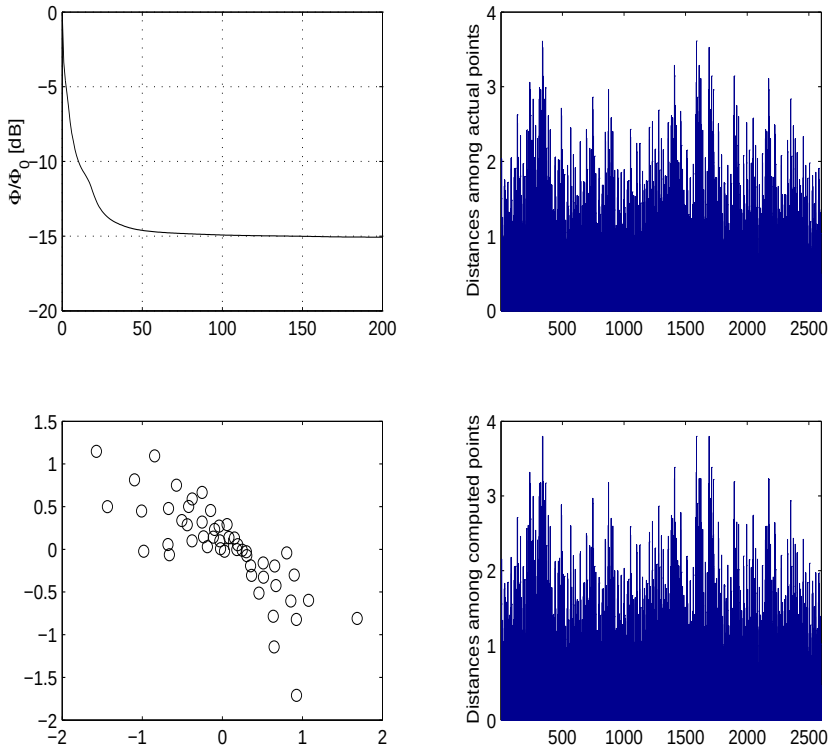

Figure 8: Illustrative application about finding a $\mathbb{S}^+(9) \to \mathbb{R}^2$ MDS map. The data-constellation is visualized by open circles. The average symmetric positive-definite matrix computed by the algorithm proposed in [20] is visualized by a '+' symbol.

data-constellation refers to a set of symmetric positive-definite matrices to

average. The average symmetric positive-definite matrix computed by the algorithm proposed in [20] is visualized as well. Visual inspection suggests that the computed average symmetric positive-definite matrix locates amidst the data-matrices and lays as close as possible to every element in the constellation.

4. Conclusion

The present paper suggests the use of multidimensional scaling as a visualization tool for high-dimensional Riemannian-manifolds-valued data encountered in machine learning. Visualization tools are useful in machine learning and machine-learning-based signal processing as they help inspecting the distribution of high-dimensional Riemannian-manifold-valued elements or trajectories on Riemannian manifolds. The MDS-based visualization tool captures the pattern of proximity among high-dimensional manifold-valued elements and computes a set of 2-dimensional or 3-dimensional coordinate-vectors that retain the given pattern of proximity as much as possible.

Multidimensional scaling is a known non-linear dimensionality reduction technique, which, in some context, is paired with manifold learning. Manifold learning algorithms take a finite number of high-dimensional samples and try to discover the structure of the underlying manifold and to approximate the distances among samples on the manifold. The problem addressed by the present paper differs from the dimensionality reduction problems involving "manifold learning", as the structure of the data-manifold is known and the distances on these manifolds may be calculated in closed form.

Some specific cases of manifolds of interest in the scientific literature are discussed in details, namely the manifold of special orthogonal matrices, the hypersphere and the manifold of symmetric positive-definite matrices. The geometry of these spaces has been briefly surveyed and closed-form expressions of distances on these spaces have been recalled.

Numerical experiments were performed on toy problems to illustrate the numerical features of multidimensional scaling as a visualization tool for high-

dimensional data. Results of numerical experiments were also illustrated on two practical problems, namely, the visualization of the trajectory of an independent-component-analysis learning algorithms whose parameter space is the set of 9×9 rotations, and the visualization of the behavior of an algorithm that computes the average of a set of 9×9 symmetric positive-definite matrices. The obtained graphical results clearly show the usefulness of the proposed visualization tool in the context of machine learning.

References

[1] S. Affes and Y. Grenier, *A signal subspace tracking algorithm for speech acquisition and noise reduction with a microphone array*, Proc. of IEEE/IEE Workshop on Signal Processing Methods in Multipath Environments, pp. 64 – 73, 1995

[2] A. Agarwal, J.M. Phillips and S. Venkatasubramanian, *A unified algorithmic framework for multi-dimensional scaling*. Submitted, 2010 (Available at http://arxiv.org/abs/1003.0529.)

[3] S. D. Bartlett, B. Sanders, S. Braunstein and K. Nemoto, *Efficient classical simulation of continuous variable quantum information processes*, Physical Review Letters, Vol. 88, 097904/1-4, 2002

[4] R.W. Brockett, *Dynamical Systems that Sort Lists, Diagonalize Matrices and Solve Linear Programming Problems*, Linear Algebra and Its Applications, Vol. 146, pp. 79 – 91, 1991

[5] J.F. Cardoso and B. Laheld, *Equivariant adaptive source separation*, IEEE Trans. on Signal Processing, Vol. 44, No. 12, pp. 3017 – 3030, 1996

[6] Y. Chen and J.E. McInroy, *Estimation of symmetric positive-definite matrices from imperfect measurements*, IEEE Trans. on Automatic Control, Vol. 47, No. 10, pp. 1721 – 1725, October 2002

[7] A. Ciaramella, S. Cocozza, F. Iorio, G. Miele, F. Napolitano, M. Pinelli, G. Raiconi and R. Tagliaferri, *Interactive data analysis and clustering of genomic data*, Neural Networks, Vol. 21, No.s 2-3, pp. 368 – 378, March-April 2008

[8] T. Cox and M. Cox, *Multidimensional Scaling*, Chapman & Hall (London, UK), 1994

[9] J. de Leeuw, *Applications of convex analysis to multidimensional scaling*, in Recent developments in statistics (Ed.s F. Brodeau, G. Romie, et al.), pp. 133 – 145, 1977

[10] F.M. Dopico and C.R. Johnson, *Complementary bases in symplectic matrices and a proof that their determinant is one*, Linear Algebra and its Applications, Vol. 419, No.s 2-3, pp. 772 – 778, December 2006

[11] A.J. Draft, F. Neri, G. Rangarajan, D.R. Douglas, L.M. Healy and R.D. Ryne, *Lie algebraic treatment of linear and nonlinear beam dynamics*, Annual Review of Nuclear and Particle Science, Vol. 38, pp. 455 – 496, December 1988

[12] T. Dwyer, Y. Koren and K. Marriott, *Stress majorization with orthogonal ordering constraints*, Proceedings of the 13$^{\text{th}}$ International Symposium on Graph Drawing (GD'05, Limerick, Ireland), pp. 141 – 152, LNCS 3843, Springer, 2006

[13] L. Eldén and H. Park, *A Procrustes problem on the Stiefel manifold*, Numerical Mathematics, Vol. 82, pp. 599 – 619, 1999

[14] Y. Ephraim and L. Van Trees, *A signal subspace approach for speech enhancement*, IEEE Trans. on Speech and Audio Processing, Vol. 3, No. 4, pp. 251 – 266, 1995

[15] S. Fiori, F. Grimani and P. Burrascano, *Novel neural network feature selection procedure by generalization maximization with applica-*

tion to automatic robot guidance, International Journal of Smart Engineering System Design, Vol. 4, No. 2, pp. 91 – 106, June 2002

[16] S. Fiori, *Neural minor component analysis approach to robust constrained beamforming*, IEE Proceedings - Vision, Image and Signal Processing, Vol. 150, No. 4, pp. 205 – 218, August 2003

[17] S. Fiori, *A fast fixed-point neural blind deconvolution algorithm*, IEEE Trans. on Neural Networks, Vol. 15, No. 2, pp. 455 – 459, March 2004

[18] S. Fiori, *Quasi-geodesic neural learning algorithms over the orthogonal group: A tutorial*, Journal of Machine Learning Research, Vol. 6, pp. 743 – 781, May 2005

[19] S. Fiori, *Geodesic-based and projection-based neural blind deconvolution algorithms*, Signal Processing, Vol. 88, No. 3, pp. 521 – 538, March 2008

[20] S. Fiori and T. Tanaka, *An algorithm to compute averages on matrix Lie groups*, IEEE Trans. on Signal Processing, Vol. 56, No. 12, pp. 4734 – 4743, December 2009

[21] M. Fisher, D.P. Mandic, J.A. Bangham and R. Harvey, *Visualising error surfaces for adaptive filters and other purposes*, in Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'2000, Istanbul, Turkey), Vol. VI, pp. 3522 – 3525, 2000

[22] M. Fréchet, *Les élements aléatoires de nature quelconque dans un espace distancié*, Annales de l'Institut Henri Poincaré, Vol. 10, pp 215 – 310, 1948

[23] I.M. Gelfand and S.V. Fomin, *Calculus of Variations*, Dover Publications, 2000

[24] V. Guillemin and S. Sternberg, *Symplectic Techniques in Physics*, Cambridge University Press, 1984

[25] W.F. Harris, *Paraxial ray tracing through noncoaxial astigmatic optical systems, and a* 5×5 *augmented system matrix*, Optometry and Vision Science, Vol. 71, No. 4, pp. 282 – 285, 1994

[26] W.F. Harris, *The average eye*, Ophthalmic and Physiological Optics, Vol. 24, pp. 580 – 585, 2004

[27] S. Haykin, *Foundations of Cognitive Dynamic Systems*, Cambridge University Press, 2009

[28] K.A. Hoffman, *Methods for determining stability in continuum elastic-rod models of DNA*, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, Vol. 362, No. 1820, pp. 1301 – 1315, July 2004

[29] J. Karhunen and J. Joutsensalo, *Learning of robust principal component subspace*, Proceedings of the International Joint Conference on Neural Networks (IJCNN'1993, Como, Italy), pp. 2409 – 2412, 1993

[30] J.B. Kruskal, *Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis*, Psychometrika, Vol. 29, pp. 1 – 27, 1964

[31] J.B. Kruskal and M. Wish, *Multidimensional scaling*, Sage University Paper series on Quantitative Application in the Social Sciences, 07-011. Beverly Hills and London: Sage Publications, 1978

[32] T.-W. Lee, *Independent Component Analysis: Theory and Applications*, Kluwer Academic Publishers, September 1998

[33] L.J.P. van der Maaten, E.O. Postma and H.J. van den Herik, *Dimensionality reduction: A comparative review*, Tilburg University Technical Report, TiCC-TR 2009-005, 2009

[34] C.S. MacInnes and R.J. Vaccaro, *Tracking direction-of-arrival with invariant subspace updating*. Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'1996, Atlanta, (GA) USA), pp. 2896 – 2899, 1996

[35] R.S. Manning and G.B. Bulman, *Stability of an elastic rod buckling into a soft wall*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, Vol. 461, No. 2060, pp. 2423 – 2450, August 2005

[36] F. Napolitano, G. Raiconi, R. Tagliaferri, A. Ciaramella, A. Staiano and G. Miele, *Clustering and visualization approaches for human cell cycle gene expression data analysis*, International Journal of Approximate Reasoning, Vol. 47, No. 1, pp. 70 – 84, January 2008

[37] E. Oja, *Neural networks, principal components, and subspaces*, International Journal of Neural Systems, Vol. 1, pp. 61 – 68, 1989

[38] P. Pajunen and M. Girolami, *Implementing decisions in binary decision trees using independent component analysis*, Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation (ICA'2000, Helsinki, Finland), pp. 477 – 481, 2000

[39] S.J. Pan and Q. Yang, *A survey on transfer learning*, IEEE Trans. on Knowledge and Data Engineering, Vol. 22, No. 10, pp. 1345 – 1359, October 2010

[40] A. Paraschiv-Ionescu, C. Jutten and G. Bouvier, *Neural network based processing for smart sensor arrays*, Proceedings of International Conference on Artificial Neural Networks (ICANN'1997, Lausanne, Switzerland), pp. 565 – 570, 1997

[41] N. Prabhu, H.-C. Chang and M. Deguzman, *Optimization on Lie manifolds and pattern recognition*, Pattern Recognition, Vol. 38, No. 12, pp. 2286 – 2300, December 2005

[42] M.J. Prentice, *Fitting smooth paths to rotation data*, The Journal of the Royal Statistical Society Series C (Applied Statistics), Vol. 36, No. 3, pp. 325 – 331, 1987

[43] I.U. Rahman, I. Drori, V.C. Stodden, D.L. Donoho and P. Schröder, *Multiscale representations for manifold-valued data*, Multiscale Modeling and Simulation, Vol. 4, No. 4, pp 1201 – 1232, 2005

[44] R.P.N. Rao and D.L. Ruderman, *Learning Lie groups for invariant visual perception*, Advances in Neural Information Processing Systems (NIPS) 11, pp. 810 – 816, 1999

[45] J. Salencon, *Handbook of Continuum Mechanics*, Springer-Verlag, Berlin, 2001

[46] O. Shalvi and E. Weinstein, *Super-exponential methods for blind deconvolution*, IEEE Trans. on Information Theory, vol. 39, pp. 504 – 519, March 1993

[47] N.G. Stephen, *Transfer matrix analysis of the elastostatics of one-dimensional repetitive structures*, Proceedings of the Royal Society A, Vol. 462, No. 2072, pp. 2245 – 2270, August 2006

[48] M. Spivak, *A Comprehensive Introduction to Differential Geometry*, Volume 1, 2nd Edition, Berkeley, CA: Publish or Perish Press, 1979

[49] A. Srivastava, U. Grenander, G.R. Jensen and M.I. Miller, *Jump-diffusion Markov processes on orthogonal groups for object pose estimation*, Journal of Statistical Planning and Inference, Vol. 103, pp. 15 – 37, 2002

[50] K. Tsuda, G. Rätsch and M.K. Warmuth, *Matrix exponentiated gradient updates for on-line learning and Bregman projection*, Journal of Machine Learning Research, Vol. 6, pp. 995 – 1018, 2005

[51] C. Ye, J. Liu, C. Chen, M. Song and J. Bu, *Speech emotion classification on a Riemannian manifold*, Proceedings of Advances in Multimedia Information Processing (PCM'2008, 9th Pacific Rim Conference on Multimedia, Tainan, Taiwan), Lecture Notes in Computer Science, Volume 5353/2008, pp. 61 – 69, Springer Berlin/Heidelberg, 2008

[52] J. Wallner and H. Pottmann, *Intrinsic subdivision with smooth limits for graphics and animation*. Techincal Report 120, Geometry Preprint Series, Technische Universität Wien, 2004

[53] J. Woelfel and G.A. Barnett, *Multidimensional scaling in Riemann space*, Quality and Quantity, Vol. 16, No. 6, pp. 469 – 491, 1982

[54] Q. Wu, J. Guinney, M. Maggioni and S. Mukherjee, *Learning gradients: Predictive models that infer geometry and statistical dependence*, Journal of Machine Learning Research, Vol. 11, pp. 2175 – 2198, 2010