Manifold learning for multi-dimensional auto-regressive dynamical models

Fabio Cuzzolin

1 Introduction

Manifold learning [2, 5, 28, 32, 36, 12] has become a popular topic in machine learning and computer vision in the last few years, as many objects of interests (like natural images, or sequences representing walking persons), in spite of their apparent high dimensionality, live in a non-linear space of usually limited dimension. Many unsupervised algorithms (e.g. locally linear embedding [27]) take an input dataset and embed it into some other space, implicitly learning a metric. Extensions learning a full metric for the whole input space have been recently formulated [4].

In particular, videos or image sequences are often represented as realizations of some sort of dynamical model, either stochastic (e.g. HMMs) or deterministic (e.g. ARMA). Such an approach has proven to be effective in problems such as video coding (e.g. dynamic textures [11]), action recognition [7], or identity recognition from gait [31]. Several metrics or distance functions on linear systems have been introduced [6] in the context of system identification [21, 35, 30]. A vast literature also exists on dissimilarity measures between Markov models [10, 33], mainly concerning variants of the Kullback-Leibler divergence [18].

Consider, though, the problem of classifying a dynamical model (as the representative of an input image sequence). Since models (or sequences) can be endowed with different labeling, while maintaining the same geometrical structure, no single distance function can possibly outperform all the others in each and every classification problem. A reasonable approach, when possessing some a-priori information, consists therefore in trying to *learn* in a supervised fashion the "best" distance function for a specific classification problem [2, 5, 28, 32, 36, 12]. A natural optimization criterion seeks to maximize the classification performance achieved by means of the learnt metric. Efforts have been done in this sense in the linear case [29, 34].

Fabio Cuzzolin
Department of Computing, Oxford Brookes University, OXFORD, UK OX33 1HX, e-mail: fabio.cuzzolin@brookes.ac.uk

However, as even linear dynamical models live in a nonlinear space, the need for a principled way of learning Riemannian metrics from such data naturally arises. A tool is provided by the formalism of *pullback metrics*. If the models belong to a Riemannian manifold $\mathscr{M}$, any diffeomorphism of $\mathscr{M}$ onto itself (or *automorphism*) induces such a metric on $\mathscr{M}$. By designing a suitable parameterized family of automorphisms we obtain a family of pullback metrics on $\mathscr{M}$ we can optimize upon.

In this paper we propose a general framework for learning the optimal pullback metric for a data-set D of dynamical models. Assume each input observation sequence is mapped to a model of a certain class by parameter identification. If such models belong to a Riemannian manifold (for instance endowed with the Fisher metric [1]) we can design a parametric family of automorphisms which induce a family of pullback metrics. If the training set of models is labeled, we can then find the parameter of the metric which optimizes classification performance by cross-validation [8]. Otherwise, the metric which optimizes some purely geometric objective function can be sought (like, for instance, the inverse volume of the manifold around the data-points in D [19]).
In particular, we consider here the class $\mathscr{AR}(2)$ of multidimensional autoregressive models of order 2. We study the Riemannian structure of their manifold, and design a number of automorphisms inducing families of parameterized pullback metrics on $\mathscr{AR}(2)$. We apply this framework to identity recognition from gait. We use the video sequences of the Mobo database [14] to prove that classifiers based on an optimal pullback Fisher metric between stochastic models significantly improve classification scores with respect to what obtained by standard, a-priori distance functions.

2 Learning pullback metrics for linear models

Let us suppose a data-set of dynamical models is available. Suppose also that such models live on a Riemannian manifold $\mathscr{M}$ of some sort, i.e, a Riemannian *metric* is defined in any point of the manifold. Any automorphism (a differentiable map) from $\mathscr{M}$ to itself induces a new metric, called "pullback metric". The use of pullback metrics has been recently proposed by Lebanon [19] in the context of document retrieval. However, pullback metrics are a well studied notion of differential geometry [16], which has found several applications in computer vision [17].

2.1 Pullback metrics

Formally, consider a family of automorphisms between the Riemannian manifold $\mathscr{M}$ in which the data-set $D = \{m_1, ..., m_N\} \subset \mathscr{M}$ resides and itself:

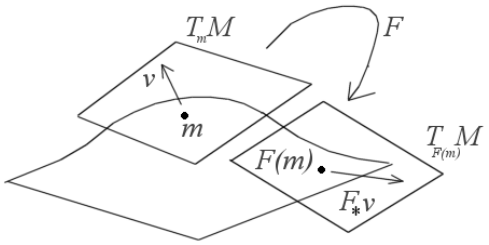

Fig. 1 The push-forward map F_* associated with an auto-morphism F on a Riemannian manifold $\mathscr{M}$

$$\begin{aligned} F_p : \mathscr{M} &\to \mathscr{M} \\ m \in \mathscr{M} &\mapsto F_p(m). \end{aligned}$$

Let us denote by $T_m\mathscr{M}$ the tangent space to $\mathscr{M}$ in m. Any such automorphism F is associated with a "push-forward" map

$$\begin{aligned} F_* : T_m\mathscr{M} &\to T_{F(m)}\mathscr{M} \\ v \in T_m\mathscr{M} &\mapsto F_* v \in T_{F(m)}\mathscr{M} \end{aligned}$$

defined as $F_* v(f) = v(f \circ F)$ for all smooth functions f on $\mathscr{M}$ (see Figure 1). Consider now a Riemannian metric

$$g : T\mathscr{M} \times T\mathscr{M} \to \mathbb{R}$$

on $\mathscr{M}$. Roughly speaking, g determines how to compute scalar products of tangent vectors $v \in T_m\mathscr{M}$. The map F induces a *pullback* metric g_* on $\mathscr{M}$:

$$g_{*m}(u,v) \doteq g_{F(m)}(F_* u, F_* v). \tag{1}$$

The scalar product of two vectors u, v of $T_m\mathscr{M}$ according to g_* is computed as the scalar product with respect to the *original* metric g of the images $F_* u, F_* v$ of the vectors u, v under the push-forward map F_*. The pullback geodesic between any two points m_1, m_2 of the manifold $\mathscr{M}$ is the geodesic connecting their images with respect to the original metric. If we manage to define an entire class of such automorphisms depending on some parameter λ, we get a corresponding family of pullback metrics on $\mathscr{M}$, also depending on λ. We can then define an optimization problem over such a family in order to select an "optimal" metric, which in turn determines the desired manifold. The nature of this manifold will depend on the objective function we choose to optimize.

2.2 Fisher metric for linear models

To apply the pullback metric framework to dynamical models, we first need to define a structure of Riemannian manifold on them. The study of the geometrical structure

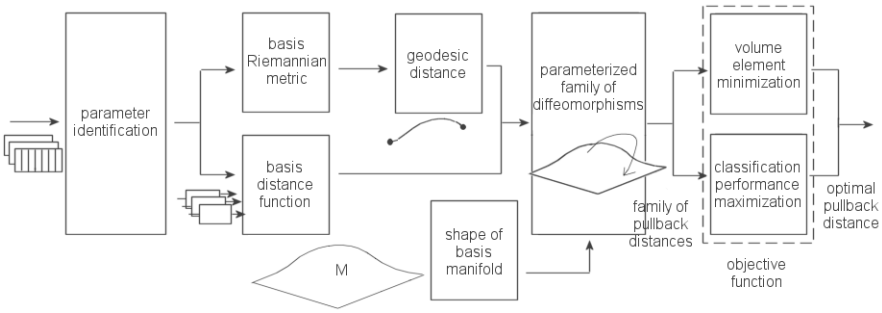

Fig. 2 A bird's eye view of the proposed framework for learning pullback metrics on dynamical models.

of the space formed by a family of probability distribution is due to Rao, and has been developed by Nagaoka and Amari [1]. A family S of probability distributions $p(x,\xi)$ depending on a n-dimensional parameter ξ can be regarded in fact as an n-dimensional manifold. If the Fisher information matrix

$$g_{ij} \doteq E\Big[\frac{\partial logp(x,\xi)}{\partial \xi_i}\frac{\partial logp(x,\xi)}{\partial \xi_j}\Big]$$

is nondegenerate, $G=[g_{ij}]$ is a Riemannian metric, and S is a Riemannian manifold. The Fisher information matrix for several manifolds of linear MIMO systems has been computed in [15].

2.3 General framework

As linear dynamical models do live in a Riemannian space, we can apply to them the pullback metric formalism and obtain a family of metrics on which to optimize. Itoh *et al* [16] have indeed recently done some work on pullbacks of Fisher information metrics.

Here we design a general framework for learning an optimal pullback metric from a training set of dynamical models, as depicted in Figure 2.

1. A data-set Y of observation sequences $\{y_k=[y_k(t),t=1,...,T_k],k=1,...,N\}$ of variable length T_k is available;
2. from each sequence, a dynamical model m_i of a certain class $\mathscr{C}$ is estimated by parameter identification, yielding a data-set of such models $D=\{m_1,...,m_N\}$;
3. models of a certain class $\mathscr{C}$ belong to a manifold $\mathscr{M}_{\mathscr{C}}$; its atlas of coordinate charts has to be known[1];

[1] In the case considered here, a single coordinate chart actually spans the whole manifold.

4. to measure distances between pairs of models on the manifold $\mathscr{M}_{\mathscr{C}}$, either a distance function $d_{\mathscr{M}}$ or a proper Riemannian metric $g_{\mathscr{M}}$ have to be defined on it;
5. in the case of a Riemannian metric, it is necessary to know the geodesic path between two models in order to compute the associated geodesic distance;
6. a family F_λ of automorphisms from $\mathscr{M}_{\mathscr{C}}$ onto itself, parameterized by a vector λ, is designed to provide a search space of metrics from which to select the optimal one;
7. F_λ induces a family of pullback metrics (1) $g_{*\lambda.\mathscr{M}}$ or distances $d_{*\lambda.\mathscr{M}}$ on $\mathscr{M}$, respectively;
8. we optimize over this family of pullback distances/metrics in order to find the optimal such function/metric, according to some objective function;
9. this yields an optimal pullback metric $\hat{g}_*$ or distance function $\hat{d}_*$;
10. in the metric case, knowing the geodesics of $\mathscr{M}$ suffices to compute the geodesic distances on $\mathscr{M}$ based on $\hat{g}_*$;
11. the optimal distance function can finally be used to cluster or classify new "test" models/sequences.

2.4 Objective functions: classification performance and inverse volume

When the data-set of models is *labeled*, we can exploit this information to determine the optimal metric/distance function. In particular, we can use cross-validation [8] to optimize its classification performance by dividing the overall sample into a number v of folds. The models belonging to $v-1$ folds are used as training sample, the remaining fold as testing sample, and the parameter of the pullback metric which optimizes the correct classification rate on fold v given the training set is selected. As the classification score is hard to describe analytically, we can extract a number of random samples from the parameter space and pick the maximal performance sample.

When the training set is *unlabeled*, manifold learning has to be based on purely geometrical considerations. G. Lebanon [19] has recently suggested in the context of document retrieval an approach that seeks to *maximize the inverse volume element* associated with a metric around the given training set of points [22]:

$$\mathscr{O}(D) = \prod_{k=1}^{N} \frac{(\det g(m_k))^{-\frac{1}{2}}}{\int_{\mathscr{M}} (\det g(m))^{-\frac{1}{2}} dm} \tag{2}$$

where $g(m_k)$ denotes the Riemannian metric in the point m_k of the data-set D living on a Riemannian manifold $\mathscr{M}$. This amounts to finding a lower dimensional representation of the dataset, in a similar fashion to LLE [27] or laplacian eigenmaps [3], where dimensionality reduction is often considered a factor in improving classification.

The computation of (2) requires that of the *Gramian* $\det g$. To find the expression of the Gramian associated with a pullback metric (1) we first need to choose a base of the space $T_m\mathscr{M}$ tangent to $\mathscr{M}$ in m. Let us denote by $\{\partial_i, i = 1, ..., \dim \mathscr{M}\}$ the base of $T_m\mathscr{M}$. The push-forward of the vectors of this base yields a base for $T_{F(m)}\mathscr{M}$. By definition, the push-forward $F_{*\lambda}$ of a vector $v \in T_m\mathscr{M}$ under a an automorphism F_λ with parameter λ is given by [19]:

$$F_{*\lambda}(v) \doteq \frac{d}{dt} F_\lambda(m + t \cdot v)\big|_{t=0}, \quad v \in T_m\mathscr{M}.$$

The automorphism F_λ induces a base for the space of vector fields on $\mathscr{M}$, $w_i \doteq \{F_{*\lambda}(\partial_i)\}$, for $i = 1, ..., \dim \mathscr{M}$. We can rearrange these vectors as rows of a matrix:

$$J = [w_1; \cdots; w_{\dim \mathscr{M}}].$$

The volume element of the pullback metric g_* in a point $m \in \mathscr{M}$ is the determinant of the Gramian [19]: $\det g_*(m) \doteq \det[g(F_{*\lambda}(\partial_i), F_{*\lambda}(\partial_j))]_{ij} = \det(J^T g J)$. If J is a square matrix (as in the rest of this paper) we get simply:

$$\det g_*(m) = \det(J)^2 \cdot \det g(m). \tag{3}$$

After plugging (3) into (2) we obtain the function to minimize.

3 Pullback metrics for multidimensional autoregressive models

In virtue of their importance as a class of dynamical models, and their relative simplicity, we consider here the class of stable autoregressive discrete-time processes of order 2, $\mathscr{M} = \mathscr{AR}(2)$, in a stochastic setting in which the input signal is a Gaussian white noise with zero mean and unit variance.

3.1 The basis manifold

This set can be given a Riemannian manifold structure under Fisher metric. A natural parametrization uses as coordinates the non-unit coefficients (a_1, a_2) of the denominator of the transfer function:

$$h(z) = \frac{z^2}{z^2 + a_1 z + a_2} \tag{4}$$

which corresponds to the AR difference: $y(k) = -a_1 y(k-1) - a_2 y(k-2)$.

The basis manifold $\mathscr{M}$ and the associated Fisher metric *in the scalar case* have been studied in the context of control theory [23, 25]. We build here on these results to determine a coordinate chart and a product Fisher metric on the manifold

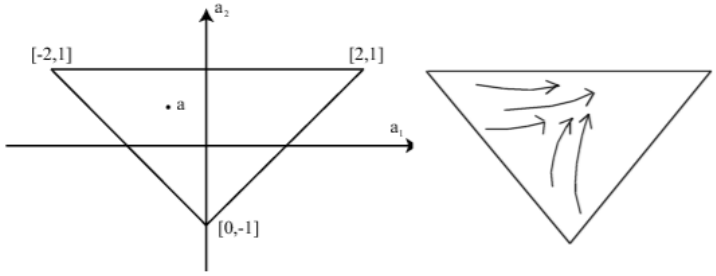

Fig. 3 The manifold of stable scalar autoregressive systems of order 2, $\mathscr{AR}(2,1)$, with the non-unit coefficients of the denominator of $h(z)$ as parameters. It forms a simplex with vertices $[-2,1],[2,1],[0,-1]$. Right: Effect of an automorphism of the form (10) on the $\mathscr{AR}(2,1)$ simplex.

$\mathscr{AR}(2,p)$ of *p-dimensional* AR models. We will then be able to design two different families of automorphisms on $\mathscr{AR}(2,p)$, and use the framework of Section 2 to determine there two families of pullback distance functions.

3.1.1 The basis manifold $\mathscr{AR}(2,1)$ in the scalar case

Let us consider first the scalar case $p = 1$ of a single output channel. To impose the stability of the transfer function (4) the necessary conditions are $1+a_1+a_2 > 0$, $1-a_1+a_2 > 0$, and $1-a_2 > 0$. The manifold of stable AR(2,1) systems is then composed by a single connected component (see Figure 3-left).
The Fisher tensor is [23]:

$$g(a_1,a_2) = \frac{1}{(1+a_1+a_2)(1-a_1+a_2)(1-a_2)} \begin{pmatrix} 1+a_2 & a_1 \\ a_1 & 1+a_2 \end{pmatrix} \tag{5}$$

with volume element

$$\det g_{\mathscr{AR}(2,1)}(a_1,a_2) = \frac{1}{(1-a_2)^2(1-a_1+a_2)(1+a_1+a_2)}. \tag{6}$$

3.1.2 The multidimensional case

In the multidimensional case, an autoregressive model is composed of p separate channels, each characterized by a transfer function

$$h^i(z) = \frac{z^2}{z^2 + a_1^i z + a_2^i}$$

(assuming their independence). When using two coefficients a_1^i, a_2^i to describe each channel, each p-dimensional AR system has coordinates $\mathbf{a} = [a_1^i, a_2^i : i = 1, ..., p]'$. $\mathscr{AR}(2,p)$ is therefore the *product manifold*

$$\mathscr{AR}(2,p) = \times_{i=1}^{p} \mathscr{AR}(2,1), \tag{7}$$

i.e., the Cartesian product of the manifolds associated with the individual channels. As a Cartesian product of a number of simplices (triangles), $\mathscr{AR}(2,p)$ turns out to be a *polytope* in $\mathbb{R}^{2p}$. Such polytope has in general $n_1 \times \cdots \times n_p$ vertices, the product of the number of vertices of the individual simplices. In our case, $\mathscr{AR}(2,p)$ is a polytope with 3^p vertices:

$$\mathscr{AR}(2,p) = Cl(\mathbf{v}_{i_1,...,i_p}, i_j = 1, .., 3 \; \forall j = 1, ..., p).$$

Each p-dimensional AR system $\mathbf{a}$ also possesses, therefore, a vector of simplicial coordinates *in the polytope* $\mathscr{AR}(2,p)$:

$$\mathbf{m} = [m_{i_1,...,i_p}, i_j = 1, .., 3 \; \forall j = 1, ..., p]' \tag{8}$$

such that

$$\mathbf{a} = \sum_{i_j=1, j=1,...,p}^{3} m_{i_1,...,i_p} \mathbf{v}_{i_1,...,i_p}.$$

3.1.3 Product metric

On the Cartesian product $\mathscr{M}_1 \times \mathscr{M}_2$ of two Riemannian manifolds with metrics $g_p^{\mathscr{M}_1}$ and $g_q^{\mathscr{M}_2}$, respectively, one can define the *product metric* on $\mathscr{M}_1 \times \mathscr{M}_2$ as

$$g_{(p,q)}^{\mathscr{M}_1 \times \mathscr{M}_2} : T_{(p,q)}(\mathscr{M}_1 \times \mathscr{M}_2) \times T_{(p,q)}(\mathscr{M}_1 \times \mathscr{M}_2),$$

with

$$(u,v) \mapsto g_p^{\mathscr{M}_1}(T_{(p,q)}\pi_1(u), T_{(p,q)}\pi_1(v)) + g_q^{\mathscr{M}_2}(T_{(p,q)}\pi_2(u), T_{(p,q)}\pi_2(v))$$

where $\pi_i : \mathscr{M}_1 \times \mathscr{M}_2 \rightarrow \mathscr{M}_i$ is the natural projection of a point of the Cartesian product onto one of the component manifolds. The definition can be extended to any finite number of manifolds. The product metric $g_{\mathscr{AR}(2,p)}$ is described by a $2p \times 2p$ block diagonal matrix, whose p 2×2 blocks are copies of the metric (5) valid in the scalar case:

$$g_{\mathscr{AR}(2,p)}(\mathbf{a}) = diag\big(g_{\mathscr{AR}(2,1)}(a_1^i, a_2^i)\big).$$

Its volume element $\det g_{\mathscr{AR}(2,p)}$ is (given the expression (6) of the scalar volume element $\det g_{\mathscr{AR}(2,1)}$):

$$\begin{aligned}\det g_{\mathscr{AR}(2,p)}(\mathbf{a}) &= \prod_{i=1}^{p} \det g_{\mathscr{AR}(2,1)}(a_1^i, a_2^i) \\ &= \prod_{i=1}^{p} \frac{1}{(1-a_2^i)^2(1-a_1^i+a_2^i)(1+a_1^i+a_2^i)}.\end{aligned} \tag{9}$$

3.1.4 Geodesics

To compute the distance between two points of a Riemannian manifold (and therefore, in particular, between two dynamical models) the metric is not sufficient. It is necessary to compute (analytically or numerically) the shortest path connecting any such pair of points on the manifold (geodesic). All the geodesics of stable $AR(2,1)$ systems endowed with the Fisher metric (5) as a function of the Schur parameters $\gamma_1 = a_1/(1+a_2)$, $\gamma_2 = a_2$ have been analytically computed by Rijkeboer [25]:

$$4 \cdot (\ddot{\gamma}_1 + \ddot{\gamma}_2) + \frac{1}{1+(\gamma_2)^2} \cdot \dot{\gamma}_1 \dot{\gamma}_2 + \frac{\gamma_2}{1-(\gamma_2)^2} \cdot (\dot{\gamma}_2)^2 - \frac{1}{1+\gamma_1} \cdot (\dot{\gamma}_1)^2 = 0.$$

In the general case $AR(2,p)$, unfortunately, the manifold's geodesics are not analytically known. However [24]:

Proposition 1. *The sub-manifolds of a product manifold are geodesic, i.e., all geodesic paths on the individual sub-manifolds are geodesics of the product manifold too.*

In our case, as $\mathscr{AR}(2,p)$ is itself a product manifold (7), the (known) geodesics of the "scalar" manifold $\mathscr{AR}(2,1)$ are also geodesics of $\mathscr{AR}(2,p)$. As an approximation, therefore, we can measure the geodesic distance between two generic p-dimensional autoregressive models by applying a generalization of Phytagoras' theorem $d(\mathbf{a},\mathbf{a}') = \sqrt{\sum_p d_i(\mathbf{a},\mathbf{a}')^2}$, where $d_i(\mathbf{a},\mathbf{a}')$ is the distance of their projections on the i-th sub-manifold.

3.2 An automorphism for the scalar case

To build a parameterized family of Riemannian metrics for $\mathscr{AR}(2,p)$ it is necessary to choose a family of automorphisms of the manifold onto itself (Section 2). The more sophisticated is the set of automorphisms, the larger is the search space to optimize the metric on.

One possible choice for an automorphism of $\mathscr{AR}(2,1)$ is suggested by the triangular form of the manifold, which has three vertices (see Figure 3-left). Let $\mathbf{m} = [m_1, m_2, m_3]'$ collect the "simplicial" coordinates of a system $\mathbf{a} \in \mathscr{AR}(2,1)$ in the manifold:

$$\mathbf{a} = [a_1, a_2]' = m_1[0,-1]' + m_2[2,1]' + m_3[-2,1]', \qquad m_1 + m_2 + m_3 = 1.$$

A natural automorphism of a simplex onto itself is given by "stretching" the simplicial coordinates of its point by a set of weights $\lambda = [\lambda_1, \lambda_2, \lambda_3]'$ such that $\sum_j \lambda_j = 1$, $\lambda_j \geq 0$:

$$F_\lambda(\mathbf{m}) = F_\lambda([m_1, m_2, m_3]') = \frac{[\lambda_1 m_1, \lambda_2 m_2, \lambda_3 m_3]'}{\lambda \cdot \mathbf{m}}, \tag{10}$$

where $\lambda \cdot \mathbf{m}$ denotes the scalar product of the two vectors λ, $\mathbf{m}$. The application (10) stretches the triangle towards the vertex with the largest λ_i (Figure 3-right).

3.3 Product and global automorphisms for $\mathscr{AR}(2,p)$

A *product* automorphism for the whole manifold $\mathscr{AR}(2,p)$ of multidimensional, p channel autoregressive models can be obtained by using (10), designed for scalar systems, as a building block. If we denote by $\mathbf{m}^i = [m_1^i, m_2^i, m_3^i]'$ the simplicial coordinates of a system **a** in the i-th sub-manifold, such a system will be identified by a vector $\mathbf{m} = [\mathbf{m}^i, i = 1, ..., p]'$ of $3p$ such coordinates.
The mapping

$$F_{\lambda^i, i=1,...,p}(\mathbf{m}) = [F_{\lambda^i}(\mathbf{m}^i), i = 1, ..., p]' \tag{11}$$

with $3p$ parameters applies an automorphism (10) with parameter λ^i to the projection of $\mathbf{m}$ onto each sub-manifold.

In alternative, the global geometry of the product manifold $\mathscr{AR}(2,p)$ inspires a *global* automorphism which acts on the manifold as a whole, by multiplying its "polytopial" coordinates (8) by a set of convex weights

$$\mu = [\mu_{i_1,...,i_p}, i_j = 1, .., 3 \; \forall j = 1, ..., p]'.$$

We obtain, up to normalization:

$$F_\mu(\mathbf{m}) \propto [\mu_{i_1,...,i_p} \cdot m_{i_1,...,i_p}, i_j = 1, .., 3 \; \forall j = 1, ..., p]'. \tag{12}$$

3.3.1 Volume element for $\mathscr{AR}(2,p)$ under product automorphism

Assume that the data-set of models is unlabeled. To select an optimal pullback metric for p-dimensional autoregressive models by volume minimization (2), we need to find the analytical expression of the determinant of the Gramian $det g_{*\lambda}$ (3) as a function of the parameter vector λ. By plugging it into the expression for the inverse volume (2) we obtain the objective function to optimize. We consider here the relatively simpler product diffeomorphism (11).
Notice that, in the product simplicial coordinates $\mathbf{m} = [\mathbf{m}^i, i = 1, ..., p]'$ of a system of $\mathscr{AR}(2,p)$, the volume element of the product Fisher metric (9) reads as:

$$\det g_{\mathscr{AR}(2,p)}(\mathbf{m}) = \prod_{i=1}^{p} \frac{1}{(m_1^i)^2 m_2^i m_3^i}.$$

Theorem 1. *The volume element of the Fisher pullback metric on $\mathscr{AR}(2,p)$ induced by the automorphism (11) is:*

$$detg_{*\lambda}(\lambda,\mathbf{m}) = \frac{1}{2^{2p}}\prod_{i=1}^{p}\frac{(\lambda_1^i\lambda_2^i\lambda_3^i)^2}{(\lambda^i\cdot\mathbf{m}^i)^6}\cdot\frac{1}{(m_1^i)^2m_2^im_3^i}. \tag{13}$$

Proof. We need to compute the Gramian $detg_{*\lambda}$ (3) of the pullback metric under the automorphism (11). Following the procedure of Section 2, we need to choose a basis of the tangent space

$$T\mathscr{AR}(2,p) = T\mathscr{AR}(2,1)\oplus\cdots\oplus T\mathscr{AR}(2,1)$$

of the product manifold (7). The size $2p$ vectors

$$\partial_1^i = [0,\cdots,0,1/2,1/2,0,\cdots,0]',\quad \partial_2^i = [0,\cdots,0,-1/2,1/2,0,\cdots,0]'$$

whose only non-zero entries are in positions $2i-1$, $2i$, form such a basis. Let us express the product automorphism (11) in coordinates $\mathbf{a} = [a_1^1,a_2^1,\cdots,a_1^p,a_2^p]$. We have that, $\forall i = 1,...,p$:

$$\begin{aligned} &a_1^i = 2(m_2^i - m_3^i),\ \ a_2^i = m_2^i + m_3^i - m_1^i,\\ &m_2^i = \frac{1+a_1^i+a_2^i}{4},\ m_3^i = \frac{1-a_1^i+a_2^i}{4},\quad m_1^i = \frac{1-a_2^i}{2}. \end{aligned}$$

It follows that:

$$\begin{aligned} F_{\lambda^1,\ldots,\lambda^p}(\mathbf{a})[2i-1,2i] &= \tfrac{1}{\Delta_i}\big[\ 2\lambda_2^i(1+a_1^i+a_2^i) - 2\lambda_3(1-a_1^i+a_2^i),\\ &\qquad \lambda_2(1+a_1^i+a_2^i) + \lambda_3(1-a_1^i+a_2^i) - 2\lambda_1(1-a_2^i)\big]',\\ F_{\lambda^1,\ldots,\lambda^p}(\mathbf{a})[h,l] &= 0 \quad (h,l)\neq(2i-1,2i), \end{aligned} \tag{14}$$

where $\Delta_i = 2\lambda_1^i(1-a_2^i) + \lambda_2^i(1+a_1^i+a_2^i) + \lambda_3^i(1-a_1^i+a_2^i)$.
We seek for all channels $i = 1,...,p$ the push-forward tangent vectors

$$\mathbf{w}_1^i = \frac{d}{dt}F_{\{\lambda^j,j\}}(\mathbf{a}+t\partial_\mathbf{1}^\mathbf{i})\big|_{t=0},\quad \mathbf{w}_2^i = \frac{d}{dt}F_{\{\lambda^j,j\}}(\mathbf{a}+t\partial_\mathbf{2}^\mathbf{i})\big|_{t=0}.$$

We get:

$$\begin{aligned} \mathbf{w}_1^i[2i-1,2i] &= \big[\quad -2\lambda_1^i\lambda_3^i(3-a_2^i+a_1^i) + 2\lambda_2^i(\lambda_1^i-2\lambda_3^i)(1+a_1^i+a_2^i)\ ,\\ &\qquad 2\lambda_1^i\lambda_3^i(3-a_2^i+a_1^i) + 2\lambda_1^i\lambda_2^i(1+a_1^i+a_2^i)\quad\big],\\ \mathbf{w}_1^i[h,l] &= 0 \qquad (h,l)\neq(2i-1,2i) \end{aligned} \tag{15}$$

and a similar expression for $\mathbf{w}_2^i$.
The matrix J formed by the stacked collection of the row vectors $\mathbf{w}_{1,2}^i$,

$$J = [\mathbf{w}_1^1; \mathbf{w}_2^1; ...; \mathbf{w}_1^p; \mathbf{w}_2^p]$$

is clearly block diagonal. Its determinant is therefore the product of the determinants of the blocks:

$$\det J = \frac{1}{2^p} \prod_{i=1}^{p} \frac{\lambda_1^i \lambda_2^i \lambda_3^i}{(\lambda^i \cdot \mathbf{m}^i)^3}.$$

By plugging the expressions for $\det J$ and $\det g_{\mathscr{A}\mathscr{R}(2,p)}$ into that (3) of the pullback volume element, we get (13). □

The function to maximize is finally obtained by plugging (13) in the general expression (2). The normalization factor $I(\lambda) = \int_{\mathscr{M}} (\det g_{*\lambda}(m))^{-\frac{1}{2}} dm$ can be approximated as:

$$I(\lambda) \simeq \sum_{k=1}^{N} det g_{*\lambda}(\lambda, \mathbf{m}_k)^{-\frac{1}{2}}.$$

In the labeled case, instead, we find the optimal parameters λ by optimizing the classification performance on the available data-set by cross-validation.

4 Tests on identity recognition

To test the actual, empirical effect of our approach to manifold learning on the classification of dynamical models, we considered the problem of recognizing actions and identities from image sequences. We used the Mobo database [14], a collection of 600 image sequences of 25 people walking on a treadmill in four different variants (slow walk, fast walk, walk on a slope, walk carrying a ball), seen from 6 different viewpoints. We selected all the sequences associated with the gaits "slow walk" and "walking on inclined slope", simulating this way the impact of nuisance factors actually present in gait identification, and making recognition more challenging.

4.1 Identification of a AR(2,p) model for each sequence

As the Mobo database comes with pre-processed silhouettes attached to each image, we decided to use silhouette based-features to represent images. However, this is by no means a limitation of the proposed approach. Indeed, more sophisticated 3D pose estimation methods could be used to run tests on the 3D setup [26]. We plan to run such tests in the near future. Given a silhouette we detect its center of mass, re-scale it to the associated bounding box, and project its contours on to a sheaf of 18 equally spaced lines passing through its center of mass. This proved superior to alternative representations [20].

According to the scheme of Figure 2 each input sequence has to be represented by a dynamical model, in particular, an autoregressive model of order 2.

Each component of the feature/observation vector, then, is associated with a different output channel of the AR(2,p) model (4). We used the Matlab routine $M = ARX(DATA, ORDERS)$ to identify by least-squares optimization the parameters a_1^i, a_2^i for each output channel $i = 1, ..., p$. For comparison, for each input sequence we also identified a hidden Markov model [13] by applying the Expectation Maximization [9] algorithm, yielding a model of the form $x(t+1) = Ax(t) + v(t)$, $y(t) = Cx(t) + w(t)$ described by a transition matrix $A[i, j] = P(x(t) = i | x(t-1) = j)$ and a state-output matrix C.

4.2 Performances of optimal pullback metrics

To classify the test models, we adopted standard nearest neighbor classification: each testing sequence is attributed the label of the closest model in the training set. Note that *by no means* this is a limitation of the proposed approach: *any* advanced classification technique (Guassian kernels, SVMs, etc) can be used in cascade to our metric learning procedure. The classification performance was measured as the percentage of correctly classified sequences. For each run we randomly selected a training and a testing set in the database.

Figure 4 plots the average classification performance (over 10 runs) of the following metrics on the space of autoregressive models of order 2 with p outputs: 1 - product Fisher metric $g_{\mathscr{AR}(2,p)}$; 2 - pullback Fisher metric induced by the product automorphism (11) optimizing the classification performance after cross-validation on the training set; 3 - pullback Fisher metric induced by the *global* automorphism (12) for the same objective function; 4 - pullback Fisher metric induced by (11) with optimal inverse volume; 5 - Frobenius[2] distance between HMMs:

$$\|H_1 - H_2\| = \|A_1 - A_2\|_F + \|C_1 - C_2\|_F. \tag{16}$$

Fifteen identities are here considered, with the parameter space sampled 200 times to detect the optimal parameters. The optimal classification pullback Fisher metrics induced by the proposed automorphisms are clearly superior to the standard Fisher distance over all experiments. The improvement margin ranges from 5% up to even 20%.

Figure 6 plots instead the classification score of the different competing metrics versus the number of identities considered in the experiments. We ran different tests involving a number of subjects ranging from 10 up to 22. Again, the performance of both pullback Fisher metrics obtained by maximizing classification score by n-fold validation (solid red and dashed red lines) is widely superior to that of the original Fisher distance (in solid black), or the naive Frobenius distance between HMMs (dashed black). The approach displays an interesting robustness to the expected

[2] The Frobenius distance of two matrices X, X' is $\|X - X'\| = \sqrt{Tr((X - X')(X - X')^T)}$ where $Tr(X) = \sum_{ii} X[i, i]$ is the trace.

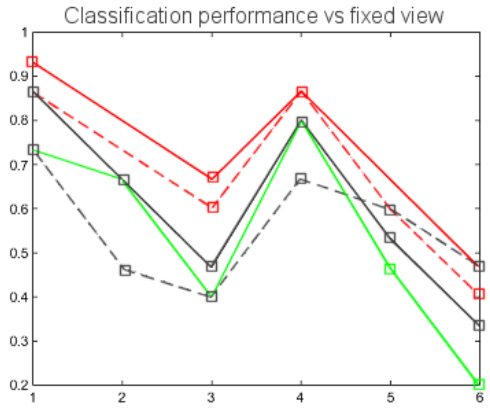

Fig. 4 In six separate experiments, the classification performance of the following metrics has been computed for image sequences coming from a single view, from 1 to 6. Fifteen identities, 200 samples extracted from the parameter space. Line styles: basis Fisher geodesic distance, solid black; Frobenius distance between HMMs (16), dashed black; optimal pullback Fisher under automorphism (11), solid red; optimal pullback Fisher under automorphism (12), dashed red; inverse volume optimal Fisher metric with automorphism (11), solid green.

decreasing performance as the problem grows more difficult, as optimal pullback classification rates are remarkably stable compared to those of classical metrics.

4.3 Influence of parameters

It is natural to conjecture that, when optimizing the classification performance in the cross-validation procedure described in Section 2, a larger training set should lead to identify more effective automorphism parameters. Indeed, Figure 5-left shows the behavior of the considered metrics as a function of the size of the training set on which the optimal parameters are learnt. We can notice two facts here. First, as expected, optimization over larger training sets delivers better metrics, i.e, better classification scores. Second, with its higher-dimensional parameter space, the global automorphism (12) of the the $\mathscr{AR}(2,p)$ polytope generates more performing metrics, with a margin over the simpler, product automorphism ranging from 10% up to a remarkable 25%.

Finally, Figure 5-right illustrates how sampling more densely the parameter space when looking for the pullback metric that optimizes the n-fold classification score improves the performance of the resulting classifier. As an example, here the optimal pullback Fisher metric under product automorphism (11) is analyzed and compared with the baseline results obtained by using the basis Fisher geodesic distance between $AR(2,p)$ models. As expected, the margin of improvement increases quite steadily as more samples are assessed in the parameter space. Here all 25 identities

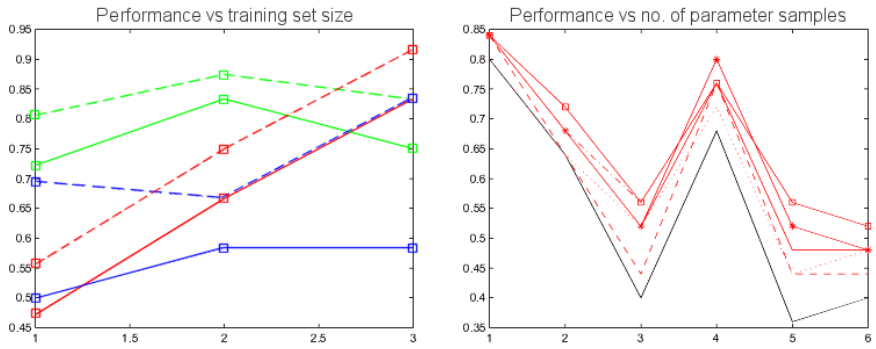

Fig. 5 Left: Classification performance of the two optimal pullback Fisher metrics plotted versus increasing size of the training set over which the parameters of the optimal automorphism are learnt. Here 12 identities, 200 parameter samples are chosen. Abscissa: 1 - 28 sequences in the training set 2 - 40 sequences 3 - 52 sequences. Red: experiment on view 1; green: view 4; blue: view 5. Solid: optimal metric induced by product automorphism (11); dashed: global automorphism (12). Right: Classification performance of the optimal pullback Fisher metric induced by product automorphism, for experiments run on all viewpoints. All 25 identities are considered. Different classification scores are plotted for a number of samples varying from 10 (red dotted line) to 200 (solid red with squares). The score of the basis Fisher metric is plotted in black for comparison. When the optimal parameter is the result of a more extensive search, the classification performance is generally better.

are considered, the margin ranging from a minimum of 5% for view 1 (for which the best silhouettes are available) to 15% for view 3, to a very substantial 23% for view 5, in which case the basis metric has the worst performance.

5 Perspectives and conclusions

In this paper we proposed a differential-geometric framework for manifold learning given a data-set of linear dynamical models, based on optimizing over a family of pullback metrics induced by automorphisms. We adopted as basis metric tensor the classical Fisher information matrix, and showed tests on identity recognition that attest the improvement in classification performance one can gain from such a learnt metric. The method is fully general, and easily extendible to other classes of dynamical models or more sophisticated classifiers. For several classes of multi-dimensional linear systems both the Fisher metric and its geodesics can still be computed by means of an iterative numerical scheme [23, 15]. The extension to another popular class of stochastic model, hidden Markov models [13] requires an interesting analysis of its manifold structure and is already under way. Last but not least, the incorporation into the framework of objective functions that take into account

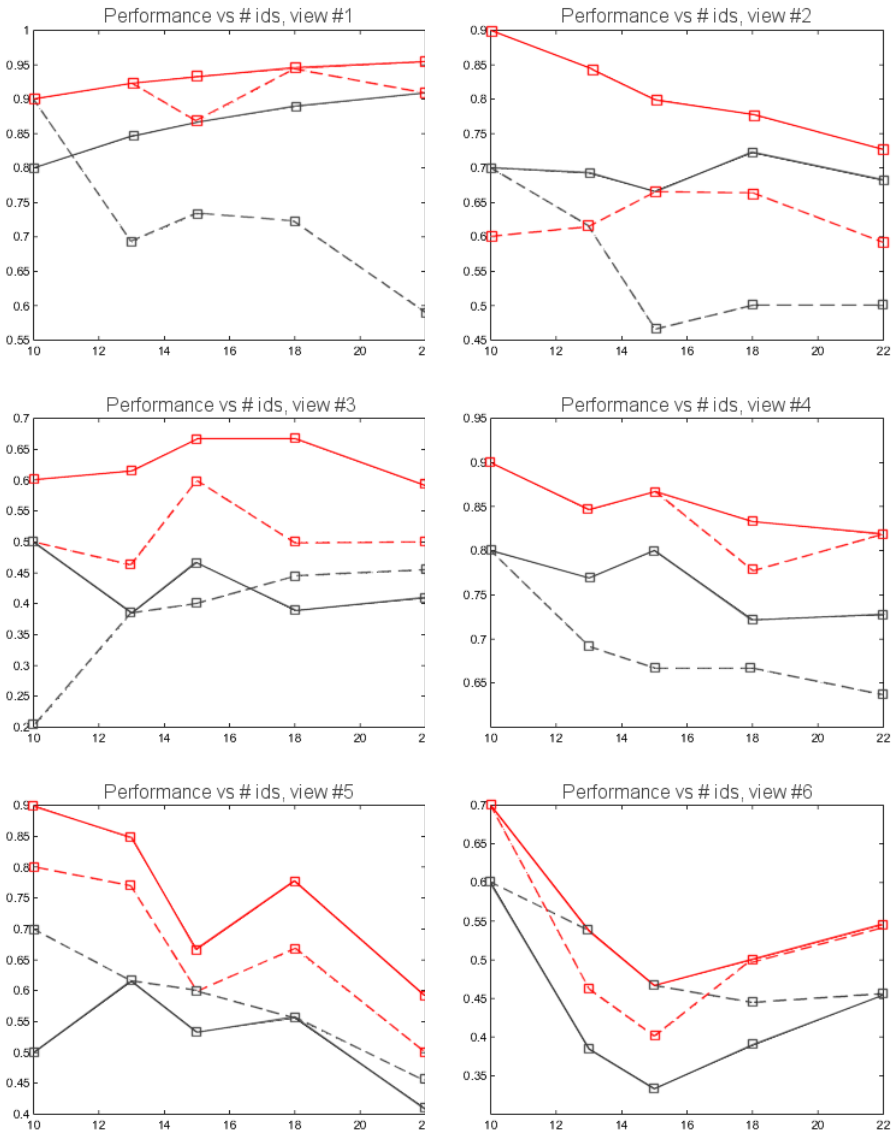

Fig. 6 Metrics' classification scores plotted versus the number of identities (from 10 to 22) in the testing set, for all viewpoints from 1 to 6. Solid black: Frobenius HMM distance. Dashed black: classical Fisher geodesic distance between $AR(2,p)$ models. Solid red: optimal pullback Fisher geodesic distance induced by (11). Dashed red: optimal pullback Fisher induced by (12).

a-priori knowledge on the training set, such as similarity relations [34], is highly desirable, and will be pursuit in the near future.

References

1. S.-I. Amari, *Differential geometric methods in statistics*, Springer-Verlag, 1985.
2. A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, *Learning distance functions using equivalence relations*, ICML03, 2003, pp. 11–18.
3. M. Belkin and P. Niyogi, *Semi-supervised learning on riemannian manifolds*, MLJ **56** (2004), 209–239.
4. Y. Bengio, J.-F. Paiement, and P. Vincent, *Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering*, Tech. report, 2003.
5. M. Bilenko, S. Basu, and R.J. Mooney, *Integrating constraints and metric learning in semi-supervised clustering*, Proc. of ICML'04, 2004.
6. A. Bissacco, A. Chiuso, and S. Soatto, *Classification and recognition of dynamical models: The role of phase, independent components, kernels and optimal transport*, IEEE Trans. PAMI **29** (2007), no. 11, 1958–1972.
7. M. Brand, N. Oliver, and A. Pentland, *Coupled hidden markov models for complex action recognition*, 1997, pp. 994–999.
8. P. Burman, *A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods*, Biometrika **76(3)** (1989), 503–514.
9. A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society B **39**.
10. M.N. Do, *Fast approximation of Kullback-Leibler distance for dependence trees and hidden Markov models*, IEEE Signal Processing Letters **10** (2003), no. 4, 115 – 118.
11. G. Doretto, A. Chiuso, Y.N. Wu, and S. Soatto, *Dynamic textures*, Journal International Journal of Computer Vision **51** (2003), no. 2, 91–109.
12. C.F. Eick, A. Rouhana, A. Bagherjeiran, and R. Vilalta, *Using clustering to learn distance functions for supervised similarity assessment*, ICML and Data Mining, 2005.
13. R. Elliot, L. Aggoun, and J. Moore, *Hidden markov models: estimation and control*, Springer Verlag, 1995.
14. R. Gross and J. Shi, *The CMU motion of body (Mobo) database*, Tech. report, CMU, 2001.
15. B. Hanzon and R.L.M. Peeters, *Aspects of Fisher geometry for stochastic linear systems*, Open Problems in Mathematical Systems and Control Theory, 2002, pp. 27–30.
16. M. Itoh and Y. Shishido, *Fisher information metric and poisson kernels*, Differential Geometry and its Applications **26** (2008), no. 4, 347 – 356.
17. M. Kerckhove, *Computation of ridges via pullback metrics from scale space*, Book Scale-Space Theories in Computer Vision, LNCS, vol. 1682/1999, 1999, pp. 82–92.
18. S. Kullback and R. A. Leibler, *On information and sufficiency*, Annals of Math. Stat. **22** (1951), 79–86.
19. G. Lebanon, *Metric learning for text documents*, IEEE Tr. PAMI **28** (2006), no. 4, 497–508.
20. L. Lee and W. Grimson, *Gait analysis for recognition and classification*, AFGR'02, 2002, pp. 155–162.
21. R. J. Martin, *A metric for ARMA processes*, IEEE Trans. on Signal Processing **48(4)** (April 2000), 1164–1170.
22. M.K. Murray and J.W. Rice, *Differential geometry and statistics*, CRC Press, 1993.
23. R.L.M. Peeters and B. Hanzon, *On the Riemannian manifold structure of classes of linear systems*, Equadiff, 2003.
24. G. Pitis, *On some submanifolds of a locally product manifold*, Kodai Math. J. **9** (1986), no. 3, 327–333.
25. A.L. Rijkeboer, *Differential geometric models for time-varying coefficients of autoregressive processes*, PhD thesis, Tilburg University, 1994.
26. G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P.H.S. Torr, *Randomized trees for human pose detection*, CVPR'08.
27. S. Roweis and L. Saul, *Nonlinear dimensionality reduction by locally linear embedding*, Science **290(5500)** (2000), 2323–2326.

28. M. Schultz and T. Joachims, *Learning a distance metric from relative comparisons*, NIPS, 2004.
29. N. Shental, T. Hertz, D. Weinshall, and M. Pavel, *Adjustment learning and relevant component analysis*, ECCV'02, 2002.
30. A.J. Smola and S.V.N. Vishwanathan, *Hilbert space embeddings in dynamical systems*, Proc. of IFAC'03, August 2003, pp. 760 – 767.
31. Aravind Sundaresan, Amit Roy Chowdhury, and Rama Chellappa, *A hidden markov model based framework for recognition of humans from gait sequences*, PROC. ICIP, 2003, pp. 93–96.
32. I.W. Tsang, J.T. Kwok, C.W. Bay, and H. Kong, *Distance metric learning with kernels*, Proceedings of the International Conference on Artificial Intelligence, 2003.
33. L. Xie, A. Ugrinovskii, and I.R. Petersen, *Probabilistic distances between finite-state finite-alphabet hidden Markov models*, Proc. of CDC'03, 2003, pp. 5347–5352.
34. E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russel, *Distance metric learning with applications to clustering with side information*, Advances in Neural Information Processing Systems, 15, The MIT Press, 2003.
35. G. Zames and A. K. El-Sakkary, *Unstable systems and feedback: The gap metric*, Proc. 18th Allerton Conference on Communications, Control, and Computers, Urbana, IL, October 1980, pp. 380–385.
36. Z. Zhang, *Learning metrics via discriminant kernels and multidimensional scaling: Toward expected euclidean representation*, ICML'03, Hong Kong, 2003.