

Learning pullback manifolds of dynamical models

Fabio Cuzzolin

Abstract

In this paper we present a general differential-geometric framework for learning Riemannian metrics or distance functions for dynamical models, given a training set which can be either labeled or unlabeled. Given a training set of models, the optimal metric is selected among a family of pullback metrics induced by a parameterized automorphism of the space of models. The problem of classifying motions, encoded as dynamical models of a certain class, can then be posed on the learnt manifold. As significant case studies, in virtue of their applicability to gait identification and action recognition, we consider the class of multidimensional autoregressive models of order 2 and that of hidden Markov models. We study their manifolds and design automorphisms there which allow to build parametric families of metrics we can optimize upon. Experimental results concerning action and identity recognition are presented, which show how such optimal pullback Fisher metrics greatly improve classification performances.

F. Cuzzolin is Lecturer and Early Career Fellow with the Department of Computing, Oxford Brookes University, Oxford, United Kingdom. URL: <http://cms.brookes.ac.uk/staff/FabioCuzzolin/>. E-mail: fabio.cuzzolin@brookes.ac.uk Phone: +44 (0)1865 484526.

Keywords: *Dynamical models, distance learning, pullback metrics, autoregressive models, hidden Markov models, identity recognition.*

I. INTRODUCTION

Recognizing human activities from video is a natural application of computer vision. Though the formulation of the problem is simple and intuitive, activity recognition is a much harder problem than it may look. Motions inherently possess an extremely high degree of variability. Movements quite different from each can in fact carry the same meaning, or represent the same gesture. Motions are subject to a large number of nuisance factors [29], such as illumination, background, viewpoint [31]. Locality [50] is a critical factor too as, for instance, a person can walk and wave at the same time. When several persons move in the field of view, occlusion problems arise. Interestingly, the presence of one or more (static) objects in the vicinity of the motion can effectively help to disambiguate the activity class (recognition “in context”, [20], [32], [47]). The actions of interest have to be temporally segmented from a video sequence: we need to know when an action/activity starts or stops. Actions of sometimes very different lengths have to be encoded in a homogeneous fashion in order to be compared (“time warping”).

Recently, methods which neglect to take into account action dynamics for recognition have proven very effective. Typically, these approaches extract spatio-temporal features from the 3D volume associated with a video [24], [54]. The recognition of actions from static images [22] is also enjoying renewed interest.

Encoding the dynamics of videos or image sequences by means of some sort of dynamical model, however, can be useful in situations in which the dynamics is critically discriminative. Furthermore, the actions of interest have to be temporally segmented from a video sequence: we need to know when an action/activity starts or stops. Actions of sometimes very different lengths have to be encoded in

a homogeneous fashion in order to be compared (“time warping”). Dynamical representations are very effective in coping with time warping or action segmentation [45]. Dynamic textures [15] based on deterministic linear dynamical systems (LDS), for instance, have proven to be effective in video coding. Furthermore, in limit situations in which a significant number of people move or ambulate in the field of view (as it is common in surveillance scenarios), the attention has necessarily to move from single objects/bodies to approaches which consider the monitored crowd some sort of fluid, and describe its behavior in a way similar to the physical modeling of fluids []. Dynamical models are well equipped to deal with such scenarios [].

In these scenarios, action (or identity) recognition reduces to classifying dynamical models.

Hidden Markov models [17] have been indeed widely employed in action recognition [37], [45] and gait identification [48], [8]. HMM classification can happen either by evaluating the likelihood of a new sequence with respect to the learnt models, or by learning a new model for the test sequence, measuring its distance from the old models, and attributing to it the label of the closest model. The latter approach applies in principal to any kind of dynamical model.

Indeed, many researchers have explored the idea of encoding motions via linear [5], nonlinear [?], stochastic [?], [?] or chaotic [?] dynamical systems, and classifying them by measuring distances in their space. Chaudry et al [10], for instance, have used nonlinear dynamical systems (NLDS) to model times series of histograms of oriented optical flow, measuring distances between NLDS by means of Cauchy kernels, while Wang and Mori [51], have proposed sophisticated max-margin conditional random fields to address locality by recognizing actions as constellations of local motion patterns.

Sophisticated graphical models can be useful to learn in a bottom-up fashion the temporal structure or plot of a footage, or to describe causal relationships in complex activity patterns [30]. Gupta et al [19] work on determining the plot of a video by discovering causal relationships between actions,

Programming is used for storyline extraction on baseball footage.

DISTANCE-BASED RECOGNITION

As a matter of fact, a number of distance functions between linear systems have indeed been introduced [5], in particular in the context of system identification [33], [55], [46], and a vast literature about dissimilarity measures between Markov models also exists [14], [52], mostly about variants of the Kullback-Leibler divergence [25]. Whatever the class of dynamical model we decide to employ to represent motions, though, no single distance function can possibly outperform all the others in each and every classification problem, as models (or image sequences) can be endowed with different labeling while maintaining the same geometrical structure.

A reasonable approach when possessing some a-priori information is therefore trying to *learn* in a supervised fashion the “best” distance function for a specific classification problem [2], [4], [43], [49], [56], [16]. A natural optimization criterion consists on maximizing the classification performance achieved by the learnt metric, a problem which has elegant solutions in the case of linear mappings [44], [53]. Xing et al. [53] have recently proposed a way to solve this optimization problem for *linear* maps $y = A^{1/2}x$, when some pairs of points are known to be “similar”, leading to a parameterized family of Mahalanobis distances $\|x - y\|_A$. Shental et al. [44] have posed a similar optimization problem in the framework of information theory.

However, as even the simplest linear dynamical models live in a nonlinear space, the need for a principled way of learning Riemannian metrics from such data naturally arises. An interesting tool is provided by the formalism of *pullback metrics*. If the models belong to a Riemannian manifold \mathcal{M} , any diffeomorphism of \mathcal{M} onto itself or “automorphism” induces such a metric on \mathcal{M} . By designing a suitable family of automorphisms depending on a parameter λ , we obtain a family of pullback metrics

Pullback metrics are a well studied notion of differential geometry [23], which has found several applications in computer vision. Their use has been recently proposed by Lebanon [27] in the context of document retrieval, where a proper Fisher metric is available: instead of optimization classification rates, the inverse volume of the pullback manifold are there maximized. In [11] pullback Fisher metrics for simple autoregressive models of order 2 are learned. However, only one-dimensional observations are allowed, making the approach impractical for action recognition. In general such metrics, as they optimize a geometric quantity unrelated to classification, deliver rather modest classification performances.

A. Contribution and paper outline

In this paper we propose a 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. The framework is introduced in its generality in Section II. 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 [7]. 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 [27]).

We first consider the class $\mathcal{AR}(2)$ of multi-dimensional autoregressive models of order 2 (Section III). After studying the Riemannian structure of their manifold, we design a number of automorphisms which in turn induce families of parameterized pullback metrics on $\mathcal{AR}(2)$. We apply this framework to identity recognition from gait (Section VI-A). We use the video sequences of the Mobo database

significantly improve classification scores with respect to what obtained by a-priori distance functions.

For important classes of dynamical models used in action recognition, however, such as HMMs or variable length Markov models (VLMMs), a manifold structure is not analytically known, nor a proper metric has yet been identified. In order to learn optimal pullback distances between HMMs (a relevant case in the action recognition perspective) we need to relax the constraint of having a proper manifold structure, extending the pullback learning technique to apply to mere distance functions or divergences in the space of models.

In Section IV, after studying the product manifold structure of the space of HMMs, we design an automorphism inducing a family of parameterized pullback distances between Markov models. In Section VI-A we conduct tests on the KTH and Weizmann datasets to demonstrate the dramatic improvement in classification rates (with respect to any chosen base distance, and in absolute terms) caused by employing optimal pullback distances between HMMs.

II. LEARNING PULLBACK METRICS FOR DYNAMICAL MODELS

Let us suppose a data-set of dynamical models is available. Suppose also that such models live on a Riemannian manifold \mathcal{M} of some sort, i.e, a Riemannian *metric* is defined in any point of the manifold. Any automorphism (a differentiable map) from \mathcal{M} to itself induces a new metric, called “pullback metric”.

A. Metrics, geodesics, and distance functions

DEFINITIONS HERE

Formally, consider a family of automorphisms between the Riemannian manifold \mathcal{M} in which the data-set $D = \{m_1, \dots, m_N\} \subset \mathcal{M}$ resides and itself: $F_p : \mathcal{M} \rightarrow \mathcal{M}, m \mapsto F_p(m), m \in \mathcal{M}$. Let us denote by $T_m\mathcal{M}$ the tangent space to \mathcal{M} in m . Any such automorphism F is associated with a “push-forward” map

$$\begin{aligned} F_* : T_m\mathcal{M} &\rightarrow T_{F(m)}\mathcal{M} \\ v \in T_m\mathcal{M} &\mapsto F_*v \in T_{F(m)}\mathcal{M} \end{aligned} \quad (1)$$

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

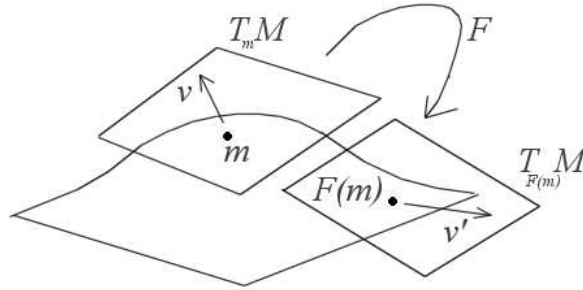


Fig. 1. The push-forward map associated with an automorphism on a Riemannian manifold \mathcal{M} .

Riemannian metric $g : T\mathcal{M} \times T\mathcal{M} \rightarrow \mathbb{R}$ on \mathcal{M} . Roughly speaking, g determines how to compute scalar products of tangent vectors $v \in T_m\mathcal{M}$. The map F induces a *pullback* metric on \mathcal{M} :

$$g_{*m}(u, v) \doteq g_{F(m)}(F_*u, F_*v). \quad (2)$$

The scalar product of two vectors u, v according to g_{*m} is computed as the scalar product with respect to the *original* metric g of the pair of vectors F_*u, F_*v which are mapped onto u, v by F_* . The pullback geodesic between two points 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 \mathcal{M} , also depending on λ . We can then define an optimization problem over such 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.

Given a pullback metric on a manifold \mathcal{M} , the corresponding *pullback distance* between two points $m, m' \in \mathcal{M}$ will simply be the geodesic distance between the two points according to the obtained pullback metric.

C. Fisher metrics for dynamical 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 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

$$g_{ij} \doteq E\left[\frac{\partial \log p(x, \xi)}{\partial \xi_i} \frac{\partial \log p(x, \xi)}{\partial \xi_j}\right] \quad (3)$$

is non-degenerate, $G = [g_{ij}]$ 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 [21].

* EXTEND: MORE DETAILS ON DIFFERENT CLASSES, METRICS/GEODESICS FOR SOME, ONLY DISTANCES/DIVERGENCES FOR OTHERS

* INTEGRATE WITH EXISTING MATERIAL, MOVE SEC IV-D HERE?

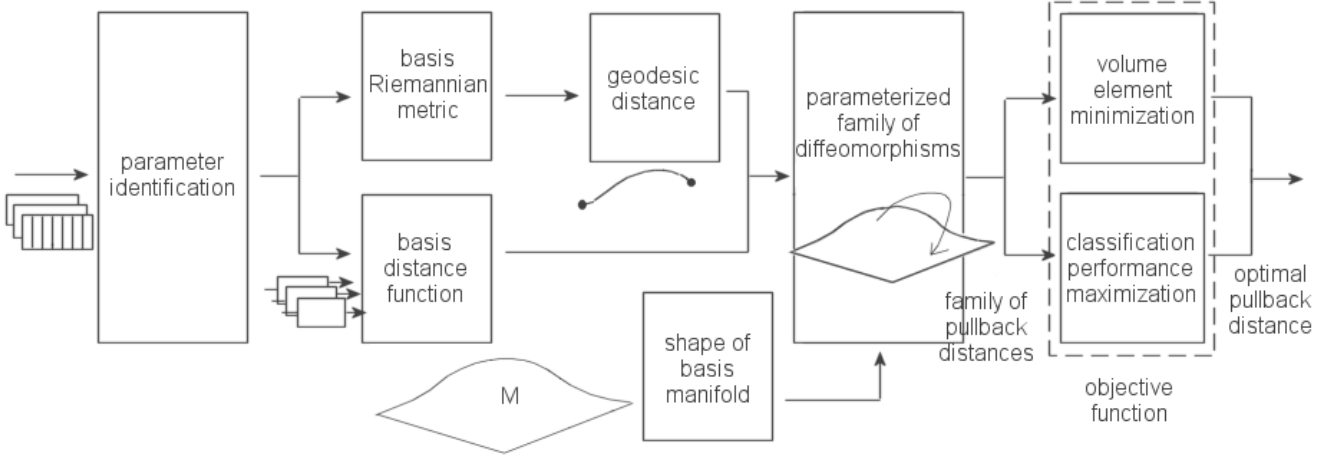


Fig. 2. A bird's eye view of the general framework for learning pullback metrics on dynamical models proposed here.

D. General framework

As many important types of 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. As a matter of fact, Itoh et al [23] have recently done some work on pullbacks of Fisher information metrics. However, as we have seen that other important classes of models do not admit a proper Riemannian structure, the formalism needs to be made more flexible, in order to be extended to cases in which mere distance functions or even divergences are available.

Accordingly, we present here a general framework for learning an optimal pullback metric/distance from a training set of dynamical models, as depicted in Figure 2. The learning procedure involves the following steps:

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

to be known¹. In alternative, \mathcal{M}_C can be a space endowed with an arbitrary distance function;

4. in order to measure distances between pairs of models on the \mathcal{M}_C , either a distance function $d_{\mathcal{M}}$ or a proper Riemannian metric $g_{\mathcal{M}}$ have to be defined there;
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 (see Section II-A);
6. a family F_{λ} of automorphisms from \mathcal{M}_C onto itself, parameterized by a vector λ , is then designed to provide a search space of metrics (the variable in our optimization scheme) from which to select the optimal one;
7. such family F_{λ} of mappings induces a family of pullback metrics (2) $g_{*\lambda\mathcal{M}}$ or distance functions $d_{*\lambda\mathcal{M}}$ on \mathcal{M} , respectively;
8. we can therefore optimize over this family of pullback distances/metrics in order to find the optimal such function/metric, according to some appropriate objective function;
9. this yields an optimal pullback metric \hat{g}_* or distance function \hat{d}_* ;
10. in the former case, knowing the geodesics of \mathcal{M} we can compute optimal geodesic distances on \mathcal{M} based on \hat{g}_* ;
11. the optimal distance function can finally be used to cluster or classify new “test” models/sequences.

E. The choice of the objective function

1) *Classification performance by cross validation:* When the data-set of models is *labeled*, we can exploit this information to determine the optimal metric/distance function.

As we have mentioned in the introduction, in the linear case (in which we look for a linear transformation of the original dataset into a new space), analytical solutions can actually be achieved by

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

means of simple convex optimization techniques [2], [4], [43], [49], [56], [16], [44], [53]. In the case of dynamical models, which live in a non-linear space, obtaining a closed-form solution for an optimal distance function is extremely complicated.

To avoid this problem, we can use cross-validation [7] to optimize the classification performance of the metric. The idea is to divide the overall sample into a number of v folds. The models belonging to $v - 1$ folds are used as training sample, the remaining fold as testing sample, and parameter of the pullback metric which optimizes the correct classification rate on fold v given the training set is selected. This is done for each possible choice of the testing sample.

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. EXPLAIN BETTER!

2) *Volume element minimization*: When the training set is *unlabeled*, distance function learning has to be based on purely geometrical considerations. G. Lebanon [27] 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 [34]:

$$\mathcal{O}(D) = \prod_{k=1}^N \frac{(\det g(m_k))^{-\frac{1}{2}}}{\int_{\mathcal{M}} (\det g(m))^{-\frac{1}{2}} dm} \quad (4)$$

where $g(m_k)$ denotes the Riemannian metric in the point m_k of the data-set D living on a Riemannian manifold \mathcal{M} . This amounts to finding a lower dimensional representation of the dataset, in a similar fashion to LLE [41] or laplacian eigenmaps [3], where dimensionality reduction is often considered a factor in improving classification.

The computation of (4) requires that of the *Gramian* $\det g$. To find the expression of the Gramian associated with a pullback metric (2) we first need to choose a base of the tangent space $T_m\mathcal{M}$ to \mathcal{M} . Let us denote by $\{\partial_i, i = 1, \dots, \dim \mathcal{M}\}$ the base of $T_m\mathcal{M}$.

The push-forward of the vectors of this base yields a base for $T_{F(m)}\mathcal{M}$. By definition, the push-forward of a vector $v \in T_m\mathcal{M}$ is [27]

$$F_p(v) \doteq \left. \frac{d}{dt} F_p(m + t \cdot v) \right|_{t=0}, \quad v \in T_m\mathcal{M}. \quad (5)$$

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

$$J = [w_1; \dots; w_{\dim \mathcal{M}}]. \quad (6)$$

The volume element of the pullback metric g_* in a point $m \in \mathcal{M}$ is the determinant of the Gramian [27]: $\det g_*(m) \doteq \det[g(F_{*p}(\partial_i), F_{*p}(\partial_j))]_{ij} = \det(J^T g J)$. If J is a square matrix (as in the following) we get simply

$$\det g_*(m) = \det(J)^2 \cdot \det g(m). \quad (7)$$

Plugging (7) in (4) we obtain the function to minimize.

F. Three relevant classes of dynamical models

AR2, ARMA \rightarrow dynamic texture; HMM \rightarrow action, gait id; hierarchical MM \rightarrow activity

III. PULLBACK METRICS FOR AUTOREGRESSIVE MODELS

* In virtue of their importance as a class of dynamical models, and their relative simplicity, ...

MOTIVATE BETTER

We consider first the class of stable autoregressive discrete-time processes of order 2, $\mathcal{M} = \mathcal{AR}(2)$, in a stochastic setting in which the input signal is a Gaussian white noise with zero mean and unit variance. This set can be given a Riemannian manifold structure under Fisher metric (3). Such systems,

in the case of scalar observation processes, are characterized by a transfer function of the form

$$h(z) = z^2 / (z^2 + a_1 z + a_2), \quad (8)$$

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

The basis manifold \mathcal{M} and the associated Fisher metric *in the scalar case* have been studied in the context of control theory [36], [39]. A natural parametrization uses the non-unit coefficients (a_1, a_2) of the denominator of the transfer function as coordinates on such manifolds. We build here on these results to determine a coordinate chart and a product Fisher metric on the manifold $\mathcal{AR}(2, p)$ of p -dimensional AR models. We will then design two different families of automorphisms on $\mathcal{AR}(2, p)$, and use the framework of Section II to determine there two families of pullback distance functions.

A. The basis manifold $\mathcal{AR}(2, 1)$ 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 (8) 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 therefore composed by a single connected component (see Figure 3-left). The Riemannian Fisher tensor is [36]:

$$g(a_1, a_2) = \frac{\begin{pmatrix} 1 + a_2 & a_1 \\ a_1 & 1 + a_2 \end{pmatrix}}{(1 + a_1 + a_2)(1 - a_1 + a_2)(1 - a_2)} \quad (9)$$

with volume element:

$$\det g_{\mathcal{AR}(2,1)}(a_1, a_2) = \frac{1}{(1 - a_2)^2(1 - a_1 + a_2)(1 + a_1 + a_2)}. \quad (10)$$

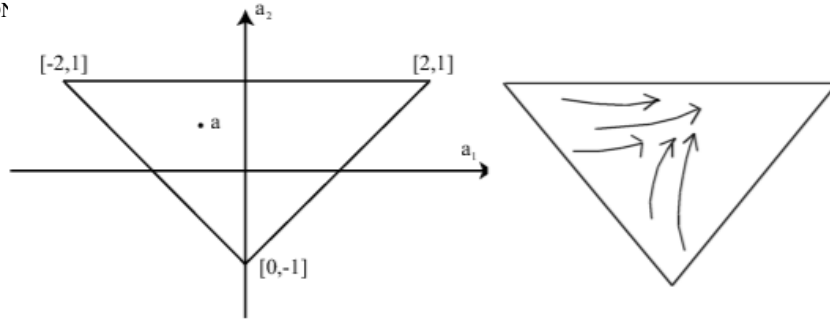


Fig. 3. Left: The manifold of stable scalar autoregressive systems of order 2, $\mathcal{AR}(2, 1)$, with the non-unit coefficients of the denominator of $h(z)$ as parameters. It forms a triangle (a simplex) with vertices $[-2, 1]$, $[2, 1]$, $[0, -1]$. Right: Effect of an automorphism of the form (15) on the $\mathcal{AR}(2, 1)$ simplex.

B. The basis manifold in the multidimensional case $\mathcal{AR}(2, p)$

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). If we use two coefficients a_1^i, a_2^i to describe each channel, each p -dimensional AR system will have coordinates $\mathbf{a} = [a_1^i, a_2^i : i = 1, \dots, p]'$. $\mathcal{AR}(2, p)$ is therefore the *product manifold*

$$\mathcal{AR}(2, p) = \times_{i=1}^p \mathcal{AR}(2, 1), \quad (11)$$

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

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

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

$$\mathbf{m} = [m_{i_1, \dots, i_p}, i_j = 1, \dots, 3 \forall j = 1, \dots, p]' \quad (12)$$

such that $\mathbf{a} = \sum_{i,j=1,\dots,p} m_{i_1,\dots,i_p} \mathbf{v}_{i_1,\dots,i_p}$. It is easy to check that $m_{i_1,\dots,i_p} \propto m_{i_1}^1 \cdot \dots \cdot m_{i_p}^p$ (up to normalization).

C. Product metric

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

$$g_{(p,q)}^{\mathcal{M}_1 \times \mathcal{M}_2} : T_{(p,q)}(\mathcal{M}_1 \times \mathcal{M}_2) \times T_{(p,q)}(\mathcal{M}_1 \times \mathcal{M}_2) \rightarrow \mathbb{R}$$

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

where $\pi_i : \mathcal{M}_1 \times \mathcal{M}_2 \rightarrow \mathcal{M}_i$ 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 $g_{\mathcal{AR}(2,p)}$ is described by a $2p \times 2p$ block diagonal matrix, whose p 2×2 blocks are copies of the metric (9) valid in the scalar case:

$$g_{\mathcal{AR}(2,p)}(\mathbf{a}) = \text{diag}\left(g_{\mathcal{AR}(2,1)}(a_1^i, a_2^i)\right). \quad (13)$$

Accordingly, its volume element $\det g_{\mathcal{AR}(2,p)}$ is (remembering the expression (10) of the scalar volume element)

$$\det g_{\mathcal{AR}(2,p)}(\mathbf{a}) = \prod_{i=1}^p \det g_{\mathcal{AR}(2,1)}(a_1^i, a_2^i) = \prod_{i=1}^p \frac{1}{(1 - a_2^i)^2(1 - a_1^i + a_2^i)(1 + a_1^i + a_2^i)}. \quad (14)$$

D. On the geodesics of the base manifold

To compute the distance between two points of a Riemannian manifold (and in particular two dynamical models) the metric is not sufficient. It is necessary to compute (analytically or numerically) the shortest path connecting them on the manifold (geodesic). All the geodesics of stable $\mathcal{AR}(2,1)$

$\gamma_2 = a_2$) have been analytically computed by Rijkeboer [39]:

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

In the general case $AR(2, p)$ the manifold's geodesics are not analytically known. However [38]:

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 $\mathcal{AR}(2, p)$ is itself a product manifold (11), the (known) geodesics of the “scalar” manifold $\mathcal{AR}(2, 1)$ are also geodesics of $\mathcal{AR}(2, p)$. As an approximation, therefore, we can measure the geodesic distance between two generic p -dimensional autoregressive models by applying a generalization of Pythagoras' 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.

E. Automorphisms

1) *An automorphism for the scalar case $\mathcal{AR}(2, 1)$:* To build a parameterized family of Riemannian metrics for $\mathcal{AR}(2, p)$ it is necessary to choose a family of automorphisms of the manifold onto itself (Section II). The more sophisticated the set of automorphisms, the larger is the search space to optimize the metric on. One possible choice for an automorphism of $\mathcal{AR}(2, 1)$ is suggested by the triangular form of the manifold, which has three vertices (see Figure 3-left). Let us $\mathbf{m} = [m_1, m_2, m_3]'$ collect the “simplicial” coordinates of a system $\mathbf{a} \in \mathcal{AR}(2, 1)$ in the manifold: $\mathbf{a} = [a_1, a_2]' = m_1[0, -1]' + m_2[2, 1]' + m_3[-2, 1]'$.

A natural automorphism of a simplex onto itself is given by “stretching” the simplicial coordinates

of its point by a set of weights $\lambda = [\lambda_1, \lambda_2, \lambda_3]$ with $\sum_j \lambda_j = 1, \lambda_j \geq 0$:

$$F_\lambda(\mathbf{m}) = F_\lambda([m_1, m_2, m_3]') = \frac{[\lambda_1 m_1, \lambda_2 m_2, \lambda_3 m_3]'}{\lambda \cdot \mathbf{m}} \quad (15)$$

where $\lambda \cdot \mathbf{m}$ denotes the scalar product of the two vectors. The application (15) stretches the triangle towards the vertex with the largest λ_i (Figure 3-right).

2) *A product automorphism for $\mathcal{AR}(2, p)$* : A product automorphisms for the whole manifold $\mathcal{AR}(2, p)$ of multidimensional, p channel autoregressive models can be obtained by using (15), 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 \mathbf{a} in the i -th sub-manifold, such a system will be identified by a vector $\mathbf{m} = [\mathbf{m}^i, i = 1, \dots, p]'$ of $3p$ such coordinates. The mapping

$$F_{\lambda^i, i=1, \dots, p}(\mathbf{m}) = [F_{\lambda^i}(\mathbf{m}^i), i = 1, \dots, p]' \quad (16)$$

with $3p$ parameters applies an automorphism (15) with parameter λ^i to the projection of \mathbf{m} onto each sub-manifold.

3) *A global automorphism for $\mathcal{AR}(2, p)$* : In alternative, the global geometry of the product manifold $\mathcal{AR}(2, p)$ inspires a *global* automorphism which acts on the manifold as a whole, by multiplying its “polytopial” coordinates (12) by a set of convex weights $\mu = [\mu_{i_1, \dots, i_p}, i_j = 1, \dots, 3 \forall j = 1, \dots, p]'$, to obtain (up to normalization)

$$F_\mu(\mathbf{m}) \propto [\mu_{i_1, \dots, i_p} \cdot m_{i_1, \dots, i_p}, i_j = 1, \dots, 3 \forall j = 1, \dots, p]' \quad (17)$$

F. Volume element for $\mathcal{AR}(2, p)$ under product automorphism

* LABELED CASE, CROSS VALIDATION (BRIEF MENTION)

Assume instead that the data-set of models is unlabeled. To select an optimal pullback metric

for p -dimensional autoregressive models by volume minimization (4), we need to find the analytical

expression of the determinant of the Gramian $detg_{*\lambda}$ (7) as a function of the parameter vector λ . By plugging it into the expression for the inverse volume (4) we obtain the objective function to optimize.

We consider here the relatively simpler product diffeomorphism (16).

Notice that, in the product simplicial coordinates $\mathbf{m} = [\mathbf{m}^i, i = 1, \dots, p]'$ of a system of $\mathcal{AR}(2, p)$, the volume element of the product Fisher metric (14) reads as:

$$\det g_{\mathcal{AR}(2,p)}(\mathbf{m}) = \prod_{i=1}^p \frac{1}{(m_1^i)^2 m_2^i m_3^i}. \quad (18)$$

Theorem 1: The volume element of the Fisher pullback metric on $\mathcal{AR}(2, p)$ induced by the automorphism (16) is:

$$detg_{*\lambda}(\lambda, \mathbf{m}) = \frac{1}{2^{2p}} \prod_{i=1}^p \frac{(\lambda_1^i \lambda_2^i \lambda_3^i)^2}{(\lambda^i \cdot \mathbf{m}^i)^6} \cdot \frac{1}{(m_1^i)^2 m_2^i m_3^i}. \quad (19)$$

Proof: We need to compute the Gramian $detg_{*\lambda}$ (7) of the pullback metric under the automorphism (16). Following the procedure of Section II, we need to choose a basis of the tangent space $T\mathcal{AR}(2, p) = T\mathcal{AR}(2, 1) \oplus \dots \oplus T\mathcal{AR}(2, 1)$ of the product manifold (11), such as the size $2p$ vectors

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

whose only non-zero entries are in positions $2i - 1, 2i$.

Let us express the product automorphism (16) in coordinates $\mathbf{a} = [a_1^1, a_2^1, \dots, a_1^p, a_2^p]$. As for all

$i = 1, \dots, p$

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

$$= \frac{1}{\Delta_i} [2\lambda_2^i(1 + a_1^i + a_2^i) - 2\lambda_3(1 - a_1^i + a_2^i), \lambda_2(1 + a_1^i + a_2^i) + \lambda_3(1 - a_1^i + a_2^i) - 2\lambda_1(1 - a_2^i)]', \quad (20)$$

$F_{\lambda^1, \dots, \lambda^p}(\mathbf{a})[h, l] = 0$ elsewhere, 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 $i = 1, \dots, p$ the push-forward tangent vectors

$$\mathbf{w}_1^i = \left. \frac{d}{dt} F_{\{\lambda^j, j\}}(\mathbf{a} + t\partial_1^i) \right|_{t=0}, \quad \mathbf{w}_2^i = \left. \frac{d}{dt} F_{\{\lambda^j, j\}}(\mathbf{a} + t\partial_2^i) \right|_{t=0}.$$

We get $\mathbf{w}_1^i[2i-1, 2i] =$

$$= [-2\lambda_1^i \lambda_3^i (3 - a_2^i + a_1^i) + 2\lambda_2^i (\lambda_1^i - 2\lambda_3^i) (1 + a_1^i + a_2^i), 2\lambda_1^i \lambda_3^i (3 - a_2^i + a_1^i) + 2\lambda_1^i \lambda_2^i (1 + a_1^i + a_2^i)],$$

$\mathbf{w}_1^i[h, l] = 0$ elsewhere. AND $W_2^i??$

The matrix J (6) formed by the stacked collection of the row vectors \mathbf{w}_j^i , $J = [\mathbf{w}_1^1; \mathbf{w}_2^1; \dots; \mathbf{w}_1^p; \mathbf{w}_2^p]$, is clearly block diagonal. Its determinant is therefore simply the product of the determinants of the blocks:

$$\det J = \frac{1}{2^p} \prod_{i=1}^p \frac{\lambda_1^i \lambda_2^i \lambda_3^i}{(\lambda^i \cdot \mathbf{m}^i)^3} \quad (21)$$

By plugging (21) and (18) into the general expression (7) of the pullback volume element, we get (19). ■

The objective function to maximize (the inverse volume element) is finally obtained by plugging (19) in its general expression (4). The normalization factor $I(\lambda) = \int_M (\det g_{*\lambda}(m))^{-\frac{1}{2}} dm$ can be approximated as $I(\lambda) \simeq \sum_{k=1}^N \det g_{*\lambda}(\lambda, \mathbf{m}_k)^{-\frac{1}{2}}$ [27].

We will see in Section VI-A the empirical effects of the two approaches (cross validation and volume minimization) on gait recognition performances.

* USE OF LDSs * HMMs EVEN MORE IMPORTANT

A. Hidden Markov models

A *hidden Markov model* is a statistical model whose states $\{X_k\}$ form a *Markov chain*, and in which only a corrupted version $y_k \in \mathcal{Y} \subset \mathbb{R}^D$ of the state (“observation process”) is observable. If we associate its n states $X_k \in \mathcal{X}$ (\mathcal{X} is the finite state space) with versors $e_i = [0, \dots, 0, 1, 0, \dots, 0]' \in \mathbb{R}^n$ [17] we can write the model⁴ as

$$\begin{cases} X_{k+1} = AX_k + V_{k+1} \\ y_{k+1} = CX_k + \text{diag}(W_{k+1})\Sigma X_k. \end{cases} \quad (22)$$

Given a state $X_k = e_j$, the observations are assumed to have Gaussian distribution $p(y_{k+1}|X_k = e_j)$ centered around a vector $C_j = E[p(y_{k+1}|X_k = e_j)]$ which is the j -th column of the matrix C . The parameters of a hidden Markov model (22) are therefore the *transition matrix* $A = [a_{ij}] = P(X_{k+1} = e_i|X_k = e_j)$, the matrix C collecting the means C_j of the state-output distributions $p(y_{k+1}|X_k = e_j)$, and the matrix Σ of their variances. Given a sequence of observations $\{y_1, \dots, y_T\}$ they can be identified by means of the EM algorithm [13], [17].

To apply the pullback learning technique of Section II to HMMs we need to understand the structure of the space \mathcal{H} they live in (point **3.**), in order to design an appropriate parametric family of automorphisms (point **6.**).

⁴Here $\{V_{k+1}\}$ is a sequence of martingale increments and $\{W_{k+1}\}$ is a sequence of i.i.d. Gaussian noises $\mathcal{N}(0, 1)$.

In most practical cases the state-output covariance Σ is assumed to be fixed, as this assumption helps the convergence of the EM algorithm without excessively jeopardizing the generative power of a HMM. In that case each HMM of the form (22) is completely determined by the two matrices A and C , and the space $\mathcal{H} = \{(A, C)\}$ of HMMs with fixed covariance matrix is a product space $\mathcal{H} = \mathcal{M}_A \times \mathcal{M}_C$, where \mathcal{M}_A denotes the space of all transition matrices $n \times n$, while \mathcal{M}_C is the (at this point still unknown) space of state-output matrices.

1) *The product of simplices of all transition matrices:* Transition matrices (also called “stochastic matrices”) have $n(n - 1)$ free parameters, as they are subject to n constraints enforcing that the elements of each column sum to 1: $\sum_{i=1}^n P(X_{k+1} = e_i | X_k = e_j) = 1$, as the j -th column of A is the (conditional) probability distribution $P(X_{k+1} | X_k = e_j)$. But probability distributions on a finite set of n elements (the states of the Markov model) live in a simplex

$$Cl(p_1, \dots, p_n) = \left\{ \alpha_1 p_1 + \dots + \alpha_n p_n, \sum_i \alpha_i = 1 \right\},$$

with as n vertices p_i the probability distributions $p_i : \mathcal{X} \rightarrow [0, 1]$ such that $p_i(e_i) = 1$, $p_i(e_j) = 0 \forall j \neq i$ and dimension $n - 1$. As a consequence, transition matrices live in the product of n such simplices, one for each column of A (see Figure 4):

$$\mathcal{M}_A = \times_{j=1}^n Cl(p_1^j, \dots, p_n^j).$$

The simplicial coordinates of a transition matrix A in each of these individual polytopes $j = 1, \dots, n$ are simply given by the entries of the j -th column A_j itself: $A_j = \sum_i a_{i,j} p_i^j$. The overall coordinate vector of A in \mathcal{M}_A is then formed by the collection of the simplicial coordinates $(a_{i,j}, i = 1, \dots, n)$ of all its columns j in the respective polytopes, i.e., the stacked vector of its columns.

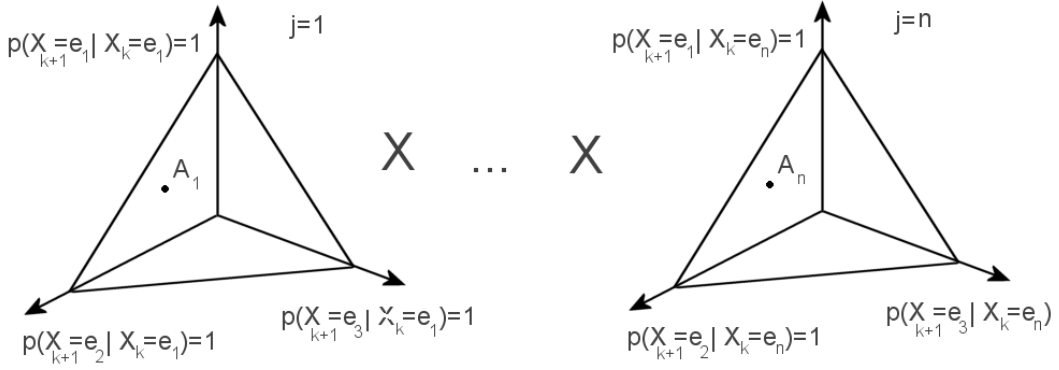


Fig. 4. Left: The manifold \mathcal{M}_A of transition matrices with n states is the product of n simplices, the j -th simplex being the probability simplex of all the conditional probabilities $P(X_{k+1} = e_i | X_k = e_j)$. The j -th column A_j of a transition matrix A lives in the j -th component of \mathcal{M}_A .

2) *Learning an approximate observation space:* The situation is quite more complex when it comes to the state-output matrix C . Obviously, C matrices are n -plets of vectors of the observation space \mathcal{Y} of the Markov model. In general, when no a-priori information on the nature of the observation is available, we can only assume that the observation space is \mathbb{R}^D itself (where D is the size of the observation vectors). However, when a training set of models is available, we can use that information to build an approximation of the actual observation space. It has been observed (for instance in the gait ID context) that for specific classes of motions the observation vectors live in a pretty well defined lower-dimensional manifold of \mathbb{R}^D . It makes then sense to conduct this analysis in a more adequate lower-dimensional space, using for instance unsupervised nonlinear embeddings such as LLE [41]. By applying LLE to the collection of the columns of the C matrices of the training set of HMMs we get a cloud of d -dimensional embedded columns approximating the actual shape of the observation space for the specific training set of models at hand (Figure 5-left).

Even after a sensible LLE embedding of the training set of observation columns, however, any analytical description of such space remains elusive. We need to “coordinatize” the obtained lower-dimensional observation manifold, i.e., find a way to associate each element of the manifold with a vector of real numbers (its coordinates). This involves finding an atlas of coordinate charts covering

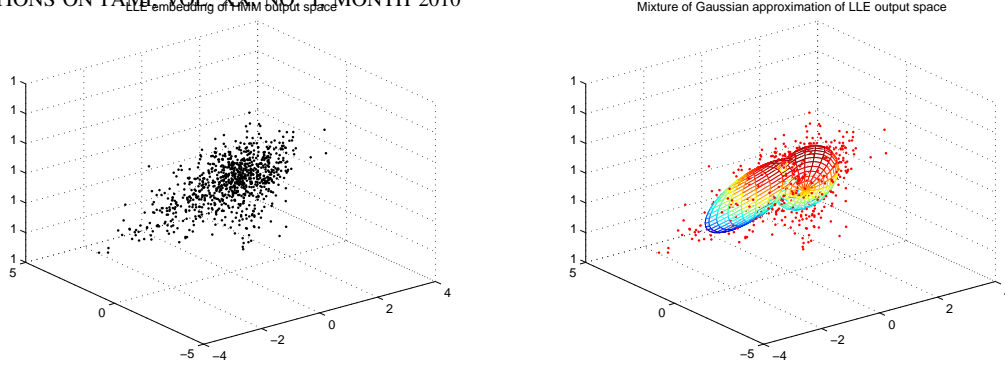


Fig. 5. Left: the embedded columns of the C matrices of all the HMMs encoding image sequences of a specific class in the Weizmann dataset form a decent approximation $\tilde{\mathcal{Y}}$ of the unknown observation space. Right: the corresponding mixture of Gaussian fitting the embedded cloud provides an atlas of coordinates charts in the embedded (approximate) observation space.

the approximate manifold. This atlas will of course have to be learned from the available training data.

A sensible way of learning such coordinate charts consists on estimating via EM the mixture of Gaussian distributions $\{\Gamma^k, k\}$ which best fits the available lower-dimensional approximate observation manifold $\tilde{\mathcal{Y}}$ obtained by dimensionality reduction of the set of columns of the C matrices of all the HMMs in the training set (Figure 5-right). As coordinates of each observation vector y we may then use the unique set of probability densities associated with them: $y \mapsto \{\Gamma^k(y), k\}$.

C. An automorphism of \mathcal{H}

As the space \mathcal{H} of hidden Markov models is a product space, we can design a product automorphism on \mathcal{H} , of the form $F(h) = F((A, C)) = (F_A(A), F_C(C))$, with $F_A : \mathcal{M}_A \rightarrow \mathcal{M}_A$, $F_C : \mathcal{M}_C \rightarrow \mathcal{M}_C$ two automorphisms of the product simplex \mathcal{M}_A of all transition matrices and of the approximate observation manifold (in the embedding space) \mathcal{M}_C , respectively.

1) *An automorphism of transition matrices:* A natural automorphism for the space \mathcal{M}_A of transition matrices can be derived by exploiting its form as the product of n probability polytopes, one for each state. Again, each probability distribution p in a probability simplex $\mathcal{P} = Cl(p_1, \dots, p_n)$ with n vertices

is a point whose simplicial coordinates are its probability values: $p = \sum_i p(e_i)p_i$.

A simple automorphism on such a simplex can be obtained by “stretching” the simplicial coordinates $\mathbf{p} = [p(e_1), \dots, p(e_n)]'$ of its points by means of a set of normalized weights $\lambda = [\lambda_1, \lambda_2, \lambda_3]$ with $\sum_j \lambda_j = 1, \lambda_j \geq 0$:

$$F_\lambda(p) = F_\lambda(\mathbf{p}) = \frac{[\lambda_1 p(e_1), \lambda_2 p(e_2), \dots, \lambda_n p(e_n)]'}{\lambda \cdot \mathbf{p}}, \quad (23)$$

where $\lambda \cdot \mathbf{p}$ denotes the scalar product of the two vectors. Intuitively, (23) stretches the triangle towards the vertex with the largest λ_i .

A *product* automorphisms for the whole manifold $\mathcal{M}_A = \times_{j=1}^n Cl(p_1^j, \dots, p_n^j)$ (similarly to the case of the $\mathcal{AR}(2, p)$ manifold, Section III-E.2) can be derived by using (23) as building block. Each of the columns $A_j = \sum_i a_{i,j} p_i^j$ of A is a (conditional) probability distribution, with coordinates given by its own entries $a_{i,j}$. The mapping: $F_{\lambda^j, j=1, \dots, n}(A) = [F_{\lambda^j}(A_j), j = 1, \dots, n]'$ (with $n \cdot (n-1)$ parameters) applies the automorphism (23) with parameter λ^j to each column A_j of the transition matrix.

2) *A mapping of the approximate observation space:* The mixture of Gaussians approximation of the observation space which allows us to introduce coordinates charts in $\tilde{\mathcal{Y}}$ can be also exploited to work out an interesting automorphism of $\tilde{\mathcal{Y}}$ itself.

Again, each output vector y has as coordinates its set of densities with respect to the mixture: $y \mapsto \{\Gamma^k(y), k\}$. For each of these Gaussian components Γ^k there exists a “dominated” region of the (embedded) observation space in which Γ^k is greater than all other components’ densities: $O^k = \{w : \Gamma^k(w) \geq \Gamma^l(w) \forall l \neq k\}$. Now, take the (threshold) maximum value attained by any other component in this part of the observation space: $T^k = \max_{l \neq k, w \in O^k} \Gamma^l(w)$, and define the “proprietary” region R^k in which the pdf Γ^k is above this threshold: $R^k = \{w : \Gamma^k(w) \geq T^k\}$. Figure 6 illustrate such regions R^1, R^2 and R^3 in the case of a one-dimensional output space and 3 Gaussian components.

We define an automorphism of the approximate observation space $\tilde{\mathcal{Y}}$ which acts non-trivially only

this way, we can ensure the injectivity of the mapping.

Suppose that we computed for each Gaussian component with mean μ_k its proprietary region R^k and threshold value T^k . An automorphism $F_{\tilde{\mathcal{Y}}} : \tilde{\mathcal{Y}} \rightarrow \tilde{\mathcal{Y}}$ can be defined as follows. Given an observation vector $y \in \tilde{\mathcal{Y}}$:

1. take the Gaussian component with highest pdf value: $K = \arg \max_k \Gamma^k(y)$;
2. if y does not belong to the proprietary region R^K of Γ^K , leave it unchanged: $y' = F_C(y) = y$;
3. if it does, its density value $\Gamma^K(y)$ must belong to the interval $[T^K, \Gamma^K(\mu_K)]$, with some convex coordinates $\mu, (1 - \mu)$, $\mu \in [0, 1]$:

$$\Gamma^K(y) = \mu T^K + (1 - \mu) \Gamma^K(\mu_K);$$

4. if $\lambda \in [0, 1]$ is the parameter of the desired automorphism, map $\Gamma^K(y)$ to the new density value

$$\Gamma^K(y') = \frac{\lambda \mu}{\lambda \mu + (1 - \lambda)(1 - \mu)} T^K + \frac{(1 - \lambda)(1 - \mu)}{\lambda \mu + (1 - \lambda)(1 - \mu)} \Gamma^K(\mu_K)$$

in the same interval $[T^K, \Gamma^K(\mu_K)]$;

5. within R^K there exists an entire hypersphere of observation vectors with such a density value $\Gamma^K(y')$ (just two points in symmetric positions wrt μ_k in the 1-dim example of Figure 6);

6. we take as mapped observation vector (the output $y' = F_{\tilde{\mathcal{Y}}}(y)$ of the automorphism of $\tilde{\mathcal{Y}}$) the *unique* vector y' which has pdf $\Gamma^K(y')$ and lies on the half-line joining μ_k and the original vector y .

3) *Mapping of C matrices and permutation issue:* * MAPPING OF C MATRICES

* ISSUE WITH STATE PERMUTATION

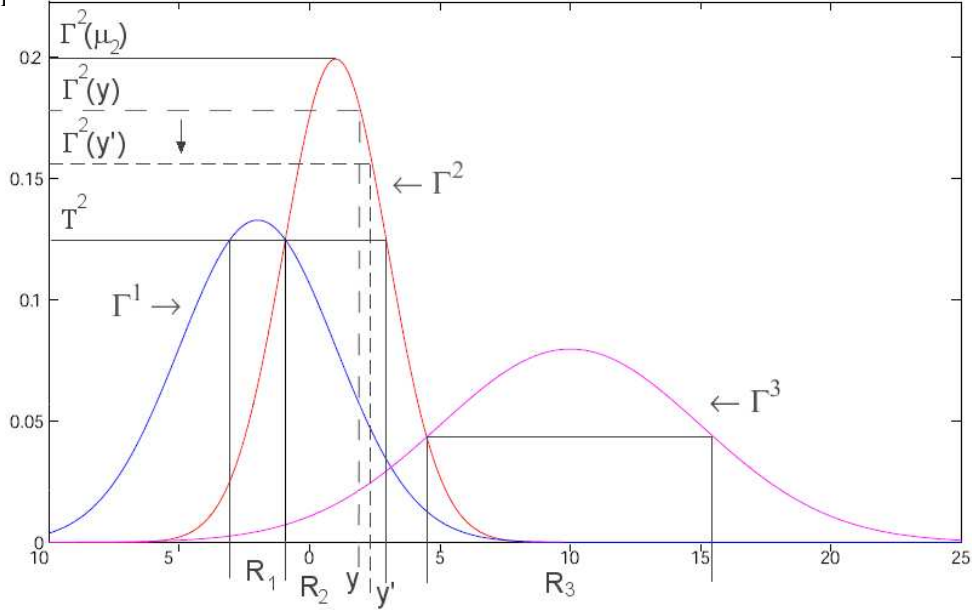


Fig. 6. Mapping observation vectors in the approximate output space given by a mixture of Gaussians: a 1-dimensional example.

D. Some base distances between HMMs

* RECALL PULLBACK DISTANCE

* NEED SOME BASE DISTANCE/DIVERGENCE

As base distance function between Markov models a variety of options is offered by the past literature. Arguably the simplest possible choice is to pick as a base metric for the product space $\mathcal{H} = \mathcal{M}_A \times \mathcal{M}_C$ the product metric obtained by applying the Frobenius norm to A and C matrices respectively: $\|H_1 - H_2\| = \|A_1 - A_2\|_F + \|C_1 - C_2\|_F$, where $\|M - M'\|_F \doteq \sqrt{\text{Tr}((M - M')(M - M')^T)}$, and $\text{Tr}(M) = \sum_{ii} M[i, i]$ is the trace of a matrix M . The Frobenius norm is inexpensive to compute, and often produces surprisingly good classification results.

A classical pseudo-distance in the space of HMMs is derived from the Kullback-Leibler divergence [25] of two probability distributions P, Q : $D_{KL}(P|Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$. Fast approximations of D_{KL} for discrete observations HMMs are available in the literature [14] CITE MORE. Computing the KL divergence for continuous observations HMMs, however, typically requires simulating its output

sequence in a number of trials, making it quite expensive to compute. An inexpensive alternative distance function derives from the Bhattacharyya distance between two Gaussian pdfs $\mathcal{G}_1, \mathcal{G}_2$ with means μ_1, μ_2 and covariances Σ_1, Σ_2 [35]:

$$d_{bhattacha}(\mathcal{G}_1, \mathcal{G}_2) = \frac{1}{8}(\mu_1 - \mu_2)' \left(\frac{\Sigma_1 + \Sigma_2}{2} \right)^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \log \frac{|(\Sigma_1 + \Sigma_2)/2|}{|\Sigma_1|^{1/2} |\Sigma_2|^{1/2}}. \quad (24)$$

Given two HMMs the average Bhattacharyya distance between pairs of output Gaussian pdfs in the two models quantifies their difference. Such distance can be further modified to include an assessment of the Kullback-inspired divergence between the two transition matrices [14]:

$$\sum_{i,j=1,\dots,n} \pi(i) A_2(i, j) \log \left(\frac{A_1(i, j)}{A_2(i, j)} \right).$$

V. THE CASE OF HIERARCHICAL HMMs

VI. TESTS ON ACTION AND IDENTITY RECOGNITION

To test the actual, empirical effect of our approach to distance function learning on the classification of dynamical models, we considered the problem of recognizing actions and identities from image sequences.

A. Pullback AR distances for identity recognition

For our tests on identity recognition, we used the Mobo database [18], 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”.

* REPHRASE! simulating this way the impact of nuisance factors actually present in gait identification, and making recognition more challenging.

1) *Observations: Silhouette-based feature sequences:* 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 [40]. Given a silhouette we

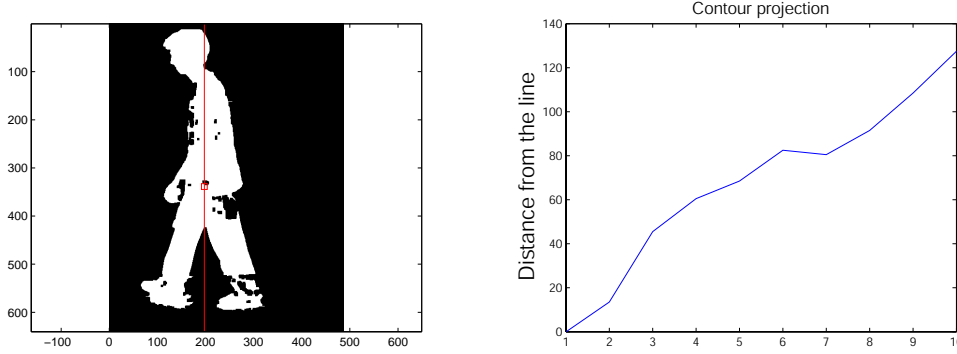


Fig. 7. Feature extraction. Left: a number of lines is drawn through the center of mass of the silhouette. Right: the distance of the points on the contour from the line is computed: these values for all the lines of the sheaf form the feature vector.

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 (see Figure 7). This proved superior to several alternative representations [28]. CITE MORE

2) *Identification of a AR(2,p) model for each sequence:* According to the pullback classification scheme of Section II, each input sequence has to be represented by a dynamical model. For these tests we decided to encode each image feature sequence as an autoregressive model of order 2. Each component of the feature/observation vector, therefore, is associated with a different output channel of the AR(2,p) model (8). 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 $i = 1, \dots, p$. For comparison, for each input sequence we also identified a hidden Markov model [17] by applying the Expectation Maximization [13] algorithm, yielding a model of the form $x(t+1) = Ax(t)+v(t), y(t) = Cx(t)+w(t)$ described by a transition $A[i, j] = P(x(t) = i|x(t-1) = j)$ and a state-output matrix C .

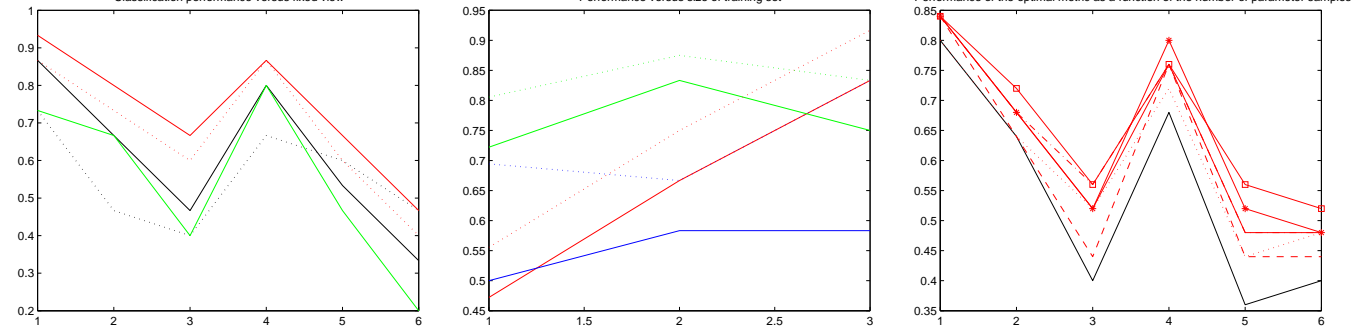


Fig. 8. Left: In six separate experiments, the classification performance of the following metrics has been computed for image sequences coming from a single view, from 1 to 6. Fifteen identities, 200 samples extracted from the parameter space. Line styles: basis Fisher geodesic distance, solid black; Frobenius distance between HMMs (??), dashed black; optimal pullback Fisher under automorphism (16), solid red; optimal pullback Fisher under automorphism (17), dashed red; inverse volume optimal Fisher metric with automorphism (16), solid green. Middle: Classification performance of 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 (16); dashed: global automorphism (17). 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 increasing.

3) *Performances of optimal pullback AR metrics:* To classify the test models, we adopted standard nearest neighbor classification: each testing sequence was 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 (Gaussian 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 8 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 - (basis) product Fisher metric (13); 2 - pullback Fisher metric induced by the product automorphism (16) optimizing the classification performance after cross-validation on the training set; 3 - pullback Fisher metric induced by the *global* automorphism (17) for the same objective function; 4 - pullback Fisher metric induced by (16) with optimal inverse volume; 5 - Frobenius distance between HMMs.

Fifteen identities have been here considered, with the parameter space of each automorphism sampled

200 times to detect the optimal parameters. The optimal classification pullback Fisher metrics induced by both 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 9 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). An interesting resistance to the expected decreasing performance as the problem grows more difficult is shown, as optimal pullback classification rates are remarkably stable compared to those of classical metrics.

4) *Influence of parameters:* It is natural to conjecture that, when optimizing the classification performance in the cross-validation procedure described in Section II, a larger training set should lead to identify more effective automorphism parameters. Indeed, Figure 8-middle shows the behavior of the considered metrics in dependence 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 (17) of the the $\mathcal{AR}(2, p)$ polytope generates more performing metrics, with a margin over the simpler, product automorphism ranging from 10% up to a remarkable 25%.

Finally, Figure 8-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 (16) is analyzed and compared with the baseline results obtained by using the basis Fisher geodesic

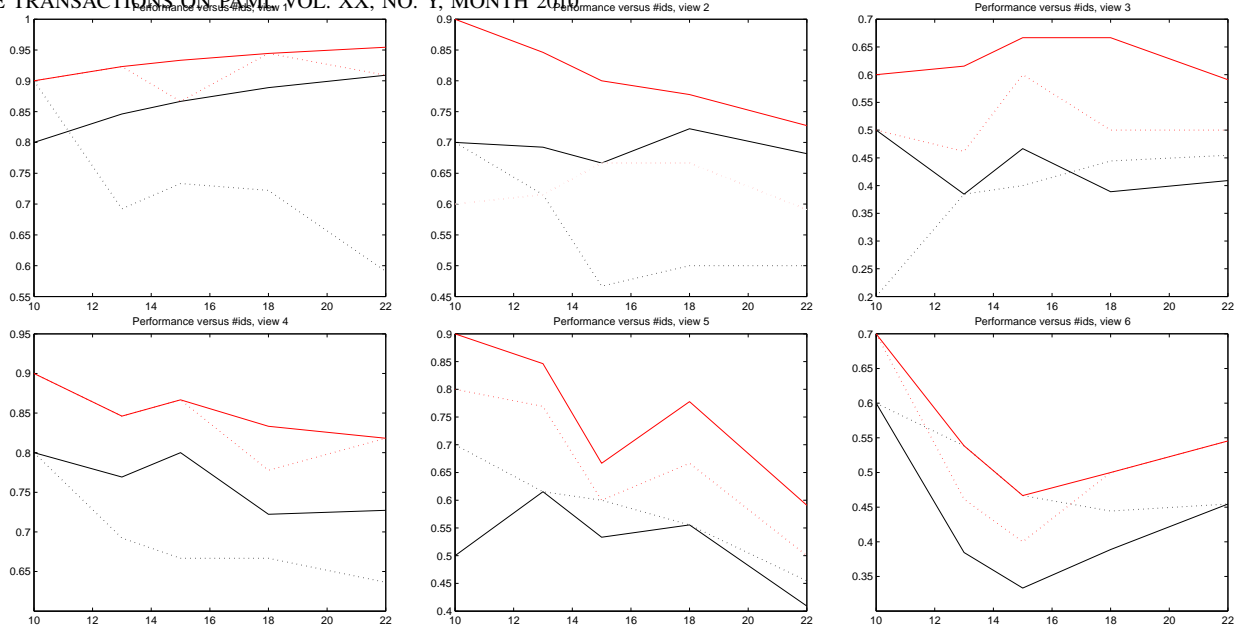


Fig. 9. The different metrics' classification scores plotted versus the number of identities (from 10 to 22) in the testing set, for all viewpoints in the Mobo database, ranging from 1 (top left) to 6 (bottom right). Solid black: Frobenius HMM distance. Dashed black: classical Fisher geodesic distance between $AR(2, p)$ models. Solid red: optimal pullback Fisher geodesic distance induced by (16). Dashed red: optimal pullback Fisher induced by (17).

distance between $AR(2, p)$ models. As expected, the margin of improvement increases quite steadily as more samples are assessed in the parameter space of the automorphism. Here all 25 identities are considered, the improvement 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, when the basis metric has the worst performance.

B. Pullback HMM distances for action recognition

We also tested the effectiveness of the proposed technique for learning optimal (classification wise) pullback distances between HMMs in the action recognition problem. Among the many available public databases on action or activity recognition we selected the Weizmann and the KTH ones to run our test on, as these are somehow more coherent with a scenario in which dynamics can play a role in recognition, besides representing standard benchmarks for action recognition approaches. The Weizmann dataset [6] includes videos of 9 different actions performed by 16 subjects, and is

considered easier as the scene is quite static and all sequences have similar characteristics. On the other hand, the KTH dataset [26] includes 6 different actions performed by 25 subjects; such dataset is more challenging, as four different scenarios are considered, subjects wear different clothes and act at different positions and distances from the camera.

1) *Feature selection: snippets*: In many approaches to action recognition, a single feature descriptor is computed for each video sequence. In this paper we use instead the feature descriptor introduced by Schindler and Van Gool in [42] (“Action snippets”), which has been proved powerful and allows us to compute one feature vector per frame, which is a prerequisite for any HMM representation of a sequence. Features are computed inside a rectangular bounding box: contrarily to silhouettes, bounding boxes can be reliably obtained in most scenarios, by using person detectors [9] or trackers [12]. Each frame of the sequence is processed in two independent pipelines associated with shape and motion features, respectively. The former computes Gabor filter responses at different orientation and scales, while the latter represents motion by computing the optical flow at different directions, velocities and scales.

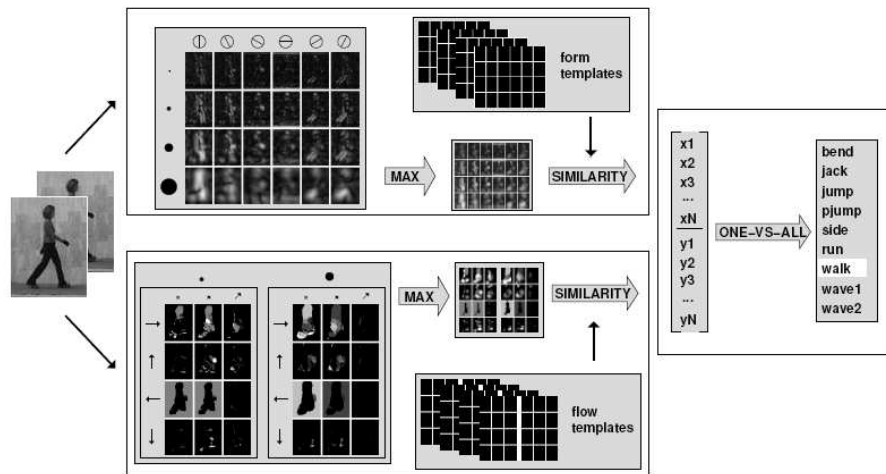


Fig. 10. The motion and shape pipelines of the action snippets' feature extraction procedure.

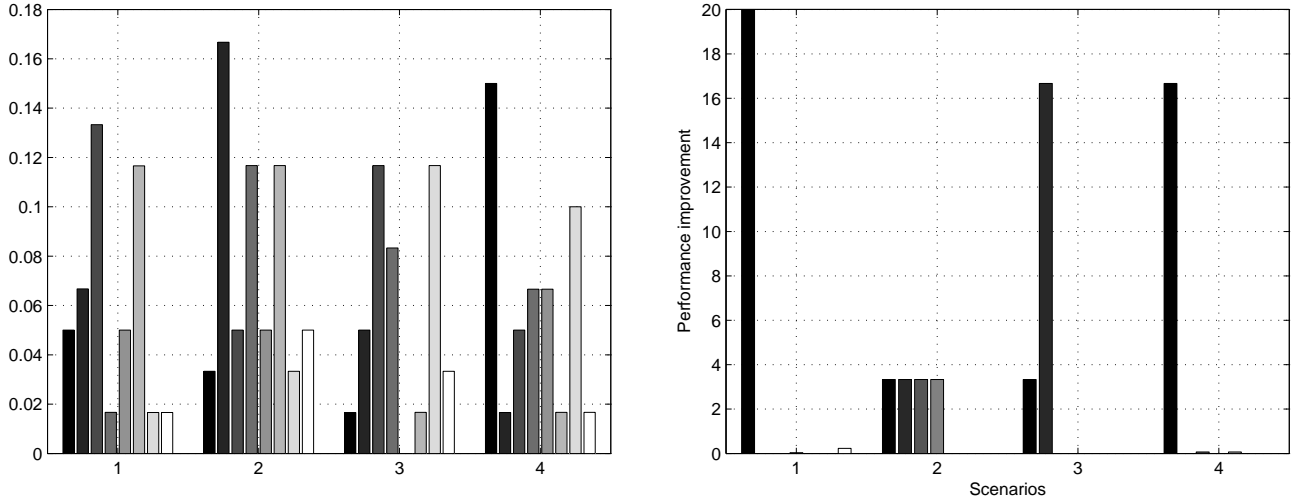


Fig. 11. Improvement on action recognition rates on the KTH dataset associated with pullback Frobenius (left) and pullback modified Bhattacharyya (right) distances. Only 6 training and 60 testing sequences were selected from the dataset to make these tests more challenging. Eight random training, validation and test sets were selected in each one of the four distinct scenarios of the KTH database.

2) *HMM identification and classification tests:* For each sequence in the two datasets we selected the first 20 such features from the first pipeline. HMM parameters for each such reduced feature sequence were identified using the EM algorithm, in the version proposed by Moore et al [17]. We set the number of states to $n = 3$, and made estimation more robust by applying the EM algorithm 10 times for each sequence and selecting the parameters yielding the highest likelihood. In all the tests presented in this section we ran repeated, random selections of a training and a testing set of sequences from the KTH or Weizmann datasets, and found the parameters of the pullback distance optimizing the classification performance over that particular testing set. As it was done in the identity recognition experiments of Section VI-A, to each test sequence we attributed the class of the closest model in the training set, according to the chosen distance. Then, parts of the remaining sequences in the dataset were selected as second test bed to measure the classification rate of the resulting, optimal pullback distance.

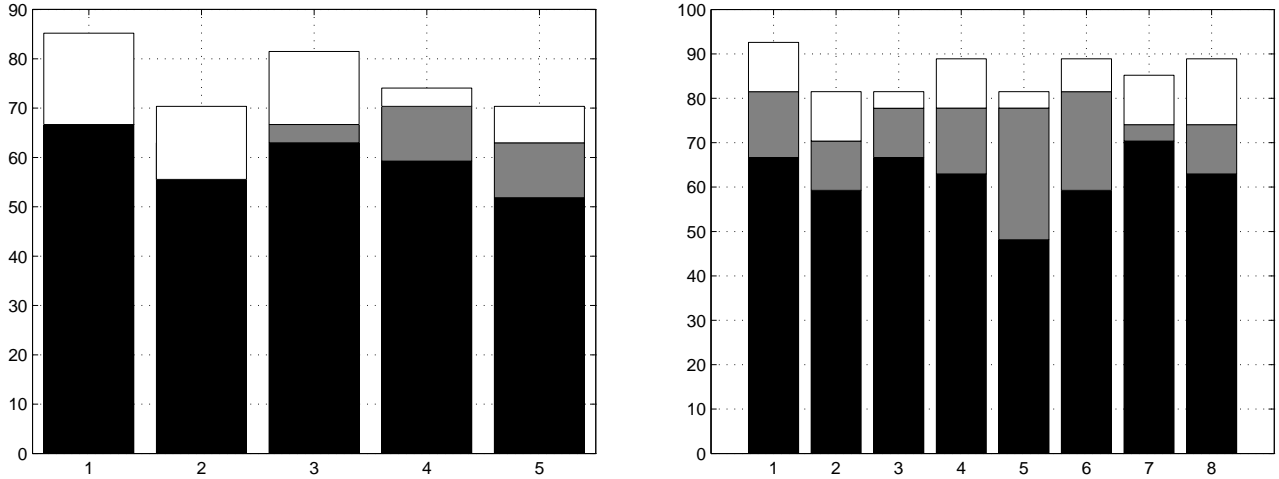


Fig. 12. Effect of pullback distance learning on action recognition rates on random subsets of the Weizmann dataset with 27 training and test sequences. Left: pullback modified Bhattacharyya (white), base modified Bhattacharyya (gray) and Frobenius (black) classification rates over 5 runs. The pullback distance was optimized over 1705 samples in the parameter space of the diffeomorphism. Right: pullback Frobenius (white), base Frobenius (gray) and modified Bhattacharyya (black) classification rates over 8 runs.

3) *Effect of pullback learning on classification performance:* Figure 11 illustrates the effect of pullback distance function learning on classification performance for the four distinct scenarios of the KTH dataset taken separately, and the two chosen base distances “Frobenius” and “modified Bhattacharyya”. In the figure for each of the 32 randomly generated runs (8 for each of the different scenarios of the KTH dataset), a bar plots the difference between the classification rate achieved by the optimal pullback distance and that achieved by the base distance on the same training and testing set. In each run 6 training and 60 testing sequences were randomly selected for cross validation optimization. The parameter space of the HMM automorphism was sampled in 55 points for the transition matrix and 51 points for the output space, yielding 2805 overall samples over which to optimize the performance of the pullback HMM distance in the cross validation stage.

The consistent, dramatic improvement in performance can be easily appreciated. In the case of the Frobenius base distance it ranged from 2 up to 16% over the base function performance. In the modified Bhattacharyya case the improvement is still visible even though less marked: this distance measure

has considerable performances to start with, and there is little room for improvement when recognition rates approach 90-95%. We noticed in our experiments that modified Bhattacharyya typically produces recognition rates equal or superior to that of the Kullback-Leibler divergence, whilst being tens of times less computationally expensive. Therefore, we decided not to run any more tests involving the KL divergence.

Figure 12 plots, instead, action recognition results on randomly selected subsets of the Weizmann dataset, with 27 training sequences (three for each action category) and 27 randomly selected test sequences. Results for both modified Bhattacharyya (left) and Frobenius (right) base distances are shown. In the former case, the pullback distance function was optimized over 55 samples in the parameter space for the transition matrix diffeomorphism F_A and 31 samples in the parameter space of the output space automorphism F_C , for a total of 1705 samples. In the latter, 28 and 11 samples were used, respectively. In both cases the improvement in recognition performance (recognizable as a white strip on the diagrams' bars) is dramatic, ranging from 4.30% to 18.52%.

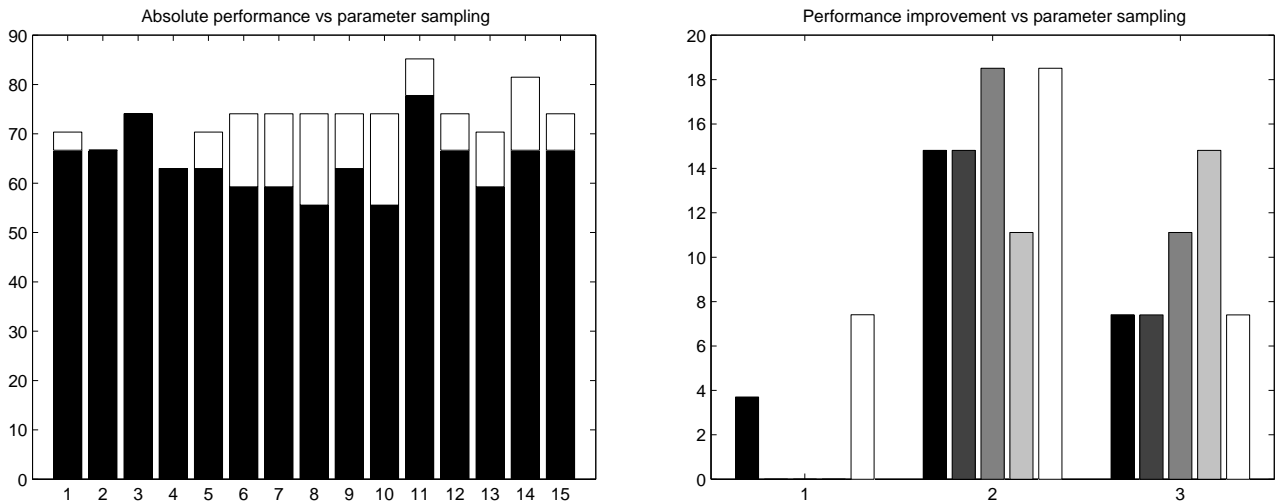


Fig. 13. The effect of the density of sampling in the automorphism's parameter space. Weizmann dataset, 9 training sequences, 27 test sequences. The Frobenius norm was chosen as base distance. Left: absolute classification rates of base Frobenius (in black) and pullback Frobenius (in white) over 15 runs: first five with 110 parameter samples, the second five with 588 parameter samples, the last five with 1705 parameter samples. Right: corresponding performance improvements over the three sets of runs (1,2,3).

Notice that, on the full Weizmann dataset (which is not an extremely challenging test bed), equipped with a large enough training set, Frobenius, KLD or Bhattacharyya distances can indifferently produce recognition results close to 100%. Selecting a smaller portion of the available sequences makes the problem artificially more difficult, highlighting the effect of pullback learning on recognition performances. On the other hand, as optimization by cross validation is rather expensive, this allows to sample more densely the parameter space of the HMM automorphism.

4) *Influence of parameters: Influence of parameter sampling.* Figure 13 shows the effect of the density of the sampling in the parameter space of the automorphism on classification performance. As the number of samples increase (from 110 to 588, to 1705) both the absolute classification performance and the relative improvement on the base distance (in this case the Frobenius norm) get better. Results of test on a subset of the Weizmann dataset with 9 training and 27 test sequences are shown. It can be noticed how pullback learning brings the absolute performance of this rather naive norm in line with the benchmark for this dataset. Improvements in classification rates for significant sampling of the diffeomorphism parameter space are very substantial, ranging from 7% up to even 18%.

Influence of training set size. When optimizing the classification performance in the cross-validation procedure described in Section II, it is natural to guess that larger training sets should lead to identify more effective automorphism parameters. Figure 14 indeed illustrates the dependence from the size of the training set over which the optimal parameters are learnt of, for instance, the pullback Frobenius norm on the Weizmann dataset. A clear correlation between larger training sets and better classification scores emerges.

VII. PERSPECTIVES AND CONCLUSIONS

* DESIGN OF AUTOMORPHISMS AUTOMATIC?

* DIFFEO TO OTHER METRIC SPACES

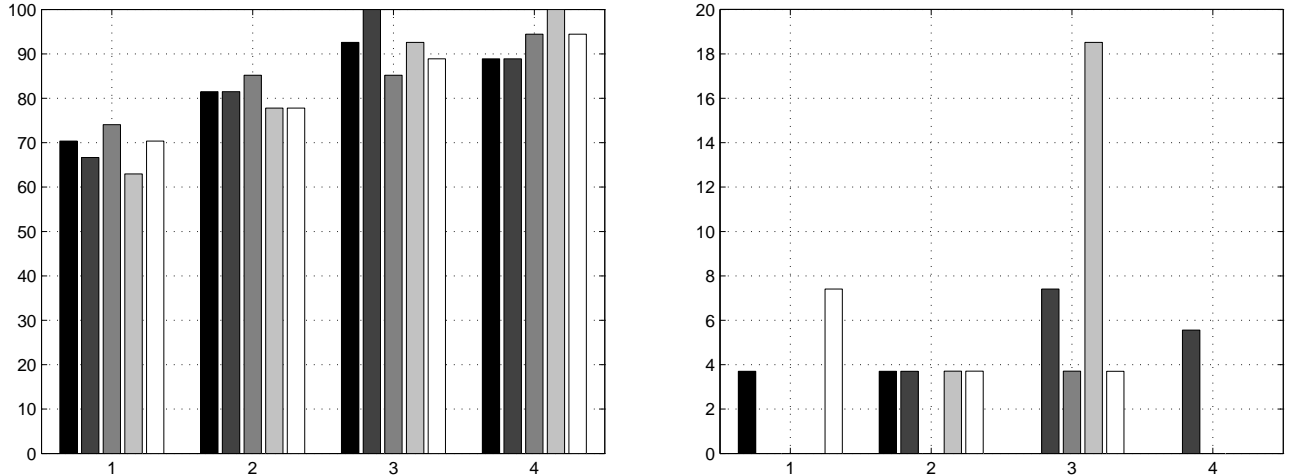


Fig. 14. Effect of the size of the training set of the pullback learning procedure on performance improvement. Left: results on the Weizmann dataset. A number of training sequences ranging from 9 to 27, to 45, to 90 was considered (1,2,3,4 on the abscissa). 27 test sequences, five runs per size. Obtained for pullback Frobenius distance, 11 samples in the output diffeomorphism F_C , transition diffeomorphism on the first column of A only, 10 samples. Right: corresponding improvements in the classification rates.

* METRIC/DISTANCE

* OBJECTIVE FUNCTION, ANALYTICAL CLASS PERF?

In this paper we proposed a differential-geometric framework for manifold learning given a dataset 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 [36], [21]. The extension to another popular class of stochastic model, hidden Markov models [17] 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 a-priori knowledge on the training set such as similarity relations [53] is highly desirable, and will

A. Fisher metric for other classes of multidimensional systems

In the case of multi-dimensional linear systems of a number of different classes, both Fisher metric and its geodesics can still be computed by means of an iterative numerical scheme [36], [21]. The extension of the pullback manifold learning scheme to multi-dimensional systems is then straightforward. In any case, using the Fisher information metric as basis Riemannian metric is not mandatory. We can as easily adopt the standard Euclidean metric as initial distance, and build a family of pullback Euclidean metrics to optimize upon.

B. Extension to similarity relations

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. In *ICML03*, pages 11–18, 2003.
- [3] M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. *MLJ*, 56:209–239, 2004.
- [4] M. Bilenko, S. Basu, and R. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proc. of ICML'04*, 2004.
- [5] A. Bissacco, A. Chiuso, and S. Soatto. Classification and recognition of dynamical models: The role of phase, independent components, kernels and optimal transport. *PAMI*, 29(11):1958–1972, 2007.
- [6] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. 2005.
- [7] P. Burman. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76(3):503–514, 1989.
- [8] N. Carter, D. Young, and J. Ferryman. Supplementing markov chains with additional features for behavioural analysis. pages 65–65, 2006.
- [9] A. Casile and M. Giese. Critical features for the recognition of biological motion. *J Vision*, 5:348–360.
- [10] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *Proc. of CVPR'09*, pages 1932–1939, 2009.

- [11] F. Cuzzolin. Learning pullback metrics for linear models. In *Workshop on Machine Learning for Vision-based Motion Analysis MLVMA*, 2008.
- [12] N. Dalai and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 886–893, 2006.
- [13] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Statistical Society B*, 39.
- [14] M. Do. Fast approximation of Kullback-Leibler distance for dependence trees and hidden Markov models. *IEEE Signal Processing Letters*, 10(4):115 – 118, 2003.
- [15] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *Int. J. Comput. Vision*, 51(2):91–109, 2003.
- [16] C. Eick, A. Rouhana, A. Bagherjeiran, and R. Vilalta. Using clustering to learn distance functions for supervised similarity assessment. In *ICML and Data Mining*, 2005.
- [17] R. Elliot, L. Aggoun, and J. Moore. *Hidden Markov models: estimation and control*. Springer Verlag, 1995.
- [18] R. Gross and J. Shi. The CMU motion of body (Mobo) database. Technical report, CMU, 2001.
- [19] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *CVPR*, pages 2012–2019. IEEE, 2009.
- [20] D. Han, L. Bo, and C. Sminchisescu. Selection and context for action recognition. In *Proc. of ICCV'09*, 2009.
- [21] B. Hanzon and R. Peeters. Aspects of Fisher geometry for stochastic linear systems. In *Open Problems in Mathematical Systems and Control Theory*, pages 27–30. 2002.
- [22] N. Ikizler-Cinbis, R. Cinbis, and S. Sclaroff. Learning actions from the web. In *Proc. of ICCV'09*, 2009.
- [23] M. Itoh and Y. Shishido. Fisher information metric and poisson kernels. *Differential Geometry and its Applications*, 26(4):347 – 356, 2008.
- [24] T. Kim and R. Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. 31(8):1415–1428, August 2009.
- [25] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Math. Stat.*, 22:79–86, 1951.
- [26] I. Laptev and T. Lindeberg. Local descriptors for spatiotemporal recognition. 2003.
- [27] G. Lebanon. Metric learning for text documents. *PAMI*, 28(4):497–508, 2006.
- [28] L. Lee and W. Grimson. Gait analysis for recognition and classification. In *AFGR'02*, pages 155–162, 2002.
- [29] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos 'in the wild'. pages 1996–2003, 2009.
- [30] C. Loy, T. Xiang, and S. Gong. Modelling activity global temporal dependencies using time delayed probabilistic graphical model. In *Proc. of ICCV'09*, 2009.
- [31] C. Loy, T. Xiang, and S. Gong. Multi-camera activity correlation analysis. pages 1988–1995, 2009.

- [32] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. *CVPR'09*, 2009.
- [33] R. J. Martin. A metric for ARMA processes. *IEEE Trans. on Signal Processing*, 48(4):1164–1170, April 2000.
- [34] M. Murray and J. Rice. *Differential Geometry and Statistics*. CRC Press, 1993.
- [35] M. Naito, L. Deng, and Y. Sagisaka. Speaker clustering for speech recognition using vocal tract parameters. *Speech Communication*, 36(3-4), 2002.
- [36] R. Peeters and B. Hanzon. On the Riemannian manifold structure of classes of linear systems. In *Equadiff*, 2003.
- [37] M. Piccardi and O. Perez. Hidden markov models with kernel density estimation of emission probabilities and their use in activity recognition. pages 1–8, 2007.
- [38] G. Pitis. On some submanifolds of a locally product manifold. *Kodai Math. J.*, 9(3):327–333, 1986.
- [39] A. Rijkeboer. *Differential geometric models for time-varying coefficients of autoregressive processes*. PhD thesis, Tilburg University, 1994.
- [40] G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P. Torr. Randomized trees for human pose detection. In *CVPR'08*.
- [41] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
- [42] K. Schindler and L. van Gool. Action snippets: How many frames does human action recognition require? In *Proc. of CVPR'08*, 2008.
- [43] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *NIPS*. 2004.
- [44] N. Shental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis. In *ECCV'02*, 2002.
- [45] Q. Shi, L. Wang, L. Cheng, and A. Smola. Discriminative human action segmentation and recognition using semi-markov model. pages 1–8, 2008.
- [46] A. Smola and S. Vishwanathan. Hilbert space embeddings in dynamical systems. In *Proc. of IFAC'03*, pages 760 – 767, August 2003.
- [47] J. Sun, X. Wu, S. Yan, L. Cheong, T. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. pages 2004–2011, 2009.
- [48] A. Sundaesan, A. Roy Chowdhury, and R. Chellappa. A hidden markov model based framework for recognition of humans from gait sequences. pages II: 93–96, 2003.
- [49] I. Tsang, J. Kwok, C. Bay, and H. Kong. Distance metric learning with kernels. In *Proceedings of the International Conference on Artificial Intelligence*, 2003.
- [50] Y. Wang and G. Mori. Human action recognition by semilattent topic models. *PAMI*, 31(10):1762–1774, October 2009.
- [51] Y. Wang and G. Mori. Max-margin hidden conditional random fields for human action recognition. pages 872–879, 2009.
- [52] L. Xie, A. Ugrinovskii, and I. Petersen. Probabilistic distances between finite-state finite-alphabet hidden Markov models. In *Proc. of CDC'03*, pages 5347–5352, 2003.

- [53] E. Xing, A. Ng, M. Jordan, and S. Russel. Distance metric learning with applications to clustering with side information. In *NIPS'03*. 2003.
- [54] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. pages 2442–2449, 2009.
- [55] G. Zames and A. K. El-Sakkary. Unstable systems and feedback: The gap metric. In *Proc. 18th Allerton Conference on Communications, Control, and Computers*, pages 380–385, Urbana, IL, October 1980.
- [56] Z. Zhang. Learning metrics via discriminant kernels and multidimensional scaling: Toward expected euclidean representation. In *ICML'03*, Hong Kong, 2003.