Adaptive filtering for estimation of a low-rank positive semidefinite matrix

S. Bonnabel, G. Meyer and R. Sepulchre

Abstract— **In this paper, we adopt a geometric viewpoint to tackle the problem of estimating a linear model whose parameter is a fixed-rank positive semidefinite matrix. We consider two gradient descent flows associated to two distinct Riemannian quotient geometries that underlie this set of matrices. The resulting algorithms are non-linear and can be viewed as a generalization of Least Mean Squares that instrically constrain the parameter within the manifold search space. Such algorithms designed for low-rank matrices find applications in high-dimensional distance learning problems for classification or clustering.**

I. INTRODUCTION

In this paper, we adopt a differential geometry viewpoint to tackle the problem of estimating a model whose parameter is a fixed-rank positive semidefinite matrix. Given data $\mathbf{x} \in \mathbb{R}^d$ and observations $y \in \mathbb{R}$, the problem amounts to identify the unknown parameter $W \in S_+(r,d)$ in the linear model

$$y(t) = \mathrm{Tr}\,(WX(t)) + \nu(t) \qquad (1)$$

over the nonlinear search space

$$S_+(r,d) = \{W \in \mathbb{R}^{d\times d} \text{ s.t. } W = W^T \succeq 0,\ \mathrm{rank}(W) = r\}$$

that represents the set of rank-r positive semidefinite matrices; $X(t)$ is a symmetric input matrix, and $\nu(t)$ is an observation noise.

An important application that motivates the problem of interest is the learning of a distance function between data samples. This task is a central issue for many machine learning applications where a data-specific distance has to be constructed, or where an existing distance needs to be improved based on additional side information [1], [2], [3], [4], [5]. When the distance is represented as a kernel function or as a Mahalanobis distance, it follows a quadratic model parameterized by a positive semidefinite matrix, and the learning problem can be formulated as an estimation of W from a sequence of observations (1), where X is a rank-one positive semidefinite matrix, and $\nu(t)$ represents classification uncertainties.

Low-rank learning has attracted considerable interest in the recent literature. Learning low-rank matrices is a typical solution to reduce the computational cost of subsequent algorithms. Indeed, the complexity generally decreases from $O(d^3)$ to $O(dr^2)$ where the approximation rank r is generally much smaller than the problem size d. One obvious solution is to fix the range space and to apply algorithms developed for the full rank case in that subspace [5], [6]. This amounts to decouple the data reduction problem from the distance learning problem. Recently, we proposed a general framework for the simultaneous learning of the subspace and of the (low-rank) distance. In short, the approach consists to generalize the stochastic gradient learning framework to the Rieamannian manifold of fixed-rank positive semidefinite matrices. The approach recovers several existing algorithms known for full-rank distance learning and allows for a smooth extension to rank-deficient distance learning problems [7].

In this paper, we consider the continuous-time formulation of the algorithms proposed in [7] in order to explore connections with the framework of invariant observer design. Observers can be used for parameter estimation choosing the parameter space as the state space, and using the dynamical model: $\frac{d}{dt}W = 0$. In this sense, learning algorithms can be regarded as nonlinear observers. The invariant observer design theory is shown to be useful for the practically relevant gain tuning problem. We propose a normalized version of our algorithms with the help of the recent theory of Symmetry-Preserving Observers [8]. This normalized observer is robust to scalings of the inputs and parameters. Such normalized algorithms are easy-to-tune as the learning rate (the gain) is a dimensionless quantity, meaning that a normalized observer with a gain properly tuned can be applied to a large class of problems having a similar signal-to-noise ratio.

The algorithms developed in this paper rely on a Riemannian geometric approach. They automatically maintain the parameter within the search space of interest, scale to high-dimensional problems and enjoy important invariance properties.

In Section II, the considered framework is introduced through well-known examples of Least Mean Squares (LMS) problems, and subspace tracking. In Section III, two algorithms to estimate matrices in $S_+(r,d)$ from a linear model are presented. They rely on two Riemannian geometries that have recently appeared. Section IV presents convergence results. The problem of gain tuning is discussed in Section V. Numerical experiments are presented in Section VI.

This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors. G.M. is supported as an FRS-FNRS research fellow (Belgian Fund for Scientific Research).

S. Bonnabel is with Centre de Robotique, Mathématiques et Systèmes, Mines ParisTech, Boulevard Saint-Michel 60, 75272 Paris, France. silvere.bonnabel@mines-paristech.fr

G. Meyer and R. Sepulchre are with Departement of Electrical Engineering and Computer Science, University of Liège, 4000 Liège, Belgium. {g.meyer,r.sepulchre}@ulg.ac.be

ISBN 978-963-311-370-7

II. Least mean squares

A. Least mean squares on $\mathbb{R}^d$

Let $\mathbf{x}(t) \in \mathbb{R}^d$ be the input vector, and $y(t)$ be the output where

$$y(t) = w^T \mathbf{x}(t) + \nu(t) \tag{2}$$

where the unknown vector $w \in \mathbb{R}^d$ is to be identified (filter weights), and $\nu(t)$ is a noise. LMS filters use steepest descent to find the parameter w which minimizes a cost function. Let $\hat{w}$ be the estimated parameter. Let

$$\hat{y}(t) = \hat{w}^T \mathbf{x}(t)$$

be the estimated output. The cost function is the mean-square output error (MSE)

$$C(t) = \frac{1}{2}\mathbb{E}[(\hat{y}(t) - y(t))^2] \tag{3}$$

where $\mathbb{E}$ denotes the expected value. Applying the steepest descent leads to the following algorithm :

$$\frac{d}{dt}\hat{w} = -\eta \mathbb{E}[(\hat{w}^T \mathbf{x}(t) - y(t))\mathbf{x}(t)], \tag{4}$$

Usually, the update for the LMS algorithm is performed replacing the estimated value $\mathbb{E}[(\hat{y}(t) - y(t))^2]$ by the current value $\hat{y}(t) - y(t))^2$, yielding to the stochastic gradient algorithm applied to the MSE criterion:

$$\frac{d}{dt}\hat{w} = -\eta(\hat{w}^T \mathbf{x}(t) - y(t))\mathbf{x}(t). \tag{5}$$

B. Extension of LMS update to a Riemannian manifold

The least mean squares algorithm (LMS) is a stochastic gradient descent method as the filter is only adapted based on the error at the current time. In this paper we extend LMS algorithms to some Riemannian manifolds. Indeed, when the parameter belongs to a Riemannian manifold, one can always define a stochastic gradient descent the following way:

- define a cost function on the manifold based on the estimation error
- approximate the cost function by the error at current time
- define a Riemannian metric
- compute the gradient on the manifold according to the chosen metric [9].

The next sections illustrate this construction on various nonlinear spaces.

C. LMS on the Grassman manifold

As a preliminary example, we consider revisit the well-known subspace tracking algorithm by Oja [10]. Let $\mathbf{x}(t) \in \mathbb{R}^d$ be the input vector, and $y(t)$ be the output where

$$y(t) = VV^T \mathbf{x}(t) + \nu(t) \tag{6}$$

where the unknown matrix $V \in \mathrm{St}(r,d) = \{V \in \mathbb{R}^{d\times r} \text{ s.t. } V^T V = I\}$. The matrix VV^T is a projector, and it can be identified to a r-dimensional subspace of $\mathbb{R}^d$. Indeed the set of rank r projectors can be identified to the Grassman manifold of r-dimensional subspaces:

$$\mathrm{Gr}(r,d) = \{P \in \mathbb{R}^{d\times d} \text{ s.t. } P^T = P,\ P^2 = P,\ \mathrm{Tr}\,(P) = r\}.$$

The cost function is then

$$C(t) = \frac{1}{2}\mathbb{E}[(UU^T \mathbf{x}(t) - y(t))^2] \tag{7}$$

It is invariant to rotations $U \mapsto UO, O \in \mathcal{O}(r)$. The state-space is therefore the set of equivalence classes

$$[U] = \{UO \text{ s.t. } O \in \mathcal{O}(r)\}.$$

This set is denoted by $St(r,d)/\mathcal{O}(r)$. It is a *quotient representation* of the Grassman manifold $\mathrm{Gr}(r,d)$. The quotient geometry of the Grassman manifold has been well-studied in [14]. The tangent space to $U \in St(r,d)$ can be decomposed into a vertical space, which is tangent to the fiber $[U]$, and a horizontal space, orthogonal to it. The tangent space to $\mathrm{Gr}(r,d)$ at $[U]$ is generally identified to the horizontal space at each representative of $[U]$. The projector on the horizontal space (orthogonal to the fiber) at U is

$$\Pi_U : \Delta \mapsto (I - UU^T)\Delta, \quad \Delta \in \mathbb{R}^{d\times r}$$

The metric

$$g_{[U]}(\xi_{[U]}, \zeta_{[U]}) \triangleq \bar{g}_U(\bar{\xi}_U, \bar{\zeta}_U)$$

is induced by the standard metric in $\mathbb{R}^{d\times r}$,

$$\bar{g}_U(\Delta_1, \Delta_2) = \mathrm{Tr}(\Delta_1^T \Delta_2),$$

which is invariant along the set of equivalence classes. Therefore, the gradient admits the simple horizontal representation

$$\overline{\mathrm{grad} f(U)} = \Pi_U \ \mathrm{grad} f(U), \tag{8}$$

where $\mathrm{grad} f(U)$ is defined by the identity

$$Df(U)[\Delta] = \bar{g}_U(\Delta, \mathrm{grad} f(U)).$$

Applying the steepest descent, replacing the estimated value by the current value, and projecting the correction term onto the horizontal space, leads to the following algorithm :

$$\frac{d}{dt}U = -\eta(I - UU^T)\mathbf{x}\mathbf{x}^T U, \tag{9}$$

which is known as the Oja's vector field for subspace tracking.

III. LMS on the set of positive semidefinite matrices

We now consider a generalization of the problem of Section (II-A) where $X(t) \in \mathbb{R}^{d\times d}$ is an input matrix, $y(t)$ is the output, and the matrix counterpart of the linear noisy model (6) is the linear model

$$y(t) = \mathrm{Tr}\left(W^T X(t)\right) + \nu(t) \tag{10}$$

where the unknown matrix $W \in \mathbb{R}^{d\times d}$ is to be identified (filter weights), and $\nu(t)$ is a noise. From now on, motivated by applications, we will assume that X is a rank-one symmetric positive matrix, and W is a symmetric positive semidefinite matrix. This leads to the following problem:

$$y(t) = \mathrm{Tr}\left(W\mathbf{x}(t)\mathbf{x}(t)^T\right) + \nu(t) = \mathbf{x}(t)^T W\mathbf{x}(t) + \nu(t) \tag{11}$$

where $\mathbf{x} \in \mathbb{R}^d$. In the sequel, we are going to study several cases where the parameter W belongs to a submanifold of the set of positive definite matrices.

The cost function is the mean-square output error (MSE)

$$C(t) = \frac{1}{2}\mathbb{E}[(\hat{y}(t) - y(t))^2] \tag{12}$$

where $\mathbb{E}$ denotes the expected value and is systematically approximated in the sequel by its current value.

A. LMS on the cone of positive definite matrices $S_+(d)$

The quotient geometries of $S_+(d)$ are rooted in the matrix factorization

$$W = GG^T, \quad G \in \mathrm{GL}(d),$$

where $\mathrm{GL}(d)$ is the set of all invertible $d \times d$ matrices. Because the factorization is invariant by rotation, $G \mapsto GO$, $O \in \mathcal{O}(d)$, the search space is identified to the quotient

$$S_+(d) \simeq \mathrm{GL}(d)/\mathcal{O}(d),$$

which represents the set of equivalence classes

$$[G] = \{GO \text{ s.t. } O \in \mathcal{O}(d)\}.$$

We will equip this quotient with two meaningful Riemannian metrics.

1) Flat metric on the square-root factor: The metric on the quotient

$$g_{[G]}(\xi_{[G]}, \zeta_{[G]}) \triangleq \bar{g}_G(\bar{\xi}_G, \bar{\zeta}_G),$$

is induced by the standard metric in $\mathbb{R}^{d\times d}$,

$$\bar{g}_G(\Delta_1, \Delta_2) = \mathrm{Tr}(\Delta_1^T \Delta_2) \tag{13}$$

which is invariant by rotation, that is, along the set of equivalence classes. As a consequence, it induces a metric $g_{[G]}$ on $S_+(d)$. With this geometry, a tangent vector $\xi_{[G]}$ at $[G]$ is represented by a horizontal tangent vector $\bar{\xi}_G$ at G by

$$\bar{\xi}_G = \mathrm{Sym}(\Delta)G, \quad \Delta \in \mathbb{R}^{d\times d},$$

where $\mathrm{Sym}(\cdot)$ extracts the symmetric part of its argument, $\mathrm{Sym}(A) = (A + A^T)/2$. The horizontal gradient of the approximation at time t of (12)

$$f(G) = \frac{1}{2}(\mathrm{Tr}(GG^T\mathbf{x}\mathbf{x}^T) - y)^2, \tag{14}$$

is the unique horizontal vector $\overline{\mathrm{grad} f(G)}$ that satisfies

$$Df(G)[\Delta] = \bar{g}_G(\Delta, \overline{\mathrm{grad} f(G)}).$$

Elementary computations yields

$$\overline{\mathrm{grad} f(G)} = 2(\hat{y} - y)\mathbf{x}\mathbf{x}^T G.$$

Those formulas lead to the online gradient algorithm

$$\frac{d}{dt}G = -\eta(\hat{y} - y)\mathbf{x}\mathbf{x}^T G \tag{15}$$

2) Affine-invariant metric: As the parameter space $S_+(d) \simeq \mathrm{GL}(d)/\mathcal{O}(d)$ is the quotient of two Lie groups, its (reductive) geometric structure can be further exploited. In particular, the natural metric at identity

$$g_I(\xi_I, \zeta_I) = \mathrm{Tr}(\xi_I^T \zeta_I)$$

can be extended to the entire space to satisfy the invariance property

$$g_I(\xi_I, \zeta_I) = g_W(W^{\frac{1}{2}}\xi_I W^{\frac{1}{2}}, W^{\frac{1}{2}}\zeta_I W^{\frac{1}{2}}) = g_W(\xi_W, \zeta_W).$$

The resulting metric on $S_+(d)$ is defined by

$$g_W(\xi_W, \zeta_W) = \mathrm{Tr}(\xi_W W^{-1}\zeta_W W^{-1}). \tag{16}$$

With this geometry, tangent vectors ξ_W are expressed as

$$\xi_W = W^{\frac{1}{2}}\mathrm{Sym}(\Delta)W^{\frac{1}{2}}, \quad \Delta \in \mathbb{R}^{d\times d}$$

The gradient $\mathrm{grad} f(W)$ is given by the identity

$$Df(W)[\Delta] = g_W(\Delta, \overline{\mathrm{grad} f(W)}).$$

Applying this formula to the approximation at current value of (12) yields

$$\overline{\mathrm{grad} f(\hat{W})} = (\hat{y} - y)\hat{W}\mathrm{Sym}(X)\hat{W}. \tag{17}$$

which leads to the following online gradient algorithm:

$$\frac{d}{dt}\hat{W} = -\eta(\hat{y} - y)\hat{W}\mathbf{x}\mathbf{x}^T\hat{W} \tag{18}$$

B. LMS on the set of fixed-rank positive semidefinite matrices

Consider now the problem above where W belongs to the set symmetric semidefinite positive matrices of fixed rank r:

$$S_+(r, d) = \{W \in \mathbb{R}^{d\times d} \text{ s.t. } W = W^T \succeq 0, \mathrm{rank}(W) = r\}.$$

In this paper we propose to derive the LMS algorithm for two different Riemannian metrics of $S_+(r, d)$ corresponding to two parametrizations.

1) Flat metric on the square root factor: The generalization of the results of Section III-A.1 to the set $S_+(r, d)$ is a straightforward consequence of the factorization

$$W = GG^T, \quad G \in \mathbb{R}_*^{d\times r},$$

where $\mathbb{R}_*^{d\times r} = \{G \in \mathbb{R}^{d\times r} \text{ s.t. } \mathrm{rank}(G) = r\}$. The flat quotient geometry of $S_+(d) \simeq \mathrm{GL}(d)/\mathcal{O}(d)$ is generalized to the quotient geometry of $S_+(r, d) \simeq \mathbb{R}_*^{d\times r}/\mathcal{O}(r)$ by a mere adaptation of matrix dimension, leading to the following update:

$$\frac{d}{dt}G = -\eta(\hat{y} - y)\mathbf{x}\mathbf{x}^T G. \tag{19}$$

2) Riemannian metric for a polar factorization of the matrix: In contrast to the flat geometry, the affine-invariant geometry of $S_+(d) \simeq \mathrm{GL}(d)/\mathcal{O}(d)$ does not generalize directly to $S_+(r,d) \simeq \mathbb{R}_*^{d\times r}/\mathcal{O}(r)$ because $\mathbb{R}_*^{d\times r}$ is not a group. A partial generalization is however possible by considering the polar matrix factorization:

$$G = UR, \quad U \in \mathrm{St}(r,d),\ R \in S_+(r).$$

It is obtained from the singular value decomposition of $G = Z\Sigma V^T$ as $U = ZV^T$ and $R = V\Sigma V^T$ [13]. The polar parametrization

$$W = UR^2U$$

leads to the quotient representation

$$S_+(r,d) \simeq (\mathrm{St}(r,d) \times S_+(r))/\mathcal{O}(r), \tag{20}$$

based on the invariance of W to the transformation $(U,R^2) \mapsto (UO, O^TR^2O)$, $O \in \mathcal{O}(r)$. It thus describes the set of equivalence classes

$$[(U,R^2)] = \{(UO, O^TR^2O) \text{ s.t. } O \in \mathcal{O}(r)\}.$$

The cost function is now given by

$$f(U,R^2) = \frac{1}{2}(\mathrm{Tr}(UR^2U^T\mathbf{x}\mathbf{x}^T) - y)^2. \tag{21}$$

The Riemannian geometry of (20) has been recently studied [11]. A tangent vector $\xi_{[W]} = (\xi_U, \xi_{R^2})_{[U,R^2]}$ at $[U,R^2]$ is described by a horizontal tangent vector $\bar{\xi}_W = (\bar{\xi}_U, \bar{\xi}_{R^2})_{(U,R^2)}$ at (U,R^2) by

$$\bar{\xi}_U = \Pi_U\Delta,\ \Delta \in \mathbb{R}^{d\times r}, \quad \bar{\xi}_{R^2} = R\mathrm{Sym}(\Psi)R,\ \Psi \in \mathbb{R}^{r\times r}.$$

The metric

$$\begin{aligned} g_{[W]}(\xi_{[W]}, \zeta_{[W]}) &\triangleq \bar{g}_W(\bar{\xi}_W, \bar{\zeta}_W) \qquad (22)\\ &= \frac{1}{\lambda}\ \bar{g}_U(\bar{\xi}_U, \bar{\zeta}_U) + \frac{1}{1-\lambda}\ \bar{g}_{R^2}(\bar{\xi}_{R^2}, \bar{\zeta}_{R^2}), \end{aligned}$$

where $\lambda \in (0,1)$, is induced by the normal metric of $\mathrm{St}(r,d)$ and the affine-invariant metric of $S_+(r)$,

$$\begin{aligned} \bar{g}_U(\Delta_1, \Delta_2) &= \mathrm{Tr}(\Delta_1^T\Delta_2),\\ \bar{g}_{R^2}(\Psi_1, \Psi_2) &= \mathrm{Tr}(\Psi_1R^{-2}\Psi_2R^{-2}). \end{aligned}$$

It is invariant along the set of equivalence classes and thus induces a quotient Riemannian structure on $S_+(r,d)$. For the sake of simplicity of notation, let $B =: R^2 \in S_+(r)$. Computing the gradient of (21) with the Riemannian metric leads to the update:

$$\begin{aligned} \frac{d}{dt}U &= -2\eta\lambda(\hat{y} - y)(I - UU^T)\mathbf{x}\mathbf{x}^TUB, \qquad (23)\\ \frac{d}{dt}B &= -\eta(1-\lambda)(\hat{y} - y)BU^T\mathbf{x}\mathbf{x}^TUB. \qquad (24) \end{aligned}$$

C. Connection to subspace tracking and discussion

Using a polar decomposition of W, the model (11) can be viewed as a model whose parameter is a positive definite matrix of rank r and where the data $x \in \mathbb{R}^d$ are projected in the subspace span(W). A proper tuning of the parameter λ in the definition of metric (22) allows to place more emphasis either on identifying the subspace span(W), or on identifying the positive definite matrix B in that subspace. In the case $\lambda = 1$, the algorithm only performs subspace learning. Conversely, in the case $\lambda = 0$, the algorithm tries to identify a rank-r matrix in a fixed subspace of reduced dimension (all the data are projected via the projector UU^T). The problem of identifying a positive definite matrix from observations (11) corresponds to a distance learning problem that has many applications (see [1], [2], [3], [4], [5]). Thus, intermediate values of λ continuously interpolates between the subspace learning problem and the distance learning problem at fixed range space. In particular, it is interesting to note that when the matrix to be identified W is a projector, and a projector is seeked (i.e. $\lambda = 1, B = I_r$), algorithm (23) writes

$$\frac{d}{dt}U = -2\eta\lambda(\hat{y} - y)(I - UU^T)\mathbf{x}\mathbf{x}^TU,$$

and it can be viewed as a generalization of the Oja's subspace tracking algorithm of Section II-C to the case where the model is $y = \mathbf{x}^TVV^T\mathbf{x} = \|V^T\mathbf{x}\|^2$ (instead of $\mathbf{y} = VV^T\mathbf{x}$).

IV. Convergence issues

The parameter estimation problem from the model (11) is a linear problem. When using either polar or square-root parameterizations to enforce the rank constraint, it becomes nonlinear and nonconvex. Experimentally, both algorithms (19) and (23)-(24) are well-behaved, and their convergence properties only depend on the distribution of the inputs. In this section, we consider a simplified version of the problem of Section III. We assume that the cost function (12) is available, and that the input matrices X are generated by Gaussian vectors $\mathbf{x}$ with a zero mean and an identity covariance matrix. The proof of convergence on this simplified problem suggests the good behavior of algorithms observed in practice.

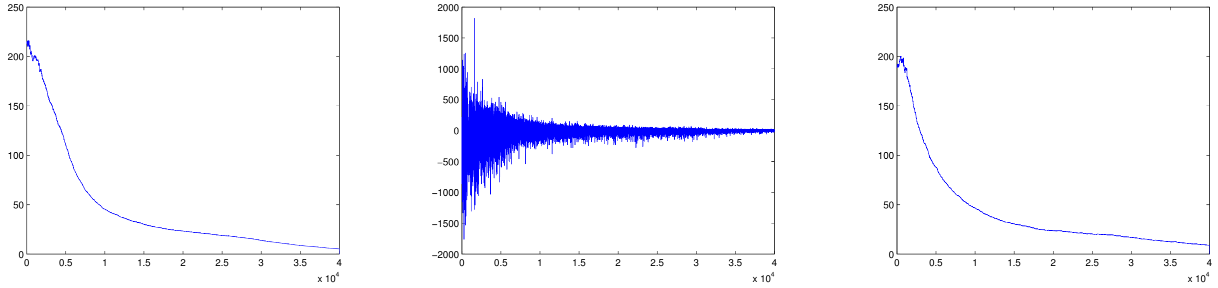
A. Algorithm based on the flat metric:

Proposition 1: Let $\mathbf{x}$ be a Gaussian vector with zero mean and identity covariance matrix. Consider the model (11) with noise turned off (no interference). Then the algorithm $\dot{G} = -\mathbb{E}(\hat{y} - y)\mathbf{x}\mathbf{x}^TG$ asymptotically converges to a fixed matrix G_∞ satisfying:

$$\forall \mathbf{x} \in \mathrm{span}(G_\infty) \quad \mathrm{Tr}\left(\mathbf{x}\mathbf{x}^TW\right) - \mathrm{Tr}\left(\mathbf{x}\mathbf{x}^TG_\infty G_\infty^T\right) = 0 \tag{25}$$

This proposition proves that if the outputs are generated by a matrix W of rank r, and if G has converged to a matrix of rank r indeed, then necessarily GG^T asymptotically converges to W.

Proof: The gradient algorithm $\dot{G} = -\mathbb{E}(\hat{y} - y)\mathbf{x}\mathbf{x}^TG$ converges to a critical point of the gradient, i.e. $\dot{G} = 0$. Writing $\hat{y} - y = \mathrm{Tr}\left(\mathbf{x}\mathbf{x}^T(GG^T - W)\right)$ we see



(a) (b) (c)

Fig. 1. (a): $\|GG^T - W\|_2$ versus number of iterations for algorithm (19) with model (11) and noise turned off $\nu = 0$. (b) $\hat{y} - y$ versus number of iterations for algorithm (19) with model (11) and noise turned off $\nu = 0$. (c) $\|GG^T - W\|_2$ versus number of iterations for algorithm (19) with model (11) and a Gaussian white noise of amplitude 10% of the mean value of y.

that span(G_∞) is included in the kernel of the matrix $\mathbb{E}$ Tr $\left(\mathbf{x}\mathbf{x}^T(G_\infty(G_\infty)^T - W)\right)\mathbf{x}\mathbf{x}^T$. Consider the (matrix) map:

$$M \mapsto \mathbb{E}(\mathrm{Tr}\left(\mathbf{x}\mathbf{x}^T M\right)\mathbf{x}\mathbf{x}^T)$$

It is a linear map. As for any $1 \leq i, j \leq d$ with $i \neq j$ the coordinates of a Gaussian vector with null mean and variance 1 satisfy $\mathbb{E}(x^i x^j) = 0$, $\mathbb{E}((x^i)^2) = 1$ and $\mathbb{E}((x^i)^4) = 3$ we can prove easily that

$$\mathbb{E}(\mathrm{Tr}\left(\mathbf{x}\mathbf{x}^T M\right)\mathbf{x}\mathbf{x}^T) = M + M^T + \mathrm{Tr}\,(M)\,Id$$

Thus $\mathbb{E}(\mathrm{Tr}\left(\mathbf{x}\mathbf{x}^T M\right)\mathbf{x}\mathbf{x}^T) = 0$ implies that $M = 0$. As a consequence, $\mathrm{Tr}\left(\mathbf{x}\mathbf{x}^T(G_\infty G_\infty^T - W)\right) = 0$ for $\mathbf{x} \in$ span$(G_\infty G_\infty^T)$. ■

B. Algorithm based on the polar decomposition

Proposition 2: Let $\mathbf{x}$ be a Gaussian vector with zero mean and identity covariance matrix. Consider the model (11) with noise turned off. Then the algorithm $\dot{U} = -2\lambda\mathbb{E}((\hat{y}-y)(I - UU^T)\mathbf{x}\mathbf{x}^T UB)$, $\dot{B} = -(1-\lambda)\mathbb{E}((\hat{y}-y)BU^T\mathbf{x}\mathbf{x}^T UB)$ asymptotically converges to fixed matrices U_∞, B_∞ wich are such that for any $\mathbf{x} \in$ span(U) we have $\mathrm{Tr}\left(\mathbf{x}\mathbf{x}^T W\right) - \mathrm{Tr}\left(\mathbf{x}\mathbf{x}^T U_\infty B_\infty U_\infty^T\right) = 0$.

Proof: The proof holds by the same token as in the square-root algorithm case. ■

This proposition proves that if the outputs are generated by a matrix W (of any rank), the matrices W and UBU^T coincide on the span of U (subspace of dimension r).

V. NLMS on the set of fixed-rank positive semidefinite matrices

A practically important limitation of the LMS algorithm is that it is sensitive to the scaling of the input $x(t)$. In practice it is very hard to find a gain η that guarantees stability of the algorithm [15]. The Normalised least mean squares filter (NLMS) is a variant of the LMS algorithm that avoids this problem by normalizing with the power of the input. Such a normalization allows the gain to be dimensionless, and thus makes the tuning more robust to changes of the input magnitude over the time, as well as changes of units or types of data. The NLMS algorithm on $\mathbb{R}^d$ can be written as a modification of update (5) in order to make it invariant to scalings:

$$\frac{d}{dt}\hat{w} = -\eta(\hat{w}^T\mathbf{x}(t) - y(t))\frac{\mathbf{x}(t)}{\mathbf{x}(t)^T\mathbf{x}(t)}. \qquad (26)$$

The analog of a NLMS algorithm for fixed-rank positive semidefinite matrices is highly desirable: in distance learning applications considered in [7], the algorithm must cope with a wide variety of types of data, that sometimes do not have physical units (binary data etc.). Making the algorithms robust to such heterogeneous types of data is therefore an issue of practical importance. However, when the learning algorithm is nonlinear, normalizing with the power of the input does not guarantee that the gain η becomes dimensionless. To achieve the analog robustness property for the nonlinear algorithms considered in this paper, we use the theory of symmetry-preserving observers, that indicates how to achieve invariance properties for a given nonlinear observer [8].

A. Homogeneous functions and symmetries of the problem

We use the notations of [16]. For fixed $\alpha, \beta > 0$, consider the family of transformations

$$g^s(W, x) = (e^{\alpha s}W, e^{\beta s}x)$$

that correspond to changes of units on the input and output data. The output is a homogeneous function of degree $\gamma = \alpha + 2\beta$ since

$$h(g^s(W, x)) = e^{\gamma s}h(W, x).$$

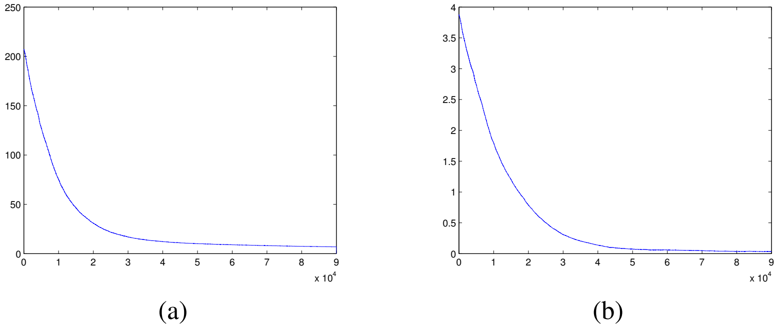
The observer equations (19) or (23)-(24) have the form

$$\frac{d}{dt}\hat{W} = f(\hat{W}, x, \hat{y}, y, \eta).$$

where $\hat{W}$ is the estimated matrix and $\hat{y} = h(\hat{W}, x)$. We would like the gain tuning to be insensitive to changes of units, which means in the terminology of [8] that the observer should be invariant to the transformations above, i.e.:

$$\frac{d}{dt}(e^{\alpha s}\hat{W}) = e^{\alpha s}f(g^s(\hat{W}, x), h(g^s(\hat{W}, x)), h(g^s(W, x)), \eta). \qquad (27)$$

It means that the dynamical behaviour of the observer (and therefore the gain tuning) will not be affected by a change of



(a) (b)

Fig. 2. (a) $\|UBU^T - W\|_2$ versus number of iterations for algorithm (23)-(24) with model (11) and noise turned off $\nu = 0$. (b) $\|UU^T - \Pi_W\|_2$ versus number of iterations for algorithm (23)-(24) with model (11) and noise turned off $\nu = 0$.

units. Indeed it we let $V = e^{\alpha s} W, \hat{V} = e^{\alpha s} \hat{W}$ and $X, \hat{Y}, Y$ represent the variables $x, \hat{y}, y$ in expressed in the new units, the equation of such an observer writes:

$$\frac{d}{dt}\hat{V} = f(\hat{V}, X, \hat{Y}, Y, \eta).$$

In other words, the gains of an invariant observer can be tuned independently of the data units.

In the case of observer (19) the group action is: $g^s(G, x) = (e^{\alpha s} G, e^{\beta s} x)$. In the case of observer (23)-(24) the group action is: $g^s(U, B, x) = (U, e^{\alpha s} G, e^{\beta s} x)$. Note that U is not affected by a change of units since it is an orthonormal basis representing a subspace.

B. Symmetry-preserving observers

Such an invariance to scalings was treated in [8] in a simpler case. As a recap of the whole theory of invariant observers goes beyond the scope of this paper, we briefly recall the necessary ingredients in order to "invariantize" the observer and make it robust to change of units. In order to build such an observer we need

- Invariant output error : $E = \hat{y}/y$. It is unaffected by a change of units $y \mapsto e^{ds} y$.
- Scalar invariants I: for any fixed i, j, a complete set of scalar invariants is given by the coordinates of $\mathbf{x}/\mathbf{x}^i$ where i denotes the i-th coordinate, and by the coordinates of W/W^{ij} where ij denotes the ij-th entry of the matrix.
- Invariant frame : a set of n^2 vector fields $w_i(W, \mathbf{x})$ such that $w_i(e^{\alpha s} W, e^{\beta s}\mathbf{x}) = e^{\alpha s} w_i(W, \mathbf{x})$.

If I denotes the set of scalar invariants, the theory [8] states that *any* observer which is invariant to changes of units defined above writes

$$\frac{d}{dt}\hat{W} = \sum_{i=1}^{n^2} \mathcal{L}_i(E, I) w_i(W, \mathbf{x}). \tag{28}$$

C. Examples of NLMS algorithms

Let us apply this theory to the observers above in order to define a NLMS version of them. Indeed note that $\mathbf{x}/\|\mathbf{x}\|$ is a function of the set of scalar invariants I. Thus one way to invariantize observer (19) above so that it has the form (28) is:

$$\frac{d}{dt}G = -\log(\hat{y}/y)\frac{\mathbf{x}\mathbf{x}^T}{\|\mathbf{x}\|^2} G. \tag{29}$$

Similarly, observer (23)-(24) becomes:

$$\begin{aligned} \frac{d}{dt}U &= -2\lambda \log(\hat{y}/y)(I - UU^T)\frac{\mathbf{x}\mathbf{x}^T}{\|\mathbf{x}\|^2} U \frac{B}{\|B\|^2}, \quad (30) \\ \frac{d}{dt}B &= -(1-\lambda)\log(\hat{y}/y)\frac{B}{\|B\|^2} U^T \frac{\mathbf{x}\mathbf{x}^T}{\|\mathbf{x}\|^2} UB. \quad (31) \end{aligned}$$

The gain tuning of such observers suits a larger class of types of data. From the theory we have a guarantee that such algorithms are insensitive to the scalings defined above.

VI. Simulations

In this section, we generate input vectors in discrete time $\mathbf{x} \in \mathbb{R}^{50}$ that are Gaussian vectors with zero mean and identity covariance matrix. The output is generated via model (11) where $W \in \mathbb{R}^{50 \times 50}$ is a matrix of rank $r = 9$. The results are given in the following figures. We see that both algorithms perform well, and algorithm (19) converges twice as fast as algorithm (23)-(24). In particular we see on Fig.2a that UU^T converges to the projector Π_W on the span of W, and thus allows to track the subspace span(W) simultaneously with the identification of W itself. It was also verified experimentally that the normalized algorithms of Section V-C are totally insensitive to arbitrary scalings of both the inputs and the parameter W to be identified.

References

[1] K. Tsuda, G. Ratsch, and M. Warmuth. Matrix exponentiated gradient updates for on-line learning and bregman projection. *Journal of Machine Learning Research*, 6:995–1018, 2005.

[2] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press, Cambridge, MA, 2002.

[3] S. Shalev-Shwartz, Y. Singer, and A. Ng. Online and batch learning of pseudo-metrics. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, 2004.

[4] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, 2007.

[5] B. Kulis, M. Sustik, and I. S. Dhillon. Low-rank kernel learning with bregman matrix divergences. *Journal of Machine Learning Research*, 10:341–376, 2009.

[6] J. V. Davis and I. S. Dhillon. Structured metric learning for high dimensional problems. In *Proceedings of the 14th ACM SIGKDD conference on Knowledge Discovery and Data Mining*, 2008.

[7] G. Meyer, S. Bonnabel and R. Sepulchre. Regression on fixed-rank positive semidefinite matrices: a geometric approach. *Submitted*, 2010.

[8] S. Bonnabel, Ph. Martin, and P. Rouchon. Symmetry-preserving observers. *IEEE Trans. Automatic Control*, 53(11) 2514- - 2526, 2008.

[9] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

[10] E. Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, 5:927–935, 1992.

[11] S. Bonnabel and R. Sepulchre. Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1055–1070, 2009.

[12] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization for semidefinite convex problems. *SIAM Journal on Matrix Analysis and Applications (in press)*, 2010.

[13] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.

[14] A. Edelman, T.A. Arias, and S.T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[15] S. Haykin. *Adaptive Filter Theory.* Prentice Hall, 2002.

[16] V.I. Arnold. *Geometrical Methods in the Theory of Ordinary Differential Equations* Springer-Verlag, 1983.