FROM SUBSPACE LEARNING TO DISTANCE LEARNING: A GEOMETRICAL OPTIMIZATION APPROACH

Gilles Meyer[*]*, Michel Journée*[*]*, Silvère Bonnabel*[†] *and Rodolphe Sepulchre*[*]

[*]Dept. of Electrical Engineering and Computer Science
University of Liège, Belgium
{g.meyer,m.journee,r.sepulchre}@ulg.ac.be

[†]CAOR, Mathématiques et Systèmes
Mines ParisTech, France
silvere.bonnabel@mines-paristech.fr

ABSTRACT

In this paper, we adopt a differential-geometry viewpoint to tackle the problem of learning a distance online. As this problem can be cast into the estimation of a fixed-rank positive semidefinite (PSD) matrix, we develop algorithms that exploits the rich geometry structure of the set of fixed-rank PSD matrices. We propose a method which separately updates the subspace of the matrix and its projection onto that subspace. A proper weighting of the two iterations enables to continuously interpolate between the problem of learning a subspace and learning a distance when the subspace is fixed.

***Index Terms*—** Kernel and metric learning, low-rank approximation, online learning, manifold-based optimization.

1. INTRODUCTION

The choice of an appropriate distance measure between data objects is a central issue for many classification and clustering algorithms. As this choice depends strongly on the application of interest, algorithms have been proposed to learn a distance from data [1, 2, 3, 4, 5, 6, 7]. When this distance is represented as a kernel function or a Mahalanobis distance, the problem reduces to learning a positive semidefinite (PSD) matrix.

Since most classification and clustering algorithms poorly scale as $O(n^3)$ in the problem size, which prevents their use in a growing number of large-scale problems, recent research has been devoted to the learning of distances represented by low-rank PSD matrices. This reduces the complexity to $O(nr^2)$ where r is the rank of the matrix.

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. GM and MJ are supported as an FNRS research fellow (Belgian Fund for Scientific Research).

Whereas the full-rank case amounts to solve a convex optimization problem on the set of PSD matrices [1, 2, 3, 5], the convexity is lost and local minima are introduced as soon as the rank is constrained. To circumvent that problem, existing algorithms first project the data on a r-dimensional subspace, and then solve a convex problem for a full-rank PSD matrix of dimension r-by-r [4, 6]. Poor results might however be obtained if an inappropriate subspace is chosen in the first place.

In this paper, we thus discuss a gradient-descent method to learn a fixed-rank PSD matrix W that leaves its range space free to evolve over time. Besides learning *simultaneously* the subspace spanned by W and its projection onto that subspace, the proposed method results in *separate* iterations for these two tasks. This viewpoint is inspired by the recent paper [8], which treats the set of fixed-rank PSD matrices

$$S_+(r,n) = \{W \in I\!R^{n\times n} : W \succeq 0,\ \mathrm{rank}(W) = r\},$$

as product of the Grassmann manifold $\mathrm{Gr}(r,n)$ and the cone of strictly positive definite matrices $S_+(r,r)$. Each iteration of the proposed algorithm involves two distinct updates: one on $\mathrm{Gr}(r,n)$ and one on $S_+(r,r)$. We furthermore introduce a weighting factor on these iterations in order to continuously interpolate between the subspace learning problem (defined solely on $\mathrm{Gr}(r,n)$) and the distance learning at fixed range space problem (defined solely on $S_+(r,r)$). Tuning of this parameter puts more emphasis on the former or the latter problem. Such tuning can be of interest when for instance a good estimate of the subspace is already available.

The paper is organized as follows. We first expose the distance learning problem as an optimization on PSD matrices (Section 2). We then discuss the geometry of fixed-rank PSD matrices as the product of the Grassmann manifold with the cone of positive definite matrices (Section 3). Gradient-descent algorithms are proposed in Section 4, and their connection with the subspace learning and full-rank distance learning problems are highlighted in Section 5.

2. ONLINE DISTANCE LEARNING PROBLEM

The problem of learning a distance function from sequentially received information about distances between data points has received considerable attention in the recent years [2, 5, 7]. We summarize the exposition of [9].

The problem is formulated as the computation of a PSD matrix W that minimizes the expected loss

$$L(W) = \mathbb{E}[l(\mathbf{y}, W)], \tag{1}$$

where $l(\mathbf{y}, W)$ is a loss function that quantifies the discrepancy between an observed distance y and its estimation with respect to a model parameterized by W, e.g.,

$$\hat{y} = \mathbf{z}^T W \mathbf{z}, \tag{2}$$

where $\mathbf{z}$ is given. Online learning means that each update of the model W is determined from a new pair $\mathbf{y}_t = (\mathbf{z}_t, y_t)$ which specifies a target value y_t and a vector $\mathbf{z}_t$ that is used to compute the estimate $\hat{y}_t$. A quadratic loss function is given by

$$l(\mathbf{y}, W) = (\mathbf{z}^T W \mathbf{z} - y)^2. \tag{3}$$

Inequalities $\hat{y} \leq y$ and $\hat{y} \geq y$ can be easily handled by treating them as equalities when they are not satisfied.

The distance model (2) is often used in kernel-based methods [10], which transform the data samples $\mathbf{x}_1, ..., \mathbf{x}_n \in I\!R^d$ by a nonlinear mapping ϕ in order to facilitate pattern detection and data analysis. A kernel function κ is defined as the dot product between any two elements in the new feature space,

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle,$$

and is in practice encoded by an n-by-n PSD matrix K such that the entry K_{ij} equals $\kappa(\mathbf{x}_i, \mathbf{x}_j)$. This inner product information is used solely to compute distances in the feature space,

$$\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 = \mathbf{z}^T K \mathbf{z},$$

where $\mathbf{z}$ is a n-dimensional vector having components $z_i = 1$, $z_j = -1$ and zeros everywhere else. In this example, each new observation corresponds to a new pairwise distance in the feature space. The distance model (2) is also compatible with the Mahalanobis distance

$$d_A(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T A (\mathbf{x}_i - \mathbf{x}_j),$$

where A is a d-by-d PSD matrix. See [5] for details.

3. GEOMETRY OF FIXED-RANK PSD MATRICES

We first present the geometry of fixed-rank PSD matrices discussed in [8]. The set of fixed-rank PSD matrices,

$$S_+(r, n) = \{W \in I\!R^{n \times n} : W \succeq 0, \ \mathrm{rank}(W) = r\},$$

has a rich Riemannian geometry structure that can be exploited for algorithmic purposes. Because of its smooth manifold structure, virtually every unconstrained optimization algorithm in $I\!R^n$ can be extended to the set $S_+(r, n)$ [11].

Given a matrix $W \in S_+(r, n)$, this paper focus on the matrix factorization

$$W = UBU^T, \tag{4}$$

where U is an element of the Stiefel manifold

$$\mathrm{St}(n, r) = \{U \in I\!R^{n \times r} : U^T U = I\},$$

and $B \in S_+(r, r)$. The parameterization (4) is invariant with respect to the group action

$$(U, B) \mapsto (UO, O^T B O),$$

for any element O of the orthogonal group

$$\mathcal{O}(r) = \{O \in I\!R^{r \times r} : O^T O = I\}.$$

Hence, the set $S_+(r, n)$ admits the quotient manifold representation

$$S_+(r, n) \simeq (\mathrm{St}(n, r) \times S_+(r, r))/\mathcal{O}(r). \tag{5}$$

The geometry (5) defines any element W of $S_+(r, n)$ by an r-dimensional subspace of $I\!R^n$ and a matrix of $S_+(r, r)$, i.e., this representation allows a separate treatment for the range space of W and its projection onto that subspace. The tangent space to the manifold (5) at a point (U, B) is naturally represented by the set

$$T_{(U,B)} S_+(r, n) = \{(\Delta, D) : U_\perp M, M \in I\!R^{(n-r) \times r}, D \in \mathrm{Sym}(p)\}$$

where $U_\perp \in \mathrm{St}(n, n-r)$, $U_\perp^T U = 0$ and $\mathrm{Sym}(p)$ is the set of symmetric matrices in $I\!R^{p \times p}$. We finally endow this geometry with the Riemannian metric

$$g_{(U,B)}((\Delta_1, \Delta_2), (D_1, D_2)) = \frac{1}{\lambda} \mathrm{tr}(\Delta_1^T \Delta_2) + \frac{1}{1-\lambda} \mathrm{tr}(D_1 B^{-1} D_2 B^{-1}), \tag{6}$$

which is a weighted sum of the natural metrics of $\mathrm{St}(n, r)$ and $S_+(r, r)$. The parameter $\lambda \in (0, 1)$ controls the relative importance given to the learning of the subspace compared to the estimation of B. The metric (6) is invariant by rotation (change of orthogonal frame) and scaling (change of units).

In order to obtain separate iterations on the two manifolds $\mathrm{Gr}(r, n) = \mathrm{St}(n, r)/\mathcal{O}(r)$ and $S_+(r, r)$, we define particular curves $W(\gamma) \in S_+(r, n)$ emanating from $W(0) = UBU^T$ and tangent to the direction $(\Delta, D) \in T_{(U,B)} S_+(r, n)$:

$$W(\gamma) = U(\gamma) B(\gamma) U(\gamma)^T, \tag{7}$$

where the curves $U(\gamma) \in \mathrm{St}(n,r)$ and $B(\gamma) \in S_+(r,r)$ are tangent to Δ at U and to D at B, respectively, e.g.,

$$U(\gamma) = \mathrm{qf}(U + \gamma\Delta),$$

and

$$B(\gamma) = B^{\frac{1}{2}} \exp(\gamma B^{-\frac{1}{2}} D B^{-\frac{1}{2}}) B^{\frac{1}{2}},$$

where qf denotes the Q factor of the QR decomposition. Note that further explicit characterizations of curves on $\mathrm{St}(n,r)$ are available (see, e.g., [8, 11]).

The algorithmic complexity of this update is $O(nr^2 + r^3)$. The computational cost is dominated by the computation of a SVD for U and the computation of the exponential for B.

In this paper, we propose an alternative to the geometry (5) that is expected to be more effective from an algorithmic point of view. If $B \in S_+(r,r)$ is factorized as the square of an r-by-r full-rank symmetric matrix $R \in \mathrm{Sym}_*(r)$, i.e., $B = RR$, it induces the parameterization

$$W = URRU^T \tag{8}$$

for any $W \in S_+(r,n)$. Again, the representation (8) is invariant with respect to the group action

$$(U, R) \mapsto (UO, O^T R O),$$

for any $O \in \mathcal{O}(r)$, which leads to the quotient manifold representation

$$S_+(r,n) \simeq (\mathrm{St}(n,r) \times \mathrm{Sym}_*(r))/\mathcal{O}(r), \tag{9}$$

We define the tangent space to (9) at a point (U, R) by

$$T_{(U,R)} S_+(r,n) = \\ \{(\Delta, D) : U_\perp M, M \in I\!R^{(n-r)\times r}, D \in \mathrm{Sym}(p)\}.$$

As previously, we endow the geometry (9) with the Riemannian metric

$$g_{(U,R)}((\Delta_1, \Delta_2), (D_1, D_2)) = \\ \frac{1}{\lambda}\mathrm{tr}(\Delta_1^T \Delta_2) + \frac{1}{1-\lambda}\mathrm{tr}(D_1 R^{-1} D_2 R^{-1}), \tag{10}$$

which is invariant by rotation and scaling. Accordingly, curves $W(\gamma) \in S_+(r,n)$ emanating from $W(0) = UR^2U^T$ and tangent to $(\Delta, D) \in T_{(U,R)} S_+(r,n)$ are given by

$$W(\gamma) = U(\gamma) R(\gamma)^2 U(\gamma)^T, \tag{11}$$

with the curves $U(\gamma) \in \mathrm{St}(n,r)$ and $R(\gamma) \in \mathrm{Sym}_*(r)$ that are tangent to Δ at U and to D at R, respectively. Because $\mathrm{Sym}(r)$ is a vector space, the simplest expression for $R(\gamma)$ is probably

$$R(\gamma) = R + \gamma D,$$

which is full-rank for generic γ and D.

4. STOCHASTIC GRADIENT ALGORITHMS

We now derive algorithms to minimize online the expected loss (1) by considering both geometries (5) and (9) of the fixed-rank PSD matrices. We focus in this context on stochastic gradient descent algorithms [12], which minimize the expected loss by performing each gradient step with respect to a single observation at a time, i.e., the update law is written as follows in the case of vector spaces,

$$W_{t+1} = W_t - \gamma_t \nabla_W l(\mathbf{y}_t, W_t). \tag{12}$$

The line-search iteration (12) is generalized to a general nonlinear manifold $\mathcal{M}$ by the update

$$W_{t+1} = \mathcal{R}_{W_t}(-\gamma_t \nabla_W l(\mathbf{y}_t, W_t)), \tag{13}$$

where the gradient $\nabla_W l(\mathbf{y}_t, W_t)$ belongs to the tangent space $T_{W_t}\mathcal{M}$ at the current iterate $W_t \in \mathcal{M}$ and the retraction $\mathcal{R}_W : T_W\mathcal{M} \to \mathcal{M}$ is a mapping such that the curve $\Gamma(\gamma) = \mathcal{R}_W(\gamma Z)$ passes through W and is tangent to $Z \in T_W\mathcal{M}$ at $\gamma = 0$. More details on the adaptation of line-search algorithms to manifolds can be found in [11] and references therein.

In case of the geometry (5) endowed with the metric (6), the objective to minimize is

$$l(\mathbf{y}, U, B) = (\mathbf{z}^T U B U^T \mathbf{z} - y)^2.$$

The gradient $(\nabla_U l, \nabla_B l) \in T_{(U,B)} S_+(r,n)$ with respect to the chosen metric are given by

$$\nabla_U l(\mathbf{y}, U, B) = 4\lambda(\hat{y} - y)(I - UU^T)\mathbf{z}\mathbf{z}^T U B, \tag{14}$$
$$\nabla_B l(\mathbf{y}, U, B) = 2(1-\lambda)(\hat{y} - y) B U^T \mathbf{z}\mathbf{z}^T U B.$$

Similarly, for the geometry (9) and the metric (10), the objective becomes

$$l(\mathbf{y}, U, R) = (\mathbf{z}^T U R^2 U^T \mathbf{z} - y)^2,$$

whose gradient $(\nabla_U l, \nabla_R l) \in T_{(U,R)} S_+(r,n)$ is

$$\nabla_U l(\mathbf{y}, U, R) = 4\lambda(\hat{y} - y)(I - UU^T)\mathbf{z}\mathbf{z}^T U R^2, \tag{15}$$
$$\nabla_R l(\mathbf{y}, U, R) = 4(1-\lambda)(\hat{y} - y) R\,\mathrm{sym}(R U^T \mathbf{z}\mathbf{z}^T U) R,$$

where $\mathrm{sym}(M) = \frac{M + M^T}{2}$ extracts the symmetric part of the square matrix M.

Possible retractions in (13) are for instance obtained by moving the iterate along the curves (7) and (11). The retraction associated to the geometry (9) and derived from the curve (11) is probably the most efficient from a computational point of view.

The step size γ_t can either be fixed to a small value or adjusted by backtracking or bisection techniques.

The convergence proof of the resulting stochastic gradient algorithms is the topic of ongoing research.

5. CONNECTION WITH EXISTING METHODS

The parameter λ introduced in both metrics (6) and (10) acts as a weighting factor on the gradients with respect to U and B (or R). This allows to tune the resulting search direction in order to place more emphasis on the subspace learning problem ($\lambda \to 1$), or on the estimation of the full-rank PSD projection onto that subspace ($\lambda \to 0$). We now focus on both limit cases and highlight the connection of the proposed algorithms with existing methods.

5.1. Connection with subspace learning

Online subspace learning is the problem of tracking the r-dimensional subspace that spans at best a sequence of received vectors $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t \in \mathbb{R}^n$. The problem is usually formulated as minimization of the expected loss

$$F(U) = \mathbb{E}[\|\mathbf{z} - UU^T\mathbf{z}\|^2], \tag{16}$$

with $U \in \mathrm{St}(n, r)$ that spans the subspace of interest [13, 14]. Note that (16) remains invariant by the transformation $U \to UO$ with $O \in \mathcal{O}(r)$. The subspace learning problem is hence defined on the Grassmann manifold $\mathrm{Gr}(r, n) = \mathrm{St}(n, r)/\mathcal{O}(r)$. In case of the Euclidean metric $g_U(\Delta_1, \Delta_2) = \mathrm{tr}(\Delta_1^T \Delta_2)$, a stochastic gradient algorithm to minimize (16) uses the gradient

$$\nabla_U f(\mathbf{z}, U) = -2(I - UU^T)\mathbf{z}\mathbf{z}^T U, \tag{17}$$

of the instantaneous cost function $f(U) = \|\mathbf{z} - UU^T\mathbf{z}\|^2$.

In the limit case $\lambda \to 1$, the gradient of (3) is

$$\nabla_W l(\mathbf{y}, W) = (\nabla_U l, 0)$$

where $\nabla_U l$ is given by either (14) or (15) and spans the same subspace as the gradient (17). In that setting, the method proposed in Section 4 can thus be seen as an adaptive line-search algorithm for subspace learning for the sequence of received vectors $\mathbf{z}_i$.

5.2. Connection with full-rank distance learning

In the limit case $\lambda \to 0$, the gradient of (3) is

$$\nabla_W l(\mathbf{y}, W) = (0, \nabla_B l) \quad \text{or} \quad \nabla_W l(\mathbf{y}, W) = (0, \nabla_R l),$$

i.e., the range space of W is not allowed to evolve over time. The problem amounts to learn the projection of W onto that subspace, i.e., to learn an r-by-r positive definite matrix. This problem inherits nice convexity properties as well as a well-characterized convergence. We can thus draw a parallel between our algorithm and methods that learn PSD matrices at fixed range space (e.g., [4]).

6. CONCLUSION

In this paper, we propose a flexible algorithmic framework to estimate a fixed-rank PSD matrix. The main idea is to provide a separate iteration for the span of the matrix and its projection onto that span. This separation results from a parameterization of the set of fixed-rank PSD matrices as the product of the Grassmann manifold with either the cone of positive definite matrices or the set of symmetric matrices. By weighting differently the two updates, more emphasis can be either placed on learning the subspace or learning a distance for a fixed range space. Connections with existing algorithms have been identified in the two limit cases. Numerical experiments of the proposed method are in progress.

7. REFERENCES

[1] E. Xing, A. Ng, M. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," in *Proceedings of the NIPS conference*, 2002.

[2] S. Shalev-Shwartz, Y. Singer, and A.Y. Ng, "Online and batch learning of pseudo-metrics," in *Proceedings of the ICML conference*, 2004.

[3] K. Tsuda, G. Ratsch, and M. Warmuth, "Matrix exponentiated gradient updates for on-line learning and bregman projection," *JMLR*, vol. 6, pp. 995–1018, 2005.

[4] B. Kulis, M. Sustik, and I.S. Dhillon, "Low-rank kernel learning with bregman matrix divergences," *JMLR*, vol. 10, pp. 341–376, 2009.

[5] J. Davis, B. Kulis, P. Jain, S. Sra, and I.S. Dhillon, "Information-theoretic metric learning," in *Proceedings of the ICML conference*, 2007.

[6] J. Davis and I.S. Dhillon, "Structured metric learning for high dimensional problems," in *Proceedings of the KDD conference*, 2008.

[7] P. Jain, B. Kulis, I.S. Dhillon, and K. Grauman, "Online metric learning and fast similarity search," in *Proceedings of the NIPS conference*, 2008.

[8] S. Bonnabel and R. Sepulchre, "Riemannian distance and geometric mean for positive semi-definite matrices of fixed rank," *SIAM Journal on Matrix Analysis and Applications, in press.*

[9] G. Meyer, S. Bonnabel, and R. Sepulchre, "Stochastic learning of fixed-rank positive semidefinite matrices," *Submitted to NIPS*, 2009.

[10] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.

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

[12] L. Bottou, "Stochastic learning," in *Advanced Lectures on Machine Learning*, LNAI-3176, pp. 146–168. Springer, 2004.

[13] J. Karhunen and J. Joutsensalo, "Generalizations of principal component analysis, optimization problems, and neural networks," *Neural Networks*, vol. 8, no. 4, pp. 549–562, 1995.

[14] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, 2001.