

Low-rank optimization for distance matrix completion

B. Mishra, G. Meyer and R. Sepulchre

Abstract—This paper addresses the problem of low-rank distance matrix completion. This problem amounts to recover the missing entries of a distance matrix when the dimension of the data embedding space is possibly unknown but small compared to the number of considered data points. The focus is on high-dimensional problems. We recast the considered problem into an optimization problem over the set of low-rank positive semidefinite matrices and propose two efficient algorithms for low-rank distance matrix completion. In addition, we propose a strategy to determine the dimension of the embedding space. The resulting algorithms scale to high-dimensional problems and monotonically converge to a global solution of the problem. Finally, numerical experiments illustrate the good performance of the proposed algorithms on benchmarks.

I. INTRODUCTION

Completing the missing entries of a matrix under low-rank constraint is a fundamental and recurrent problem in many modern engineering applications (see [1] and references therein). Recently, the problem has gained much popularity thanks to collaborative filtering applications and the Netflix challenge [2].

This paper focuses on an important variant of the problem, that is, completing the missing entries of a Euclidean distance matrix (EDM) under low-rank constraint. Typical applications include data visualization [3], dimensionality reduction in behavioral sciences and economics [4], molecular conformation problems [5], [6], just to name a few.

A Euclidean distance matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$ contains the (squared) pairwise distances between n data points $\mathbf{y}_i \in \mathbb{R}^r$, $i = 1, \dots, n$. This matrix is symmetric and has a zero diagonal. Its entries are non-negative and satisfy the triangle inequality. These properties are readily verified by examining the entries of the distance matrix,

$$D_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|_2^2.$$

The set $\text{EDM}(n)$ of n -by- n Euclidean distance matrices forms a convex cone which has a well-studied geometry (see [7], [8], and references therein). One property of a Euclidean distance matrix is that, it is rank deficient. The rank of \mathbf{D} is upper bounded by $r + 2$ (and the rank is generically $r + 2$), which in many problems is very small compared to n , the number of data points.

Given a set of pairwise distances or dissimilarities between data points, the goal of low-rank distance matrix completion algorithms is to recover a full Euclidean distance matrix from a restrictive set of given distances. Inference on the unknown

entries is possible thanks to the low-rank property which models the redundancy between the available data.

A closely related problem is multidimensional scaling (MDS) for which all pairwise distances are available up front. A solution to this problem is the classical multidimensional scaling algorithm (CMDS), which relies on singular value decomposition to find a globally optimum embedding of fixed-rank. The CMDS algorithm minimizes the total quadratic error on scalar products between data points. Other algorithms have focused on variant cost functions, see the paper [9] for a survey in this area.

In contrast to the classical multidimensional scaling formulation, the problem of Euclidean distance matrix completion involves missing distances. The problem can be considered as a variant of multidimensional scaling problem with binary weights [9], [10]. The low-rank distance matrix completion problem is known to be NP-hard in general [11], [12], but convex relaxations have been proposed to render the problem tractable [13], [14]. Typical convex relaxations cast the EDM completion problem into a convex optimization problem on the set of positive semidefinite matrix, resulting in semidefinite programming techniques [15]. This convex formulation is nevertheless a large-scale problem when n is large.

Imposing the rank constraint in the problem formulation is an appealing way of reducing the size of the search space. However, it results in a non-convex optimization problem. Although convergence results are only local, the approach performs well in practice [16]. Both first-order [17], [18] and second order [6], [19], [10], [20] optimization methods have been considered and heuristics for finding a good low-rank initialization have been proposed [20].

A difficulty encountered by second order optimization algorithms is the intrinsic invariance properties of the data representation due to rotations. This issue may indeed prevent second order optimization algorithms to converge [21]. Several authors have resolved this issue at the extra cost of normalizing the data representation [10] or adding a penalization term to the objective function [19]. In this paper, the invariance to rotations is lifted in the problem formulation and is free of additional computational cost (see Section III). A survey of low-rank distance matrix completion algorithms can be found in the recent papers [22], [20].

Although, the problem is not new and is well-studied, a practical limitation of most of existing algorithms is that they do not scale to high-dimensional problems. Moreover, the problem of choosing a priori an appropriate dimension for the data embedding is still an open research question.

In this paper, the focus is on efficient algorithms that

The authors are with the Department of Electrical Engineering and Computer Science, University of Liège, Montefiore Institute, Sart-Tilman, 4000 Liège, Belgium (e-mails: {b.mishra, g.meyer, r.sepulchre}@ulg.ac.be).

scale to high-dimensional problems. Following a number of previous contributions in the literature, we recast the problem into an optimization problem over the set of low-rank (fixed-rank) positive semidefinite matrices. We adopt the geometric framework of optimization over Riemannian matrix manifolds [23] and exploit recent results on low-rank optimization for positive semidefinite matrices [24]. In particular, an efficient strategy for estimating the dimension of the embedding space is proposed. The resulting algorithms have linear complexity in the problem size and in the number of available distances. Using the strategy for estimating the optimal embedding dimension, the proposed algorithms converge monotonically to the global (low-rank) solution of the problem.

The paper is organized as follow. Section II presents the problem of interest and its different formulations. Section III describes the chosen optimization framework and introduces the main geometrical objects required by our algorithms. Section IV is devoted to the design of efficient algorithms for low-rank distance matrix completion. Finally, Section VI presents some numerical simulations.

II. LOW-RANK DISTANCE MATRIX COMPLETION

Given a set of dissimilarities $\tilde{\mathbf{D}}_{ij} \geq 0$ between n data points, distance matrix completion algorithms solve

$$\min_{\mathbf{D} \in \text{EDM}(n)} \|\mathbf{H} \odot (\mathbf{D} - \tilde{\mathbf{D}})\|_F^2, \quad (1)$$

where \mathbf{H} is a symmetric matrix with binary entries and the operator \odot denotes elementwise multiplication. If \mathcal{D} is the set of given entries (i, j) in $\tilde{\mathbf{D}}$ such that $i < j$, then

$$\mathbf{H}_{ij} = \mathbf{H}_{ji} = \begin{cases} 1 & \text{if } (i, j) \in \mathcal{D}, \\ 0 & \text{otherwise.} \end{cases}$$

The number of elements in the set \mathcal{D} is denoted by d . Although, d is at most equal to $n(n-1)/2$, in most applications, it is of order $O(nr)$, where r is the optimal embedding dimension. Dissimilarities potentially differ from distances in that they are not required to satisfy triangle inequality. For instance, this takes into account the fact that observation noise could make $\tilde{\mathbf{D}}$ different from a valid EDM.

A convenient alternative formulation of (1) is to cast this problem into an optimization problem on the set of positive semidefinite matrices [13]. The reformulation is given by

$$\min_{\mathbf{X} \succeq 0} \|\mathbf{H} \odot (\kappa(\mathbf{X}) - \tilde{\mathbf{D}})\|_F^2, \quad (2)$$

where κ is a mapping from the set of positive semidefinite matrices to the set of Euclidean distance matrices

$$\kappa(\mathbf{X}) = \text{Diag}(\mathbf{X})\mathbf{1}^T + \mathbf{1}\text{Diag}(\mathbf{X})^T - 2\mathbf{X}.$$

The function $\text{Diag}(\cdot)$ extracts the diagonal of its argument, and $\mathbf{1}$ denotes a vector with all entries equal to one.

A practical advantage of (2) compared to (1) is that the rank of \mathbf{X} identifies with the dimension of the embedding space. When no restriction is imposed on the rank of \mathbf{X} , problem (2) is convex and thus presents a global solution.

In this paper, we consider the case where the global solution \mathbf{X}^* of (2) is low-rank that is,

$$\text{rank}(\mathbf{X}^*) = r \ll n. \quad (3)$$

Following [24], we solve a sequence of nonconvex problems of increasing dimension until the actual value of the rank r is reached. Each nonconvex problem consists in solving the following rank-constrained optimization problem

$$\min_{\mathbf{X} \succeq 0} \|\mathbf{H} \odot (\kappa(\mathbf{X}) - \tilde{\mathbf{D}})\|_F^2, \quad \text{s.t. } \text{rank}(\mathbf{X}) = p. \quad (4)$$

By screening values from $p = 1$ to $p = r$, the results presented in [24] guarantee a monotonic convergence to a solution of the original problem (2). The proposed strategy for finding the actual rank r is detailed in Section IV-C.

Problem (4) is solved efficiently by exploiting a low-rank parametrization of the search space. The proposed approach hinges on the fact that any rank- p positive semidefinite matrix admits a factorization

$$\mathbf{X} = \mathbf{Y}\mathbf{Y}^T,$$

where $\mathbf{Y} \in \mathbb{R}_*^{n \times p} = \{\mathbf{Y} \in \mathbb{R}^{n \times p} : \det(\mathbf{Y}^T \mathbf{Y}) \neq 0\}$.

To exploit this factorization, we adopt the geometric framework of optimization on Riemannian manifolds [23]. Basic concepts and notations are introduced in the next section. See the book [23] for further details on optimization on matrix manifolds and for a state-of-the-art in this area.

III. MANIFOLD-BASED OPTIMIZATION

An intrinsic property of the factorization $\mathbf{X} = \mathbf{Y}\mathbf{Y}^T$ is that it is invariant with respect to the transformation

$$\mathbf{Y} \mapsto \mathbf{Y}\mathbf{Q},$$

where $\mathbf{Q} \in \mathcal{O}(p) = \{\mathbf{Q} \in \mathbb{R}^{p \times p} : \mathbf{Q}^T \mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}\}$.

This invariance property renders the minima of a cost function $f(\mathbf{Y}\mathbf{Y}^T)$ not isolated. This issue is not harmful for first order-methods such as gradient descent algorithms but greatly affects the convergence properties of second-order methods [23], [21].

To circumvent this issue, we reformulate the problem of interest as an optimization problem on the quotient manifold

$$\mathcal{M} \triangleq S_+(p, n) \simeq \mathbb{R}_*^{n \times p} / \mathcal{O}(p), \quad (5)$$

which represents the set of equivalence classes

$$[\mathbf{Y}] = \{\mathbf{Y}\mathbf{Q} : \mathbf{Q} \in \mathcal{O}(p)\}. \quad (6)$$

The set $S_+(p, n)$ is the set of rank- p symmetric positive semidefinite matrices of size n , that is,

$$S_+(p, n) = \{\mathbf{X} \in \mathbb{R}^{n \times n} : \mathbf{X} = \mathbf{X}^T \succeq 0, \text{rank}(\mathbf{X}) = p\}.$$

This set has a rich Riemannian manifold geometry which can be exploited for algorithmic purposes [25], [26], [27].

Problem (4) is now reformulated as an unconstrained optimization problem over the set of equivalence classes (6),

$$\min_{[\mathbf{Y}] \in \mathcal{M}} f([\mathbf{Y}]), \quad (7)$$

for the cost function

$$f([\mathbf{Y}]) = \|\mathbf{H} \odot (\kappa(\mathbf{Y}\mathbf{Y}^T) - \tilde{\mathbf{D}})\|_F^2. \quad (8)$$

To develop optimization algorithms on the quotient manifold, the space (5) is endowed with the Riemannian metric

$$g_{\mathbf{Y}}(\xi_{\mathbf{Y}}, \eta_{\mathbf{Y}}) = \text{Tr}(\xi_{\mathbf{Y}}^T \eta_{\mathbf{Y}}), \quad \xi_{\mathbf{Y}}, \eta_{\mathbf{Y}} \in T_{\mathbf{Y}}\mathcal{M},$$

which is inherited from the natural metric of $\mathbb{R}^{n \times p}$. With this metric, the tangent space $T_{\mathbf{Y}}\mathcal{M}$ at a given point \mathbf{Y} is decomposed into the sum of two complementary spaces,

$$T_{\mathbf{Y}}\mathcal{M} = \mathcal{V}_{\mathbf{Y}}\mathcal{M} \oplus \mathcal{H}_{\mathbf{Y}}\mathcal{M}.$$

The vertical space $\mathcal{V}_{\mathbf{Y}}\mathcal{M}$ contains the set of directions that are tangent to the set of equivalence classes (6), that is,

$$\mathcal{V}_{\mathbf{Y}}\mathcal{M} = \{\mathbf{Y}\Omega : \Omega^T = -\Omega \in \mathbb{R}^{p \times p}\}.$$

The horizontal space $\mathcal{H}_{\mathbf{Y}}\mathcal{M}$ contains the directions $\bar{\xi}_{\mathbf{Y}}$ that are orthogonal to the set of equivalence classes,

$$\mathcal{H}_{\mathbf{Y}}\mathcal{M} = \{\bar{\xi}_{\mathbf{Y}} \in \mathbb{R}^{n \times p} : \bar{\xi}_{\mathbf{Y}}^T \mathbf{Y} = \mathbf{Y}^T \bar{\xi}_{\mathbf{Y}}\}.$$

With such a construction, the directions of interest can be restricted to horizontal directions $\bar{\xi}_{\mathbf{Y}}$. Indeed, displacements along vertical directions leave the cost function unchanged.

The projection of a direction $\mathbf{Z} \in \mathbb{R}^{n \times p}$ onto the horizontal space is given by $\Pi_{\mathcal{H}_{\mathbf{Y}}}\mathbf{Z} = \mathbf{Z} - \mathbf{Y}\Omega$, where $\Omega \in \mathbb{R}^{p \times p}$ is skew-symmetric and satisfies the Sylvester equation

$$\Omega \mathbf{Y}^T \mathbf{Y} + \mathbf{Y}^T \mathbf{Y} \Omega = \mathbf{Y}^T \mathbf{Z} - \mathbf{Z}^T \mathbf{Y}.$$

Overall, projecting a direction $\mathbf{Z} \in \mathbb{R}^{n \times p}$ onto the horizontal space requires $O(np^2 + np + p^3)$ operations (computing matrices $\mathbf{Y}^T \mathbf{Y}$, $\mathbf{Y}^T \mathbf{Z}$, and $\mathbf{Y}\Omega$ requires $O(np^2)$ operations, solving the Sylvester equation is performed in $O(p^3)$ operations and the projection requires $O(np)$ operations).

To update our search variable, we require a local mapping from tangent space to the manifold. Such a mapping is called a retraction. For the manifold of interest, a retraction is provided by the simple and efficient formula

$$\mathbf{R}_{\mathbf{Y}}(\bar{\xi}_{\mathbf{Y}}) = \mathbf{Y} + \bar{\xi}_{\mathbf{Y}}. \quad (9)$$

which gives a full-rank matrix for generic direction $\bar{\xi}_{\mathbf{Y}}$.

IV. ALGORITHMS

In this section, we exploit the concepts presented in the previous section to develop both a gradient descent algorithm and a trust-region algorithm to solve (7).

A. Gradient descent algorithm

The gradient of a smooth cost function $f : \mathcal{M} \rightarrow \mathbb{R}$ is the unique tangent vector $\text{grad}f(\mathbf{Y}) \in T_{\mathbf{Y}}\mathcal{M}$ that satisfies

$$g_{\mathbf{Y}}(\xi_{\mathbf{Y}}, \text{grad}f(\mathbf{Y})) = Df(\mathbf{Y})[\xi_{\mathbf{Y}}], \quad \forall \xi_{\mathbf{Y}} \in T_{\mathbf{Y}}\mathcal{M}. \quad (10)$$

The quantity $Df(\mathbf{Y})[\xi_{\mathbf{Y}}]$ is the directional derivative of f in the direction $\xi_{\mathbf{Y}}$, that is,

$$Df(\mathbf{Y})[\xi_{\mathbf{Y}}] = \lim_{t \rightarrow 0} \frac{f(\mathbf{Y} + t\xi_{\mathbf{Y}}) - f(\mathbf{Y})}{t}.$$

Applying formula (10) to the cost (8) gives us the gradient

$$\text{grad}f(\mathbf{Y}) = 2\kappa^*(\mathbf{H} \odot (\kappa(\mathbf{Y}\mathbf{Y}^T) - \tilde{\mathbf{D}}))\mathbf{Y}, \quad (11)$$

where $\kappa^*(\mathbf{A})$ is the adjoint operator of κ defined by

$$\kappa^*(\mathbf{A}) = 2(\text{Diag}(\mathbf{A}\mathbf{1}) - \mathbf{A}).$$

Combining the gradient (11) with the retraction (9) gives us the gradient descent algorithm

$$\mathbf{Y}_{t+1} = \mathbf{Y}_t - 2s_t \kappa^*(\mathbf{H} \odot (\kappa(\mathbf{Y}_t \mathbf{Y}_t^T) - \tilde{\mathbf{D}}))\mathbf{Y}_t, \quad (12)$$

where $s_t > 0$ is the gradient step size. We select s_t using the Armijo criterion [28], that is, a step size s_A that satisfies

$$f(\mathbf{Y}_t - s_A \text{grad}f(\mathbf{Y}_t)) \leq f(\mathbf{Y}_t) - c s_A \|\text{grad}f(\mathbf{Y}_t)\|_F^2,$$

where $c \in (0, 1)$ is a constant (we choose the value $c = 0.5$).

The asymptotic computational cost of an iteration (12) is $O(dp + np)$, where d is the number of known entries of $\tilde{\mathbf{D}}$. The memory requirement is $O(dp + np)$. The computationally most demanding step is the computation of the gradient, which requires $O(dp)$ operations. This low computational complexity and memory requirement allows us to handle potentially large data sets. A drawback is however that the gradient descent algorithm only guarantees a linear convergence rate. We can achieve a superlinear convergence rate by means of a Riemannian trust-region algorithm which exploits second-order information.

B. Trust-region algorithm

Trust-region algorithms sequentially solve the problem

$$\begin{aligned} \min_{\bar{\xi} \in \mathcal{H}_{\mathbf{Y}}\mathcal{M}} & f(\mathbf{Y}) + g_{\mathbf{Y}}(\bar{\xi}, \text{grad}f(\mathbf{Y})) + \frac{1}{2}g_{\mathbf{Y}}(\bar{\xi}, \text{Hess}f(\mathbf{Y})[\bar{\xi}]), \\ \text{s.t.} & g_{\mathbf{Y}}(\bar{\xi}, \bar{\xi}) \leq \delta^2, \end{aligned}$$

which amounts to minimize a quadratic model of the cost function on a trust-region radius of size δ . Once a search direction $\bar{\xi}$ is identified, the search variable is updated as

$$\mathbf{Y}_{t+1} = \mathbf{R}_{\mathbf{Y}_t}(\bar{\xi}). \quad (13)$$

The trust-region radius δ vary according to the quality of the iterate. When a good solution is found within the trust-region, then the trust-region is expanded. Conversely, if the iterate is poor then the region is contracted.

More technical details on trust-region algorithms on Riemannian manifolds can be found in [29], [23]. In this paper, we adapt the generic implementation of the toolbox GenRTR to our problem of interest.¹

Trust-region algorithms require the computation of the Riemannian Hessian $\text{Hess}f(\mathbf{Y})[\bar{\eta}]$ in a given direction $\bar{\eta}$. It is obtained as

$$\text{Hess}f(\mathbf{Y})[\bar{\eta}] \triangleq \nabla_{\bar{\eta}} \text{grad}f(\mathbf{Y})$$

where $\nabla_{\bar{\eta}} \text{grad}f(\mathbf{Y})$ is the Riemannian connection of the gradient vector field in the direction $\bar{\eta}$. Riemannian connections generalize the notion of directional derivative of a

¹The software can be downloaded from <http://www.math.fsu.edu/~cbaker/GenRTR/>

vector field to Riemannian manifolds. Given a vector field ζ on \mathcal{M} that assigns to each point $\mathbf{Y} \in \mathcal{M}$ a tangent vector $\zeta_{\mathbf{Y}} \in T_{\mathbf{Y}}\mathcal{M}$, the directional derivative of ζ at $\mathbf{Y} \in \mathcal{M}$ in a direction $\bar{\eta} \in \mathcal{H}_{\mathbf{Y}}\mathcal{M}$ is given by

$$\nabla_{\bar{\eta}} \zeta_{\mathbf{Y}} = \Pi_{\mathcal{H}_{\mathbf{Y}}} \left(\lim_{t \rightarrow 0} \frac{\zeta_{\mathbf{Y}+t\bar{\eta}} - \zeta_{\mathbf{Y}}}{t} \right). \quad (14)$$

Applying this formula to the vector field $\text{grad} f(\mathbf{Y})$ gives us

$$\begin{aligned} \text{Hess} f(\mathbf{Y})[\bar{\eta}] &= 2\Pi_{\mathcal{H}_{\mathbf{Y}}}(\kappa^*(\mathbf{H} \odot (\kappa(\mathbf{Y}\bar{\eta}^T + \bar{\eta}\mathbf{Y}^T)))\mathbf{Y} \\ &\quad + \kappa^*(\mathbf{H} \odot (\kappa(\mathbf{Y}\mathbf{Y}^T) - \tilde{\mathbf{D}}))\bar{\eta}). \end{aligned}$$

The numerical cost of an iteration of the trust-region algorithm is $O(dp+np+np^2+p^3)$. The memory requirement is also $O(dp+np)$. The computational bottleneck is the computation of the Hessian. Still, the complexity is linear in both the number of available distance and in the problem size. With a proper parameter tuning, the proposed trust-region algorithm enjoys a superlinear convergence rate.

C. Strategy for estimating the optimal embedding dimension

The following section is an adaptation of the material presented in [24] to the problem of interest. To identify the (unknown) rank r of the global solution to (2), we solve a sequence of nonconvex problems (7) of increasing dimension. The approximation rank p is progressively incremented from $p = 1$ to $p = r$. Using a warm restart strategy for moving from one value of p to the next, we are able to propose a descent algorithm that converges monotonically to a global solution of the original problem (4).

This strategy efficiently exploits the previous iterations of the algorithm as opposed to earlier heuristic methods that use random restart for each value of the rank [30].

For a given rank $p < r$, the trust-region or gradient descent algorithm gives us a local minimizer \mathbf{Y}_p^* of the nonconvex problem (7). Let us consider the following initial condition for the problem of rank $p + 1$,

$$\mathbf{Y}_0 = [\mathbf{Y}_p^* | \mathbf{0}^{n \times 1}],$$

that is, \mathbf{Y}_p^* with an additional zero column appended. Since \mathbf{Y}_p^* is local minimizer for rank p , we have that \mathbf{Y}_0 is a critical point for the problem of rank $p + 1$. As \mathbf{Y}_p^* is not the sought solution to (2), this means that \mathbf{Y}_0 is a saddle point for the problem of rank $p + 1$. Therefore, by virtue of the second order KKT optimality conditions, there must exist a descent direction $\mathbf{Z} \in \mathbb{R}^{n \times p}$ such that

$$\frac{1}{2} \text{Tr}(\mathbf{Z}^T D\text{grad} f(\mathbf{Y}_0)[\mathbf{Z}]) \leq 0.$$

To escape from the saddle point, we can thus exploit the following descent direction

$$\mathbf{Z} = [\mathbf{0}^{n \times p} | \mathbf{v}],$$

where \mathbf{v} is the eigenvector associated to the smallest algebraic eigenvalue of

$$\mathbf{S}_Y = \nabla_{\mathbf{X}} f(\mathbf{Y}_p^* \mathbf{Y}_p^{*T}), \quad (15)$$

and where $\nabla_{\mathbf{X}} f(\mathbf{Y}\mathbf{Y}^T)$ is the Euclidean gradient of the convex cost function $f(\mathbf{X})$ evaluated at $\mathbf{Y}\mathbf{Y}^T$. As we have

$$\text{grad} f(\mathbf{Y}_0) = \nabla_{\mathbf{X}} f(\mathbf{Y}_0 \mathbf{Y}_0^T) \mathbf{Y}_0,$$

the proposed direction satisfies

$$\frac{1}{2} \text{Tr}(\mathbf{Z}^T D\text{grad} f(\mathbf{Y}_0)[\mathbf{Z}]) = \mathbf{v}^T \mathbf{S}_Y \mathbf{v} \leq 0.$$

The descent direction is exploited by performing a single line-search step using the Armijo rule. The resulting iterate is then used as the initial condition for the optimization algorithm that will solve the problem of rank $p + 1$.

The procedure stops at the latest when $p = n$. However, in the setting of interest, problem (2) presents a low-rank solution with $r \ll n$. We thus, expect the algorithm to stop much before reaching $p = n$.

For the proposed strategy it is important to reach a local minimum of the cost function as long as $p < r$. Although in theory convergence to saddle points cannot be excluded for gradient descent algorithms, the issue is not harmful in practice as saddle point are generally unstable from a numerical point of view.

V. DISCUSSION

We propose both a gradient descent and a trust-region algorithm for solving the fixed-rank Euclidean distance matrix completion problem. The numerical cost per iteration for the gradient descent algorithm is $O(dp+np)$ versus $O(dp+np+np^2+p^3)$ for the trust region algorithm. Although the gradient descent algorithm has a smaller computational cost per iteration, the number of iterations required to reach convergence is higher than for the trust-region algorithm.

We thus recommend the trust-region algorithm when a high optimization accuracy is required or when the observation noise is small. The gradient descent approach should be preferred for very large problems where the observation noise is high. In this setting, one is usually not interested in a solution of high-accuracy, since it generally compromises the generalization performance.

VI. NUMERICAL EXPERIMENTS

In this section, we evaluate the performance of the proposed algorithms on benchmarks. A MATLAB implementation is available from the first author's webpage.²

A. A visual example

This example is adapted from [22]. Consider $n = 121$ data points arranged in a 3-dimensional helix structure defined by

$$(x, y, z) = (4 \cos(3t), 4 \sin(3t), 2t), \quad 0 \leq t \leq 2\pi.$$

After computing the distance matrix of between these points, we randomly remove 85% of the distances uniformly and at random to generate a dissimilarity matrix $\tilde{\mathbf{D}}$. From 15% of distances the goal is to reconstruct the helix structure. We run the algorithms with the rank incremental strategy discussed in Section (IV-C). Both algorithms recover correctly

²<http://www.montefiore.ulg.ac.be/~mishra>

the helix structure (Figure 1). We only display the results for gradient descent as it coincides with the results of the trust-region algorithm. The algorithms stop when the relative or absolute variation of the cost function drops below 10^{-5} .

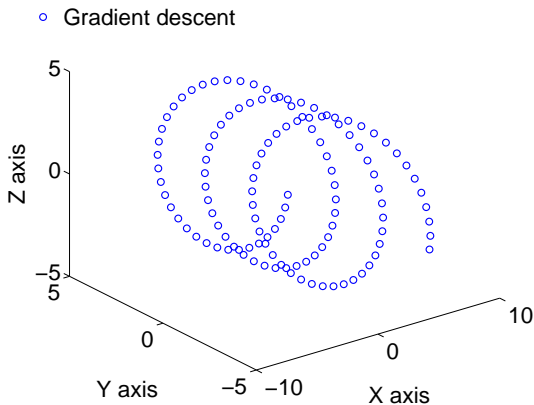


Fig. 1. The proposed algorithms correctly recover the 3D helix structure from 15% of the complete set of pairwise distances.

B. Trust-region versus gradient descent

To compare the two versions of the algorithm, we generate a random distance matrix

$$\mathbf{D}^* = \kappa(\mathbf{Y}^* \mathbf{Y}^{*T}), \quad (16)$$

where $\mathbf{Y}^* \in \mathbb{R}^{500 \times 3}$ has entries distributed according to gaussian distribution with zero mean and unit standard deviation. The fraction of unknown distances is fixed at 85%. We run the algorithms without knowing the embedding dimension. The stopping criterion is the same as for the previous experiment. The objective function is plotted against the number of iterations (Figures 2(a) and 2(b)). Both algorithms recover the correct configuration and dimensionality. The total number of iterations to reach convergence is 156 for the trust-region and 453 for the gradient descent. Observe the monotonic convergence of both algorithms to the sought solution.

C. Scaling test

We now evaluate our algorithms on larger random data sets. We vary the problem size n from 1000 to 10000. For each n , we generate a random distance matrix according to (16) with $\mathbf{Y}^* \in \mathbb{R}^{n \times 4}$. We sample 0.1 fraction of the total amount of distances and the algorithms are run by fixing the embedding dimension, $p = 4$. Results are averaged over 10 runs. The test has been performed on a single core Intel L5420 2.5 GHz with 5GB of RAM.

The time taken and number of iterations required to reach convergence is reported at Figure 3(a) and 3(b) respectively. For instance, for $n = 10000$ the number of known distances is about 5 millions (10% of 50 million total entries). The gradient descent algorithm takes about 120 iterations and 31 minutes, while the trust region algorithm solves the problem in 30 iterations and 18 minutes.

VII. CONCLUSION

In this paper two efficient numerical optimization algorithms have been presented for the distance matrix completion problem. In particular, the algorithms do not require any prior notion about the embedding and can potentially handle very large data sets. The proposed algorithms stem from a geometric view of the problem formulation. This interpretation as a manifold-based optimization problem considerably reduced the computational burden. At the same we were able to devise a superlinearly converging scheme namely, the trust-region algorithm in addition to the linearly convergent gradient descent algorithm. The numerical experiments that have been performed, are very encouraging on various parameters.

VIII. ACKNOWLEDGMENTS

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. Gilles Meyer is supported as an FRS-FNRS research fellow (Belgian Fund for Scientific Research).

REFERENCES

- [1] E. J. Candes and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, Dec. 2009.
- [2] "The Netflix prize." [Online]. Available: <http://www.netflixprize.com/>
- [3] A. Morrison, G. Ross, and M. Chalmers, "Fast multidimensional scaling through sampling, springs and interpolation," *Information Visualization*, vol. 2, no. 1, pp. 68–77, 2003.
- [4] I. Borg and P. J. Groenen, *Modern multidimensional scaling : theory and applications*, ser. Springer series in Statistics. Springer, 1997.
- [5] J. More and Z. Wu, "Distance geometry optimization for protein structures," *Journal of Global Optimization*, vol. 15, no. 3, pp. 219–234, Oct. 1999.
- [6] W. Glunt, T. Hayden, and M. Rayden, "Molecular confirmations from distance matrices," *Journal of Computational Chemistry*, vol. 14, pp. 114–120, 1993.
- [7] J. Dattorro, *Convex optimization & Euclidean distance geometry*. PO Box 12, Palo Alto, CA 94302: Meboo Publishing USA, 2005.
- [8] A. Y. Alfakih, "On the uniqueness of Euclidean distance matrix completions: the case of points in general position," *Linear Algebra Applications*, vol. 397, pp. 265–277, 2005.
- [9] J. D. Leeuw, "Multidimensional scaling," Department of Statistics, UCLA, Tech. Rep., 2001. [Online]. Available: <http://preprints.stat.ucla.edu/274/274.pdf>
- [10] A. Kearsley, R. Tapia, and M. Trosset, "The solution of the metric stress and sstress problems in multidimensional scaling by Newton's method," *Computational Statistics*, vol. 13, no. 3, pp. 369–396, 1998.
- [11] M. Laurent, "A connection between positive semidefinite and Euclidean distance matrix completion problems," *Linear Algebra and its Applications*, vol. 273, pp. 9–22, 1998.
- [12] H.-X. Huang, Z.-A. Liang, and P. M. Pardalos, "Some properties for the Euclidean distance matrix and positive semidefinite matrix completion problems," *Journal of Global Optimization*, vol. 25, pp. 3–21, January 2003.
- [13] A. Alfakih, A. Khandani, and H. Wolkowicz, "Solving Euclidean distance matrix completion problems via semidefinite programming," *Computational Optimization and Applications*, vol. 12, no. 1-3, pp. 13–30, Jan. 1999.
- [14] L. Cayton and S. Dasgupta, "Robust Euclidean embedding," in *Proceedings of the 23rd international conference on Machine learning*, 2006.
- [15] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, March 2004.

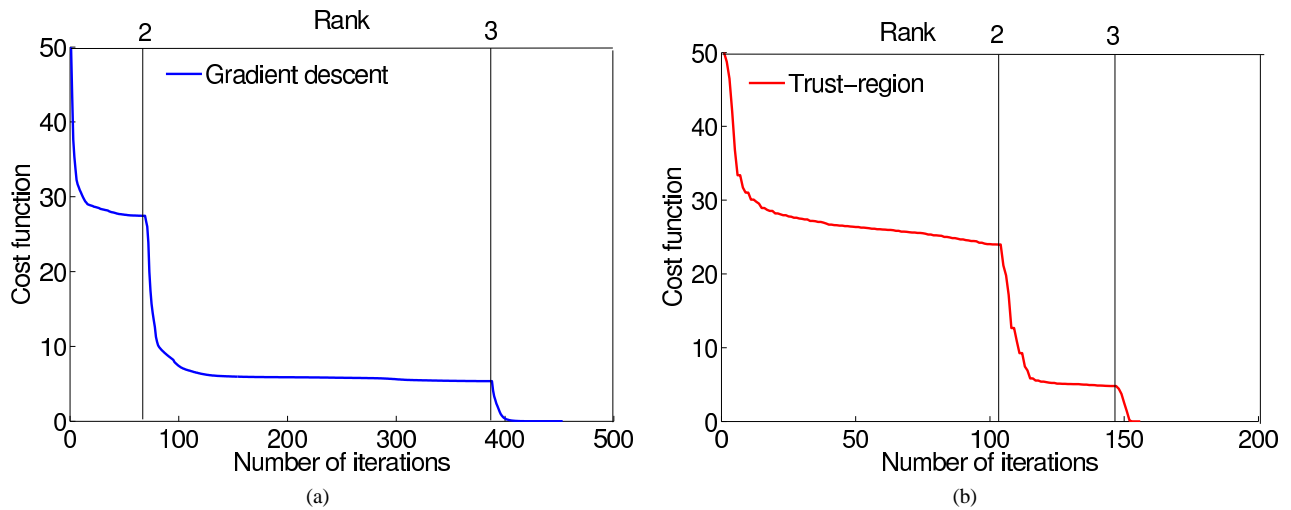


Fig. 2. Convergence plots of the algorithms.

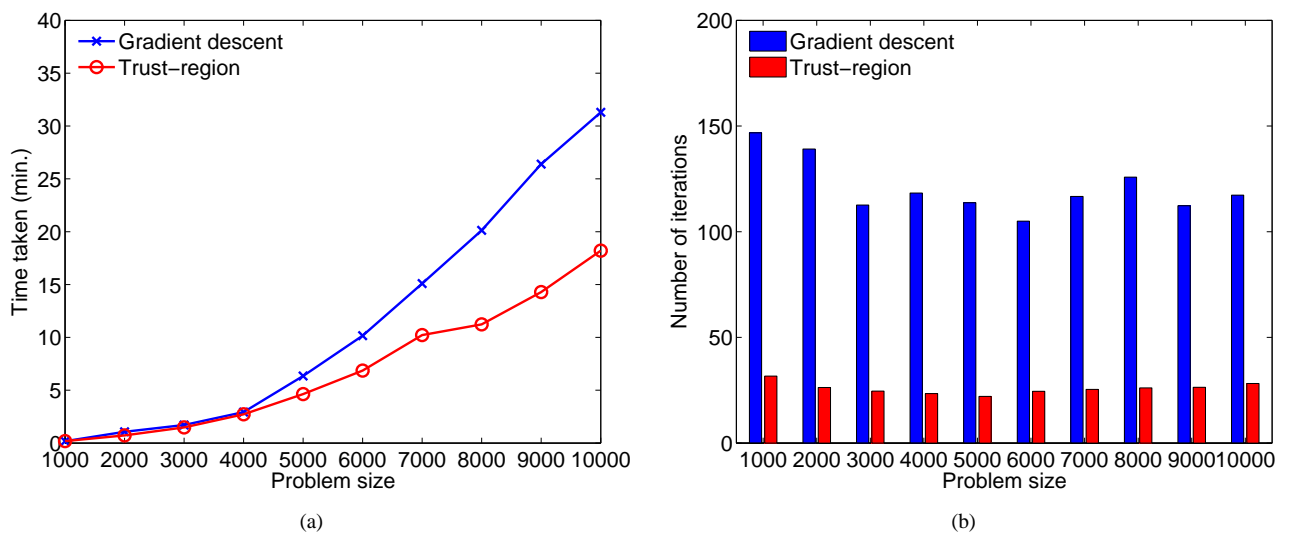


Fig. 3. Analysis of the proposed algorithms on randomly generated datasets.

- [16] M. W. Trosset, "Distance matrix completion by numerical optimization," *Comput. Optim. Appl.*, vol. 17, pp. 11–22, October 2000.
- [17] J. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, March 1964.
- [18] A. Buja and D. F. Swayne, "Interactive data visualization with multidimensional scaling," Department of Statistics, Stanford University, Tech. Rep., March 2004. [Online]. Available: [http://www-stat.stanford.edu/~sim\\$tibbs/sta306b/buja.pdf](http://www-stat.stanford.edu/~sim$tibbs/sta306b/buja.pdf)
- [19] P. Tarazaga and M. W. Trosset, "An optimization problem on subsets of the symmetric positive-semidefinite matrices," *Journal of Optimization Theory and Applications*, vol. 79, pp. 513–524, 1993.
- [20] H. Fang and D. P. O'leary, "Euclidean distance matrix completion problems," June 2010.
- [21] P. A. Absil, M. Ishteva, L. De Lathauwer, and S. Van Huffel, "A geometric Newton method for Oja's vector field," *Neural Computation*, vol. 21, no. 5, pp. 1415–1433, May 2009.
- [22] D. I. Chu, H. C. Brown, and M. T. Chu, "On least squares Euclidean distance matrix approximation and completion," Department of Mathematics, North Carolina State University, Tech. Rep., 2003. [Online]. Available: <http://www4.ncsu.edu/~mtchu/Research/Papers/distance03.pdf>
- [23] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.
- [24] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre, "Low-rank optimization on the cone of positive semidefinite matrices," *SIAM Journal on Optimization*, vol. 20, no. 5, pp. 2327–2351, 2010.
- [25] B. Vandereycken, P.-A. Absil, and S. Vandewalle, "Embedded geometry of the set of symmetric positive semidefinite matrices of fixed rank," in *Proceedings of the IEEE 15th Workshop on Statistical Signal Processing*, 2009, pp. 389–392.
- [26] S. Bonnabel and R. Sepulchre, "Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank," *SIAM J. Matrix Anal. Appl.*, vol. 31, pp. 1055–1070, August 2009.
- [27] G. Meyer, S. Bonnabel, and R. Sepulchre, "Regression on fixed-rank positive semidefinite matrices: a riemannian approach," *Journal of Machine Learning Research*, vol. 12 (Feb), 2010.
- [28] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer, 2006.
- [29] P.-A. Absil, C. G. Baker, and K. A. Gallivan, "Trust-region methods on riemannian manifolds," *Foundations of Computational Mathematics*, vol. 7, no. 3, pp. 303–330, 2007.
- [30] S. Burer and R. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Mathematical Programming*, vol. 95, no. 2, pp. 329–357, Feb. 2003.