Notice: This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.
Localization in the Internet of Things Network:  
A Low-Rank Matrix Completion Approach  

Luong Trung Nguyen, Sangtae Kim, Byonghyo Shim  
Information System Laboratory  
Department of Electrical and Computer Engineering, Seoul National University  
Email: \{ltnguyen,stkim,bshim\}@islab.snu.ac.kr

Abstract  

Location awareness, providing ability to identify the location of sensor, machine, vehicle, and wearable device, is a rapidly growing trend of hyper-connected society and one of key ingredients for internet of things (IoT). In order to make a proper reaction to the collected information from devices, location information of things should be available at the data center. One challenge for the massive IoT networks is to identify the location map of whole sensor nodes from partially observed distance information. This is especially important for massive sensor networks, relay-based and hierarchical networks, and vehicular to everything (V2X) networks. The primary goal of this paper is to propose an algorithm to reconstruct the Euclidean distance matrix (and eventually the location map) from partially observed distance information. By casting the low-rank matrix completion problem into the unconstrained minimization problem in Riemannian manifold in which a notion of differentiability can be defined, we are able to solve the low-rank matrix completion problem efficiently using a modified conjugate gradient algorithm. From the analysis and numerical experiments, we show that the proposed method, termed localization in Riemannian manifold using conjugate gradient (LRM-CG), is effective in recovering the Euclidean distance matrix for both noiseless and noisy environments.

This work was sponsored by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (2016R1A2B3015576) and Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [16ZI1100, Wireless Transmission Technology in Multi-point to Multi-point Communications].  
A part of this paper was presented at the Information Theory and Applications (ITA) workshop, 2016 [1].
I. INTRODUCTION

Recently, Internet of Things (IoT) has received much attention for its plethora of applications, such as healthcare, surveillance, automatic metering, and environmental monitoring. In sensing the environmental data (e.g. temperature, humidity, pollution density, and object movements), wireless sensor network consisting of hundreds to thousands sensor nodes is popularly used \[2, 3, 4\]. In order to make a proper reaction to the collected environmental data, location information of sensor nodes should be available at the data center (base station) \[5, 6\]. Since actions in IoT networks, such as fire alarm, energy transfer, and emergency request, are made primarily on the data center, an approach to identify the location information of whole nodes at the data center has been received much attention. In this approach, henceforth referred to as centralized localization, each sensor node measures the distance information of adjacent nodes and then sends it to the data center. Then the data center constructs a map of sensor nodes using the collected distance information \[7\]. In measuring the distance, various modalities, such as received signal strength indication (RSSI) \[8\], time difference of arrival (TDoA) \[9\], and angle of arrival (AoA) \[4\], have been popularly used. These approaches are simple and effective in measuring the short-range distance and can also be used for indoor environments.

When we consider the centralized localization in IoT, there are two major issues to be addressed. First, location information of the sensor node obtained by this approach is local, meaning that the location information is true in the relative sense. Thus, proper adjustment of the location information is needed to identify the absolute (true) location information. In fact, since the local location of a sensor might be different from the absolute location by some combinations of translations, rotations, and reflections, absolute locations of a few sensor nodes (anchor nodes) are needed to transform the local locations into the absolute locations \[5, 6\]. It has been shown that when the number of anchor nodes is enough (e.g., four or more anchor nodes in $\mathbb{R}^2$), one can identify the absolute location of sensor nodes \[10\]. For the sake of completeness, we provide a brief summary of the standard procedure to obtain the absolute location from the pairwise distance information. Let $x_i \in \mathbb{R}^{k \times 1}$ ($i = 1, \ldots, n$) be the absolute locations of $n$ sensor nodes randomly distributed in $k$-dimensional Euclidean space (typically $k = 2$ or $3$) and $d_{ij} = \|x_i - x_j\|_2$ be the pairwise distance between the sensor nodes $i$ and $j$, August 4, 2017 DRAFT
then the Euclidean distance matrix $D$ is defined as

$$D = \begin{bmatrix}
0 & d_{12}^2 & d_{13}^2 & \cdots & d_{1n}^2 \\
d_{21}^2 & 0 & d_{23}^2 & \cdots & d_{2n}^2 \\
d_{31}^2 & d_{32}^2 & 0 & \cdots & d_{3n}^2 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
d_{n1}^2 & d_{n2}^2 & d_{n3}^2 & \cdots & 0
\end{bmatrix}.$$ 

Without loss of generality, we set the first sensor node as a reference in the local coordinate system. Then a local location $\tilde{x}_i$ of sensor node $i$ is $\tilde{x}_i = x_i - x_1$ and the corresponding local location matrix is

$$\tilde{X} = \begin{bmatrix}
\tilde{x}_1 & \tilde{x}_2 & \tilde{x}_3 & \cdots & \tilde{x}_n
\end{bmatrix}^T = X - 1x_1^T, \quad (1)$$

where $1 = [1, 1, \ldots, 1]^T \in \mathbb{R}^{n \times 1}$ and $X = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^{n \times k}$ is the absolute location matrix of sensor nodes. By forming a Gramian matrix of $\tilde{X}$, one can easily find a relationship between the entries of $D$ and $\tilde{X}$. To be specific, by denoting $\tilde{Y} = \tilde{X}\tilde{X}^T$, we have

$$\tilde{y}_{ij} = \tilde{x}_i^T\tilde{x}_j = (x_i - x_1)^T(x_j - x_1) = \frac{1}{2}(\|x_j - x_1\|_2^2 + \|x_i - x_1\|_2^2 - \|x_i - x_j\|_2^2) = \frac{1}{2}(d_{i1}^2 + d_{j1}^2 - d_{ij}^2),$$

where $\tilde{y}_{ij}$ is the $(i, j)$-th entry of $\tilde{Y}$. One can easily show that

$$\tilde{Y} = \frac{1}{2}(D e_1 e_1^T + e_1^T D - D),$$

where $e_1 = [1, 0, \ldots, 0]^T$. Since $z^T \tilde{Y} z = \sum_{i,j} z_i \tilde{y}_{ij} z_j = \|z_i \tilde{x}_i\|_2^2 \geq 0$ for all vector $z$, $\tilde{Y}$ is positive semi-definite (PSD) matrix. If we express its eigendecomposition as $\tilde{Y} = Q \Lambda Q^T$, then the matrix square root $\hat{X} = Q \Lambda^{1/2}$ becomes a local location matrix of sensor nodes. Finally, when the absolute locations of a few anchor nodes are provided, absolute locations of all sensor nodes can be recovered from $\hat{X}$. Readers are referred to [5], [6] for more details.

Second and perhaps more serious issue is that the data center does not have enough distance information to identify the locations of sensor nodes. For various reasons, such as the power
outage of a sensor node or the limitation of radio communication range, only partial distance information is available at the data center. This situation can also happen in the hierarchical or relay-based IoT networks where an intermediate node sends partial distance information to the data center. Also, in the vehicle networks it might not be possible to measure the distance of all adjacent vehicles when a vehicle is located at the dead zone. To illustrate this scenario, we consider a sensor network consisting of five sensor nodes in Fig. 1. We see that only a small number of pairwise distances is measured, and hence there are many unknown entries in the

Fig. 1. Sensor nodes deployed to measure not only environment information but also their pairwise distances. The observed distances are represented by two-sided arrows. The shadow areas represent the radio communication range of the sensor nodes.
observation matrix $D_{\text{obs}}$:

$$D_{\text{obs}} = \begin{bmatrix}
0 & d_{12}^2 & d_{13}^2 & ? & ? \\
d_{21}^2 & 0 & ? & ? & ? \\
d_{31}^2 & ? & 0 & d_{34}^2 & d_{35}^2 \\
? & ? & d_{43}^2 & 0 & d_{45}^2 \\
? & ? & d_{53}^2 & d_{54}^2 & 0
\end{bmatrix},$$

where the question mark $?$ indicates unknown entries of $D$. In general, one cannot recover the original matrix $D$ from the knowledge of a subset of its entries since there are infinitely many completion options for the unknown entries. However, it is now well-known that even though we have only partial measurements on $D$, as long as $D$ is a low-rank matrix, then $D$ might be recovered from $D_{\text{obs}}$ \cite{11, 12}. As will be discussed in the next section, rank of $D$ is at most $k + 2$ and $k$ is 2 or 3 for most cases. Thus, the Euclidean distance matrix $D$ can be readily considered as a low-rank matrix. The problem to recover a low-rank matrix $D$ from the small number of known entries is expressed as

$$\min_{\tilde{D} \in \mathbb{R}^{n \times n}} \text{rank}(\tilde{D}),$$

s.t. $\mathcal{P}_E(\tilde{D}) = \mathcal{P}_E(D_{\text{obs}})$.

(2)

where $\mathcal{P}_E$ is the sampling operator given by

$$[\mathcal{P}_E(A)]_{ij} = \begin{cases}
A_{ij} & \text{if } (i, j) \in E \\
0 & \text{otherwise}.
\end{cases}$$

Let $r$ be the radio communication range, then $E = \{(i, j) : \|x_i - x_j\|_2 \leq r\}$ would be the set of observed indices. Since solving (2) is numerically infeasible due to the non-convexity of the rank function, many efforts have been made over the years to relax this problem into a tractable form to solve. Candes and Recht showed that a low-rank matrix can be recovered from partial measurements by using the nuclear norm minimization (NNM) \cite{12}. The nice feature of this approach is that the NNM can be cast as a semidefinite programming (SDP) and thus can be solved via a polynomial time complexity algorithm \cite{12, 13}. However, computational overhead is still burdensome and hardly scale to the problem size.

An alternative approach is to use the Frobenius-norm minimization given by

$$\min_{\tilde{D} \in \mathbb{R}^{n \times n}} \frac{1}{2} \|\mathcal{P}_E(\tilde{D}) - \mathcal{P}_E(D_{\text{obs}})\|_F^2,$$

s.t. $\text{rank}(\tilde{D}) \leq \eta$.

(3)
where $\eta$ is the rank of the original matrix. Since the equality constraint in (2) is replaced by the Frobenius norm-based cost function, this approach is suited for the noisy scenario. In fact, since an addition of noise is unavoidable in the distance measurement process of the IoT networks, an approach robust to the measurement noise is desired. Also, this approach is good fit for the situation where the rank constraint is known a priori. In recent years, various approaches to find a solution of (3) have been suggested. In [14], [15], approaches inspired by the greedy recovery strategy of the compressed sensing (CS) have been proposed. In these approaches, the set of rank-one matrices that best represents the original matrix is found by the iterative process. In [16], [17], alternating least squares (ALS) technique has been suggested. In this technique, a low-rank matrix is factorized into the product of lower dimensional matrices and then the problem is solved via an alternative minimization. In [18], matrix completion problem is modeled as an unconstrained optimization problem over the smooth Riemannian manifold and a nonlinear conjugate gradient method has been used to solve the problem.

The main goal of this paper is to propose a Euclidean distance matrix completion technique optimized for IoT localization. Instead of solving the Frobenius norm minimization problem in (3) as it is, we express the Euclidean distance matrix $D$ as a function of the fixed rank positive semidefinite matrix. Since the set of these matrices forms a Riemannian manifold in which the notation of differentiability can be defined, we can recycle, after a proper modifications, an optimization algorithm in the Euclidean space. In fact, in order to solve the Frobenius-norm minimization problem, we propose a modified conjugate gradient algorithm, referred to as localization in Riemannian manifold using conjugate gradient (LRM-CG), whose solution lies on the Riemannian manifold. We show from the recovery condition analysis that the sequence generated by LRM-CG converges to the original Euclidean distance matrix under suitable conditions. We also show from numerical experiments that LRM-CG is effective in recovering the Euclidean distance matrix from partial measurements and also scalable to the dimension of the matrix.

We briefly summarize notations used in this paper. $\mathbb{P}^{n \times n}$ denotes the set of $n \times n$ symmetric positive semidefinite (PSD) matrices. $\langle \beta_1, \beta_2 \rangle$ is the inner product between two vectors (or matrices) $\beta_1$ and $\beta_2$, i.e., $\langle \beta_1, \beta_2 \rangle = \text{tr}(\beta_1^T \beta_2)$. diag$(A)$ is the vector formed by the main diagonal of a matrix $A$. Sym$(A)$ and Skew$(A)$ are the matrices formed by Sym$(A) = \frac{1}{2}(A + A^T)$ and Skew$(A) = \frac{1}{2}(A - A^T)$ for any square matrix $A$, respectively. Note that $A = \text{Sym}(A) + \text{Skew}(A)$. eye$(a)$ is a diagonal matrix whose diagonal entries are elements of $a$. For an orthogonal
matrix $Q \in \mathbb{R}^{n \times k}$ with $n > k$, we define its orthogonal complement $Q_\perp \in \mathbb{R}^{n \times (n-k)}$ such that 

$$
\begin{bmatrix}
Q & Q_\perp
\end{bmatrix}
$$

forms an orthonormal matrix. Given a function $f : Y \in \mathbb{R}^{n \times n} \to f(Y) \in \mathbb{R}$, $\nabla_Y f(Y)$ is the Euclidean gradient of $f(Y)$ with respect to $Y$, i.e., $[\nabla_Y f(Y)]_{ij} = \frac{\partial f(Y)}{\partial y_{ij}}$. For a given matrix $A = [a_1 \ a_2 \ \cdots \ a_n]^T \in \mathbb{R}^{n \times n}$, the vectorization of $A$, denoted by vec($A$), is defined as

$$
\text{vec}(A) = [a_1^T \ a_2^T \ \cdots \ a_n^T]^T.
$$

Let $f : A \to f(A)$ and $g : B \to g(B)$ be mappings, then the composite $g \circ f$ of $f$ and $g$ is defined by $g \circ f : A \to (g \circ f)(A) = g(f(A))$.

$E^{(i,j)}$ is the standard element of $\mathbb{R}^{n \times n}$ whose the $(i,j)$-th element is one and the rest are zeros. For example, $E^{(1,2)} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ for $\mathbb{R}^{2 \times 2}$.

II. THE LRM-CG ALGORITHM

In this section, we present the proposed LRM-CG algorithm. By exploiting the smooth Riemannian manifold structure for the set of the low-rank symmetric PSD matrices, we formulate the matrix completion problem as an unconstrained optimization problem on the smooth Riemannian manifold. Roughly speaking, smooth manifold is a generalization of the Euclidean space on which a notion of differentiability exists. For more rigorous definition, see, e.g., [19], [20]. A smooth manifold together with an inner product, often called a Riemannian metric, forms a smooth Riemannian manifold. Since the smooth Riemannian manifold is a differentiable structure equipped with an inner product, we can use all necessary ingredients for solving optimization problems with quadratic cost function, such as Riemannian gradient, Hessian matrix, exponential map, and parallel translation [19]. Therefore, optimization techniques in Euclidean vector space (e.g., steepest descent, Newton method, conjugate gradient method) can be readily applied for solving a problem in the smooth Riemannian manifold.

A. Problem Model

From the definition of pairwise distance $d^2_{ij} = \|x_i - x_j\|^2_2 = x_i^T x_i + x_j^T x_j - 2x_i^T x_j$, we have

$$
D = g(XX^T),
$$

where $g(XX^T) = 2\text{Sym}(\text{diag}(XX^T)1^T - XX^T)$. In the example illustrated in Fig. [1] $X$ is

$$
X = \begin{bmatrix}
x_1 & x_2 & x_3 & x_4 & x_5
\end{bmatrix}^T
= \begin{bmatrix}
7 & 2 & 11 & 12 & 15 \\
9 & 7 & 7 & 4 & 6
\end{bmatrix}^T,
$$

August 4, 2017 DRAFT
and $D$ is

$$D = g(XX^T) = \begin{bmatrix}
0 & 29 & 20 & 50 & 73 \\
29 & 0 & 81 & 109 & 170 \\
20 & 81 & 0 & 10 & 17 \\
50 & 109 & 10 & 0 & 13 \\
73 & 170 & 17 & 13 & 0 \\
\end{bmatrix}. \quad (6)$$

The next lemma follows immediately from (4).

**Lemma 2.1:** If $n$ sensor nodes are distributed in $k$-dimensional Euclidean space and $n \geq k$, then $\text{rank}(D) \leq k + 2$.

**Proof:** From (4), we have $\text{rank}(D) = \text{rank}(g(XX^T))$, which gives

$$\text{rank}(D) \leq \text{rank}(2 \frac{1}{2} \text{diag}(XX^T)^T + \text{diag}(XX^T)1^T) + \text{rank}(2XX^T) \leq 2 \text{rank}(1 \text{diag}(XX^T)^T) + \text{rank}(XX^T) \leq 2 + k,$$

where the last inequality is because $\text{rank}(XX^T) = \text{rank}(X) \leq k$ and $\text{rank}(ab^T) \leq 1$ for any two vectors $a$ and $b$. 

From this lemma, (3) can be rewritten as

$$\min_{D \in \mathbb{R}^{n \times n}} \frac{1}{2} \| P_E(\tilde{D}) - P_E(D_{\text{obs}}) \|_F^2, \quad \text{s.t.} \quad \text{rank}(\tilde{D}) \leq k + 2. \quad (7)$$

Since $\text{rank}(D) = \text{rank}(g(\tilde{X}\tilde{X}^T)) \leq k + 2$ for any $\tilde{X}$, we can further simplify the problem as

$$\min_{\tilde{X} \in \mathbb{R}^{n \times k}} \frac{1}{2} \| P_E(g(\tilde{X}\tilde{X}^T)) - P_E(D_{\text{obs}}) \|_F^2. \quad (8)$$

Recalling that $Y = \tilde{X}\tilde{X}^T$, we have

$$\min_{Y \in \mathcal{Y}} \frac{1}{2} \| P_E(g(Y)) - P_E(D_{\text{obs}}) \|_F^2, \quad (9)$$

where $\mathcal{Y} = \{\tilde{X}\tilde{X}^T : \tilde{X} \in \mathbb{R}^{n \times k}\}$.

When the sensor nodes are randomly distributed in $k$-dimensional Euclidean space, rank of the location matrix $X$ is $k$ almost surely. Thus, we can strengthen the constraint set from $\mathcal{Y}$ to

\[1\text{Consider the case that sensor nodes are randomly distributed in 2D Euclidean space, then rank}(X) = 1 \text{ if and only if all of nodes are collinear. This event happens if there exists a constant } \rho \text{ such that } x_{i1} = \rho x_{i2} \text{ for any } i\text{-th row. The probability of this event } \prod_{i=1}^{n} P(x_{i1} = \rho x_{i2}) = [P(x_{i1} = \rho x_{i2})]^n \text{ becomes negligible when the number of sensor nodes are sufficiently large.}
Fig. 2. Illustration of (a) the tangent space \( T_Y \tilde{Y} \) and (b) the retraction operator \( R_Y \) at a point \( Y \) in the embedded manifold \( \tilde{Y} \).

\[
\tilde{Y} = \{ \tilde{X} \tilde{X}^T : \tilde{X} \in \mathbb{R}^{n \times k}, \text{rank}(\tilde{X}) = k \},
\]

and as a result, we have

\[
\min_{Y \in \tilde{Y}} \frac{1}{2} \| P_E(g(Y)) - P_E(D_{obs}) \|_F^2.
\]

(10)

In the sequel, we denote \( f(Y) = \frac{1}{2} \| P_E(g(Y)) - D_{obs} \|_F^2 \) for notational simplicity.

**B. Optimization over Riemannian Manifold**

Let \( S = \{ Q \in \mathbb{R}^{n \times k} : Q^T Q = I_k \}^2 \) and \( \mathcal{L} = \{ \text{eye}(\lambda_1 \cdots \lambda_k)^T : \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k > 0 \} \).

Then, for given \( Y \in \tilde{Y} \), one can express \( Y = Q \Lambda Q^T \), and thus an alternative representation of \( \tilde{Y} \) is

\[
\tilde{Y} = \{ Q \Lambda Q^T : Q \in S, \Lambda \in \mathcal{L} \}.
\]

(11)

It has been shown that \( \tilde{Y} \) is a smooth Riemannian manifold [21, Ch.5] [22]. Our approach to solve the problem in a smooth Riemannian manifold is beneficial in two major respects: First, one can easily compute the gradient of the cost function in (10) using the matrix calculus. Second, we can use an algorithm in the Euclidean space to solve the problem (10).

\footnote{S is an orthogonal Stiefel manifold embedded in \( \mathbb{R}^{n \times k} \) [20, Theorem 5.12].}
Since our work relies to a large extent on properties and operators of differential geometry, we briefly introduce tools and ingredients to describe the proposed algorithm. Since \( \widetilde{Y} \) is an embedded manifold in the Euclidean space \( \mathbb{R}^{n \times n} \), its tangent spaces are determined by the derivative of its curves, where the curve \( \gamma \) of \( \widetilde{Y} \) is a mapping from \( \mathbb{R} \) to \( \widetilde{Y} \). Put it formally, for a given point \( Y \in \widetilde{Y} \), the tangent space of \( \widetilde{Y} \) at \( Y \), denoted \( T_Y \widetilde{Y} \), is defined as \( T_Y \widetilde{Y} = \{ \gamma'(0) : \gamma \text{ is a curve in } \widetilde{Y}, \gamma(0) = Y \} \) (see Fig. 2). In the following lemma, we characterize the tangent space of \( \widetilde{Y} \).

**Lemma 2.2:** For the manifold \( \widetilde{Y} \) defined by (11), the tangent space \( T_Y \widetilde{Y} \) at \( Y \) is

\[
T_Y \widetilde{Y} = \left\{ \begin{bmatrix} Q & Q_x \\ B & C^T \\ Q_y^T \\ C & 0 \end{bmatrix} : B \in \mathbb{R}^{k \times k}, C \in \mathbb{R}^{(n-k) \times k} \right\}.
\]

**Proof:** See Appendix A.

A metric on the tangent space \( T_Y \widetilde{Y} \) is defined as the matrix inner product \( \langle B_1, B_2 \rangle = \text{tr}(B_1^T B_2) \) between two tangent vectors \( B_1, B_2 \in T_Y \widetilde{Y} \). We next define the orthogonal projection of a matrix \( A \) onto the tangent space \( T_Y \widetilde{Y} \), which will be used to find the closed-form expression of Riemannian gradient in Subsection II-C.

**Definition 2.3:** The orthogonal projection onto \( T_Y \widetilde{Y} \) is a mapping \( P_{T_Y \widetilde{Y}} : \mathbb{R}^{n \times n} \rightarrow T_Y \widetilde{Y} \) such that for a given matrix \( A \in \mathbb{R}^{n \times n} \), \( \langle A - P_{T_Y \widetilde{Y}}(A), B \rangle = 0 \) for all \( B \in T_Y \widetilde{Y} \).

The following proposition provides a closed form expression of the orthogonal projection operator.

**Proposition 2.4:** For a given matrix \( A \), orthogonal projection \( P_{T_Y \widetilde{Y}}(A) \) of \( A \) onto the tangent space \( T_Y \widetilde{Y} \) is

\[
P_{T_Y \widetilde{Y}}(A) = P_Q \text{Sym}(A) + \text{Sym}(A)P_Q - P_Q \text{Sym}(A)P_Q,
\]

where \( P_Q = QQ^T \).

**Proof:** See Appendix B.

In order to express the concept of moving in the direction of a tangent space while staying on the manifold, an operation called *retraction* is used. As illustrated in Fig. 2(b), the retraction operation is a mapping from \( T_Y \widetilde{Y} \) to \( \widetilde{Y} \) that preserves the gradient at \( Y \) [23, Definition 4.1.1].

---

3Tangent vector on the smooth Riemannian manifold is a generalization of the notion of tangent vector to curves and surfaces in Euclidean space. A collection of all tangent vectors at each point of smooth manifold forms a tangent space at that point, which is indeed a vector space.
Definition 2.5: The retraction $R_Y(B)$ of a vector $B \in T_Y\tilde{Y}$ onto $\tilde{Y}$ is defined as

$$R_Y(B) = \arg\min_{Z \in \tilde{Y}} \|Y + B - Z\|_F. \quad (14)$$

In obtaining the closed form expression of $R_Y(B)$, an operator $W_k$ keeping $k$ largest positive eigenvalues of a matrix, referred to as eigenvalue selection operator, is needed. Since the projection $R_Y(B)$ is an element of $\tilde{Y}$, $R_Y(B)$ should be a symmetric PSD matrix with rank $k$. Thus, for given square matrix $A$, we are only interested in the symmetric part $\text{Sym}(A)$. If we denote the eigenvalue decomposition (EVD) of this as $\text{Sym}(A) = P\Sigma P^T$ and the $k$ top most eigenvalues of this as $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0$, then $W_k(A)$ is defined as

$$W_k(A) = P\Sigma_k P^T, \quad (15)$$

where $\Sigma_k = \text{eye}(\begin{bmatrix} \sigma_1 & \cdots & \sigma_k & 0 & \cdots & 0 \end{bmatrix}^T)$. $R_Y(B)$ is concisely expressed using the eigenvalue selection operator $W_k$.

Lemma 2.6:

$$R_Y(B) = W_k(Y + B). \quad (16)$$

Proof: See Appendix C.

Finally, to develop the conjugate gradient algorithm over the Riemannian manifold $\tilde{Y}$, we need the Euclidean gradient of the cost function $f(Y)$.

Lemma 2.7: Euclidean gradient $\nabla_Y f(Y)$ of $f(Y)$ with respect to $Y$ is

$$\nabla_Y f(Y) = 2\text{eye}(\text{Sym}(R))1 - 2\text{Sym}(R), \quad (17)$$

where $R = \mathcal{P}_E(g(Y)) - \mathcal{P}_E(D_{obs})$.

Proof: See Appendix D.

C. Localization in Riemannian Manifold Using Conjugate Gradient (LRM-CG)

In order to solve the problem in (10), we use the conjugate gradient (CG) method. CG method is widely used to solve sparse symmetric positive definite linear systems [24].
Fig. 3. Riemannian gradient $\nabla_Y f(Y)$ is defined as the projection of the Euclidean gradient $\nabla_Y f(Y)$ onto the tangent space $T_Y \tilde{Y}$ while the Euclidean gradient is a direction for which the cost function is reduced most in $\mathbb{R}^{n \times n}$. Riemannian gradient is the direction for which the cost function is reduced most in the tangent space $T_Y \tilde{Y}$.

First, noting that $P_E$ and $g$ are linear mappings, one can easily show that

$$f(Y) = \frac{1}{2} \|P_E(g(Y)) - D_{obs}\|^2_F$$

$$= \frac{1}{2} \|P_E \left( g \left( \sum_{i,j} y_{ij} E^{(i,j)} \right) \right) - D_{obs}\|^2_F$$

$$= \frac{1}{2} \| \sum_{i,j} y_{ij} P_E(g(E^{(i,j)})) - D_{obs}\|^2_F$$

$$\overset{(a)}{=} \frac{1}{2} \left\| \sum_{i,j} y_{ij} \text{vec} \left( P_E(g(E^{(i,j)})) \right) - \text{vec}(D_{obs}) \right\|^2_2$$

$$\overset{(b)}{=} \frac{1}{2} \|A \text{vec}(Y) - b\|^2_2,$$

(18)

where (a) is because $\|M\|_F = \|\text{vec}(M)\|_2$ and (b) follows from $\text{vec}(Y) = \left[ \begin{array}{cccc} y_{11} & y_{21} & \cdots & y_{nn} \end{array} \right]^T$, $b = \text{vec}(D_{obs})$, and $A = \left[ \begin{array}{ccc} \text{vec} \left( P_E(g(E^{(1,1)})) \right) & \cdots & \text{vec} \left( P_E(g(E^{(n,n)})) \right) \end{array} \right]$.

In (18), we see that the cost function $f(Y)$ has the quadratic form of a sparse symmetric positive definite system, and thus the CG algorithm would be an efficient means to solve the
problem. The update equation of the conventional CG algorithm in the Euclidean space is

\[ Y_{i+1} = Y_i + \alpha_i P_i, \quad (19) \]

where \( \alpha_i \) is the stepsize and \( P_i \) is the conjugate direction. Note that the stepsize \( \alpha_i \) is chosen by the line minimization technique (e.g., Armijo’s rule \cite{25}, \cite{26}) and the search direction \( P_i \) of the CG algorithm is chosen as a linear combination of the gradient and the previous search direction to generate a direction which are conjugate to the previous ones. In doing so, one can avoid unnecessary searching of directions that have been searched over and achieve the speedup in the algorithm \cite{27}, \cite{24}.

Since we consider the optimization problem over the Riemannian manifold \( \tilde{Y} \), the conjugate direction \( P_i \) lies on the tangent space. Thus, to make sure that the update point \( Y_{i+1} \) lies on the manifold, we need a retraction operator. The update equation after applying the retraction operation is

\[ Y_{i+1} = R_{Y_i}(\alpha_i P_i) \]
\[ = W_k(Y_i + \alpha_i P_i). \quad (20) \]

As studied in Lemma \cite{26} the eigenvalue selection operator \( W_k \) guarantees that the updated point \( Y_{i+1} \) lies on the manifold.

We next consider the conjugate direction \( P_i \) of LRM-CG. In the conventional nonlinear CG algorithm, conjugate direction \( P_i \) is updated as

\[ P_i = -\nabla_Y f(Y_i) + \beta_i P_{i-1}, \quad (21) \]

where \( \beta_i \) is the conjugate update parameter.\footnote{There are a number of ways to choose \( \beta_i \). See, e.g., \cite{24}, \cite{28}, \cite{29}, \cite{30}.} Since we optimize over the Riemannian manifold \( \tilde{Y} \), (21) needs to be modified. First, we need to use the Riemannian gradient of \( f(Y) \) instead of the Euclidean gradient \( \nabla_Y f(Y) \) since we find the search direction on the tangent space of \( \tilde{Y} \). Riemannian gradient, denoted \( \text{grad} f(Y) \), is distinct from \( \nabla_Y f(Y) \) in the sense that it is defined on the tangent space \( T_Y \tilde{Y} \) (see Fig. \cite{3}).

**Definition 2.8:** Let \( f \) be the function differentiable everywhere in the Riemannian manifold \( \tilde{Y} \). The Riemannian gradient \( \text{grad} f(Y) \) of \( f \) at \( Y \) is defined as the unique element in \( T_Y \tilde{Y} \).
satisfying

\[ \langle B, \nabla f(Y) \rangle = \langle B, \nabla_Y f(Y) \rangle, \tag{22} \]

where \( B \) is any element in \( T_Y \hat{\mathcal{Y}} \).

As shown in Fig. 3, \( \nabla f(Y) \in T_Y \hat{\mathcal{Y}} \) is a component of the Euclidean gradient \( \nabla_Y f(Y) \) in \( T_Y \hat{\mathcal{Y}} \). In other words, \( \nabla f(Y) \) is the projection of \( \nabla_Y f(Y) \) onto \( T_Y \hat{\mathcal{Y}} \). Indeed, from Definition 2.3, \( \langle B, \nabla_Y f(Y) - P_{T_Y \hat{\mathcal{Y}}} (\nabla_Y f(Y)) \rangle = 0 \) for any matrix \( B \in T_Y \hat{\mathcal{Y}} \). Hence,

\[ \langle B, P_{T_Y \hat{\mathcal{Y}}} (\nabla_Y f(Y)) \rangle = \langle B, \nabla_Y f(Y) \rangle. \tag{23} \]

From (22) and (23), it is clear that

\[ \nabla f(Y) = P_{T_Y \hat{\mathcal{Y}}} (\nabla_Y f(Y)). \tag{24} \]

Second, since the Riemannian gradient \( \nabla f(Y_i) \) and previous conjugate direction \( P_{i-1} \) lie on two different vector spaces \( T_{Y_i} \hat{\mathcal{Y}} \) and \( T_{Y_{i-1}} \hat{\mathcal{Y}} \), we need to project \( P_{i-1} \) onto the tangent space \( T_{Y_i} \hat{\mathcal{Y}} \) before performing a linear combination between of two. In view of this, the conjugate direction update equation of LRM-CG is

\[ P_i = -\nabla f(Y_i) + \beta_i P_{T_{Y_i} \hat{\mathcal{Y}}} (P_{i-1}). \tag{25} \]

Finally, in choosing the stepsize \( \alpha_i \) in (20), we use the Armijo’s rule, a widely used line search strategy. Note that the Armijo’s rule is a simple yet effective way to find a stepsize \( \alpha_i \) minimizing the cost function \( f \), that is, \( \alpha_i \approx \min_{\alpha > 0} f(W_k (Y_i + \alpha_i P_i)) \) [25], [26].

The proposed LRM-CG algorithm is summarized in Algorithm 1.

D. Numerical Experiments

In this subsection, we investigate numerical performance of the proposed LRM-CG for both noiseless and noisy scenarios. In our simulation, we compare LRM-CG with following matrix completion algorithms:

---

5In transforming a vector from one tangent space to another, an operator called vector transport is used (see Definition 8.1.1 in [19]). For an embedded manifold of \( \mathbb{R}^{n \times n} \), vector transport is the orthogonal projection operator [19]. Hence, the vector transport of \( P_{i-1} \) is the orthogonal projection of \( P_{i-1} \) onto \( T_{Y_i} \hat{\mathcal{Y}} \).

6One can also find the matlab source code at [http://islab.snu.ac.kr/publication.html](http://islab.snu.ac.kr/publication.html).
Algorithm 1: LRM-CG algorithm

1 Input: $D_{obs}$: the observed matrix, $P_E$: the sampling operator, $\epsilon$: tolerance, $\mu \in (0, 1)$: given constant, $T$: number of iterations.

2 Initialize: $i = 1$, $Y_1 \in \tilde{Y}$: initial point, $P_1$: initial conjugate direction.

3 While $i \leq T$ do

4 $R_i = P_E(g(Y_i)) - P_E(D_{obs})$ // Generate residual matrix

5 $\nabla_{Y f}(Y_i) = 2\text{eye}(\text{Sym}(R_i))1 - 2R_i$ // Compute Euclidean gradient

6 $\text{grad}_f(Y_i) = P^T_Y Y_i \tilde{Y}(\nabla_{Y f}(Y_i))$ // Compute Riemannian gradient

7 $H_i = \text{grad}_f(Y_i) - P^T_Y Y_i \tilde{Y}(\text{grad}_f(Y_{i-1}))$

8 $h = \langle P_i, H_i \rangle$

9 $\beta_i = \frac{1}{\kappa_i} < hH_i - 2P_i\|H_i\|^2, \text{grad}_f(Y_i) >$ // Compute CG coefficient

10 $P_i = -\text{grad}_f(Y_i) + \beta_i P_{TY_i} \tilde{Y}(P_{i-1})$ // Compute conjugate direction

11 Find a stepsize $\alpha_i > 0$ such that $f(Y_i) - f(R_{Y_i}(\alpha_i P_i)) \geq -\mu\alpha_i < \text{grad}_f(Y_i), P_i >$

12 $Y_{i+1} = R_{Y_i}(\alpha_i P_i)$ // Perform retraction

13 $D_{i+1} = g(Y_{i+1})$ // Compute updated Euclidean distance matrix

14 If $\|P_E(D_{i+1}) - P_E(D_{obs})\|_F < \epsilon$ then

15 Exit from while loop

16 End If

17 Obtain $Q$ and $\Lambda$ using the eigendecomposition

$Y_{i+1} = QAQ^T$

18 $\hat{X} = QA^{1/2}$ // Find updated locations of sensor nodes

19 $i = i + 1$

20 End While

21 Output: $\hat{X}$

August 4, 2017 DRAFT
• ADMiRA [14]: this algorithm uses greedy projection to identify a set of rank-one matrices that best represents the original matrix.
• LMaFit [16]: this is a nonlinear successive over-relaxation matrix completion algorithm based on nonlinear Gauss-Seidel method.
• LRGeomCG [18]: this algorithm is essentially CG method defined over the Riemannian manifold of low rank matrices (but not necessarily positive definite).
• SDP [12]: this algorithm solves the NNM problem via semidefinite programming.
• TNN-ADMM [31]: as an improved version of the NNM problem, this algorithm solves the truncated NNM problem via an alternating direction method of multipliers.

In the noiseless scenario, we generate an \( n \times k \) location matrix \( X \) whose entries are sampled independently and identically from a uniform distribution in unit interval, and then map \( X \) into the Euclidean distance matrix \( D = g(XX^T) \). As mentioned, entries \( d_{ij} \) of \( D \) are known (observed) if \( d_{ij} \leq r \) (\( r \) is the radio communication range). In the noisy scenario, a noise matrix \( N \in \mathbb{R}^{n \times n} \) is added to the Euclidean distance matrix \( D \). Elements \( n_{ij} \) of \( N \) are sampled independently and identically from the Gaussian distribution with zero mean and variance \( \sigma^2 = \frac{\|D\|_F^2}{n(n-1)}10^{-\frac{SNR}{10}} \). In obtaining the performance for each algorithm, we perform at least 1000 independent trials.

1) Convergence Efficiency: As a performance measures, we use two types of the mean square errors (MSE): MSE at sampled entries (\( MSE_s \)) and MSE at all entries (\( MSE_a \)), which are defined respectively as

\[
MSE_s = \frac{1}{|E|}\|P_E(\hat{D}) - P_E(D)\|_F^2,
\]

\[
MSE_a = \frac{1}{n^2}\|\hat{D} - D\|_F^2,
\]

where \(|E|\) is the cardinality of the sampling set \( E \) (see (2)). In Fig. 4 we plot the log-scale MSE as a function of the number of iterations for the 2-dimensional location vectors. Note that the performance results are obtained for the scenario where 200 sensor nodes randomly distributed. We observe that the log-scale MSE decreases linearly with the number of iteration, which in turn implies that the MSE decreases exponentially with the number of iterations. For example, if \( r = 0.5 \), it takes less than 25 iterations to achieve \( 10^{-2} \) and 53 iterations to achieve \( 10^{-4} \). Also, as expected, required number of iterations to achieve a given performance level gets smaller as \( r \) increases. To further examine the convergence speed of LRM-CG, we measured the running
time of LRM-CG for three distinct matrix dimensions (500×500, 1000×1000, and 5000×5000). The running time is measured using the MATLAB program on a personal computer (Intel Core i5 CPU with 3.4 GHz). As indicated in Table I, LRM-CG recovers 500×500 Euclidean distance matrix in a few seconds. Even for 1000×1000 matrix, it takes less than one minute to recover the matrix accurately.

2) Recovery Performance: We next investigate the recovery performance of LRM-CG for both noiseless and noisy scenarios. In Fig. 5 we plot the recovery performance of the noiseless scenario for \( k = 2 \) and 3 as a function of the sampling ratio, where the sampling ratio is defined as a ratio between the number of observed pairwise distances and total number of pairwise distances. We observe from the figure that LRM-CG performs well in all cases, achieving target performance with smaller number of measurements than conventional techniques requires. In Fig. 6 we plot the performance of LRM-CG as a function of the signal-to-noise ratio (SNR) for the noisy scenario. We observe that LRM-CG outperforms conventional approaches by a large margin and the gain increases sharply with SNR.
TABLE I. EXPERIMENTAL RESULTS FOR EUCLIDEAN DISTANCE MATRIX COMPLETION

| Size $(n \times n)$ | Euclidean dimension | $r$ | time (s) | Number of iterations | $MSE_a$ |
|----------------------|---------------------|-----|----------|----------------------|---------|
| $500 \times 500$     | 2                   | 0.5 | 9        | 90                   | $1.0 \times 10^{-6}$ |
|                      | 3                   | 0.7 | 10       | 92                   | $3.4 \times 10^{-6}$ |
| $1,000 \times 1,000$ | 2                   | 0.5 | 38       | 79                   | $1.3 \times 10^{-6}$ |
|                      | 3                   | 0.7 | 45       | 98                   | $1.4 \times 10^{-6}$ |
| $5,000 \times 5,000$ | 2                   | 0.5 | 1897     | 109                  | $4.6 \times 10^{-6}$ |
|                      | 3                   | 0.7 | 2542     | 126                  | $7.4 \times 10^{-6}$ |

Fig. 5. The MSE performance of the matrix completion algorithms for noiseless scenario for (a) 2-dimensional and (b) 3-dimensional location vectors.
Fig. 6. The MSE performance of the algorithms for noisy scenario for (a) 2-dimensional and (b) 3-dimensional location vectors.

E. Computational Complexity

In this subsection, we examine the computational complexity of LRM-CG in terms of the number of floating point operations (flops). As discussed in Section II-B, LRM-CG computes Euclidean gradient, Riemannian gradient, and the retraction in each iteration.

In order to compute the Euclidean gradient $\nabla Y f(Y_i)$ in (17), we first need to consider the computation of $Y_i$ from the $(i-1)$-th iteration. Since $Y_i = Q\Lambda Q^T$ ($Q$ is a $n \times k$ matrix and $\Lambda$ is a $k \times k$ diagonal matrix), it requires $2k$ multiplications and $(k-1)$ additions to compute $y_{ij} = \sum_{t=1}^{k} \lambda_t q_{it} q_{jt}$ so that the associated computational complexity is $(3k-1)$ flops. Further, from (4), we need to compute $[g(Y)]_{ij} = y_{ii} + y_{jj} - y_{ij}$, which requires $(9k-1)$ flops. The residual matrix $R_i = P_E(g(Y_i)) - D_{obs}$ requires $9k|E|$ flops ($|E|$ is the number of the observed entries of $D_{obs}$). In addition, since it takes $(9k+2)|E|$ flops to compute $\text{Sym}(R_i) = \frac{1}{2}(R_i + R_i^T)$, it requires at most $(9k+4)|E| + n - 1$ flops to compute $2\text{eye}(\text{Sym}(R_i)1)$. Since the cardinality of $\text{Sym}(R_i)$ is $|E|$, computational complexity of $\nabla_Y f(Y_i) = 2\text{eye}(\text{Sym}(R_i)1) - 2\text{Sym}(R_i)$ in (17) is at most $(9k+5)|E| + n - 1$. 

August 4, 2017 DRAFT
Second, recalling that the Riemannian gradient $\nabla f(Y_i)$ is an orthogonal projection of $\nabla_Y f(Y_i)$ onto the tangent space $T_{Y_i} \tilde{Y}$, we need to estimate the computational complexity of the orthogonal projection operator $P_{T_{Y_i} \tilde{Y}}$. In computing $P_{T_{Y_i} \tilde{Y}}(A)$ for an $n \times n$ matrix $A$, we need $\text{Sym}(A)$, $B = \text{Sym}(A)Q$, and $C = QT\text{Sym}(A)Q$, which require $(2k-1)n^2$, $2n^2 + (2n-1)kn$, and $(2n-1)kn + (2n-1)k^2$ flops, respectively. Then, from (13), we have

$$P_{T_{Y_i} \tilde{Y}}(A) = QQ^T\text{Sym}(A) + \text{Sym}(A)QQ^T - QQ^T\text{Sym}(A)QQ^T$$

which requires $O(kn^2 + k^2n + k^3)$ flops.

Finally, in applying Armijo’s rule to find the stepsize $\alpha_i$, we need to compute the retraction operation $R_{Y_i}(\alpha_iP_i)$. From (16), the retraction operation is obtained via the eigenvalue selection operator $\mathcal{W}_k$ and this requires the EVD of $(Y_i + P_i)$. In general, computational complexity of the EVD for a $n \times n$ matrix is expressed as $O(n^3)$. However, using $Y_i = QAQ^T$ and $P_i \in T_{Y_i} \tilde{Y}$, one can simplify the EVD operation. First, since $P_i \in T_{Y_i} \tilde{Y}$, we have

$$P_i = \begin{bmatrix} Q & Q \perp \end{bmatrix} \begin{bmatrix} B_i & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} Q^T \\ Q \perp \end{bmatrix}. \quad (26)$$

Thus,

$$Y_i + P_i = \begin{bmatrix} Q & Q \perp \end{bmatrix} \begin{bmatrix} B_i + \Lambda & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} Q^T \\ Q \perp \end{bmatrix} = \begin{bmatrix} Q & (Q \perp C_i) \end{bmatrix} \begin{bmatrix} B_i + \Lambda & I \\ I & 0 \end{bmatrix} \begin{bmatrix} Q^T \\ (Q \perp C_i)^T \end{bmatrix} = \begin{bmatrix} Q \perp C_i \end{bmatrix} K \Lambda' K^T \begin{bmatrix} Q^T \\ (Q \perp C_i)^T \end{bmatrix} = Q' \Lambda' Q'^T. \quad (27)$$

From (27), we see that the EVD of $(Y_i + P_i)$ is simplified to the EVD of the $2k \times 2k$ matrix

$$\begin{bmatrix} B_i + \Lambda & I \\ I & 0 \end{bmatrix},$$

which requires only $8k^3$ flops. Also, computing $Q \perp C_i$ and $Q' = \begin{bmatrix} Q & (Q \perp C_i) \end{bmatrix} K$ needs $(2n - 2k - 1)kn$ and $(4k - 1)kn$ flops, respectively. Thus, computational complexity of the retraction operation is $8k^3 + (2n - 2k - 1)kn + (4k - 1)kn$, which is $O(kn^2)$ for $k \ll n$. 
In summary, computational complexity of the proposed algorithm per iteration is $O(k|E| + kn^2 + k^2n + k^3) = O(kn^2)$. Since $k = 2$ or $3$ in our problem, complexity per iteration can be expressed as $O(n^2)$ flops.

III. RECOVERY CONDITION ANALYSIS

In this section, we analyze a recovery condition under which the LRM-CG algorithm recovers the Euclidean distance matrices accurately. Overall, our analysis is divided into two parts. In the first part, we analyze a condition ensuring the successful recovery of the sampled (observed) entries, i.e., $\|P_E(\hat{D}) - P_E(D)\|_F = 0$. In the second part, we investigate a condition guaranteeing the exact recovery of the Euclidean distance matrices, i.e., $\|\hat{D} - D\|_F = 0$. By exact recovery, we mean that the output $D_i$ of LRM-CG converges to the original Euclidean distance matrix $D$.

Definition 3.1: For a sequence of matrices $\{D_i\}_{i=1}^\infty$, if $\lim_{i \to \infty} \frac{\|D_{i+1} - D\|_F}{\|D_i - D\|_F} = \lambda$, we say $\{D_i\}_{i=1}^\infty$ converges linearly to $D$ with convergent rate $\lambda$ if there exists $\lambda (1 > \lambda \geq 0)$ satisfying

$$\lim_{i \to \infty} \frac{\|D_{i+1} - D\|_F}{\|D_i - D\|_F} = \lambda.$$ 

A. Convergence of LRM-CG at Sampled Entries

In this subsection, we show that $\{P_{\Omega}(D_i)\}_{i=1}^\infty$, sequence of matrices generated by LRM-CG at sampled points, converges to $P_{\Omega}(D)$. For example, if $D = \begin{bmatrix} 0 & 29 & 20 \\ 29 & 0 & 81 \\ 20 & 81 & 0 \end{bmatrix}$ and $E = \{(1,2), (1,3)\}$, then $P_E(D) = \begin{bmatrix} 0 & 29 & 20 \\ 29 & 0 & 0 \\ 20 & 0 & 0 \end{bmatrix}$. Thus, we will show that

$$\lim_{i \to \infty} P_E(D_i) = \begin{bmatrix} 0 & d_{12}^2(\infty) & d_{13}^2(\infty) \\ d_{21}^2(\infty) & 0 & 0 \\ d_{31}^2(\infty) & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 29 & 20 \\ 29 & 0 & 0 \\ 20 & 0 & 0 \end{bmatrix} = P_E(D).$$

The minimal set of assumptions used for the analytical tractability are as follows:

A1: $f(Y_i) - f(R_{Y_i}(\alpha_i P_i)) \geq -\tau \alpha_i < \text{grad} f(Y_i), P_i >$ for $\tau$ satisfying $0 < \tau < 1/2$,
A2: \[ | \langle \nabla f(R_Y(\alpha_iP_i)), P_i \rangle \rangle | \leq -\mu | \langle \nabla f(Y_i), P_i \rangle | \text{ for } \mu \text{ satisfying } \tau < \mu < 1/2, \]

A3: \[ c\|\nabla f(Y_i)\|_F \geq \|\nabla Y f(Y_i)\|_F \text{ for } c \text{ satisfying } c > 1. \]

In essence, A1 and A2 can be considered as extensions of the strong Wolfe’s conditions [32]. The assumption A1 says that the cost function \( f(Y_i) \) decreases monotonically as long as \( P_i \) is chosen in an opposite direction of \( \nabla f(Y_i) \) on the tangent space \( T_{Y_i} Y \) (i.e., \( \langle \nabla f(Y_i), P_i \rangle \leq 0 \)) (see Lemma 3.7). Note that A1 is reasonable assumption since there always exists a stepsize satisfying this assumption.

**Lemma 3.2:** There exists \( \alpha_i > 0 \) satisfying A1.

**Proof:** See Appendix E

Note that if the stepsize \( \alpha_i \) is chosen to be very small, then \( Y_{i+1} = R_Y(\alpha_i P_i) \approx R_Y(0) = Y_i \), and thus
\[
f(Y_i) - f(R_Y(\alpha_i P_i)) \approx 0.
\]
and \( -\tau \alpha_i < \nabla f(Y_i), P_i \rangle \approx 0 \). In this case, A1 holds true approximately. However, there would be almost no update of \( Y_i \) so that the algorithm will converge extremely slowly. To circumvent this pathological scenario, we use A2, which is in essence an extension of the strong Wolfe’s condition for the Riemannian manifold. Under this assumption, \( \alpha_i \) cannot be chosen to be very small since otherwise we have \( R_Y(\alpha_i P_i) \approx Y_i \), and thus
\[
| \langle \nabla f(R_Y(\alpha_i P_i)), P_i \rangle | \approx | \langle \nabla f(Y_i), P_i \rangle | \geq -\mu | \langle \nabla f(Y_i), P_i \rangle |,
\]
which contradicts the assumption A2.

The assumption A3 is needed to guarantee the global convergence of LRM-CG. We will discuss more on this in Remark 3.6.

---

7Consider an unconstrained minimization in \( \mathbb{R}^n \) with a differentiable cost function \( f(x) \) (i.e., \( \min_{x \in \mathbb{R}^n} f(x) \)). The update equation is given by \( x_{i+1} = x_i + \alpha_i p_i \) for a stepsize \( \alpha_i \) and a descent direction \( p_i \). The well-known strong Wolfe’s conditions is given by
\[
\begin{align*}
f(x_i) - f(x_{i+1}) & \geq -\tau \alpha_i \nabla_x f(x_i), p_i \rangle, \\
| \langle \nabla_x f(x_{i+1}), p_i \rangle | & \leq -\mu \nabla_x f(x_i), p_i \rangle,
\end{align*}
\]
for some constants \( 0 < \tau < \mu < 1 \).
Our first main result, stating successful recovery condition at sampled entries, is formally described in the following theorem.

**Theorem 3.3 (strong convergence of LRM-CG):** Let \( \{D_i = g(Y_i)\}_{i=1}^{\infty} \) be the sequence of the matrices generated by LRM-CG and \( D \) be the original Euclidean distance matrix. Under A1, A2, and A3, \( \{P_E(D_i)\}_{i=0}^{\infty} \) converges linearly to \( P_E(D) \).

**Remark 3.4 (strongly convergent condition in \( \mathbb{R}^n \)):** Note that

\[
\lim_{i \to \infty} \|P_E(D_i) - P_E(D)\|_F = 0
\]

is equivalent to

\[
\lim_{i \to \infty} \|\nabla_Y f(Y_i)\|_F = 0. \tag{28}
\]

This condition is often referred to as the *strongly convergent condition* of the nonlinear CG algorithms in the vector space. The equivalence can be established by the following sandwich lemma.

**Lemma 3.5:**

\[
2\|P_E(D_i) - P_E(D)\|_F \leq \|\nabla_Y f(Y_i)\|_F \leq (2\sqrt{n} + 2)\|P_E(D_i) - P_E(D)\|_F.
\]

*Proof:* See Appendix F

**Remark 3.6:** Recently, an attempt has been made to extend the convergent analysis of the conventional CG algorithms (over the Euclidean space \( \mathbb{R}^n \)) to the Riemannian manifolds. In [33, Theorem 4.3], it has been shown that under certain assumption, one can observe that the Euclidean gradient \( \nabla_Y f(Y_i) \) is replaced by the Riemannian gradient \( \text{grad} f(Y_i) \). Unfortunately, the convergence of the Riemannian gradient in (29) does not imply the convergence of Euclidean gradient in (28) because

\[
\|\nabla_Y f(Y_i)\|_F^2 = \|P_{TVY}(\nabla_Y f(Y_i))\|_F^2 + \|P_{TVY}^\perp(\nabla_Y f(Y_i))\|_F^2
\]

\[
= \|\text{grad} f(Y_i)\|_F^2 + \|P_{TVY}^\perp(\nabla_Y f(Y_i))\|_F^2, \tag{30}
\]

where \( \text{grad} f(Y_i) = P_{TVY}(\nabla_Y f(Y_i)) \) (see (24)). One can observe from this that the condition in (29) is not sufficient to guarantee (28), that is, one cannot guarantee \( \lim_{i \to \infty} \|P_E(D_i) - P_E(D)\|_F = 0 \) just from (29). However, by the introduction of A3, equivalence between (28) and (29) can
be established. We will show that the assumption A3 holds true with overwhelming probability in Section III-C.

We are now ready to prove Theorem 3.3.

Proof of Theorem 3.3: First, we show that under A1 and A2, \(\|P_E(D_i) - P_E(D)\|_F\) is non-increasing. That is, if \(\chi\) is defined by
\[
\chi = \begin{cases} 
\sup_{Y \in \{Y_i\}^\infty_{i=1}} \frac{\|P_{Y_i}^\perp(T Y_i(\nabla f(Y_i)))\|_F}{\|\nabla f(Y_i)\|_F} & \text{if } \|\nabla f(Y_i)\|_F \neq 0, \\
1 & \text{otherwise}
\end{cases},
\]
then there exists \(\gamma>0\) such that \(\gamma(1 - \chi^2) \leq 1\) and
\[
\|P_E(D_{i+1}) - P_E(D)\|^2_F \leq (1 - \gamma(1 - \chi^2)) \|P_E(D_i) - P_E(D)\|^2_F. \tag{32}
\]

We need the following lemma to prove this.

Lemma 3.7: If \(\beta_i\) is chosen based on Fletcher-Reeves’ rule\(^8\), that is,
\[
\beta_i = \frac{<\nabla f(Y_i), \nabla f(Y_i)>}{<\nabla f(Y_{i-1}), \nabla f(Y_{i-1})>}, \tag{34}
\]
then
\[
\frac{<\nabla f(Y_{i+1}), P_{i+1}>}{\|\nabla f(Y_{i+1})\|^2_F} \leq -\frac{1 - 2\mu}{1 - \mu} - \frac{\mu_i^{i+1}}{1 - \mu}.
\]

Proof: See Appendix G.

Lemma 3.8: \(\|\nabla f(Y_i)\|^2_F \geq 8(1 - \chi^2)f(Y_i)\).

Proof: See Appendix H.

\(^8\)In our simulation, we employ Hager-Zhang’s rule in the choice of \(\beta_i\) to improve the empirical performance of the CG method\[^{30}\]:
\[
\beta_i = \frac{1}{h^2} <hH_i - 2P_i||H_i||^2_F, \nabla f(Y_i)> \tag{33}
\]
where \(H_i = \nabla f(Y_i) - P_{Y_i}^\perp(\nabla f(Y_{i-1}))\) and \(h = <P_i, H_i>\). In our analysis, however, we use Fletcher-Reeves’ rule for mathematical tractability.
We are now ready to prove (32). First, from A1, we have
\[
f(Y_{i+1}) \leq f(Y_i) + \tau \alpha_i < \nabla f(Y_i), P_i >
\]
where \(a\) and \(b\) follow from Lemma 3.7 and Lemma 3.8, respectively. Let
\[
\gamma_i = 8\tau \alpha_i \left( \frac{1-2\mu}{1-\mu} \right),
\]
then \(\gamma_i > 0\) (since \(\alpha_i > 0\)) and hence
\[
f(Y_{i+1}) \leq (1 - \gamma_i(1 - \chi^2))f(Y_i).
\]
Recalling that \(f(Y_i) = \frac{1}{2}\|P_E(D_i) - P_E(D)\|_F^2\), we have
\[
\|P_E(D_{i+1}) - P_E(D)\|_F^2 \leq (1 - \gamma_i(1 - \chi^2))\|P_E(D_i) - P_E(D)\|_F^2.
\]
By choosing \(\gamma = \min_i \gamma_i\), we get the desired result.

Now, what remains is to show that \(\lim_{i \to \infty} \|P_E(D_i) - P_E(D)\|_F = 0\) under (32). From (30) and (31), we have \(0 \leq \chi \leq 1\), and thus we need to consider two cases:

1) \(\chi < 1\) case: In this case, one can easily show that \(1 > (1 - \gamma(1 - \chi^2))^{1/2}\). Using this together with (32), we have
\[
\lim_{i \to \infty} \frac{\|P_E(D_{i+1}) - P_E(D)\|_F}{\|P_E(D_i) - P_E(D)\|_F} = (1 - h(1 - \chi^2))^{1/2} < 1
\]
and hence
\[
\lim_{i \to \infty} \|P_E(D_i) - P_E(D)\|_F = 0.
\]
Thus, the sequence \(\{P_E(D_i)\}_{i=1}^{\infty}\) converges linearly to \(P_E(D)\).

2) \(\chi = 1\) case: In this case, we show that there exists \(j\) satisfying \(\|\nabla_Y f(Y_j)\|_F = 0\). As discussed in Remark 3.4 and Lemma 3.5, this is a sufficient condition to guarantee the strong convergence of LRM-CG. In this case, no further update can be made after \(j\)-th iteration (and thus linear convergence is naturally guaranteed). To show this, we use the
contradiction argument. Suppose that $\|\nabla_Y f(Y_i)\|_F \neq 0$ for all $i$. Then, from (31) we should have

$$\sup_{Y \in \{Y_i\}_{i=1}^\infty} \frac{\|P_{TY_i}^\perp \nabla_Y f(Y)\|_F}{\|\nabla_Y f(Y)\|_F} = \chi = 1.$$ 

Further, from (30), we have

$$\|P_{TY_i}^\perp \nabla_Y f(Y_i)\|_F^2 = \|\nabla_Y f(Y_i)\|_F^2 - \|\text{grad} f(Y_i)\|_F^2 \leq \|\nabla_Y f(Y_i)\|_F^2 - \frac{1}{c^2} \|\nabla_Y f(Y_i)\|_F^2,$$

where the inequality is from $A3$ ($c > 1$). Thus,

$$1 = \sup_{Y \in \{Y_i\}_{i=1}^\infty} \frac{\|P_{TY_i}^\perp \nabla_Y f(Y)\|_F^2}{\|\nabla_Y f(Y)\|_F^2} \leq \sup_{Y \in \{Y_i\}_{i=1}^\infty} \frac{\|\nabla_Y f(Y)\|_F^2 - \frac{1}{c^2} \|\nabla_Y f(Y)\|_F^2}{\|\nabla_Y f(Y)\|_F^2} = 1 - \frac{1}{c^2},$$

which is contradiction. Thus, $\|\nabla_Y f(Y_i)\|_F = 0$ for some $j$.

### B. Exact Recovery of Euclidean Distance Matrices

So far, we have shown that the output of LRM-CG converges to the original Euclidean distance matrix $D$ at sampled entries (i.e., $P_E(D_\infty) = P_E(D)$). In this subsection, we show that all entries of $D_i$ converge to that of the original Euclidean distance matrix $D$ with overwhelming probability.

Before we proceed, we briefly discuss the probability model of the sampling operator $P_E$. Let $\delta_{ij}$ be a Bernoulli random variable that takes value 1 if $d_{ij} \leq r$ (recall that $r$ is the radio communication range) and 0 otherwise. Since the distance is symmetric (i.e., $d_{ij} = d_{ji}$), we have $\delta_{ij} = \delta_{ji}$. Also, since the diagonal entries of $D$ are all zeros, we define $\delta_{ii} = 0$ for all $i$. Then, for a matrix $A$, $P_E(A)$ can be expressed as

$$P_E(A) = \sum_{i \neq j} \delta_{ij} < A, e_i e_j^T > e_i e_j^T,$$

$$= \sum_{i \neq j} \delta_{ij} a_{ij}, \quad (35)$$
where $e_i$ is the standard basis of $\mathbb{R}^n$. For example, if $A = \begin{bmatrix} 0 & 10 & 17 \\ 10 & 0 & 3 \\ 17 & 3 & 0 \end{bmatrix}$ and $E = \{(1, 2), (2, 3)\}$, then

$$P_E(A) = \begin{bmatrix} 0 & 10 & 0 \\ 10 & 0 & 3 \\ 0 & 3 & 0 \end{bmatrix} = 10 \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 3 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} + 10 \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 3 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

We now characterize the random variables $\delta_{ij}$ using $P(d_{ij} \leq r)$. Since $d_{ij} = \|x_i - x_j\|_2$, it follows $P(d_{ij} \leq r) = P(\|x_i - x_j\|_2 \leq r)$. In this work, we assume that elements of $x_i$ (locations of sensor nodes) are i.i.d. random and uniformly distributed over unit interval. By denoting $p = P(d_{ij} \leq r)$, the probability mass function (PMF) of $\delta_{ij}$ can be expressed as

$$f(\delta_{ij}; p) = p^{\delta_{ij}}(1 - p)^{1 - \delta_{ij}}. \quad (36)$$

The following lemma provides an explicit expression of $p$ in terms of the radio communication range $r$.

Lemma 3.9: If an element of the location vectors $x_i$ is i.i.d. and uniform on unit interval, then

a) If $k = 2$ (2-dimensional Euclidean space),

$$p = \begin{cases} \pi r^2 - \frac{8}{3} r^3 + \frac{1}{2} r^4 & \text{if } 0 \leq r \leq 1 \\ p_1(r) & \text{if } 1 \leq r \leq \sqrt{2}, \\ 1 & \text{else} \end{cases}$$

b) If $k = 3$ (3-dimensional Euclidean space),

$$p = \begin{cases} \frac{4}{9} \pi r^3 - \frac{3}{2} r^4 + \frac{8}{5} r^5 - \frac{1}{6} r^6 & \text{if } 0 \leq r \leq 1 \\ p_2(r) & \text{if } 1 \leq r \leq \sqrt{2} \\ p_3(r) & \text{if } \sqrt{2} \leq r \leq \sqrt{3} \\ 1 & \text{else} \end{cases}$$
where

\[ p_1(r) = -\frac{2}{3} - 2r^2 - \frac{1}{2}r^4 + \frac{1}{3}(8r^2 + 1)\sqrt{r^2 - 1} + 2r^2 \sin^{-1}\left(\frac{2}{r^2 - 1}\right) \]

\[ + \frac{1}{1 + \tan\left(\frac{\sin^{-1}\left(\frac{2}{r^2 - 1}\right)}{2}\right)}, \]  

\[ p_2(r) = -\frac{15\pi + 37}{30} + 6\pi + \frac{1}{2}r^2 - \frac{8\pi}{3}r^3 + \frac{3\pi}{2}r^4 + \frac{1}{3}r^6 + 2r^4 \sin^{-1}\sqrt{1 - \frac{1}{r^2}} \]

\[ + \left(\frac{2}{r^2} - \frac{16}{5}r^4\right)\sqrt{r^2 - 1} - 2r^4 \sin^{-1}\left(\frac{1}{r}\right) - r^4 \sin^{-1}\left(\frac{2}{r^2 - 1}\right) \]

\[ - \frac{16}{3 (1 + \tan\left(\frac{1}{2} \sin^{-1}\left(\frac{2}{r^2 - 1}\right)\right))^3} + \frac{4}{(1 + \tan\left(\frac{1}{2} \sin^{-1}\left(\frac{2}{r^2 - 1}\right)\right))^2}, \]  

\[ p_3(r) = -\frac{\pi}{2} + \frac{97}{30} - \frac{7}{2}r^2 - \frac{3}{2}r^4 - \frac{1}{6}r^6 + \left(\frac{26}{15} + \frac{44}{15}r^2 + \frac{8}{5}r^4\right)\sqrt{r^2 - 2} + \frac{16}{3}r^3 \tan^{-1}\sqrt{1 - \frac{2}{r^2}} \]

\[ + (2 - 8r^2) \tan^{-1}\sqrt{r^2 - 2} - (2r^4 - 4r^2) \sin^{-1}\sqrt{r^2 - 1} + (r^4 + 2r^2) \sin^{-1}\left(\frac{3 - r^2}{r^2 - 1}\right) \]

\[ - \frac{16}{3}r^3 \tan^{-1}\sqrt{1 - \frac{2}{r^2 - 2}} + 8r^2 \tan^{-1}\sqrt{1 - \frac{1}{r^2 - 2}} + (2r^4 - 4r^2) \sin^{-1}\sqrt{1 - \frac{1}{r^2 - 1}} \]

\[ + \frac{16}{3 (1 + \tan\left(\frac{1}{2} \sin^{-1}\left(\frac{3-r^2}{r^2-1}\right)\right))^3} - \frac{4}{(1 + \tan\left(\frac{1}{2} \sin^{-1}\left(\frac{3-r^2}{r^2-1}\right)\right))^2}. \]  

**Proof:** See Appendix II.

It is worth noting that \( p_1(r), p_2(r), \) and \( p_3(r) \) increase monotonically with \( r \) (see Fig. 7).

We now state our main result.

**Theorem 3.10:** Under the assumption A1, A2, and A3, the output sequence \( \{D_i = g(Y_i)\}_{i=1}^\infty \) of LRM-CG converges globally to the Euclidean distance matrix \( D \) \((\lim_{i \to \infty} \|D_i - D\|_F = 0)\) with the probability at least

\[ 1 - \exp \left( - \left( (1 - c) \log \left( \frac{1 - c}{1 - p} \right) + c \log \left( \frac{c}{p} \right) \right) \right) \]  

(40)

for some constant \( c \) satisfying \( 0 < c < 1 \) and \( c < p \).

**Remark 3.11:** From Lemma 3.9, we see that \( p \) gets close to 1 as the radio communication range \( r \) increases. Thus, as shown in Fig. 8, the chance of recovering \( D \) increases with \( r \).

**Remark 3.12:** Theoretical guarantee on the recovery of a matrix has been provided by Candes and Recht in [12], and later improved in [34], [35]. In short, if entries of a matrix are chosen
at random, then the $n \times n$ matrix with rank $k$ can be recovered with overwhelming probability as long as the number of measurements $m$ follows $m = \mathcal{O}(kn^{1.2} \log(n))$. The analysis in these works is based on the assumption that observed entries are sampled i.i.d. (and follow Bernoulli or uniform distribution). Whereas, our analysis does not require independence assumption among the sampled entries of $D$ since the elements of $D$ are related. In other words, random variables $\delta_{ij}$ do not need to be independent. For example, consider the scenario illustrated in Fig. 9. Since the sensor node 4 is located inside the triangle formed by three sensor nodes (nodes 1, 2, and 3), one can see that $d_{14} \leq \max(d_{12}, d_{13})$. Thus, if $d_{12}$ and $d_{13}$ are already known (i.e., $d_{12} \leq r$, $d_{13} \leq r$), then so is $d_{14}$. In other words, $P(\delta_{14} = 1|\delta_{12} = \delta_{13} = 1) = 1$, while $P(\delta_{14} = 1)$ is not necessarily one. In our work, we do not put any assumption on the independence of the entries of $D$ yet show that $D$ can be recovered exactly with overwhelming probability when $r$ is large.

Following lemma is useful to prove Theorem 3.10.

**Lemma 3.13:** For a given matrix $A$, if the diagonal entries are zeros (i.e., $a_{ii} = 0$ for all $i$) and $\|A\|_F < \infty$, then there exists a constant $t$ ($0 < t < 1$) satisfying

$$t \|A\|_F^2 \leq \|P_E(A)\|_F^2,$$

(41)
Fig. 8. The Euclidean distance matrix $D$ can be recovered with overwhelming probability in (a) 2D and (b) 3D Euclidean space when $r$ is large.

with the probability at least $1 - \exp\left(-\left(1 - mt\right)\log\left(\frac{1-mt}{1-p}\right) + mt\log\left(\frac{mt}{p}\right)\right)$ for some constant $m \geq 1$, provided that $0 < mt < p < 1$.

Proof: See Appendix J.

Proof of Theorem 3.10 Let $A = D_i - D$. Then from Lemma 3.13 we have

$$
\|D_i - D\|_F \leq \frac{1}{\sqrt{t}} \|\mathcal{P}_E(D_i) - \mathcal{P}_E(D)\|_F
$$

(42)

with the probability at least $1 - \exp\left(-\left(1 - mt\right)\log\left(\frac{1-mt}{1-p}\right) + mt\log\left(\frac{mt}{p}\right)\right)$ for some constant $m$ satisfying $m \geq 1$ and $0 < m < \frac{p}{t}$. Combining this with $\lim_{i \to \infty} \|\mathcal{P}_E(D_i) - \mathcal{P}_E(D)\|_F = 0$ (Theorem 3.3), we can conclude that

$$
\lim_{i \to \infty} \|D_i - D\|_F = 0,
$$

with the probability at least $1 - \exp\left(-\left((1 - c)\log\left(\frac{1-c}{1-p}\right) + c\log\left(\frac{c}{p}\right)\right)\right)$, where $c = mt$. 

August 4, 2017
Fig. 9. Suppose that the sensor node 4 is inside the triangle formed by three sensor nodes 1, 2, and 3. Then for a given \( r \), it can be shown that \( d_{14} \leq \max(d_{12}, d_{13}) \), and thus \( P(d_{14} \leq r | d_{12} \leq r, d_{13} \leq r) = 1 \) which is not necessarily equivalent to \( P(d_{14} \leq r) \).

C. Discussion on A3

In this section, we show that the assumption A3 \((c\|\mathrm{grad}f(Y_i)\|_F^2 + \epsilon > \|\nabla_Y f(Y_i)\|_F^2\) for some \( c > 1 \) and \( \epsilon > 0 \)) holds true with overwhelming probability when \( r \) is large. In order to show this, we first need to define the coherence, a measure of concentration in a matrix [12].

**Definition 3.14 (Coherence [12]):** Let \( Q \) be a subspace of \( \mathbb{R}^n \) of dimension \( k \) and \( P_Q \) be the orthogonal projection onto \( Q \). Then the coherence of \( Q \) is defined by

\[
\mu(Q) = \frac{n}{k} \max_{1 \leq i \leq n} \|P_Q e_i\|_2^2.
\]

Consider a matrix \( A \) of rank \( k \) whose singular value decomposition is given by

\[
A = U\Sigma V^T = \sum_{i=1}^{k} \sigma_i u_i v_i^T,
\]

where \( U = \begin{bmatrix} u_1 & \cdots & u_k \end{bmatrix} \) and \( V = \begin{bmatrix} v_1 & \cdots & v_k \end{bmatrix} \) are the matrices constructed by the left and right singular vectors, respectively, and \( \Sigma \) is the diagonal matrix whose diagonal entries are \( \sigma_i \). From (43), we see that the concentration on the vertical direction (concentration in the row) is

\[8\text{When } \epsilon \text{ is sufficiently small, one can simply put } c\|\mathrm{grad}f(Y_i)\|_F^2 \geq \|\nabla_Y f(Y_i)\|_F^2, \text{ which is the strict form of A3. Intuitively, if the generated sequence of Riemannian gradient goes to zero } (\lim_{i \to \infty} \|\mathrm{grad}f(Y_i)\|_F = 0), \text{ so does the corresponding sequence of the Euclidean gradient } (\lim_{i \to \infty} \|\nabla_Y f(Y_i)\|_F = 0). \]
determined by \( u_i \) and that on the horizontal direction (concentration in the column) is determined by \( v_i \). For example, if one of the standard basis vector \( e_i \), say \( e_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T \), lies on the space spanned by \( u_1, \cdots, u_k \) while others \((e_2, e_3, \cdots)\) are orthogonal to this space, then it is clear that nonzero entries of the matrix are only on the first row. Since we need to check the concentration on both vertical and horizontal directions, we need to investigate both \( \mu(U) \) and \( \mu(V) \). In this regard, the coherence of a matrix \( A \) is defined by

\[
\mu(A) = \max(\mu(U), \mu(V)).
\]  

(44)

In particular, if \( A \) is a positive semidefinite matrix, then it is clear that \( U = V \) and thus \( \mu(A) = \mu(U) \).

**Theorem 3.15**: Suppose \( \mu(Y_i) \leq \mu_0 \) for a given matrix \( Y_i \in \tilde{Y} \). Then, for any \( c > 1 \) and \( \epsilon > 0 \),

\[
e^2 \| \nabla f(Y_i) \|_F^2 + \epsilon > \| \nabla_Y f(Y_i) \|_F^2,
\]

with probability at least

\[
1 - \exp \left( -m\epsilon \log \left( \frac{m\epsilon}{1 - p} \right) + (1 - m\epsilon) \log \left( \frac{1 - m\epsilon}{p} \right) \right)
\]

(46)

for some constant \( m \) satisfying \( m > 0 \) and \( 0 < m < \frac{1 - p}{\epsilon} \), provided that \( n \geq 2c\mu_0k \).

**Remark 3.16**: From Lemma 3.9 we see that \( p \) gets close to 1 as \( r \) increases. Thus, as shown in Fig. 10 when \( r \) is large, (45) holds true with overwhelming probability.

Following lemmas are needed to prove the theorem.

**Lemma 3.17**:\[
\| \nabla_Y f(Y) \|_F^2 - e^2 \| \nabla f(Y) \|_F^2
\]

\[
\leq \sum_{i \neq j} \sum_{u \neq v} \delta_{ij} |<B, e_i e_j^T>| <B, e_u e_v^T> - (I - e^2 P_{TY})l(e_i e_j^T), l(e_u e_v^T) > |,
\]

(47)

where \( B = g(Y) - D \) and \( l(A) = 2\text{eye}(\text{Sym}(A))1 - 2\text{Sym}(A) \).

**Proof**: See Appendix [K].

**Lemma 3.18**: If \( n^2 \geq 4c\mu(Y)^2k^2 \) and \( i \neq j \), then

\[
| <(I - cP_{TY})l(e_i e_j^T), l(e_u e_v^T) > | \geq 4 \left( 1 - \frac{4c\mu(Y)^2k^2}{n^2} \right),
\]

(48)

where \( c > 0 \) and \( l(A) = 2\text{eye}(\text{Sym}(A))1 - 2\text{Sym}(A) \).
Fig. 10. The condition (45) holds true with overwhelming probability in (a) 2D and (b) 3D Euclidean space when the radio communication range \( r \) is large.

**Proof:** See Appendix L.

**Lemma 3.19:** Let \( \delta_1, \delta_2, \cdots, \delta_N \) be identically (not necessarily independently) distributed Bernoulli random variables with \( P(\delta_i = 1) = p \) and \( P(\delta_i = 0) = 1 - p \). Also, let \( a_1, a_2, \cdots, a_N \) be positive values. Let \( q \) be the largest integer obeying \( 2^q \leq N \). Then, for any \( \epsilon > 0 \),

\[
P\left( \sum_{i=1}^{N} \delta_i a_i \geq \epsilon \right) \leq \exp\left( - \left( m\epsilon \log\left( \frac{m\epsilon}{1-p} \right) + (1 - m\epsilon) \log\left( \frac{1 - m\epsilon}{p} \right) \right) \right),
\]

(49)

where \( m\epsilon = \frac{\sum_{i=1}^{N} a_i - \epsilon}{2q a_{\min}} \) with \( a_{\min} = \min_{i} a_i \), provided that \( 0 < m\epsilon < 1 - p \).

**Proof:** See Appendix M.

Now, we are ready to prove Theorem 3.15.

**Proof of Theorem 3.15:** Let

\[
I = \| \nabla \mathbf{v f}(\mathbf{Y}_i) \|_{F}^{2} - \epsilon^2 \| \text{grad} f(\mathbf{Y}_i) \|_{F}^{2},
\]

\[
s_{ij} = < \mathbf{g}(\mathbf{Y}_i) - \mathbf{D}, \mathbf{e}_i \mathbf{e}_j^T >,
\]
and

\[ g_{ij} = \sum_{u \neq v} |s_{ij} s_{uv} < (I - c^2 P_{TV}^T \tilde{\delta}^T l(e_i e_j^T), l(e_u e_v^T)) |. \]

In this proof, we will show that \( P(I \leq \epsilon) \) is lower bounded by the quantity in (46). First, since \( I \leq \sum_{i \neq j} \delta_{ij} g_{ij} \) from Lemma 5.17 and hence \( P(I \leq \epsilon | \sum_{i \neq j} \delta_{ij} g_{ij} \leq \epsilon) = 1 \), we have

\[
P(\sum_{i \neq j} \delta_{ij} g_{ij} \leq \epsilon) = P(\sum_{i \neq j} \delta_{ij} g_{ij} \leq \epsilon | \sum_{i \neq j} \delta_{ij} g_{ij} \leq \epsilon) \\
= P(\sum_{i \neq j} \delta_{ij} g_{ij} \leq \epsilon) \\
= P(\sum_{i \neq j} \delta_{ij} g_{ij} \leq \epsilon | I \leq \epsilon) P(I \leq \epsilon) \\
\leq P(I \leq \epsilon).
\]  

(50)

What remains is to find out a lower bound of \( P(\sum_{i \neq j} \delta_{ij} g_{ij} \leq \epsilon) \). Equivalently, we find an upper bound of \( P(\sum_{i \neq j} \delta_{ij} g_{ij} \geq \epsilon) \). First, in order to use Lemma 3.19, we need to find a lower bound of \( g_{ij} \) for \((i, j) \in \Omega (\Omega = \{(i, j) : s_{ij} \neq 0\})\). Let \( s = \min_{(i,j)\in\Omega} |s_{ij}| \), then

\[
g_{ij} = \sum_{u \neq v} |s_{ij} s_{uv} < (I - c^2 P_{TV}^T \tilde{\delta}^T l(e_i e_j^T), l(e_u e_v^T)) | \\
\geq |s_{ij}^2 < (I - c^2 P_{TV}^T \tilde{\delta}^T l(e_i e_j^T), l(e_i e_j^T)) | \\
\geq \frac{c^2}{4} s^2 (1 - 4c^2 \mu(Y)^2 k^2 / n^2) \\
\geq 4s^2 (1 - 4c^2 \mu(Y)^2 k^2 / n^2),
\]

where (a) follows from Lemma 5.18. Now using Lemma 3.19 we have

\[
P(\sum_{i \neq j} \delta_{ij} g_{ij} \geq \epsilon) \leq \exp \left( -m \epsilon \log \left( \frac{m \epsilon}{1 - p} \right) + (1 - m \epsilon) \log \left( \frac{1 - m \epsilon}{p} \right) \right),
\]

(51)

where \( m = (\sum_{(i,j)\in\Omega} g_{ij} - \epsilon)/(2^a c_1) \) (\( c_1 = 4s^2 (1 - 4c^2 \mu(Y)^2 k^2 / n^2) \), provided that \( 0 < m < \frac{1 - \epsilon}{\epsilon} \) and \( n \geq 2c_0 \)). Here, \( q \) is the largest integer obeying \( 2^q \leq |\Omega| \) (\(|\Omega| \) being the cardinality of \( \Omega \)). From (50) and (51), and noting that \( P(\sum_{i \neq j} \delta_{ij} g_{ij} \leq \epsilon) = 1 - P(\sum_{i \neq j} \delta_{ij} g_{ij} \geq \epsilon) \), we get the desired result.

IV. CONCLUSION

In this paper, we have proposed an algorithm to recover the Euclidean distance matrix (and therefore the location map) from partially observed distance information. In solving the Frobenius
norm minimization problem with a rank constraint, we expressed the Euclidean distance matrix as a function of the fixed rank positive semidefinite matrix. By capitalizing on the Riemannian manifold structure for this set of matrices, we could solve the low-rank matrix completion problem using the modified conjugate gradient algorithm. We have shown from the recovery condition analysis that the proposed LRM-CG algorithm recovers the original Euclidean distance matrix with overwhelming probability under some suitable conditions. We have also demonstrated from numerical experiments that the LRM-CG algorithm is effective in recovering the original Euclidean distance matrix while exhibiting computational cost scalable to the matrix dimension. Given the importance of the location-aware applications and services in the IoT era, we believe that the proposed LRM-CG algorithm will be a useful tool for various localization problems.

While our work focused primarily on the centralized localization scenario, extension to the distributed and/or cooperative scenarios would also be interesting direction worth pursuing. We also would like to point out that our recovery guarantee analysis is not so quantitative due to the probabilistic distance measurement model (Section III.B). By introducing the non-probabilistic model (e.g., fixed ratio communication range model), we might come up with more quantitative and insightful analytic results. We leave these interesting explorations for our future work.
APPENDIX A
PROOF OF LEMMA 2.2

Proof: Let \( \dot{Y} \) be a tangent vector at \( Y \in \tilde{Y} \), i.e., \( \dot{Y} \in T_Y \tilde{Y} \). Also, let

\[
S = \left\{ \begin{bmatrix} Q & Q_\perp \\ \ast & \ast \end{bmatrix} \begin{bmatrix} B & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} Q^T \\ Q_\perp \end{bmatrix} : B^T = B \in \mathbb{R}^{k \times k}, C \in \mathbb{R}^{(n-k) \times k} \right\}.
\]

Then, what we need to show is that \( \dot{Y} \) is an element in \( S \). By the definition of \( T_Y \tilde{Y} \), there exists a curve \( \gamma(t) \) in \( \tilde{Y} \) such that \( Y = \gamma(0) \) and \( \dot{Y} = \left. \frac{d}{dt} \gamma(t) \right|_{t=0} \). For convenience, we denote \( \gamma(t) = Z(t) \). Using the eigenvalue decomposition \( Z(t) = Q(t) \Lambda(t) Q(t)^T \), we have

\[
\dot{Y} = \left. \frac{d}{dt} Z(t) \right|_{t=0} = \dot{Q} \Lambda Q^T + Q \dot{\Lambda} Q^T + Q \Lambda \dot{Q}^T.
\]

Since \( Q^T Q = I \), \( Q \) is an element of the Stiefel manifold \( Q = \{ A : A^T A = I, A \in \mathbb{R}^{n \times k} \} \). The tangent vector of \( Q \) at the point \( Q \) is given by \[19\] Example 3.5.2]

\[
\dot{Q} = Q \Omega + Q_\perp K,
\]

where \( \Omega \) is the \( k \times k \) skew-symmetric matrix (i.e., \( \Omega = -\Omega^T \)) and \( K \) is the \( (n-k) \times (n-k) \) matrix. From (53) and (54), we have

\[
\dot{Y} = (Q \Omega + Q_\perp K) \Lambda Q^T + Q \Lambda Q^T
\]

\[
+ Q \Lambda (Q \Omega + Q_\perp K)
\]

\[
= \begin{bmatrix} Q & Q_\perp \end{bmatrix} \begin{bmatrix} \Omega \Lambda + \dot{\Lambda} + \Lambda \Omega^T & \Lambda K^T \\ K \Lambda & 0 \end{bmatrix} \begin{bmatrix} Q^T \\ Q_\perp \end{bmatrix}.
\]

If we denote \( B = \Omega \Lambda + \dot{\Lambda} + \Lambda \Omega^T \) and \( C = K \Lambda \), then one can easily see that \( \dot{Y} \in T_Y \tilde{Y} \subseteq S \).

To complete the proof, we need to show that \( S = T_Y \tilde{Y} \). This implies that two vector spaces \( S \) and \( T_Y \tilde{Y} \) have the same dimension. Indeed, from (52), we can easily check that the dimension\(^{10}\) of \( S \) is \( \frac{1}{2} k (2n - k + 1) \), which is the dimension of \( \tilde{Y} \) \[21\] Proposition 1.1.

---

\(^{10}\) The dimension of \( S \) is obtained by counting the number of independent entries of an element in \( S \). Since \( B \) is a \( k \times k \) symmetric matrix, the number of independent entries of \( B \) is \( \frac{k(k+1)}{2} \). In addition, since \( C \) is an arbitrary \( (n-k) \times k \) matrix, the number of independent entries of \( C \) is \( (n-k)k \). Thus, the dimension of \( S \) is \( \frac{k(k+1)}{2} + (n-k)k = \frac{1}{2} k (2n - k + 1) \).
APPENDIX B

PROOF OF PROPOSITION 2.4

Proof: First, we partition a matrix $A$ into two parts: $A = A_1 + A_2$ where $A_1 \in T_Y \tilde Y$ and $A_2 \in (T_Y \tilde Y)^\perp$. Then, it is clear that $P_{T_Y \tilde Y}(A) = A_1$ and thus the goal is to find out the closed form expression of $A_1$. From Lemma 2.2, there exist a symmetric matrix $B \in \mathbb{R}^{k \times k}$ and a matrix $C \in \mathbb{R}^{(n-k) \times k}$ such that

$$A_1 = \begin{bmatrix} Q & Q^\perp \end{bmatrix} \begin{bmatrix} B & C^T \end{bmatrix} \begin{bmatrix} Q^T \\ C & 0 \end{bmatrix} \begin{bmatrix} Q^T \\ Q^\perp \end{bmatrix}. \quad (55)$$

Since $< A_1, A_2 > = 0$, we have

$$0 = < A_1, A_2 >$$

$$= < A_1, A - A_1 >$$

$$= < \begin{bmatrix} Q & Q^\perp \end{bmatrix} \begin{bmatrix} B & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} Q^T \\ Q^\perp \end{bmatrix}, A - \begin{bmatrix} Q & Q^\perp \end{bmatrix} \begin{bmatrix} B & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} Q^T \\ Q^\perp \end{bmatrix} >$$

$$= < \begin{bmatrix} B & C^T \\ C & 0 \end{bmatrix}, \begin{bmatrix} Q^T \\ Q^\perp \end{bmatrix} \left( A - \begin{bmatrix} Q & Q^\perp \end{bmatrix} \begin{bmatrix} B & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} Q^T \\ Q^\perp \end{bmatrix} \right) \begin{bmatrix} Q & Q^\perp \end{bmatrix} >$$

$$= < \begin{bmatrix} B & C^T \\ C & 0 \end{bmatrix}, \begin{bmatrix} Q^T \\ Q^\perp \end{bmatrix} A \begin{bmatrix} Q & Q^\perp \end{bmatrix} - \begin{bmatrix} B & C^T \\ C & 0 \end{bmatrix} >. \quad (56)$$

Let

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} Q^T \\ Q^\perp \end{bmatrix} A \begin{bmatrix} Q & Q^\perp \end{bmatrix} = \begin{bmatrix} Q^T AQ & Q^T AQ^\perp \\ Q^\perp AQ & Q^\perp AQ^\perp \end{bmatrix},$$

then we have

$$0 = < \begin{bmatrix} B & C^T \\ C & 0 \end{bmatrix}, \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} - \begin{bmatrix} B & C^T \\ C & 0 \end{bmatrix} >$$

$$= < B, A_{11} - B > + < C, \frac{1}{2} (A_{21} + A_{12}^T) - C >,$$

where the equality is because

$$< \begin{bmatrix} \alpha_1 & \alpha_2 \\ \alpha_3 & \alpha_4 \end{bmatrix}, \begin{bmatrix} \beta_1 & \beta_2 \\ \beta_3 & \beta_4 \end{bmatrix} > = < \alpha_1 , \beta_1 > + < \alpha_2 , \beta_2 > + < \alpha_3 , \beta_3 > + < \alpha_4 , \beta_4 >.$$
Since \( B \) and \( C \) are chosen arbitrarily, we should have
\[
< B, A_{11} - B > = 0, \tag{58}
\]
\[
< C, \frac{1}{2}(A_{21} + A_{12}^T) - C > = 0. \tag{59}
\]

First, it is clear from (59) that
\[
C = \frac{1}{2}(A_{21} + A_{12}^T). \tag{60}
\]

Next, noting that \( A_{11} = \text{Sym}(A_{11}) + \text{Skew}(A_{11}) \), (58) becomes
\[
0 = < B, A_{11} - B > = < B, \text{Sym}(A_{11}) - B > + < B, \text{Skew}(A_{11}) > \tag{a}
\]
\[
= < B, \text{Sym}(A_{11}) - B > + < \text{Sym}(B), \text{Skew}(A_{11}) > \tag{b}
\]
where (a) is because \( B \) is the symmetric matrix (i.e., \( B = \text{Sym}(B) \)), and (b) is because \( < \text{Sym}(C), \text{Skew}(D) > = 0 \) for any matrices \( C \) and \( D \). Since \( B \) is any symmetric matrix, we should have
\[
B = \text{Sym}(A_{11}) = \frac{1}{2}(A_{11} + A_{11}^T). \tag{61}
\]

Substituting (60) and (61) into (55), we have
\[
A_1 = \begin{bmatrix}
Q & Q_{\perp}
\end{bmatrix}
\begin{bmatrix}
\frac{1}{2}(A_{11} + A_{11}^T) & \frac{1}{2}(A_{21}^T + A_{12}) \\
\frac{1}{2}(A_{21} + A_{12}^T) & 0
\end{bmatrix}
\begin{bmatrix}
Q^T \\
Q_{\perp}^T
\end{bmatrix}, \tag{62}
\]
where \( A_{11}, A_{12}, \) and \( A_{21} \) are the components of \( A \) (see (57)).

Now, what remains is to find a closed form expression for \( A_1 \) in terms of \( A \). First, we can rewrite (62) as
\[
A_1 = \frac{1}{2}
\begin{bmatrix}
Q & Q_{\perp}
\end{bmatrix}
\begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix}
\begin{bmatrix}
Q^T \\
Q_{\perp}^T
\end{bmatrix}
+ \frac{1}{2}
\begin{bmatrix}
Q & Q_{\perp}
\end{bmatrix}
\begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix}^T
\begin{bmatrix}
Q^T \\
Q_{\perp}^T
\end{bmatrix}
- \frac{1}{2} Q_{\perp} (A_{22} + A_{22}^T) Q_{\perp}^T. \tag{63}
\]
Substituting (57) into (63), we have
\[
A_1 = \frac{1}{2} \begin{bmatrix} Q & Q_{\perp} \end{bmatrix} \begin{bmatrix} Q^T \\ Q_{\perp}^T \end{bmatrix} A \begin{bmatrix} Q & Q_{\perp} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} Q & Q_{\perp} \end{bmatrix} A^T \begin{bmatrix} Q & Q_{\perp} \end{bmatrix} \]
\[
- \frac{1}{2} Q_{\perp} (Q_{\perp}^T A Q_{\perp} + Q_{\perp}^T A^T Q_{\perp}) Q_{\perp}^T
\]
\[
= \frac{1}{2} (A + A^T) - \frac{1}{2} Q_{\perp} Q_{\perp}^T (A + A^T) Q_{\perp} Q_{\perp}^T
\]
\[
= \text{Sym}(A) - Q_{\perp} Q_{\perp}^T \text{Sym}(A) Q_{\perp} Q_{\perp}^T.
\]
Since $QQ^T + Q_{\perp} Q_{\perp}^T = I$ and $P_Q = QQ^T$, we have
\[
A_1 = \text{Sym}(A) - (I - QQ^T) \text{Sym}(A) (I - QQ^T)
\]
\[
= P_Q \text{Sym}(A) + \text{Sym}(A) P_Q - P_Q \text{Sym}(A) P_Q,
\]
which is the desired result.

**APPENDIX C**

**PROOF OF LEMMA 2.6**

**Proof:** Our goal is to find a simple expression of the retraction operator $R_Y(B)$. First, since $Z = \text{Sym}(Z)$ for $Z \in \tilde{Y}$, we have
\[
\|Y + B - Z\|_F^2 = \|Y + B - \text{Sym}(Z)\|_F^2
\]
\[
= \|\text{Skew}(Y + B) + \text{Sym}(Y + B) - \text{Sym}(Z)\|_F^2
\]
\[
= \|\text{Skew}(Y + B)\|_F^2 + \|\text{Sym}(Y + B) - \text{Sym}(Z)\|_F^2
\]
\[
= \|\text{Skew}(Y + B)\|_F^2 + \|\text{Sym}(Y + B) - Z\|_F^2,
\]
where (64) is because $\text{Sym}(A) + \text{Skew}(A) = A$ and (65) is because $<\text{Skew}(C), \text{Sym}(D)> = 0$ for any $C$ and $D$. Since the first term in (66) is unrelated to $Z$, it is clear that
\[
R_Y(B) = \arg\min_{Z \in \tilde{Y}} \|\text{Sym}(Y + B) - Z\|_F.
\]
Using the eigenvalue decomposition $\text{Sym}(Y + B) = K \Sigma K^T$, we have
\[
R_Y(B) = \arg\min_{Z \in \tilde{Y}} \|K \Sigma K^T - Z\|_F
\]
\[
= \arg\min_{Z \in \tilde{Y}} \|K (\Sigma - K^T Z K) K^T\|_F
\]
\[
= \arg\min_{Z \in \tilde{Y}} \|\Sigma - K^T Z K\|_F
\]
\[
= \arg\min_{Z \in \tilde{Y}} \|\Sigma - K^T Z K\|_F,
\]
where (a) is because $\|KUK^T\|_F^2 = tr(KU^TK^TKUK^T) = \|U\|_F^2$ for any matrix $U$. Now let $R_Y(B) = Z^*, \Sigma^* = KZ^*K^T$, and $Q = K^TZEK$, then

$$\Sigma^* = KZ^*K^T = \arg\min_Q \|\Sigma - Q\|_F.$$  \hfill (67)

Since $\Sigma$ is a diagonal matrix, $\Sigma^*$ should also be a diagonal matrix. Also, $\Sigma^* \succeq 0$ and $\text{rank}(\Sigma^*) = k$. Thus, $\Sigma^*$ is a diagonal matrix with only $k$ positive entries and the rest being zero. That is,

$$\Sigma^* = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \sigma_k & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad (68)$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0$. Recalling that $\text{Sym}(Y + B) = K\Sigma K^T$, we finally have

$$R_Y(B) = K\Sigma^*K^T = \mathcal{W}_k(Y + B),$$

where the last equality is from (15). \hfill \blacksquare

APPENDIX D

PROOF OF LEMMA 2.7

Proof: In general, Euclidean gradient $\nabla_Y f(Y)$ can be obtained by taking partial derivatives with respect to each coordinate of the Euclidean space. Since $\nabla_Y f(Y)$ is interpreted as a matrix whose inner product with an arbitrary matrix $H$ becomes the Frechet differential $Df(Y)[H]$ of $f$ at $Y$ \footnote{Since $Z^* \in \tilde{Y}$, $Z^* \succeq 0$ and also $\Sigma^* = K^TZ^*K \succeq 0$ and $\text{rank}(\Sigma^*) = \text{rank}(K^TZ^*K) = \text{rank}(Z^*) = k.$}, that is,

$$Df(Y)[H] = \sum_{ij} h_{ij} \frac{\partial}{\partial y_{ij}} f(Y),$$

it is convenient to compute $\nabla_Y f(Y)$ as a unique element of $\mathbb{R}^{n \times n}$ that satisfies

$$< \nabla_Y f(Y), H >= Df(Y)[H],$$  \hfill (69)
for all \( H \). We first compute \( Df(Y)[H] \) and then use (69) to obtain the expression of \( \nabla_Y f(Y) \).

Note that the cost function \( f(Y) = \frac{1}{2}\|P_E(g(Y) - D_{obs})\|_F^2 \) can be expressed as \( f(Y) = h(k(Y)) = (h \circ k)(Y) \) where

\[
\begin{align*}
  h(R) &= \frac{1}{2}\|R\|_F^2, \quad \text{(70)} \\
  k(Y) &= P_E(g(Y)) - P_E(D) \\
  &= (P_E \circ g)(Y) - P_E(D). \quad \text{(71)}
\end{align*}
\]

Thus,

\[
Df(Y)[H] = D(h \circ k)(Y)[H] = Dh(k(Y))[Dk(Y)[H]]. \quad \text{(72)}
\]

For any two square matrices \( R \) and \( A \), we have

\[
\begin{align*}
  Dh(R)[A] &= \sum_{i,j} a_{ij} \frac{\partial}{\partial r_{ij}} h(R) \\
  &= \sum_{i,j} a_{ij} \frac{\partial}{\partial r_{ij}} \left( \frac{1}{2}\|R\|_F^2 \right) \\
  &= \sum_{i,j} a_{ij} \frac{\partial}{\partial r_{ij}} \left( \frac{1}{2} \sum_{p,q} r_{pq}^2 \right) \\
  &= \sum_{i,j} a_{ij} r_{ij} \\
  &= <R, A>.
\end{align*}
\]

By choosing \( R = k(Y) \) and \( A = Dk(Y)[H] \) in (73), we can rewrite (72) as

\[
Df(Y)[H] = <k(Y), Dk(Y)[H]> \quad \begin{array}{ll}
(a) & <k(Y), D((P_E(g(Y)) - P_E(D))[H]> \\
(b) & <k(Y), D(P_E \circ g)(Y)[H]> \\
(c) & <k(Y), DP_E(g(Y))[Dg(Y)[H]]>
\end{array}
\]

where (a) follows (70), (b) is because \( P_E(D) \) is not a function of \( Y \) and thus the Frechet differential of this is zero, and (c) is due to the chain rule.
Before we proceed, we remark that if $S$ is a linear operator (i.e., $S(\alpha_1 A_1 + \alpha_2 A_2) = \alpha_1 S(A_1) + \alpha_2 S(A_2)$), then
\[
DS(A)[B] = S(B)
\] (74)
for all matrices $A$ and $B$ (see Example 4.4.2 \cite{37}).

Since $P_E$ is a linear operator, $D P_E(g(Y))[Dg(Y)[H]] = P_E([Dg(Y)[H]])$ and hence
\[
Df(Y)[H] = < k(Y), P_E(Dg(Y)[H]) >
\]
\[\begin{align*}
(a) & = < P_E(k(Y)), Dg(Y)[H] > \\
(b) & = < k(Y), Dg(Y)[H] > \\
(c) & = < k(Y), g(H) > \\
(d) & = < k(Y), 2 \text{Sym}(1 \text{diag}(H)^T) - 2 \text{Sym}(H) > \\
& = 2 < k(Y), \text{Sym}(1 \text{diag}(H)^T) > -2 < k(Y), \text{Sym}(H) >,
\end{align*}
\] (75)
where (a) is because $P_E$ is a self-adjoint operator, (b) is because $P_E(k(Y)) = P_E((P_E \circ g)(Y) - P_E(Y)) = (P_E \circ g)(Y) - P_E(Y) = k(Y)$, (c) is because $g$ is also a linear function and thus $Dg(Y)[H] = g(H)$, and (d) is due to (4).

Now, the first term in (75) is
\[
2 < k(Y), \text{Sym}(1 \text{diag}(H)^T) > \stackrel{(a)}{=} 2 < \text{Sym}(k(Y)), 1 \text{diag}(H)^T > \\
\stackrel{(b)}{=} 2 < \text{Sym}(k(Y)), \text{diag}(H)1^T > \\
\stackrel{(c)}{=} 2 < \text{Sym}(k(Y))1, \text{diag}(H) > \\
\stackrel{(d)}{=} 2 < \text{eye}(\text{Sym}(k(Y)))1, H >,
\] (76)
where (a) is because $\text{Sym}()$ is a self-adjoint operator, (b) is because $< U, W >= < U^T, W^T >$, (c) is because $< A, b1^T > = tr(A^T b1^T) = tr((A1^T)b) = < A1, b >$, and (d) is because $\text{eye}()$ is the adjoint operator of $\text{diag}()$. Next, the second term in (75) is
\[
-2 < k(Y), \text{Sym}(H) > = -2 < \text{Sym}(k(Y)), H >.
\] (77)

\[\text{Let } A \text{ and } B \text{ be two linear operators in } \mathbb{R}^{n \times n}. \text{ If } < A(A), B >= < A, B(B) >, \text{ we say } A \text{ and } B \text{ are adjoint to each other in } \mathbb{R}^{n \times n}. \text{ Further, if } A \equiv B, \text{ then we say it is a self-adjoint operator.}\]
From (75), (76), and (77), we have
\[
D f(Y)_H = 2 \langle \text{eye}(\text{Sym}(k(Y)))1, H \rangle - 2 \langle \text{Sym}(k(Y)), H \rangle
= \langle 2\text{eye}(\text{Sym}(k(Y)))1 - 2\text{Sym}(k(Y)), H \rangle
\]  
(78)
From (69) and (78), we have
\[
\nabla_Y f(Y) = 2\text{eye}(\text{Sym}(k(Y)))1 - 2\text{Sym}(k(Y)),
\]
which is the desired result.

\section*{APPENDIX E}
\textbf{PROOF OF LEMMA 3.2}

\textbf{Proof:} If \( Y_i \) is the optimal point (i.e, \( Y_i = \arg \min_Y f(Y) \)), then \( \nabla f(Y_i) = 0 \) and \( Y_{i+1} = Y_i \). For all \( \alpha_i \geq 0 \), we have
\[
f(\mathcal{R}_{Y_i}(\alpha_i P_i)) = f(Y_{i+1}) = f(Y_i) + \tau \alpha_i < \nabla f(Y_i), P_i >,\]
satisfying A1. Next, we consider the case where \( Y_i \neq \arg \min_Y f(Y) \). First, we let
\[
g(\alpha) = f(\mathcal{R}_{Y_i}(\alpha P_i)),
\]
\[
h(\alpha) = f(Y_i) + \tau \alpha < \nabla f(Y_i), P_i >.\]  
(79)
Note that \( \langle \nabla f(Y_i), P_i \rangle \geq 0 \) (see Lemma 3.7) and \( g(\alpha) \geq 0 \). Since \( g(0) = f(\mathcal{R}_{Y_i}(0)) = f(Y_i) = h(0) \), \( g(\alpha) \) and \( h(\alpha) \) intersect at \( \alpha = 0 \). Also, when \( \tau \) varies from 0 to 1, the slope of \( h(\alpha) \) varies from 0 to \( |\langle \nabla f(Y_i), P_i \rangle| \). Since \( \frac{dg(\alpha)}{d\alpha} \bigg|_{\alpha=0} = \langle \nabla f(Y_i), P_i \rangle \), \( h(\alpha) \) is the tangential curve of \( g(\alpha) \) at \( \alpha = 0 \) when \( \tau = 1 \). Thus, there exits \( 0<\tau<1/2 \) such that \( h(\alpha) \) intersects \( g(\alpha) \) at some point \( \alpha > 0 \), which means that there exist \( \alpha_i > 0 \) satisfying
\[
f(\mathcal{R}_{Y_i}(\alpha_i P_i)) = g(\alpha_i) \leq h(\alpha_i) = f(Y_i) + \tau \alpha_i < \nabla f(Y_i), P_i >,\]
which completes the proof.
Proof: First, a lower bound of $\|\nabla_Y f(Y)\|_F$ is given by

$$\|\nabla_Y f(Y)\|_F \overset{(a)}{=} \|\text{eye}(R_i) - 2R_i\|_F^2,$$

$$\overset{(b)}{=} \|\text{eye}(R_i)\|_F^2 + \|2R_i\|_F^2$$

$$\geq \|2R_i\|_F^2$$

$$= 4\|P_E(D_i) - P_E(D)\|_F^2,$$  \hspace{1cm} (80)

where (a) is from (17) and (b) is from the fact that diagonal entries of $R_j$ are all zeros and $\text{eye}(R_j1)$ is a diagonal matrix. That is, positions of nonzero elements in $\text{eye}(R_i1)$ and $R_i$ are disjoint. An upper bound is obtained as follows.

$$\|\nabla_Y f(Y)\|_F \leq \|\text{eye}(R_i)\|_F + \|2R_i\|_F$$

$$\overset{(a)}{\leq} \|2R_i\|_F + \|2R_i\|_F$$

$$\overset{(b)}{\leq} 2\|R_i\|_F + 2\|R_i\|_F$$

$$\leq (2\sqrt{n} + 2)\|R_i\|_F$$

$$\leq (2\sqrt{n} + 2)\|P_E(D_i) - P_E(D)\|_F,$$  \hspace{1cm} (81)

where (a) is because $\|\text{eye}(b)\|_F = \|b\|_2$ for any vector $b$, and (b) is because $\|Ab\|_2 \leq \|A\|_F \|b\|_2$ for any matrix $A$ and any vector $b$. By combining (80) and (81), we obtain the desired result.

\[\square\]

APPENDIX G

PROOF OF LEMMA 3.7

Proof: Recall from (25) that we have

$$P_{i+1} = -\nabla f(Y_{i+1}) + \beta_{i+1}P_{TV_{i+1}}\tilde{\gamma}(P_i).$$

Thus,

$$< -\nabla f(Y_{i+1}), P_{i+1} > = \| -\nabla f(Y_{i+1})\|_F^2 + \beta_{i+1} < -\nabla f(Y_{i+1}), P_{TV_{i+1}}\tilde{\gamma}(P_i) >$$

$$= \| -\nabla f(Y_{i+1})\|_F^2 + \beta_{i+1} < -P_{TV_{i+1}}\tilde{\gamma}(\nabla f(Y_{i+1})), P_i >$$

$$\overset{(a)}{=} \| -\nabla f(Y_{i+1})\|_F^2 + \beta_{i+1} < -\nabla f(Y_{i+1}), P_i >$$
where (a) is because $\nabla f(Y_{i+1}) \in T_{Y_{i+1}}\hat{Y}$. Then we have

$$
|<\nabla f(Y_{i+1}), P_{i+1}> + \|\nabla f(Y_{i+1})\|_F^2| = \beta_{i+1} |<\nabla f(Y_{i+1}), P_i>| \leq \beta_{i+1}\mu <\nabla f(Y_i), P_i>,
$$

where (a) is from the assumption A2.

If we denote $\zeta_i = -\frac{<\nabla f(Y_i), P_i>}{\|\nabla f(Y_i)\|_F^2}$, then

$$
\left| -\zeta_{i+1}\|\nabla f(Y_{i+1})\|_F^2 + \|\nabla f(Y_{i+1})\|_F^2 \right| \leq \beta_{i+1}\mu \zeta_i \|\nabla f(Y_i)\|_F^2,
$$

and also

$$
| -\zeta_{i+1} + 1 | \leq \mu \beta_{i+1} \zeta_i. \tag{82}
$$

From Fletcher-Reeves rule in (34), we have $\beta_{i+1} \|\nabla f(Y_{i+1})\|_F^2 = 1$ and thus

$$
| -\zeta_{i+1} + 1 | \leq \mu \zeta_i.
$$

In other words,

$$
\zeta_{i+1} \geq 1 - \mu \zeta_i, \tag{83}
$$

and

$$
\zeta_{i+1} \leq 1 + \mu \zeta_i,
$$

$$
\zeta_i \leq 1 + \mu \zeta_{i-1}
$$

$$
\vdots
$$

$$
\zeta_2 \leq 1 + \mu \zeta_1,
$$

where we set $\zeta_1 = 1$. Thus,

$$
\zeta_i \leq \sum_{j=0}^{i-1} \mu^j = \frac{1 - \mu^i}{1 - \mu}. \tag{84}
$$

From (83) and (84), we finally have

$$
\zeta_{i+1} \geq 1 - \mu \frac{1 - \mu^i}{1 - \mu} = \frac{1 - 2\mu + \mu^{i+1}}{1 - \mu},
$$

which is the desired result.
APPENDIX H

PROOF OF LEMMA 3.8

Proof: From (24), we have
\[ \text{grad}\ f(Y_i) = P_{T_{Y_i}, \tilde{Y}}(\nabla_{\text{E}} f(Y_i)), \]
where \( \nabla_{\text{E}} f(Y_i) \) is the Euclidean gradient. Let \( P_{T_{Y_i}, \tilde{Y}}^\perp \) be the orthogonal operator on the complement space of \( T_{Y_i, \tilde{Y}} \), then we obtain
\[ \| \nabla_{\text{E}} f(Y_i) \|_F^2 = \| P_{T_{Y_i}, \tilde{Y}}(\nabla_{\text{E}} f(Y_i)) \|_F^2 + \| P_{T_{Y_i}, \tilde{Y}}^\perp(\nabla_{\text{E}} f(Y_i)) \|_F^2, \]
and hence
\[ \| \text{grad}\ f(Y_i) \|_F^2 = \| P_{T_{Y_i}, \tilde{Y}}(\nabla_{\text{E}} f(Y_i)) \|_F^2 \]
\[ = \| \nabla_{\text{E}} f(Y_i) \|_F^2 - \| P_{T_{Y_i}, \tilde{Y}}^\perp(\nabla_{\text{E}} f(Y_i)) \|_F^2. \]

Now, we define
\[ \chi = \begin{cases} \sup_{Y \in \{Y_i\}_{i=1}^\infty} \frac{\| P_{T_{Y_i}, \tilde{Y}}^\perp(\nabla_{\text{E}} f(Y)) \|_F}{\| \nabla_{\text{E}} f(Y) \|_F} & \text{if } \| \nabla_{\text{E}} f(Y) \|_F \neq 0 \\ 1 & \text{otherwise} \end{cases}. \]

Note that \( 1 \geq \chi \geq 0 \) because \( \| \nabla_{\text{E}} f(Y) \|_F \geq \| P_{T_{Y_i}, \tilde{Y}}^\perp(\nabla_{\text{E}} f(Y)) \|_F \) (see (85)). From (86) and (87), we have
\[ \| \text{grad}\ f(Y_i) \|_F^2 = \left( 1 - \frac{\| P_{T_{Y_i}, \tilde{Y}}^\perp(\nabla_{\text{E}} f(Y_i)) \|_F^2}{\| \nabla_{\text{E}} f(Y_i) \|_F^2} \right) \| \nabla_{\text{E}} f(Y_i) \|_F^2 \]
\[ \geq (1 - \chi^2) \| \nabla_{\text{E}} f(Y_i) \|_F^2. \]

Now, what remains is to show that \( \| \nabla_{\text{E}} f(Y_i) \|_F^2 \geq 8 f(Y_i) \). Indeed, from Lemma 2.7, we have
\[ \| \nabla_{\text{E}} f(Y_i) \|_F^2 = \| \text{eye}(\{R + R^T \}) 1 - 2R \|_F^2, \]
where \( R = \mathcal{P}_E(g(Y)) - \mathcal{P}_E(D) \). Noting that \( R \) is symmetric with zero diagonal entries \( r_{ii} = 0 \), we have

\[
\frac{1}{4} \| \nabla f(Y_i) \|_F^2 = \| \text{eye}(R1) \|_F^2 + \| R \|_F^2 - 2 < \text{eye}(R1), R > \\
= \| R1 \|_2^2 + \| R \|_F^2 - 2 \sum_i \left( \sum_j r_{ij} \right) r_{ii} \\
= \| R1 \|_2^2 + \| R \|_F^2 \\
\geq \| R \|_F^2 \\
= 2f(Y_i). \tag{90}
\]

By substituting (90) into (89), we obtain the desired result. 

\[\blacksquare\]

**APPENDIX I**

**PROOF OF LEMMA 3.9**

**Proof:**

By denoting \( x_i = [x_{i1} \ x_{i2} \ \cdots \ x_{ik}]^T \), we have

\[
p = P(d_{ij} \leq r) \\
= P(\|x_i - x_j\|_2^2 \leq r^2) \\
= P(\sum_{t=1}^{k}(x_{it} - x_{jt})^2 \leq r^2).
\]

Before finding the general form of \( p \), we compute the distribution of \( Y = (X_1 - X_2)^2 \) where \( X_1 \) and \( X_2 \) are i.i.d. uniformly distributed random variables at unit interval. Let \( Z = X_1 - X_2 \), then the cdf of \( Z \) is given by

\[
F_Z(z) = P(Z \leq z) \\
= P(X_1 - X_2 \leq z) \\
= \int_{0}^{1} P(X_1 \leq z + x_2 | X_2 = x_2) f_{X_2}(x_2) dx_2 \\
= \int_{0}^{1} P(X_1 \leq z + x_2) f_{X_2}(x_2) dx_2 \\
= \int_{0}^{1} F_{X_1}(z + x_2) f_{X_2}(x_2) dx_2.
\]
Thus,
\[
f_Z(z) = \frac{d}{dz} F_Z(z) = \begin{cases} 
1 - |z| & \text{if } |z| \leq 1 \\
0 & \text{otherwise}
\end{cases}
\]

Now, we can easily obtain the cdf of \( Y = Z^2 \)
\[
F_Y(y) = \begin{cases} 
1 & \text{if } 1 \leq y \\
2\sqrt{y} - y & \text{if } 0 \leq y \leq 1 \\
0 & \text{if } y \leq 0
\end{cases}
\]

and also the pdf of \( Y \)
\[
f_Y(y) = \begin{cases} 
\frac{1}{\sqrt{y}} - 1 & \text{if } 0 \leq y \leq 1 \\
0 & \text{otherwise}
\end{cases}
\tag{91}
\]

Using this, we can compute \( p \) as
\[
p = Pr \left( \sum_{t=1}^{k} y_t \leq r^2 \right) \\
= \int_{\alpha_1 + \alpha_2 + \ldots + \alpha_k \leq r^2} \cdots \int f_{Y_1, \ldots, Y_k}(\alpha_1, \ldots, \alpha_k) d\alpha_1 \ldots d\alpha_k \\
= \int_{\alpha_1 + \alpha_2 + \ldots + \alpha_k \leq r^2} \cdots \int f_{Y_1}(\alpha_1) \cdots f_{Y_k}(\alpha_k) d\alpha_1 \ldots d\alpha_k, \tag{92}
\]

where \( y_t = (x_{it} - x_{jt})^2 \). When the sensor nodes are located in two dimensional Euclidean space \((k = 2)\), we have
\[
p = \int \int f_{Y_1}(\alpha_1) f_{Y_2}(\alpha_2) d\alpha_1 d\alpha_2.
\]

Let \( t = \alpha_1 + \alpha_2 \), then we have
\[
p = \int \int f_{Y_1}(\alpha_1) f_{Y_2}(\alpha_2) d\alpha_1 d\alpha_2 \\
= \int_{0}^{r^2} \left[ \int_{1}^{t} f_{Y_1}(\alpha_1) f_{Y_2}(t - \alpha_1) d\alpha_1 \right] dt \\
= \int_{0}^{r^2} f_{Y_1}(t) * f_{Y_2}(t) dt. \tag{93}
\]
After some manipulations, we have
\[
f_Y(t) = \begin{cases} 
\pi - 4\sqrt{t} + t & \text{if } 0 \leq t \leq 1 \\
2\sin^{-1}\left(\frac{\sqrt{t}}{2}\right) - 1 + 4\sqrt{t - 1} - t - 2 & \text{if } 1 \leq t \leq 2 \\
0 & \text{otherwise}
\end{cases}
\] (94)

Let \( h_1(u) = \int_0^u (\pi - 4\sqrt{t} + t) \, dt \) (0 \leq u \leq 1) and \( h_2(u) = \int_1^u (2\sin^{-1}\left(\frac{\sqrt{t}}{2}\right) - 1 + 4\sqrt{t - 1} - t - 2) \, dt \) (1 \leq u \leq 2), then we have
\[
p = \begin{cases} 
h_1(r^2) & \text{if } 0 \leq r \leq 1 \\
 h_1(1) + h_2(r^2) & \text{if } 1 \leq r \leq \sqrt{2} \\
1 & \text{otherwise}
\end{cases}
\]

where \( h_1(u) \) and \( h_2(u) \) are given by
\[
h_1(u) = \pi u - \frac{8}{3}u^{3/2} + \frac{1}{2}u^2, \\
h_2(u) = \frac{8}{3}(u - 1)^{3/2} - u^2 - \frac{1}{2}u - 2 \int_1^u \sin^{-1}\left(\frac{2}{t} - 1\right) \, dt \\
= \frac{8}{3}(u - 1)^{3/2} - \frac{1}{2}(u - 1)(u + 5) + 2 \int_1^u \sin^{-1}\left(\frac{2}{t} - 1\right) \, dt \\
= \frac{8}{3}(u - 1)^{3/2} - \frac{1}{2}(u - 1)(u + 5) + 2u \sin^{-1}\left(\frac{2}{u} - 1\right) \\
+ \frac{2}{1 + \tan\left(\sin^{-1}\left(\frac{u}{2}\right)\right)} + 3\sqrt{u - 1} - \pi - 1.
\]

Denoting \( p_1(r) = h_1(1) + h_2(r^2) \), we get the desired result for \( k = 2 \).

Similarly, when the sensor nodes are located in three dimensional Euclidean space (\( k = 3 \)), we have
\[
p = \iint f_{Y_1}(\alpha_1) f_{Y_2}(\alpha_2) f_{Y_3}(\alpha_3) \, d\alpha_1 d\alpha_2 d\alpha_3 \\
= \int_0^{r^2} \left[ \int_0^1 \int_0^1 f_{Y_1}(\alpha_1) f_{Y_2}(\alpha_2) f_{Y_3}(t - \alpha_1 - \alpha_2) \, d\alpha_1 d\alpha_2 \right] \, dt \\
= \int_0^{r^2} \left[ \int_0^1 \int_0^1 f_{Y_1}(\alpha_1) f_{Y_2}(u - \alpha_1) \, d\alpha_1 \right] f_{Y_3}(t - u) \, du \, dt \\
= \int_0^{r^2} [f_{Y_1}(t) * f_{Y_2}(t)] * f_{Y_3}(t) \, dt.
\]
After some manipulations, we have

\[ [f_{Y_1}(t) * f_{Y_2}(t)] * f_{Y_3}(t) = \begin{cases} \tilde{h}_3(t) & \text{if } 0 \leq t \leq 1 \\ \tilde{h}_4(t) & \text{if } 1 \leq t \leq 2 \\ \tilde{h}_5(t) & \text{if } 2 \leq t \leq 3 \\ 0 & \text{otherwise} \end{cases}, \tag{95} \]

where

\[
\begin{align*}
\tilde{h}_3(t) &= 4t \sqrt{t} + 2\pi \sqrt{t} - 3\pi t - \frac{1}{2} t^2, \\
\tilde{h}_4(t) &= 3\pi - 4\pi \sqrt{t} + (3\pi + 4)t + \frac{1}{2} t^2 - 8t \sqrt{t - 1} - 3\sqrt{t - 1} + \frac{1}{2} (t - 1)^2 \\
&\quad - 4t \sin^{-1} \sqrt{\frac{1}{t}} + 4t \sin^{-1} \sqrt{1 - \frac{1}{t}} - 2t \sin^{-1} \left( \frac{2}{t} - 1 \right) - \frac{2}{1 + \tan \left( \frac{\sin^{-1} \left( \frac{2}{t} - 1 \right)}{2} \right)}, \\
\tilde{h}_5(t) &= 3\sqrt{t - 2} + 4t \sqrt{t - 2} - 4t - \frac{1}{2} (t - 1)^2 - 3 + 8\sqrt{t} \tan^{-1} \sqrt{\frac{t - 2}{t}} \\
&\quad - 8 \tan^{-1} \sqrt{t - 2} - 4(t - 1) \sin^{-1} \sqrt{\frac{t - 2}{t - 1}} + 2(t + 1) \sin^{-1} \left( \frac{3 - t}{t - 1} \right) \\
&\quad - 8\sqrt{t} \tan^{-1} \sqrt{\frac{1}{t(t - 2)}} + 8 \tan^{-1} \sqrt{\frac{1}{t - 2}} + 4(t - 1) \sin^{-1} \sqrt{\frac{1}{t - 1}} \\
&\quad + \frac{2}{1 + \tan \left( \frac{\sin^{-1} (\sqrt{t - 2})}{2} \right)}.
\end{align*}
\]

Now, we let \( h_3(u) = \int_0^u \tilde{h}_3(t) \, dt \) (0 ≤ u ≤ 1), \( h_4(u) = \int_1^u \tilde{h}_4(t) \, dt \) (1 ≤ u ≤ 2), and \( h_5(u) = \int_2^u \tilde{h}_5(t) \, dt \) (2 ≤ u ≤ 3), then we have

\[
p = \begin{cases} 
\begin{cases} h_3(r^2) & \text{if } 0 \leq r \leq 1 \\
h_3(1) + h_4(r^2) & \text{if } 1 \leq r \leq \sqrt{2} \\
h_3(1) + h_4(2) + h_5(r^2) & \text{if } \sqrt{2} \leq r \leq \sqrt{3} \\
1 & \text{otherwise} \end{cases} 
\end{cases}, \tag{96} \]
where \( h_3(u), h_4(u), \) and \( h_5(u) \) are given by

\[
\begin{align*}
    h_3(u) &= \frac{4\pi}{3} u \sqrt{u} + \frac{8}{5} u^2 \sqrt{u} - \frac{3\pi}{2} u^2 - \frac{1}{6} u^3, \\
    h_4(u) &= 3\pi u - \frac{8\pi}{3} u \sqrt{u} + \frac{3\pi + 4}{2} u^2 + \frac{1}{6} u^3 + \frac{1}{6} (u - 1)^3 - \frac{\pi}{3} - \frac{5}{2} - 6\sqrt{u - 1} \\
        &\quad - \frac{28}{3} (u - 1) \sqrt{u - 1} - \frac{16}{5} (u - 1)^2 \sqrt{u - 1} - 2u^2 \sin^{-1} \sqrt{\frac{1}{u}} \\
        &\quad + 2u^2 \sin^{-1} \sqrt{1 - \frac{1}{u} - u^2 \sin^{-1} \left( \frac{2}{u} - 1 \right)} - \frac{16}{3} \left( 1 + \tan \left( \frac{1}{2} \sin^{-1} \left( \frac{2}{u} - 1 \right) \right) \right)^2, \\
    h_5(u) &= 2(u - 2) \sqrt{u - 2} - 2u^2 - \frac{1}{6} (u - 1)^3 - 3u + \frac{29}{2} + \frac{8}{5} (u - 2)^2 \sqrt{u - 2} \\
        &\quad + \frac{22}{3} (u - 2) \sqrt{u - 2} + 14 \sqrt{u - 2} + \left( \frac{16\sqrt{2}}{3} - 12 \right) \pi + 2 \tan^{-1} \sqrt{u - 2} \\
        &\quad - 8u \tan^{-1} \sqrt{u - 2} - \frac{16}{3} u \sqrt{u} \tan^{-1} \sqrt{u - 2} - 2u(u - 2) \sin^{-1} \sqrt{\frac{u - 2}{u - 1}} \\
        &\quad + u(u + 2) \sin^{-1} \left( \frac{3 - u}{u - 1} \right) - \frac{16}{3} u \sqrt{u} \tan^{-1} \sqrt{\frac{1}{u(u - 2)}} + 8u \tan^{-1} \sqrt{\frac{1}{u - 2}} \\
        &\quad + 2u(u - 2) \sin^{-1} \sqrt{\frac{1}{u - 1}} + \frac{16}{3} \left( 1 + \tan \left( \frac{1}{2} \sin^{-1} \left( \frac{3 - u}{u - 1} \right) \right) \right)^2 \\
        &\quad - \frac{4}{(1 + \tan \left( \frac{1}{2} \sin^{-1} \left( \frac{3 - u}{u - 1} \right) \right))^2}.
\end{align*}
\]

By denoting \( p_2(r) = h_3(1) + h_4(r^2) \) and \( p_3(r) = h_3(1) + h_4(2) + h_5(r^2) \), we get the desired result for \( k = 3 \).

\[ \blacksquare \]

**APPENDIX J**

**PROOF OF LEMMA 3.13**

*Proof:* Since \( \|A\|_F^2 = \|P_E(A)\|_F^2 + \|P_E^\perp(A)\|_F^2 \), we can rewrite \( t\|A\|_F^2 \leq \|P_E(A)\|_F^2 \) as

\[
t\|A\|_F^2 \leq \|A\|_F^2 - \|P_E^\perp(A)\|_F^2,
\]

and also

\[
\|P_E^\perp(A)\|_F^2 \leq (1 - t)\|A\|_F^2.
\]

August 4, 2017 DRAFT
To show that (97) holds true with overwhelming probability, we first have
\[
P(\|P_E^\perp(A)\|_F^2 \geq (1-t)\|A\|_F^2) = P(\exp(\epsilon\|P_E^\perp(A)\|_F^2) \geq \exp(\epsilon(1-t)\|A\|_F^2))
\]
\[
\leq \exp(-\epsilon(1-t)\|A\|_F^2) E \left[ \exp(\epsilon\|P_E^\perp(A)\|_F^2) \right]
\]
\[
= \exp(-\epsilon(1-t)\|A\|_F^2) E \left[ \prod_{i \neq j} \exp(\epsilon(1-\delta_{ij})a_{ij}^2) \right], \quad (98)
\]
for any $\epsilon > 0$, where (a) follows from the Markov inequality and (b) is from $\|P_E^\perp(A)\|_F^2 = \sum_{i \neq j} (1-\delta_{ij})a_{ij}^2$ (see (35)). Let $\Omega = \{(i, j) : a_{ij} \neq 0\}$ (i.e., $\Omega$ is the index set of nonzero entries of $A$), and $N = 2^{\left\lfloor \log_2 |\Omega| \right\rfloor}$ ($|\Omega|$ is the cardinality of $\Omega$). Also, let $\tilde{\Omega}$ be a subset of $\Omega$ such that $|\tilde{\Omega}| = N$, then
\[
E \left[ \prod_{i \neq j} \exp(\epsilon(1-\delta_{ij})a_{ij}^2) \right] \leq \prod_{(i, j) \in (\Omega, \tilde{\Omega})} \exp(\epsilon a_{ij}^2) E \left[ \prod_{(i, j) \in \tilde{\Omega}} \exp(\epsilon(1-\delta_{ij})a_{ij}^2) \right]^{1/N}
\]
\[
= \prod_{(i, j) \in (\Omega, \tilde{\Omega})} \exp(\epsilon a_{ij}^2) \left( \prod_{(i, j) \in \tilde{\Omega}} E \left[ \exp(N\epsilon(1-\delta_{ij})a_{ij}^2) \right] \right)^{1/N}
\]
\[
= \prod_{(i, j) \in (\Omega, \tilde{\Omega})} \exp(\epsilon a_{ij}^2) \prod_{(i, j) \in \tilde{\Omega}} (1-p) \exp(N\epsilon a_{ij}^2) + p)^{1/N}
\]
\[
= \prod_{(i, j) \in \tilde{\Omega}} \exp(\epsilon a_{ij}^2) \prod_{(i, j) \in \tilde{\Omega}} \exp(\epsilon a_{ij}^2) \left( 1-p + p \exp(-N\epsilon a_{ij}^2) \right)^{1/N}
\]
\[
= \exp(\epsilon\|A\|_F^2) \prod_{(i, j) \in \tilde{\Omega}} \left( 1-p + p \exp(-N\epsilon a_{ij}^2) \right)^{1/N}
\]
\[
\leq \exp(\epsilon\|A\|_F^2) \prod_{(i, j) \in \tilde{\Omega}} \left( 1-p + p \exp(-N\epsilon a_{\min}^2) \right)^{1/N}
\]
\[
\leq \exp(\epsilon\|A\|_F^2) \left( 1-p + p \exp(-N\epsilon a_{\min}^2) \right), \quad (99)
\]
where (a) is because $E \left[ \prod_{i=1}^M A_i \right] \leq \left( \prod_{i=1}^M E \left[ A_i^M \right] \right)^{1/M}$ for positive random variable $A_i$ and $M = 2^q \geq 1$, (b) is because $a_{\min} = \min_{(i, j) \in \Omega} |a_{ij}|$, and (c) is because $|\tilde{\Omega}| = N$.

In summary, we have
\[
P(\|P_E^\perp(A)\|_F^2 \geq (1-t)\|A\|_F^2) \leq g(\epsilon),
\]
where \( g(\epsilon) = \exp(mtN\epsilon_2^{a_{min}})(1-p+p\exp(-N\epsilon_2^{a_{min}})) \) \((m = ||A||_F^2/(a_{min}^2N))\). If \(0<mt<1\), we obtain the minimum value of \(g(\epsilon)\) at \(\epsilon^* = 1/(Na_{min}^2N)\log((1-mp)/(1-pmt))\). Thus,

\[
P(\|P_E(A)\|_F^2 \geq (1-t)\|A\|_F^2) \leq g(\epsilon^*)
\]

\[
= \left(1 - \frac{p}{1 - mt}\right)^{1 - mt} \left(\frac{p}{mt}\right)^{mt}
\]

\[
= \exp\left(-\left((1 - mt) \log\left(\frac{1 - mt}{1 - p}\right) + mt \log\left(\frac{mt}{p}\right)\right)\right),
\]

which is the desired result.

**APPENDIX K**

**PROOF OF LEMMA 3.17**

*Proof:* We first denote \(B = g(Y) - D\), and then express \(B = \sum_{ij} <B,e_ie_j^T,e_i^T, e_j^T,\) which gives

\[
P_E(B) = \sum_{i \neq j} \delta_{ij} <B,e_ie_j^T,e_i^T,
\]

Recalling that \(l(A) = 2\text{eye}(\text{Sym}(A)1) - 2\text{Sym}(A)\), we have

\[
\nabla_Y f(Y) = l(P_E(B))
\]

\[
= l(\sum_{i \neq j} \delta_{ij} <B,e_ie_j^T,e_i^T)\]

\[
= l(\sum_{i \neq j} \delta_{ij} <B,e_i e_j^T,l(e_i^T,e_j^T))
\]

\[
= \sum_{i \neq j} \delta_{ij}s_{ij}l(e_i^T,e_j^T),
\]

where \(s_{ij} = <B,e_i e_j^T>\) and (a) is because \(l(\alpha C + \beta D) = \alpha l(C) + \beta l(D)\). Now, if we let

\[
I = ||\nabla_Y f(Y)||_F^2 - c^2||\text{grad} f(Y)||_F^2,
\]

then we have

\[
I = \sum_{i \neq j} \sum_{u \neq v} \delta_{ij}\delta_{uv}s_{ij}s_{uv} < l(e_i^T,e_j^T), l(e_u^T,e_v^T),
\]

\[
- c^2 \sum_{i \neq j} \sum_{u \neq v} \delta_{ij}\delta_{uv}s_{ij}s_{uv} < P_{TY\tilde{Y}}(l(e_i^T)), P_{TY\tilde{Y}}(l(e_u^T))>
\]

\[
= \sum_{i \neq j} \sum_{u \neq v} \delta_{ij}\delta_{uv}s_{ij}s_{uv} < l(e_i^T,e_j^T), l(e_u^T,e_v^T)>
\]

\[
- c^2 \sum_{i \neq j} \sum_{u \neq v} \delta_{ij}\delta_{uv}s_{ij}s_{uv} < P_{TY\tilde{Y}}(l(e_i^T)), l(e_u^T,e_v^T)>
\]

August 4, 2017  DRAFT
\[ \sum_{i \neq j} \delta_{ij} \delta_{uv} s_{ij} s_{uv} < (\mathcal{I} - c^2 P_{\mathcal{Y}^\perp}) l(e_i e_j^T), l(e_u e_v^T) > \]
\[ \leq \sum_{i \neq j} \delta_{ij} \delta_{uv} |s_{ij} s_{uv} < (\mathcal{I} - c^2 P_{\mathcal{Y}^\perp}) l(e_i e_j^T), l(e_u e_v^T) > | \]
\[ \leq \frac{1}{2} \sum_{i \neq j} \sum_{u \neq v} (\delta_{ij}^2 + \delta_{uv}^2) |s_{ij} s_{uv} < (\mathcal{I} - c^2 P_{\mathcal{Y}^\perp}) l(e_i e_j^T), l(e_u e_v^T) > | \]
\[ = \frac{1}{2} \sum_{i \neq j} \sum_{u \neq v} (\delta_{ij} + \delta_{uv}) |s_{ij} s_{uv} < (\mathcal{I} - c^2 P_{\mathcal{Y}^\perp}) l(e_i e_j^T), l(e_u e_v^T) > | \]
\[ = \frac{1}{2} \sum_{i \neq j} \sum_{u \neq v} |s_{ij} s_{uv} < (\mathcal{I} - c^2 P_{\mathcal{Y}^\perp}) l(e_i e_j^T), l(e_u e_v^T) > | \]
\[ + \frac{1}{2} \sum_{i \neq j} \sum_{u \neq v} |s_{ij} s_{uv} < (\mathcal{I} - c^2 P_{\mathcal{Y}^\perp}) l(e_u e_v^T), l(e_i e_j^T) > | \]
\[ = \sum_{i \neq j} \sum_{u \neq v} |s_{ij} s_{uv} < (\mathcal{I} - c^2 P_{\mathcal{Y}^\perp}) l(e_i e_j^T), l(e_u e_v^T) > |, \]

where (a) is because \( < P_{\mathcal{Y}^\perp}(E), F > = < P_{\mathcal{Y}^\perp}(E), P_{\mathcal{Y}^\perp}(F) > + < P_{\mathcal{Y}^\perp}(E), P_{\mathcal{Y}^\perp}(F) > = < P_{\mathcal{Y}^\perp}(E), P_{\mathcal{Y}^\perp}(F) > \), (b) is because \( x^2 + y^2 \geq 2xy (x, y \geq 0) \), and (c) is because \( (\mathcal{I} - c^2 P_{\mathcal{Y}^\perp}) \) is a self-adjoint operator.

\[ \text{APPENDIX L} \]

\[ \text{PROOF OF LEMMA 3.18} \]

\[ \text{Proof:} \] Recalling that \( l(A) = 2\text{eye}(\text{Sym}(A)1) - 2\text{Sym}(A) \), we have

\[ l(e_i e_j^T) = 2\text{eye}(\text{Sym}(e_i e_j^T)1) - 2\text{Sym}(e_i e_j^T) \]
\[ = \text{eye}(e_i e_j^T1 + e_j e_i^T1) - e_i e_j^T - e_j e_i^T \]
\[ \overset{(a)}{=} \text{eye}(e_i + e_j) - e_i e_j^T - e_j e_i^T \]
\[ = e_i e_j^T + e_j e_i^T - e_i e_j^T - e_j e_i^T \]
\[ = (e_i - e_j)(e_i - e_j)^T, \]

August 4, 2017 DRAFT
where (a) is because \( \text{eye}(e_i e_j^T 1) = \text{eye}(e_i) \). Let \( \delta = e_i - e_j \), then \( l(e_i e_j^T) = \delta \delta^T \). Also, if \( n^2 \geq 4c\mu(Y)^2 k^2 \), then
\[
\begin{align*}
| & < (I - cP_{TVY}) l(e_i e_j^T), l(e_i e_j^T) > | \\
= & | < (I - cP_{TVY}) \delta \delta^T, \delta \delta^T > | \\
= & | < \delta \delta^T - cP_{TVY} (\delta \delta^T), \delta \delta^T > | \\
= & | < \delta \delta^T, \delta \delta^T > - c < P_{TVY} (\delta \delta^T), \delta \delta^T > | \\
\overset{(a)}{=} & | < \delta \delta^T, \delta \delta^T > - c < P_{Q} \delta \delta^T + \delta \delta^T P_{Q} - P_{Q} \delta \delta^T P_{Q}, \delta \delta^T > | \\
= & | < \delta \delta^T, \delta \delta^T > - c( < P_{Q} \delta \delta^T, \delta \delta^T > + < \delta \delta^T P_{Q}, \delta \delta^T > - < P_{Q} \delta \delta^T P_{Q}, \delta \delta^T > ) | \\
\overset{(b)}{=} & | \delta^T \delta \delta^T \delta - c( \delta^T P_{Q} \delta \delta^T \delta + \delta^T \delta \delta^T P_{Q} \delta - \delta^T P_{Q} \delta \delta^T P_{Q} \delta ) | \\
\overset{(c)}{=} & || \delta ||_2^2 - c(2 || P_{Q} \delta ||_2^2 || \delta ||_2^2 - || P_{Q} \delta ||_2^4 ) | \\
\overset{(d)}{=} & | 4 - 4c || P_{Q} \delta ||_2^2 + c || P_{Q} \delta ||_2^4 | \\
\overset{(e)}{\geq} & 4 - 4c \frac{4\mu(Y)^2 k^2}{n^2} + c \frac{16\mu(Y)^4 r^4}{n^4} \\
\geq & 4 \left( 1 - \frac{4c\mu(Y)^2 k^2}{n^2} \right) ,
\end{align*}
\]

where (a) follows from Proposition 2.4, (b) is because \( < X, zz^T > = \text{tr}(Xzz^T) = z^T Xz \), (c) is because \( P_{Q}^T P_{Q} = P_{Q} \) and thus \( \delta^T P_{Q} \delta = \delta^T P_{Q}^T P_{Q} \delta = || P_{Q} \delta ||_2^2 \), (d) is because \( || \delta ||_2^2 = 2 \) for \( i \neq j \), and (e) is because \( || P_{Q} \delta ||_2 = || P_{Q} e_i - P_{Q} e_j ||_2 \leq || P_{Q} e_i ||_2 + || P_{Q} e_j ||_2 \leq 2\mu(Y)r/n \) (see Definition 3.14). 

\[\square\]

**APPENDIX M**

**PROOF OF LEMMA 3.19**

**Proof:** For any \( t > 0 \), we have

\[
P(\sum_{i=1}^{N} \delta_i a_i \geq \epsilon) = P(\exp(t \sum_{i=1}^{N} \delta_i a_i) \geq \exp(t\epsilon))
\]
\[
\overset{(a)}{\leq} \exp(-t\epsilon) E \left[ \exp(t \sum_{i=1}^{N} \delta_i a_i) \right] \\
\overset{(b)}{\leq} \exp(-t\epsilon + t \sum_{i=2^n+1}^{N} a_i) E \left[ \exp(t \sum_{i=1}^{2^n} \delta_i a_i) \right]
\]

August 4, 2017 DRAFT
\[
\begin{align*}
\exp(-te + t \sum_{i = 2^{n+1}}^{N} a_i) E \left[ \prod_{i=1}^{2^n} \exp(t \delta_i a_i) \right] \\
\overset{(c)}{\leq} \exp(-te + t \sum_{i = 2^{n+1}}^{N} a_i) \left( \prod_{i=1}^{2^n} E \left[ \exp(t \delta_i a_i (2^n) \right] \right)^{1/2^n} \\
= \exp(-te + t \sum_{i = 2^{n+1}}^{N} a_i) \left( \prod_{i=1}^{2^n} \exp(t 2^n a_i \right) \left( \prod_{i=1}^{2^n} ((1 - p) \exp(-t 2^n a_i) + p) \right)^{1/2^n} \\
= \exp(-te + t \sum_{i = 2^{n+1}}^{N} a_i) \left( \prod_{i=1}^{2^n} \exp(t 2^n a_i) \right) \left( \prod_{i=1}^{2^n} ((1 - p) \exp(-t 2^n a_i) + p) \right)^{1/2^n} \\
= \exp(-te + t \sum_{i = 2^{n+1}}^{N} a_i) \left( \prod_{i=1}^{2^n} \exp(t a_i) \left( \prod_{i=1}^{2^n} ((1 - p) \exp(-t 2^n a_i) + p) \right) \right)^{1/2^n} \\
= \exp(-te + t \sum_{i = 2^{n+1}}^{N} a_i) \left( \prod_{i=1}^{2^n} ((1 - p) \exp(-t 2^n a_i) + p) \right)^{1/2^n} \\
\overset{(d)}{\leq} \exp(-te + t \sum_{i = 1}^{N} a_i) \left( \prod_{i=1}^{2^n} ((1 - p) \exp(-t 2^n a_{\text{min}}) + p) \right)^{1/2^n} \\
= \exp(-te + t \sum_{i = 1}^{N} a_i) ((1 - p) \exp(-t 2^n a_{\text{min}}) + p),
\end{align*}
\]

where (a) follows from the Markov’s inequality, (b) is because \( \delta_i \leq 1 \) for all \( i \), (c) is because \( E[\prod_{i=1}^{M} X_i] \leq (\prod_{i=1}^{M} E[X_i^M])^{1/M} \) for positive random variables \( X_i \) and \( M = 2^q \) (\( q \geq 1 \)), (d) is because \( \delta_i \) is Bernoulli random variable with \( P(\delta_i = 1) = p \), and (e) is because \( a_{\text{min}} = \min_i a_i \).

Let \( g(t) = \exp(-te + t \sum_{i = 1}^{N} a_i) ((1 - p) \exp(-t 2^n a_{\text{min}}) + p) \), then the minimum of \( g(t) \) is obtained
at $t^* = 1/(2^na_{\text{min}}) \ln((1 - m\epsilon)(1 - p)/(m\epsilon p))$ \( (m = (\sum_{i=1}^{N} a_i - \epsilon)/(2^na_{\text{min}})) \). Thus, we have

$$P(\sum_{i=1}^{N} \delta_i a_i \geq \epsilon) \leq g(t^*)$$

$$= \left( \frac{p}{1 - m\epsilon} \right)^{1-m\epsilon} \left( \frac{1 - p}{m\epsilon} \right)^{m\epsilon}$$

$$= \exp \left( - \left( m\epsilon \log \left( \frac{m\epsilon}{1 - p} \right) + (1 - m\epsilon) \log \left( \frac{1 - m\epsilon}{p} \right) \right) \right)$$

when $0 < m\epsilon < 1 - p$, which establishes the lemma.
[15] Z. Wang, M. J. Lai, Z. Lu, W. Fan, H. Davulcu, and J. Ye, “Orthogonal rank-one matrix pursuit for low rank matrix completion,” SIAM J. Sci. Comput., no. 1, pp. A488–A514, 2015.

[16] Z. Wen, W. Yin, and Y. Zhang, “Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm,” Math. Program. Comput., no. 4, pp. 333–361, 2012.

[17] T. Hastie, R. Mazumder, J. D. Lee, and R. Zadeh, “Matrix completion and low-rank svd via fast alternating least squares,” J. Mach. Learning Research, no. 16, pp. 3367–3402, 2015.

[18] B. Vandereycken, “Low-rank matrix completion by riemannian optimization,” SIAM J. Optimiz., vol. 23, no. 2, pp. 1214–1236, 2013.

[19] P. A. Absil, R. Mahony, and R. Sepulchre, Optimization Algorithms on Matrix Manifolds. Princeton Univ. Press, 2008.

[20] J. Lee, Introduction to Smooth Manifolds, 2nd ed. New York: NY:Springer, 2013, vol. 218.

[21] U. Helmke and J. B. Moore, Optimization and Dynamical Systems. London: Springer-Verlag, 1994.

[22] B. Mishra, G. Meyer, S. Bonnabel, and R. Sepulchre, “Fixed-rank matrix factorizations and riemannian low-rank optimization,” Comput. Stat., no. 3-4, pp. 591–621, 2014.

[23] P. A. Absil and J. Malick, “Projection-like retractions on matrix manifolds,” SIAM J. Optimiz., vol. 22, pp. 135–158, 2012.

[24] Y. H. Dai, “Nonlinear conjugate gradient methods,” Wiley Encyclopedia of Operations Research and Manage. Sci., 2011.

[25] L. Armijo, “Minimization of functions having lipschitz continuous first partial derivatives,” Pacific J. Math., no. 1, 1966.

[26] C. T. Kelley, Iterative methods for optimization. Philadelphia: PA: Frontiers in Applied Mathematics, 1999.

[27] M. R. Hestenes and E. Stiefel, “Methods of conjugate gradients for solving linear systems,” NBS, 1952.

[28] R. Fletcher and C. M. Reeves, “Function minimization by conjugate gradients,” J. Comput., no. 2, pp. 149–154, 1964.

[29] Y. H. Dai and Y. Yuan, “A nonlinear conjugate gradient method with a strong global convergence property,” SIAM J. Optimiz., no. 1, pp. 177–182, 1999.

[30] W. W. Hager and H. Zhang, “A new conjugate gradient method with guaranteed descent and an efficient line search,” SIAM J. Optimiz., no. 1, pp. 170–192, 2005.

[31] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, “Fast and accurate matrix completion via truncated nuclear norm regularization,” IEEE Trans. Pattern Anal. Mach. Intell., no. 9, pp. 2117–2130, Sep. 2013.

[32] P. Wolfe, “Convergence conditions for ascent methods,” SIAM Rev., no. 2, pp. 226–235, 1969.

[33] H. Sato and T. Iwai, “A new, globally convergent riemannian conjugate gradient method,” Optim. J. Math. Program. Oper. Res., no. 4, pp. 1001–1031, 2015.

[34] E. J. Candes and T. Tao, “The power of convex relaxation: Near-optimal matrix completion,” IEEE Trans. Inf. Theory, no. 5, pp. 2053–2080, May 2010.

[35] B. Recht, “A simpler approach to matrix completion,” J. Mach. Learning Research, vol. 12, pp. 3413–3430, Dec. 2011.

[36] W. Cheney, Analysis for applied mathematics. New York: Springer, 2013.

[37] V. Hutson, J. Pym, and M. Cloud, Applications of functional analysis and operator theory. Elsevier, 2005.