Pattern Recognition 111 (2021) 107661

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Bounded manifold completion

Kelum Gajamannage [a,*], Randy Paffenroth [b]

[a] Department of Mathematics and Statistics, Texas A&M University–Corpus Christi, Corpus Christi, TX 78412, USA
[b] Department of Mathematical Sciences, Department of Computer Science, Data Science Program, Worcester Polytechnic Institute, Worcester, MA 01609, USA

ARTICLE INFO

Article history:
Received 12 November 2019
Revised 24 April 2020
Accepted 18 September 2020
Available online 19 September 2020

MSC:
57M50
57N16
58D15,

Keywords:
Manifold
Low-rank matrix completion
Positive semi-definite
Truncated nuclear norm
Gramian

ABSTRACT

Nonlinear dimensionality reduction is an active area of research. In this paper, we present a thematically different approach to detect a low-dimensional manifold that lies within a set of bounds derived from a given point cloud. A matrix representing distances on a low-dimensional manifold is low-rank, and our method is based on current low-rank Matrix Completion (MC) techniques for recovering a partially observed matrix from fully observed entries. MC methods are currently used to solve challenging real-world problems such as image inpainting and recommender systems. Our MC scheme utilizes efficient optimization techniques that employ a nuclear norm convex relaxation as a surrogate for non-convex and discontinuous rank minimization. The method theoretically guarantees on detection of low-dimensional embeddings and is robust to non-uniformity in the sampling of the manifold. We validate the performance of this approach using both a theoretical analysis as well as synthetic and real-world benchmark datasets.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Modern data collection techniques impel researchers to work with nonlinear, high-dimensional, and big data, such as social and sensor networks [1], collective motion [2], and finance [3]. Analyzing such datasets is challenging or sometimes implausible without the presence of Nonlinear Dimensionality Reduction (NDR) techniques. NDR is defined as the process of transforming nonlinear high-dimensional data into a low-dimensional manifold representation [4]. Here, we propose an NDR method that utilizes concepts of low-rank Matrix Completion (MC) to guarantee better performance when the dataset consists of missing data entrees and is contaminated with noise and outliers.

Low-rank MC, a technique for estimating unobserved or partially observed (i.e., bounded) entries of a matrix that satisfies a low-rank assumption has been prominent in many real-world applications such as image inpainting [5] and recommender systems [6]. Since the data sampled from these applications are naturally high-dimensional and large volume, working with all the available data or acquiring all the information for each matrix entry is often not feasible. Thus, MC provides efficient tools to deal with partially observed or unobserved entries of a matrix and recover a low-rank estimate of the fully observed matrix. The proposed low-rank MC algorithm for NDR consists of an additional constraint that assures the recovered matrix represents distances on a manifold. This novel method is theoretically justified based upon recent work in truncated nuclear norm as a relaxation to the original rank [7] and provides superior empirical performance in challenging test problems. There are several MC schemes that use alternative optimization procedures to aforesaid standard nuclear norm relaxation. As examples, MC scheme in Ref. [8] directly minimize the original formulations of trace norm and rank norm, the MC scheme in Ref. [9] uses joint Schatten p-norm and l_p-norm to approximate the rank minimization problem, and the work proposed in Ref. [10] utilizes joint capped trace norm and capped l_1-norm to tightly approximate the rank minimization.

To set the stage for the description of our approach, we observe that classic techniques, such as Principal Component Analysis (PCA) [11] and Multi-Dimensional Scaling (MDS) [11], assume that the data lay close to a low-dimensional *linear* manifold (or subspace). Advanced techniques, such as Isometric Mapping (ISOMAP) [12], assume that each point in a dataset lay on or near a nonlinear manifold where the distances on the manifold provide faithful *geodesic* representations of the distances between the points in the ambient high-dimensional Euclidean space in which the points are embedded. Such methods rely on tight theoretical connections between the inner product matrix of the input data, often called

* Corresponding author.
E-mail addresses: kelum.gajamannage@tamucc.edu (K. Gajamannage), rcpaffenroth@wpi.edu (R. Paffenroth).

https://doi.org/10.1016/j.patcog.2020.107661
0031-3203/© 2020 Elsevier Ltd. All rights reserved.

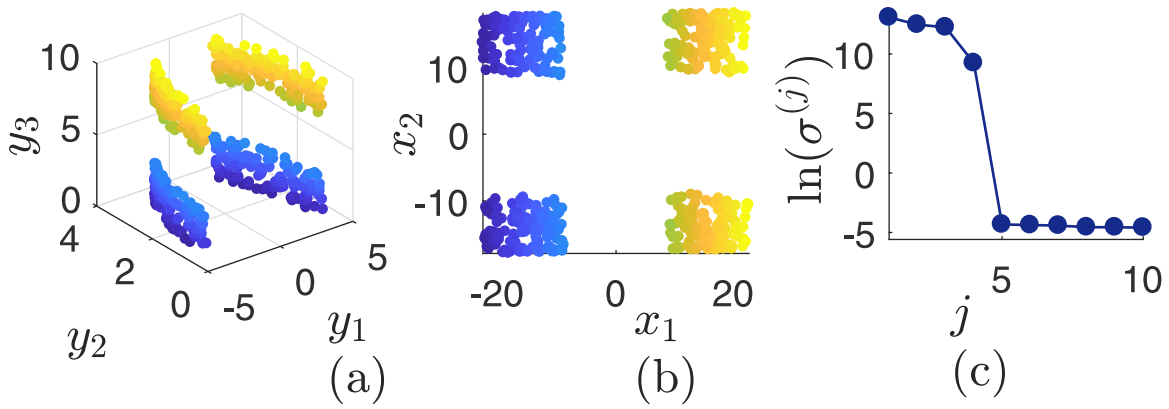


Fig. 1. Recovery of a non-uniformly sampled nonlinear manifold using BMC. (a) A dataset of 500 points sampled *non-uniformly* from a semi-cylinder having a hollow region and four isolated clusters. Here, the color spectrum shows the variation of manifold lengths along the $y^{(3)}$ axis. (b) 2-D embedding of the dataset performed using BMC where the color variation is associated with that of (a). Note, the correct 2-D embedding is accurately recovered even though the nonlinear manifold is sampled non-uniformly. (c) Semilogarithmic plot of the first 10 SVs ($\sigma^{(j)}$'s) of the recovered squared distance matrix. For this data set, the theoretically optimal embedding, by construction, has 4 non-zero singular values. Note that the fifth singular value is smaller than the fourth singular value by more than *10 orders of magnitude* and is merely an artifact of machine precision.

the Gramian matrix, and the matrix of *squared* Euclidean Distances (EDs) between the points in the data-set [11]. One recent method that is somewhat different than the above, but superficially similar in flavor to our proposed technique is Maximum Variance Unfolding (MVU) [13], which attempts to maximize the variance of the data in the presence of distance constraints between some pairs of points. However, as we will explain in detail in the sequel, our proposed method arises from a quite different theoretical perspective than MVU and extends the capabilities of current methods in several important directions. Local Spline Embedding (LSE) [14] minimizes the reconstruction error of an objective function and embeds the data using smoothing splines that map local coordinates of the underlying manifold to global coordinates. Specifically, LSE assumes the existence of a smooth low-dimensional underlying manifold that is quite similar to our proposed method that assumes existence of a low-dimensional manifold. However, LSE assumes that the data is noise free and unaffected by anomalies whereas our method is made to work on noisy data with anomalies.

With the above ideas in mind, the high-level idea of our proposed algorithm is actually quite simple. Given a collection of points in high-dimensional space, a classic MDS algorithm assumes that the EDs between all the points are a faithful representation of "meaningful" distances between the points (an idea that will be made precise in the sequel). ISOMAP relaxes that assumption by only using EDs between neighboring points and using a shortest path *graph distance* as a proxy for a "meaningful" distance between far away points. Our method relaxes this assumption even further. Like ISOMAP, we assume that the EDs between neighboring points are "meaningful", but *differing from ISOMAP, we do not assume we have any approximation for far away distances*. We only assume that there exist *rough upper and lower bounds for the distances between far away points* (perhaps provided by a maximum curvature type argument). In fact, there can be pairs of points (perhaps even many of them) for which nothing is known about their distances on the low-dimensional manifold. Such upper and lower bounds, combined with a constraint that the recovered matrix is Positive Semi-Definite (PSD), allow us to apply a MC scheme for the recovery of the unobserved distances and thereby the parameterization of the underlying low-dimensional manifold. Therefore, we name our method Bounded Manifold Completion (BMC).

Recovery of an arbitrary matrix using a low-rank principle is an ill-posed problem since such an optimization problem is non-convex and discontinuous [7]. Recent research in MC [15,16], suggests that the nuclear norm convex relaxation is an efficient alternative to rank. However, the major limitation of an optimization involving a nuclear norm is that it minimizes the sum of *all the Singular Values (SVs)*, and therefore might not correctly approximate the distances on the manifold. Thus, similar to [7], here we employ a *truncated nuclear norm* (equivalently, the sum of the smallest SVs) to achieve an accurate approximation for the partially observed distance matrix. We formulate BMC as an optimization scheme with multiple constraints where we use an Alternating Direction Method of Multipliers (ADMM) [17], to reformulate the original problem as a general separable collection of convex problems [7] with an additional constraint to enforce that the returned matrix is a distance matrix. Existing schemes utilizing nuclear norm heuristics such as SV thresholding [18], nuclear norm regularized least squares [19], and Robust Principal Component Analysis (RPCA) [15] have shown noteworthy performance. However, as we show in Fig. 1, using a truncated nuclear norm for nonlinear dimensionality reduction allows the detection of a low-dimensional embedding with *an error that is 0, to within machine-precision.*

One of the key elements of our proposed MC scheme is the MC algorithm from [16] which is capable of recovering a matrix also having partially observed data. Specifically, the input distance matrix here is given in terms of two matrices, an upper bound matrix and a lower bound matrix. In particular, the two corresponding values in the bound matrices are the same when the value at that location is fully observed and the two values in the bound matrices are not equal when the value of that location is partially observed. In particular, unobserved data can also be handled in this way by setting the lower and upper bounds to be arbitrarily small and large, respectively. This partially observed data technique is implemented in BMC which allows our method to leverage distances that are fully observed (the Manifold Distances (MDs) between two points are known), partially observed (the MDs between two points can only be bounded), and unobserved (the MDs between the points are not known).

Referring back to Fig. 1, we provide a simple example to illustrate the performance of BMC. We sample a dataset of 500 points $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{500}\}$ with $\boldsymbol{y}_i = (y_i^{(1)}, y_i^{(2)}, y_i^{(3)}) \in \mathbb{R}^3$ from a semi-cylinder such that the sample has some hollow parts of unknown size and four isolated clusters in the corners of the data [Fig. 1(a)]. We construct a *squared* ED matrix $D^e \in \mathbb{R}_{\geq 0}^{500\times 500}$ (real numbers greater than or equal to 0) such that $D^e_{ij} = \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2$. We

set the lower bound matrix (D^l) and the upper bound matrix (D^u) of the recovered *squared* distance matrix to be $D^l_{ij} = 0.1D^e_{ij}$ and $D^u_{ij} = 10D^e_{ij}$. *Note, no actual MDs are required.* A semi-cylinder represents a 2-D manifold; thus, the rank of the squared distance matrix of this dataset, when properly embedded, is four (we present this as a Theorem in Section 2.1). Then, BMC truncates the first four SVs, generates the recovered distance matrix L, and performs a 2-D embedding of the dataset in Fig. 1(b). We observe in the embedding that BMC is capable of projecting the data onto a 2-D plane such that the topology of the data is preserved even though the data is non-uniformly sampled. We compute the natural logarithm of the first 10 SVs of BMC's recovered squared distance matrix in Fig. 1(d). We observe that the first four SVs of L are nonzero while the rest of the SVs are (to within machine precision) zero and, therefore, BMC is capable of recovering the correct low-rank representation of the data.

This paper is structured as follows: In Section 2, we will detail our proposed method in three subsections; first, we present the derivation of our BMC method along with some classic MC methods that inspired BMC; then, the numerical implementation of BMC is presented; finally, we present the ADMM solver of BMC. We validate the performance of BMC in Section 3 using one synthetic dataset, a cylinder having a hollow region, and two real-life datasets, face images and handwritten digits. Here, we also present five classic DR methods that we will contrast/compare with BMC. Finally, we present discussions and conclusions in Section 4.

1.1. Contributions

Our proposed BMC approach makes the following contributions to the literature:

- BMC is a novel nonlinear dimension reduction algorithm which leverages upper and lower bounds on MDs rather than parameters such as the presumed number of neighbors of a data point.
- Even more, BMC is capable of operating when no information is available for many of the MDs.
- BMC provides theoretical guarantees as to the performance of the algorithm, leveraging ideas from [7].
- BMC provides unique capabilities for detecting low-dimensional manifolds even when the manifold is non-uniformly sampled.
- BMC is implemented using a fast ADMM solver allowing large problems to be treated with many data points.
- The idea of using bounded low-rank matrix completion has applications outside of low-dimensional manifold detection where the input data is noisy or otherwise only roughly known.

2. Methods

In this section, first, we present the details of the extent MC algorithms that inspired the BMC scheme. Then, we present the numerical implementation of this MC algorithm followed by deriving an iterative ADMM solver for the BMC algorithm. Finally, we explain the production of the low-dimensional embedding of the data using the recovered distance matrix.

2.1. Matrix completion

In general, low-rank MC schemes take a feature matrix with unobserved entries and recovers the unobserved entries of that matrix by utilizing optimization techniques. Using such MC techniques, we are interested in a low-rank recovery of distances on a manifold.

We start the derivation of BMC by assuming that we are given an arbitrary squared distance matrix of size $n \times n$, $L \in \mathbb{R}^{n\times n}_{\geq 0}$. For such a matrix L we denote by

$$\text{Gram}(L)_{ij} = -\frac{1}{2}\Big[L_{ij} - \mu_i(L_{ij}) - \mu_j(L_{ij}) + \mu_{ij}(L_{ij})\Big], \tag{1}$$

the Gramian of L, where $\mu_i(L_{ij})$, $\mu_j(L_{ij})$, and $\mu_{ij}(L_{ij})$ are the means of the i-th row, the j-th column, and the full matrix L, respectively, [11]. Ref. [11] states that L is a squared distance matrix if Gram(L) is PSD, as in Definition 2.2, leading to our use of the Gramian. We will impose this essential condition of Gram(L) is PSD, denoted by $Gram(L)\succeq 0$, as a constraint to BMC in addition to one general constraint in it, that we will discuss in the sequel.

Definition 2.1. Consider that $Diag(\lambda^{(1)}, \ldots \lambda^{(n)})$ represents a diagonal matrix formed with the vector $(\lambda^{(1)}, \ldots \lambda^{(n)})$ as its diagonal. Let $L \in \mathbb{R}^{n\times n}$ be a diagonalizable matrix (e.g. a distance matrix) then the Eigenvalue Decomposition (EVD) of L is $L = U\Lambda U^{-1}$, where $\Lambda = Diag(\lambda^{(1)}, \ldots \lambda^{(n)})$. Here, for $j = 1, \ldots, n$, $\lambda^{(j)}$ represents the j-th eigenvalue of L.

Definition 2.2. Let $\lambda^{(j)}$ is the j-th Eigenvalue (EV) of Gram(L), then the matrix Gram(L) $\in \mathbb{R}^{n\times n}$ is PSD if $\lambda^{(j)} \geq 0$ for all $j = 1, \ldots, n$. We denote PSD Gram(L) by Gram(L)$\succeq 0$.

As seen in Fig. 1, in contrast to many other DR methods, both, 1) BMC is capable of recovering a single manifold even when the sampling of the manifold is non-uniform, 2) BMC allows many of these distances to be only *partially observed*. BMC accomplishes this task by using Euclidean distances between the given points as similar to many other DR methods. Consider that the MD between two points $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ is denoted by $\mathcal{M}_{ij}$. We then define the squared lower bound matrix, denoted by $D^l = [D^l_{ij}]_{n\times n} \in \mathbb{R}^{n\times n}_{\geq 0}$, and the squared upper bound matrix, denoted by $D^u = [D^u_{ij}]_{n\times n} \in \mathbb{R}^{n\times n}_{\geq 0}$, as

$$D^l_{ij} = \begin{cases} \mathcal{M}^2_{ij}, & \text{if the MD between points } \boldsymbol{y}_i \text{ and } \boldsymbol{y}_j \text{ are observed,} \\ \alpha^l_{ij}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2, & \text{otherwise,} \end{cases} \tag{2}$$

and

$$D^u_{ij} = \begin{cases} \mathcal{M}^2_{ij}, & \text{if the MD between points } \boldsymbol{y}_i \text{ and } \boldsymbol{y}_j \text{are observed,} \\ \alpha^u_{ij}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2, & \text{otherwise,} \end{cases} \tag{3}$$

respectively. We say that $\mathcal{M}_{ij}$ is fully observed if $\mathcal{M}^2_{ij} = D^l_{ij} = D^u_{ij}$. Otherwise, we set suitable values for $\alpha^l = [\alpha^l_{ij}]_{n\times n}$ and $\alpha^u = [\alpha^u_{ij}]_{n\times n}$, where $\alpha^l_{ij} \leq \alpha^u_{ij}$ for all ij, which are user defined nonnegative scaling matrices of the squared lower bound matrix and the squared upper bound matrix, respectively. These parameters allow some MDs to only be bounded rather than known precisely. How might the values for α^u and α^l be set? Formally speaking, the BMC algorithm admits any choice the user might make. However, in practice, and as we will demonstrate in the sequel, choosing α^u and α^l based upon a bound on the *curvature* of the desired manifold leads to advantageous properties.

In particular, the scaling matrices α^l and α^u can be used to define all the three scenarios of MDs of interest to the BMC algorithm, namely *observed, partially observed*, and *unobserved*. Leveraging an observed distance is straight forward according to (2) and (3). If the MD between the points $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ is partially observed, we approximate finite values for α^l_{ij} and α^u_{ij} in (2) and (3) such that those values closely bound the squared MD between the points $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$. If the MD between the points $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ is unobserved, approximating the values for the parameters α^l_{ij} and α^u_{ij} are uncertain. Thus, we set $\alpha^l_{ij} = 0$ and $\alpha^u_{ij} = \infty$ (or some arbitrarily large number).

BMC's objective function minimizes the truncated rank of the recovered matrix L, denoted by $Rank_r(L)$, defined as the number of

nonzero values of the $(r+1)$-th tail of SVs $\{\sigma^{(r+1)}, \ldots, \sigma^{(n)}\}$ of L where $r \geq 0$ as in Definition 2.3. I.e., $Rank_r(L)$ counts the number of non-zero SVs *not counting the r largest SVs*. We are interested in minimizing the truncated rank, as suggested in Hu et al. [7], rather than the rank since it allows the user to omit large SVs from the minimization process. Note, setting $r = 0$ is the same as considering the rank of L without a truncation. Thus, the optimization scheme of BMC that we use to recover a matrix L by completing the incomplete matrix $\mathcal{M}$[2] is

$$\begin{aligned} \underset{L}{\text{minimize}} \quad & Rank_r(L), \\ & D^l \leq L \leq D^u, \\ & \text{Gram}(L) \succeq 0. \end{aligned} \tag{4}$$

Definition 2.3. Consider that $Diag(\sigma^{(1)}, \ldots \sigma^{(\min(n,m))})$ represents a diagonal matrix formed with the vector $(\sigma^{(1)}, \ldots, \sigma^{(\min(n,m))})$ as its diagonal. Let $L \in \mathbb{R}^{n \times m}$ is a matrix, and $U_{m \times m}$ and $V_{n \times n}$ are unitary matrices such that $U^T U = I$ and $V^T V = I$, respectively. Then, Singular Value Decomposition (SVD) of L is $L = U \Sigma V^T$, where $\Sigma_{m \times n} = Diag(\sigma^{(1)}, \ldots \sigma^{(\min(n,m))})$. Here, for $j = 1, \ldots, n$, $\sigma^{(j)}$ represents the j-th singular value of L and $\sigma^{(j)} \geq \sigma^{(j+1)}; \ \forall j$.

The optimization scheme in (4) is inspired by classic theorems relating the rank of distance matrices to dimensions of manifolds. In particular, for an e-D manifold, Theorem 2.1 states that the rank of the squared distance matrix L is $e+1$ or $e+2$ depending on whether the dataset represents a hyper-sphere or a non-hyper-sphere, respectively. Thus, we set $r = e+2$ in (4) for a dataset representing non-hyper-spherical e-D manifold which is the most common case of the datasets.

Theorem 2.1. *Gower [20], Let the dimensionality of a given manifold is e. Then,*

(i) *the rank of the squared distance matrix computed on the manifold is $e+1$ iff the manifold is a hyper-sphere, or*
(ii) *the rank of the squared distance matrix computed on the manifold is $e+2$ iff the manifold is not a hyper-sphere.*

Proof. See Theorem 6 and the associated derivation in [20]. □

Computing $Rank_r(L)$ in (4) is non-convex and NP-hard.[1] However, as stated in Hu et al. [7], the *truncated nuclear norm*, see Definition 2.4, provides a proxy for $Rank_r(L)$ that can be optimized efficiently. Thus, inspired by the approach in Hu et al. [7], we approximate $Rank_r(L)$ in (4) by the r-th truncated nuclear norm of L, denoted by $\|L\|_{*,r}$. With the aforementioned modification, the optimization scheme of BMC is written as

$$\begin{aligned} \underset{L}{\text{minimize}} \quad & \|L\|_{*,r}, \\ & D^l \leq L \leq D^u, \\ & \text{Gram}(L) \succeq 0. \end{aligned} \tag{5}$$

Definition 2.4. For a given matrix $L \in \mathbb{R}^{n \times n}$ and $r \in \mathbb{Z}_{\geq 0}$, the r-th truncated nuclear norm of L, denoted by $\|L\|_{*,r}$, is defined as $\sigma_{r+1} + \ldots + \sigma_n$ where $\sigma^{(j)}$ denotes the j-th SV of L.

2.2. Implementation

Computation of $\|L\|_{*,r}$ in (5) is numerically implemented by subtracting the sum of r-largest SVs from the sum of all the SVs ($\|L\|_*$ is the sum of all the SVs) as stated by Theorem 2.2. Thus, with the aid of Theorem 2.2, the modified optimization scheme is

$$\begin{aligned} \underset{L}{\text{minimize}} \quad & \|L\|_* - \text{tr}(U_r L V_r^T), \\ & D^l \leq L \leq D^u, \\ & \text{Gram}(L) \succeq 0. \end{aligned} \tag{6}$$

[1] A problem is NP-hard if an algorithm for solving it can be translated into an algorithm for solving any non-deterministic polynomial time problem.

Theorem 2.2. *Hu et al. [7]. For a given matrix $L \in \mathbb{R}^{n \times n}$; let, $L = U \Sigma V^T$ provides SVD of L. Let, U_r denote the transpose of the first r columns of the matrix U and V_r denote the transpose of the first r columns of V. Moreover, $tr(U_r L V_r^T)$ denotes the trace of $U_r L V_r^T$, that is, sum of all the diagonal entries of $U_r L V_r^T$. Then,*

$$\|L\|_{*,r} = \|L\|_* - \text{tr}(U_r L V_r^T). \tag{7}$$

Here, $\|L\|_*$ is the nuclear norm of L that is defined as the summation of all the SVs of L.

Proof. See Theorem 3.1 and the following derivation in [7]. □

To optimize (6) we utilize an ADMM [17]. In particular, as inspired by [7] we split the original decision variable L in the objective function of (6) into two decision variables, say L and K, such that $L = K$. Without loss of generality, we replace L by K in the second term of the objective function and rewrite the optimization scheme in (6) as

$$\begin{aligned} \underset{L,K}{\text{minimize}} \quad & \|L\|_* - \text{tr}(U_r K V_r^T) \\ & L = K, \\ & D^l \leq K \leq D^u, \\ & \text{Gram}(L) \succeq 0. \end{aligned} \tag{8}$$

This transformation allows a closed form solution in the case where we fix L and optimize (8) with respect to K, or in the case where we fix K and optimize (8) with respect to L, and leads to an efficient ADMM solver.

BMC's technique of leveraging the inequality constraint $D^l \leq K \leq D^u$ in (8) is influenced by [16], where a new function is introduced to imply this inequality. In BMC, we denote this new function as $E = [E_{ij}]_{n \times n} \in \mathbb{R}^{n \times n}$ and model it such that if a recovered entry K_{ij} satisfies the inequality constraint, then $E_{ij} = 0$, otherwise E_{ij} quantifies the violation of the inequality constraint. Thus,

$$E_{ij}(D^l_{ij}, D^u_{ij}, K_{ij}) = \begin{cases} K_{ij} - D^l_{ij}, & \text{if } D^l_{ij} > K_{ij}, \\ K_{ij} - D^u_{ij}, & \text{if } D^u_{ij} < K_{ij}, \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

which is shown in Fig. 2 and is often known as a shrinkage operator in the literature. Note that, for given D^l_{ij} and D^u_{ij}, E_{ij} is a function of K_{ij}. Finally, to satisfy the inequality constraint in (8), we add the constraint $E = [0]_{n \times n}$ into the optimization scheme. With

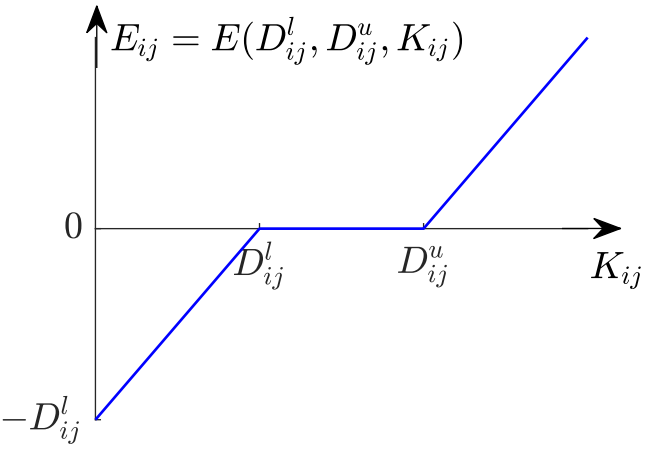


Fig. 2. Function E_{ij} versus K_{ij} for arbitrary upper bound, D^l_{ij}, and lower bound, D^u_{ij}. E_{ij} is zero if K_{ij} is between the bounds, but it increases with the same rate as K_{ij} if it is outside of the bounds.

the aforesaid modifications, our optimization scheme is

$$\begin{aligned} &\underset{L,K}{\text{minimize}} \quad \|L\|_* - \text{tr}(U_r K V_r^T) \\ &L = K, \\ &\text{Gram}(L) \ \succeq 0, \\ &E = [0]_{n\times n}, \end{aligned} \tag{10}$$

where E is defined in (9). In Section 2.3, we solve the optimization scheme in (10) using an ADMM.

2.3. ADMM solver

An ADMM classically uses Augmented Lagrangian Function (ALF) to transform a constrained optimization problem into a related unconstrained optimization problem. Thus, inspired by Boyd et al. [17], first we formulate the ALF of the optimization scheme in (10) and then we follow an ADMM scheme to minimize the ALF. Let the Frobenius norm of a matrix $A = [a_{ij}]_{n\times n} \in \mathbb{R}^{n\times n}$ be defined as

$$\|A\|_F = \sqrt{\sum_{i,j=1}^{n} |a_{ij}|^2}, \tag{11}$$

and the Frobenius inner product of two matrices $A = [a_{ij}]_{n\times n} \in \mathbb{R}^{n\times n}$ and $B = [b_{ij}]_{n\times n} \in \mathbb{R}^{n\times n}$, denoted by $\langle A, B\rangle$, be defined as

$$\langle A, B\rangle = \sum_{1\le i,j\le n} a_{ij} b_{ij}. \tag{12}$$

Let the matrices $\zeta \in \mathbb{R}^{n\times n}$ and $\eta \in \mathbb{R}^{n\times n}$ denote Lagrangian multipliers, then ALF of the optimization scheme in (10) is formed by combining all the constraints into the Lagrangian $\mathcal{L}$ as

$$\begin{aligned} &\mathcal{L}(L, K, \zeta, \eta, \rho^\zeta, \rho^\eta | \text{Gram}(L) \succeq 0) \\ &= \|L\|_* - \text{tr}(U_r K V_r^T) + \langle \zeta, L - K\rangle \\ &\quad + \frac{\rho^\zeta}{2}\|L - K\|_F^2 + \langle \eta, E\rangle + \frac{\rho^\eta}{2}\|E\|_F^2, \end{aligned} \tag{13}$$

where the scalars $\rho^\zeta > 0$ and $\rho^\eta > 0$ are penalty parameters, and E is defined according to (9). Note that, since D^l and D^u in (9) are fixed matrices for a given problem, K is the only independent variable of E that varies over each iteration. Thus, E is not listed as an independent variable of $\mathcal{L}$ in (13). The minimization of $\mathcal{L}$ is equivalent to the optimization of (10).

Small values for the penalty parameters in minimization of $\mathcal{L}$ emphasize the minimization of the objective function while large values for the parameters emphasize the feasibility (satisfying constraints). Our ADMM allows a closed-form approximation to minimization of $\mathcal{L}$ by first minimizing $\mathcal{L}$ with respect to one decision variable or one Lagrangian multiplier while keeping all the other variables and multipliers fixed. Then, ADMM uses the current solution to partially update the other variables and multipliers before their minimization and ADMM generates an iterative set of solutions for updating both the decision variables of the optimum objective function and the Lagrangian multipliers that enforce constraints.

We will minimize $\mathcal{L}$ in (13) with respect to the variables L, K, ζ, and η in Sections 2.3.1–2.3.4. We will provide additional information on numerical implementation of the method in Sections 2.3.5 and 2.5.

2.3.1. Minimization of ALF with respect to L

Let T denotes the total number of iterations. We formulate the minimization of the Lagrangian $\mathcal{L}$ for a general iteration, say m, where $1 \le m \le T$. The solution to the minimization of $\mathcal{L}$ in (13) with respect to L at the m-th iteration is considered to be L_{m+1}. That is,

$$\begin{aligned} L_{m+1} &= \underset{L}{\arg\min}\ \mathcal{L}(L, K_m, \zeta_m, \eta_m, \rho_m^\zeta, \rho_m^\eta | \text{Gram}(L) \succeq 0) \\ &= \underset{L|\text{Gram}(L)\succeq 0}{\arg\min} \left(\|L\|_* + \langle \zeta_m, L - K_m\rangle + \frac{\rho_m^\zeta}{2}\|L - K_m\|_F^2 \right), \end{aligned} \tag{14}$$

where $1 \le m \le T$. Note that, we ignored the terms independent of L in (14). We use both, 1) the relationship $\|A\|_F^2 = \langle A, A\rangle$, where A is an arbitrary matrix, between the Frobenius norm and the inner product, and 2) some identities of the inner product, to simplify (14) as

$$L_{m+1} = \underset{L|\text{Gram}(L)\succeq 0}{\arg\min} \left[\|L\|_* + \frac{\rho_m^\zeta}{2} \left\| L - \left(K_m - \frac{1}{\rho_m^\zeta}\zeta_m \right) \right\|_F^2 \right]. \tag{15}$$

Now, we provide the Definition 2.5 [7,18] of a shrinkage operator that shrinks SVs of a matrix G by the value of $\nu \ge 0$ as we are going to use this operator in Theorem 2.3. In general, ν is a small value comparative to the largest SVs. Such a shrinkage operator eliminates small SVs representing non-prominent features of the dataset such as noise and it retains only the prominent features.

Definition 2.5. Hu et al. [7], Cai et al. [18], Let $G \in \mathbb{R}^{n\times n}$ and SVD of G is $G = U\Sigma V^T$ where $\Sigma = Diag(\{\sigma^{(j)}\}_{1\le j\le n}) = Diag(\sigma^{(1)}, \ldots, \sigma^{(n)})$, see Definition 2.3 for SVD. For $\nu \ge 0$, we define SV shrinkage operator $\mathcal{D}_\nu$ of the matrix G as

$$\begin{aligned} &\mathcal{D}_\nu(G) = U\mathcal{F}(\Sigma)V^T, \text{where} \\ &\mathcal{F}(\Sigma) = \text{Diag}\big(\{\max(\sigma^{(j)} - \nu, 0)\}_{1\le j\le n}\big). \end{aligned} \tag{16}$$

Theorem 2.3. *Hu et al. [7], Cai et al. [18], For any $\nu \ge 0$ and $G \in \mathbb{R}^{n\times n}$,*

$$\mathcal{D}_\nu(G) = \underset{L}{\arg\min} \left[\nu\|L\|_* + \frac{1}{2}\|L - G\|_F^2 \right]. \tag{17}$$

Proof. See the proof of Theorem 2.1 in Cai et al. [18]. □

We use Theorem 2.3 [7,18] to get a closed-form solution for (15). Note that, G and ν in this theorem are equivalent to $K_m - (1/\rho_m^\zeta)\zeta_m$ and $1/\rho_m^\zeta$ in (15), respectively. This theorem provides an alternative implementation using shrinkage of SVs for a vital minimization problem. When $\mathcal{L}$ in (13) convergences to the true solution, the conditions $L = K$ and $\zeta = 0$ are satisfied. Thus, instead of imposing Gram(L)$\succeq$0 in (15), we will impose Gram(G)$\succeq$0 in (17) as an approximation since G and L are equivalent when (13) convergences to the true solution. Theorem 2.3 along with this approximation provides the solution for (15) as

$$\begin{aligned} &L_{m+1} = \{\mathcal{D}_\nu(G_m) | \text{Gram}(G_m) \succeq 0\}; \\ &G_m = K_m - \tfrac{1}{\rho_m^\zeta}\zeta_m \text{ and } \nu = \tfrac{1}{\rho_m^\zeta}. \end{aligned} \tag{18}$$

Now, we use a modified version of the shrinkage operator in Definition 2.5 to compute $\mathcal{D}_\nu(G_m)$ in (18) by imposing Gram(G_m)$\succeq$0. Note that the shrinkage operator in Definition 2.5 shrinks SVs of a matrix G by the value of $\nu \ge 0$ [see $\mathcal{F}(\Sigma)$] and also note that G in (16) is similar to G_m in (18). Thus, similar to the computation of $\mathcal{D}_\nu(G)$ in Definition 2.5, the shrunk SVs of G_m along with its unitary matrices, say U_m and V_m, are used to compute $\mathcal{D}_\nu(G_m)$ in (18). According to Definition 2.2, all the eigenvalues of a PSD matrix are non-negative. Thus, to incorporate the condition Gram(G_m)$\succeq$0 of (18), we perform a modification to this shrinkage definition under two steps: 1) we compute the EVD of Gram(G_m) instead of the SVD of G_m, and 2) we assign zeros for the EVs that are either negative or less than some value $\nu \ge 0$.

Step 1: We transform G_m in (18) to its Gramian using the matrix version of (1) given in (19). Let, diag(G_m) denotes a vector formed by the diagonal of G_m and $\boldsymbol{e} = [1, \ldots, 1]^T$ is a column vector of ones. Note that, $\tilde{G}_m = G_m - \text{Diag}(\text{diag}(G_m))$ represents a matrix formed by replacing the diagonal of G_m with zeros. As stated

in Ref. [20], the matrix version of (18) is

$$\text{Gram}(G_m) = -\frac{1}{2}\Big(I - \frac{1}{n}\boldsymbol{e}\boldsymbol{e}^T\Big)\tilde{G}_m\Big(I - \frac{1}{n}\boldsymbol{e}\boldsymbol{e}^T\Big). \tag{19}$$

We compute EVD of $\text{Gram}(G_m)$ using Definition 2.1 such that

$$\text{Gram}(G_m) = U_m\Lambda_m U_m^{-1};\ \ \Lambda_m = \text{Diag}\Big(\big\{\lambda_m^{(j)}\big\}_{1\le j\le n}\Big), \tag{20}$$

where $\lambda_m^{(j)}$ represents the j-th eigenvalue of $\text{Gram}(G_m)$ at the m-th iteration.

Step 2: Here, first, we modify the shrinkage operator in Definition 2.5 and then we use it to impose $\text{Gram}(G_m)$ is PSD. Second, we transform back PSD $\text{Gram}(G_m)$ to G_m to use in (18). This modified shrinkage operator guarantees that the shrunk EVs of $\text{Gram}(G_m)$ are non-negative which will assure that $\text{Gram}(G_m)$ is PSD. We are also interested in eliminating EVs that are both positive and small since they represent non-prominent features of the dataset such as noise and outliers. To incorporate both of these characteristics, we modify the shrinkage function $\mathcal{F}$ in Definition 2.5 and impose that shrunk EVs are not less than $1/\rho_m^\zeta > 0$. Thus, the new shrinkage function that is denoted by $\mathcal{F}^*$ is

$$\mathcal{F}^*(\Lambda_m) = \text{Diag}\big(\{\lambda_m^{*(j)}\}_{1\le j\le n}\big);\ \text{where}$$
$$\lambda_m^{*(j)} = \begin{cases}\lambda_m^{(j)} - \nu^*, & \text{if } \lambda_m^{(j)} - \nu^* > 0,\\ 0, & \text{otherwise,}\end{cases} \tag{21}$$

where $\nu^* = (1/\rho_m^\zeta) - \min(\lambda_m^{(1)}, \ldots, \lambda_m^{(n)}, 0)$. When all the EVs are non negative, this shrinkage function is the same as that in Definition 2.5 except of the usage of EVs here instead of SVs. But, when at least one EV is negative, this shrinks all the EVs by $\nu^* > 0$. Thus, this shrinkage function eliminates the non-prominent features as well as features related to negative EVs to ensure the recovered matrix is both low-rank and PSD. Similar to Definition 2.5, we use these shrunk EVs of $\text{Gram}(G_m)$ to formulate the shrunk Gramian, denoted by $\mathcal{D}_\nu^*(\text{Gram}(G_m))$, which is PSD. For that, we use the matrix U_m in (20) as

$$\mathcal{D}_\nu^*(\text{Gram}(G_m)) = U_m\mathcal{F}^*(\Lambda_m)U_m^{-1}. \tag{22}$$

Since (18) requires the shrunk version of G_m related to the shrunk $\text{Gram}(G_m)$, we transform $\mathcal{D}_\nu^*(\text{Gram}(G_m))$ back to its corresponding G_m, denoted by G_m^*, using the reverse formula of (19) as

$$G_m^* = \text{Diag}(\mathcal{D}_\nu^*(\text{Gram}(G_m)))\boldsymbol{e}^T + \boldsymbol{e}[\text{Diag}(\mathcal{D}_\nu^*(\text{Gram}(G_m)))]^T - 2\mathcal{D}_\nu^*(\text{Gram}(G_m)), \tag{23}$$

as in Gower [20].

Since distance matrices are PSD, the above two steps force the recovered matrix to both be a distance matrix and retain the EVs representing prominent features of the dataset. We treat G_m^* in (23) as the required $\mathcal{D}_\nu(G_m)$ in (18) as G_m^* is the EVs shrunk version related to PSD shrinkage of $\text{Gram}(G_m)$. Thus, the solution for L in (18) is given as

$$L_{m+1} = G_m^*. \tag{24}$$

2.3.2. Minimization of ALF with respect to K

Here, we minimize $\mathcal{L}$ in (13) with respect to K at a general time step, say m, and obtain a new solution for K, denoted by K_{m+1}. Since we have already minimized this Lagrangian with respect to L in Section 2.3.1, we use the current solution of L, i.e. L_{m+1}, and partially update the Lagrangian. Thus,

$$\begin{aligned}K_{m+1} &= \arg\min_K \mathcal{L}(L_{m+1}, K, \zeta_m, \eta_m, \rho_m^\zeta, \rho_m^\eta | \text{Gram}(L) \succeq 0)\\ &= \arg\min_K\Big[-\text{tr}(U_r K V_r^T) + \langle\zeta_m, L_{m+1} - K\rangle\\ &\quad + \frac{\rho_m^\zeta}{2}\|L_{m+1} - K\|_F^2 + \langle\eta_m, E_m\rangle + \frac{\rho_m^\eta}{2}\|E_m\|_F^2\Big].\end{aligned} \tag{25}$$

Again, we ignored the terms independent of K in (25). Using the relationship $\|A\|_F^2 = \langle A, A\rangle$, where A is an arbitrary matrix, and using some identities of the inner product, (25) is simplified as

$$K_{m+1} = \arg\min_K\Bigg[\frac{\rho_m^\zeta}{2}\Big\|(L_{m+1} - K) + \frac{1}{\rho_m^\zeta}(U_r^T V_r + \zeta_m)\Big\|_F^2 + \frac{\rho_m^\eta}{2}\Big\|E_m + \frac{\eta_m}{\rho_m^\eta}\Big\|_F^2\Bigg]. \tag{26}$$

We differentiate the function inside arg min in (26) with respect to K and solve it for K as

$$K_{m+1} = \frac{1}{\rho_m^\zeta}\Big[\rho_m^\zeta L_{m+1} + \big(U_r^T V_r + \zeta_m\big) - (\rho_m^\eta E_m + \eta_m)E_m'\Big]. \tag{27}$$

Note that, only K_m of the function $E_m = E_m(D^l, D^u, K_m)$ varies through iterations (e.g., as in Fig. 2 and (9)) whereas D^l and D^u are fixed through iterations and only vary from problem to problem. Thus, the derivative of the function $(E_m)_{ij}$ in (9), denoted by $(E_m')_{ij}$, is one if $D_{ij}^l > (K_m)_{ij} > D_{ij}^u$, or zero otherwise.

2.3.3. Minimization of ALF with respect to ζ

Now, we partially update $\mathcal{L}$ in (13) with the current solutions L_{m+1} and K_{m+1}, and minimize $\mathcal{L}$ with respect to ζ to obtain its new solution ζ_{m+1} as

$$\begin{aligned}\zeta_{m+1} &= \arg\min_\zeta \mathcal{L}\Big(K_{m+1}, L_{m+1}, \zeta, \eta_m, \rho_m^\zeta, \rho_m^\eta \mid \text{Gram}(L) \succeq 0\Big),\\ &= \arg\min_\zeta[\langle\zeta, L_{m+1} - K_{m+1}\rangle].\end{aligned} \tag{28}$$

Consider that we use the symbol | for "such that" in (28) and thereafter. Differentiation of (28) with respect to ζ gives

$$\zeta_{m+1} = \zeta_m + \rho_m^\zeta(L_{m+1} - K_{m+1}). \tag{29}$$

2.3.4. Minimization of ALF with respect to η

After partially updating $\mathcal{L}$ in (13) with the current solutions L_{m+1}, K_{m+1}, and ζ_{m+1}, we obtain

$$\begin{aligned}\eta_{m+1} &= \arg\min_\eta \mathcal{L}\Big(K_{m+1}, L_{m+1}, \zeta_{m+1}, \eta, \rho_m^\zeta, \rho_m^\eta \mid \text{Gram}(L) \succeq 0\Big),\\ &= \arg\min_\eta[\langle\eta, E\rangle].\end{aligned} \tag{30}$$

Differentiation of (30) with respect to η gives

$$\eta_{m+1} = \eta_m + \rho_m^\eta E_{m+1}. \tag{31}$$

2.3.5. Multiplier ρ of the parameters ρ_m^ζ and ρ_m^η.

We observed in the testing process that BMC converges fast and is numerically stable when the ADMM solver emphasizes the minimization of the objective function in the early iterations and then it emphasizes on satisfying the constraints in the later iterations. Note that, according to (13), we can control the trade-off between satisfying the objective function and satisfying the constraints by changing the values of the parameters ρ_m^ζ and ρ_m^η. Thus, to achieve those computational advantages, we set variable values for the parameters ρ_m^ζ and ρ_m^η. For that, we introduce a new parameter $\rho \ge 1$ that we use to increase the value of the parameters ρ_m^ζ and ρ_m^η through iterations such that

$$\rho_{m+1}^\zeta = \rho\ \rho_m^\zeta \text{ and } \rho_{m+1}^\eta = \rho\ \rho_m^\eta. \tag{32}$$

2.4. Low-dimensional embedding.

We perform the low-embedding of the data using the recovered distance matrix L. For that, we compute the Gramian of $L = [L_{ij}]_{n\times n}$ using (1) and then perform SVD of Gram(L) using

$$\text{Gram}(L) = U\Sigma V^T, \tag{33}$$

where U and V denote unitary matrices, and Σ denotes a diagonal matrix of SVs. p-D latent variables (collectively called an embedding) of a high-dimensional dataset is given by

$$\hat{X} = I_{p\times n}\Sigma^{1/2}V^T, \tag{34}$$

where $I_{p\times n}$ is a matrix representing the first p-rows of the identity matrix $I_{n\times n}$ [11].

2.5. The time complexity of BMC algorithm.

The details of the BMC implementation can be found in Algorithm 1. The ADMM scheme of BMC requires user input initial guesses $L_1 \in \mathbb{R}^{n\times n}$, $K_1 \in \mathbb{R}^{n\times n}$, $\zeta_1 \in \mathbb{R}^{n\times n}$, $\eta_1 \in \mathbb{R}^{n\times n}$, $\rho_1^{\zeta} \in \mathbb{R}$, and $\rho_1^{\eta} \in \mathbb{R}$. We will explain setting these initial guesses in Section 3.

As stated in early sections, BMC can be considered to be an extension of RPCA that is produced by integrating two new features: 1) the Gramian of the recovered matrix, L, is PSD and, 2) the data is input as upper and lower bounds of the distance matrix. While the first feature is significant when computing the time-complexity, the second feature does not impact the time-complexity of the algorithm. Ref. [21] states that the time-complexity of the RPCA algorithm is dominated by SVD used in the step of computing its low-rank component L of the input matrix. Note that, BMC utilizes EVD (step 1 of Algorithm 1) instead SVD for that step and then defines a new shrinkage function to filter-out negative and small EVs to impose that Gramian L is PSD. The time-complexity associated with EV shrinkage is strictly less than that with the EVD. Thus, the time-complexity of BMC is also dominated by the step that utilizes EVD to compute L as the time-complexities of both SVD and EVD of a square matrix are the same. EVD of a square matrix with n rows has a time-complexity of n^3 [22]; thus, the time-complexity of BMC algorithm is n^3.

Algorithm 1 Bounded manifold completion (BMC). Inputs: lower bound matrix (D^l) and upper bound matrix (D^u) of the squared MD matrix representing the data, truncation parameter (r), multiplier of Lagrangian multipliers (ρ), number of dimensions required in the embedding (p), and maximum number of iteration (T). Outputs: p-D embedding of the data ($\hat{X}$).

Initialize the variables L_1, K_1, ζ_1, η_1, ρ_1^{ζ}, and ρ_1^{η}.
While (T is not met)

1: Solve $L = \underset{L}{arg\,min}\ \ \mathcal{L}(L, K, \zeta, \eta, \rho^{\zeta}, \rho^{\eta}|\ \text{Gram}(L) \succeq 0)$ as explained in Sec. 2.3.1.
2: Update L, then solve $K = \underset{K}{arg\,min}\ \ \mathcal{L}(L, K, \zeta, \eta, \rho^{\zeta}, \rho^{\eta}|\ \text{Gram}(L) \succeq 0)$ as explained in Sec. 2.3.2.
3: Update L and K, then solve $\zeta = \underset{\zeta}{arg\,min}\ \ \mathcal{L}(K, L, \zeta, \eta, \rho^{\zeta}, \rho^{\eta}|\ \text{Gram}(L) \succeq 0)$ as explained in Sec. 2.3.3.
4: Update L, K, and ζ, then solve $\eta = \underset{\eta}{arg\,min}\ \ \mathcal{L}(K, L, \zeta, \eta, \rho^{\zeta}, \rho^{\eta}|\ \text{Gram}(L) \succeq 0)$ as explained in Sec. 2.3.4.
5: Produce the p-D embedding, $\hat{X}$, of the data using the recovered matrix L as explained by (33) and (34).
6: Return $\hat{X}$.

3. Performance analysis

Here, we validate BMC using three representative examples based on three datasets: one synthetic dataset that we sampled from a semi-cylinder and two real-life datasets, face images [23] and handwritten digits [24]. We compare the results of BMC with that of five DR methods from the literature, namely ISOMAP, DM, LE, LLE, and HLLE. We provide a brief description of each method below:

- *Isometric Feature Mapping* (ISOMAP) [12], has one parameter to specify the neighborhood size, say δ. ISOMAP turns an input dataset of points into a graph structure, first, by treating points as nodes, and then, by creating edges between each point and that point's δ nearest points. The weight of an edge is considered to be the ED between the adjacent two points. ISOMAP approximates the MD between any two points in the dataset as the length of the shortest path between them. Finally, this method transforms the MD matrix into an inner product matrix, as in (1), and the SVs of the inner product matrix are used to produce the latent variables.
- The DR method *Diffusion Maps* (DM) [25,26], creates a graph that is similar to that of ISOMAP which has weights of the edges computed using the Gaussian kernel function with a user input variance γ. DM defines Markov random walks on this graph for a t (an input parameter) number of time-steps to compute a proximity measure of the data points. These measures are transformed into diffusion distances and the latent variables are computed using the spectral decomposition on the diffusion distance matrix.
- *Laplacian Eigenmaps* (LE) [27], computes a low-dimensional representation of the data in which the distances between a datapoint and its δ nearest neighbors are minimized. This is done in a weighted manner using a Gaussian kernel function with a user input variance γ such that the closer the neighbor, the more weight that contributes to the cost function.
- *Local Linear Embedding* (LLE) [28], constructs the local properties of the manifold underlying the data by formulating the high-dimensional datapoints as a linear combination of their δ (an input parameter) nearest neighbors with some weights. LLE attempts to retain the weights of the linear combinations in the low-dimensional reconstruction as close as possible to that of high-dimensional representation.
- *Hessian Local Linear Embedding* (HLLE) [26], is a variant of LLE that minimizes the curvature of the high-dimensional manifold under the constraint that the low-dimensional data representation is locally isometric. This is done by an eigenanalysis of a matrix that describes the curviness, measured by means of the local Hessian, of the manifold around δ (an input parameter) nearest neighbors at each data point.

3.1. A semi-cylinder having a hollow region

We analyze the performance of BMC by embedding a dataset sampled from a semi-cylinder, a 2-D manifold, having a hollow region. First, we visually compare the embedding performed by BMC with the embedding performed by five DR methods, and then we analyze the SV spectrums of the distance matrices of the manifolds recovered by BMC and the other five DR methods.

We sample 500 points from a hollowed semi-cylinder using

$$y_1 = 4\cos(\theta), \quad y_2 = 4\sin(\theta), \quad y_3 = z, \tag{35}$$

where $\theta \in \mathbb{U}[[0, \pi/3] \cup [2\pi/3, \pi]]$ and $z \in \mathbb{U}[[0, 3] \cup [7, 10]]$ as in Fig. 3(a). Here, $\mathbb{U}[a, b]$ denotes an uniform probability distribution from a to b.

Since BMC inputs a lower bound (D^l) and a upper bound (D^u) of the squared MD matrix, we compute them using (2) and (3).

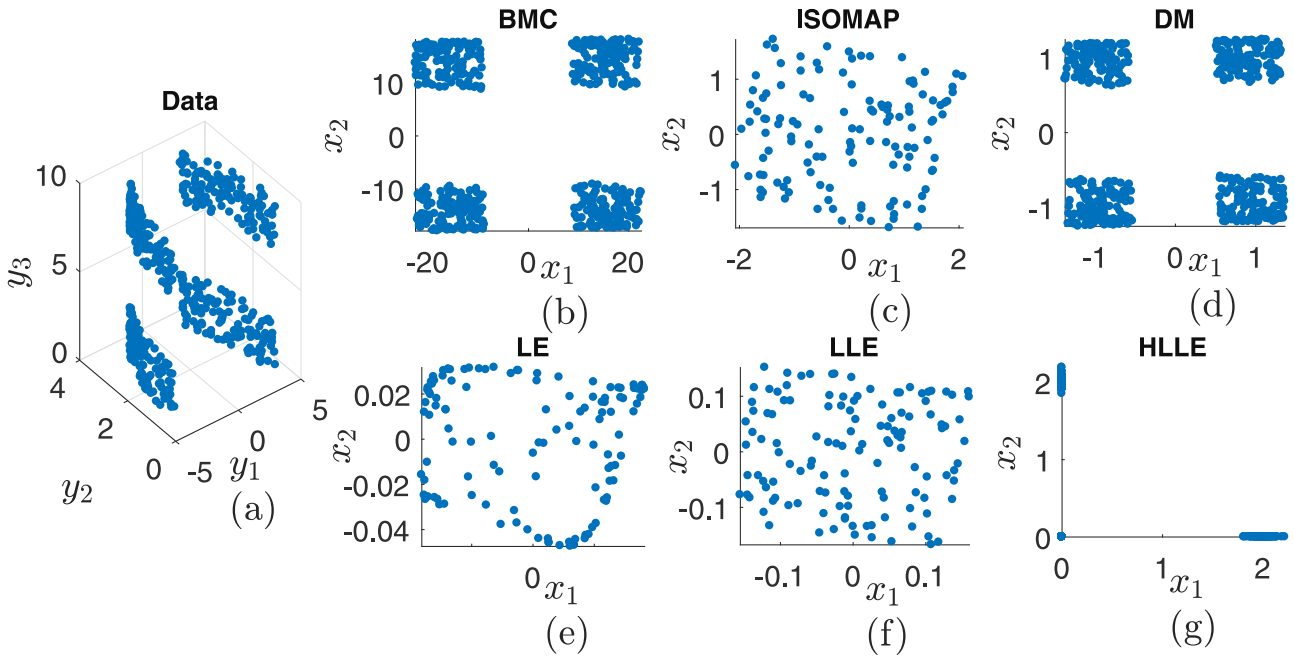


Fig. 3. 2-D embeddings of a cylindrical dataset having a hollow region. (a) A dataset of 500 points is embedded in 2-D using (b) BMC with $r = 4$, $\rho = 1.01$, $\rho_1^{\zeta} = \rho_1^{\eta} = 0.05$, and $\alpha_{ij}^{l} = 0.1$ and $\alpha_{ij}^{u} = 10$ for all ij; (c) ISOMAP with $\delta = 12$; (d) DM with $\gamma = 1$ and $t = 10$; (e) LE with $\delta = 12$ and $\gamma = 1$; (f) LLE with $\delta = 12$; and (g) HLLE with $\delta = 12$.

Even though the squared MDs of points in the same cluster of this dataset could easily be computed, we treat these MDs as unknowns to make this problem little harder. This is numerically implemented by setting unequal values for α_{ij}^{l} and α_{ij}^{u} for each ij as explained in Section 2.1. Moreover, for all ij, we set α_{ij}^{l} to a small value, say 0.1, and set α_{ij}^{u} to a large value, say 10, in (2) and (3). We set wide gaps between upper and lower bounds to make sure that their true MDs lie between them. Narrow gaps between lower and upper bounds expedite the convergence of BMC; however, the construction of valid narrow gaps is to be done in a separate research that we will summarize in Section 4.

We set the initial guesses arbitrarily as $L_1 = K_1 = \zeta_1 = \eta_1 = 0.8D^{l} + 0.2D^{u}$. Consider that any matrix between D^{l} and D^{u} would work as an initial guess. We also set $\rho_1^{\zeta} = \rho_1^{\eta} = 0.05$ and $\rho = 1.01$. As we explained in Section 2.3, setting ρ_1^{ζ} and ρ_1^{η} to a value smaller than one and setting ρ to a value slightly greater than one (this increases the parameters ρ_m^{ζ} and ρ_m^{η} in (13) through iterations) assures numerical stability and fast convergence of BMC. We run BMC with $r = 4$ and $p = 2$ for 500 iterations and obtain 2-D embedding of the dataset. Here, we set $r = 4$, since Theorem 2.1 states that the squared distance matrix of a 2-D manifold has four non-zero SVs. As explained in Section 2.4, we set $p = 2$ in BMC to get 2-D embedding of this dataset that we show in Fig. 3(b).

We also run the DR methods ISOMAP with $\delta = 12$ as in Fig. 3(c), DM with $\gamma = 1$ and $t = 10$ as in Fig. 3(d), LE with $\delta = 12$ and $\gamma = 1$ as in Fig. 3(e), LLE with $\delta = 12$ as in Fig. 3(f), and HLLE with $\delta = 12$ as in Fig. 3(g), over this dataset and compute 2-D embeddings. Consider that the used parameter values above are the default values as proposed by those methods since finding the best set of parameter values of a method to run on a given dataset is challenging and significantly time consuming. We observe that only BMC and DM preserve the geometry of the data. However, we will demonstrate on the real-world datasets that BMC provides substantial benefits over DM in more challenging problems.

Now, we analyze the SV spectrum of BMC and that of the other five DR methods. First, we compute the 10 largest SVs of the recovered matrix L. Computations of the SV spectrums of the other embeddings are performed under two steps. First, we embed the dataset in 10-D space using ISOMAP, DM, LE, LLE, and HLLE. Then, we compute the ED matrix of the embedding of each of the five methods and then compute the SV spectrum for each of the distance matrices. As the ground-truth, we compute the 10 largest SVs of the squared MD matrix of the data (MD takes curvature into consideration that can easily be computed since this data is sampled from a cylinder). For the SVs $\{\sigma^{(j)} | j = 1, \ldots, 10\}$ of the squared distance matrix of an embedding, we define the percentage of normalized SVs as

$$\hat{\sigma}^{(j)} = 100 \frac{\sigma^{(j)}}{\sum_{j=1}^{10} \sigma^{(j)}}. \tag{36}$$

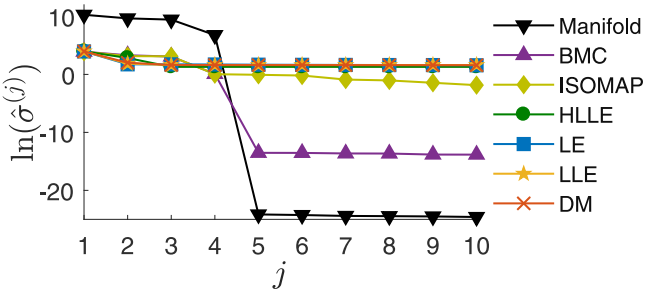


Fig. 4. Semi-log plots of percentage of normalized SVs of the squared MD matrix and that of the squared distance matrices of the embeddings of BMC and the other DR methods.

Fig. 4 shows $\ln(\hat{\sigma}^{(j)})$; $j = 1, \ldots, 10$ of all the DR methods. We observe in this figure that the first four $\hat{\sigma}^{(j)}$'s of BMC are nonzero and the rest of them are approximately zero which is similar to that of the manifold (ground-truth). However, none of the DR methods, except BMC, have zeros after $\hat{\sigma}^{(4)}$. Thus, all the methods, except for BMC, do not recover the solution guaranteed by Theorem 2.1 and that might lead to an embedding which is not representative to the dataset.

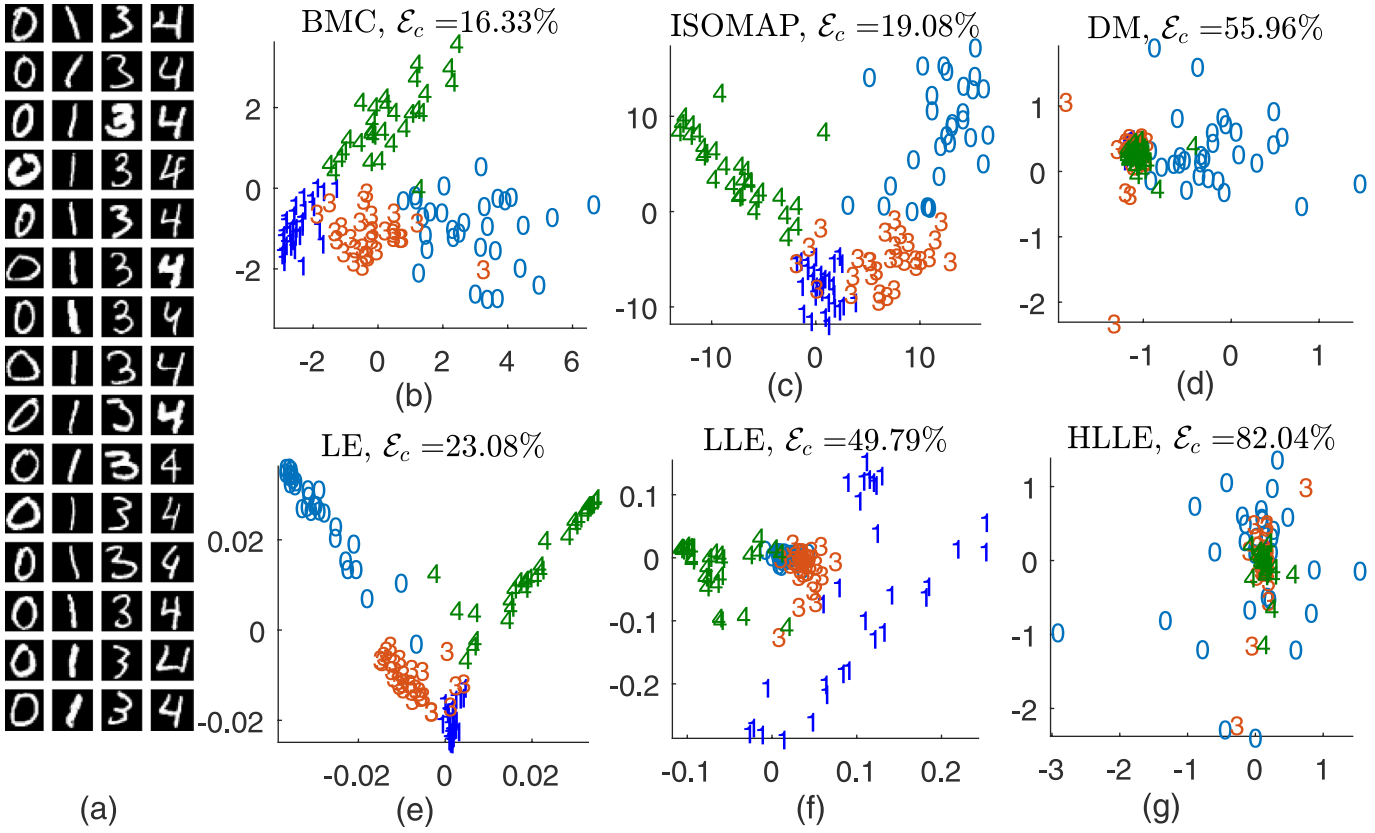


Fig. 5. 2-D embeddings and clustering errors of a image-set of 120 handwritten digits. (a) A sample of 60 images of the image-set. This image-set is embedded in 2-D using (b) BMC with $r = 4$, $\rho = 1.02$, $\rho_1^{\zeta} = \rho_1^{\eta} = 0.1$, and $\alpha_{ij}^{l} = 0.1$ and $\alpha_{ij}^{u} = 10$ for all ij; (c) ISOMAP with $\delta = 12$; (d) DM with $\gamma = 1$ and $t = 10$; (e) LE with $\delta = 12$ and $\gamma = 1$; (f) LLE with $\delta = 12$; and (g) HLLE with $\delta = 12$. The title at each embedding also display its clustering error $\mathcal{E}_c$.

3.2. Handwritten digits

Now, we embed a sample of handwritten digits obtained from the Modified National Institute of Standards and Technology (MNIST) database [24]. It is well known [24] that the low-dimensional representations of the MNIST handwritten digits display rich clustering which we will analyze using BMC and the other DR methods in this section. The MNIST database has 70,000 images of handwritten digits 0, 1, ..., 9 from which we sample (uniform random sampling) 30 images of arbitrary digits of each 0, 1, 3, and 4, (the image-set size is 120 images of digits) as in Fig. 5(a).

Each image in the MNIST database is 28 × 28 dimensions. Since each pixel represents one dimension, each image is a point in a 784-D observable space [4]. We reshape each such image into a row vector and concatenate these 120 row vectors (note that, this image-set has 120 images) vertically to produce the dataset. We are not aware of information about the data such as hollow regions or clusters like those in the semi-cylinder dataset in Section 3.1. Thus, we treat each point of this dataset as a separate cluster, and compute the lower bound (D^l) and the upper bound (D^u) of the MDs using (2) and (3) with $\alpha_{ij}^{l} = 0.1$ and $\alpha_{ij}^{u} = 10$ for all ij. Consider that computing the bounds D^l and D^u, and setting the parameters α_{ij}^{l} and α_{ij}^{u} are done in a similar manner as those in the semi-cylinder example. Here, we set the initial guesses as the same as those in the semi-cylinder example. Since we are not aware of the dimensionality of the manifold underlying the image-set, we are unable to set r based on Theorem 2.1. However, we set $r = 4$ and $p = 2$ for BMC to generate a 2-D embedding which we can easily visualize.[2] We run BMC with $\rho = 1.02$ (setting ρ is based on the same argument as that in the semi-cylinder example) for 800 iterations and produce a 2-D embedding of the dataset that we show in Fig. 5(b).

We run DR methods ISOMAP with $\delta = 12$ as in Fig. 5(c), DM with $\gamma = 1$ and $t = 10$ as in Fig. 5(d), LE with $\delta = 12$ and $\gamma = 1$ as in Fig. 5(e), LLE with $\delta = 12$ as in Fig. 5(f), and HLLE with $\delta = 12$ as in Fig. 5(g), over this dataset and compute 2-D embeddings. Similarly in Section 3.1, the parameter values used above are the default values as proposed by those methods since finding the best set of parameter values of a method to run on a given dataset is challenging and significantly time consuming. We see in Fig. 5 that while BMC, ISOMAP, and LE assure some level of clustering of digits, DM, LLE, and HLLE fail to preserve the clustering. In order to learn the clustering ability in a qualitative manner, we numerically analyze the misclustering of digits using k-Means Clustering (KMC) method available in Lloyd [29]. KMC is a data-partitioning algorithm that assigns given observations, say n, to exactly one of the given clusters, say k in total, defined by centroids where k is chosen before the algorithm starts. We set $k = 4$ in KMC as the dataset consists of four digits and find the clustering for each digit. If a digit is misclustered, then we add one penalty towards the error. The clustering error, denoted by $\mathcal{E}_c$, is defined as the percentage of misclustering in the dataset. We compute the clustering error for each embedding and display them at the titles of Fig. 5(b–g). The clustering errors justify that BMC performs well in terms of preserving the clustering in contrast to that of the other DR methods.

[2] Other choices of r are, of course, possible.

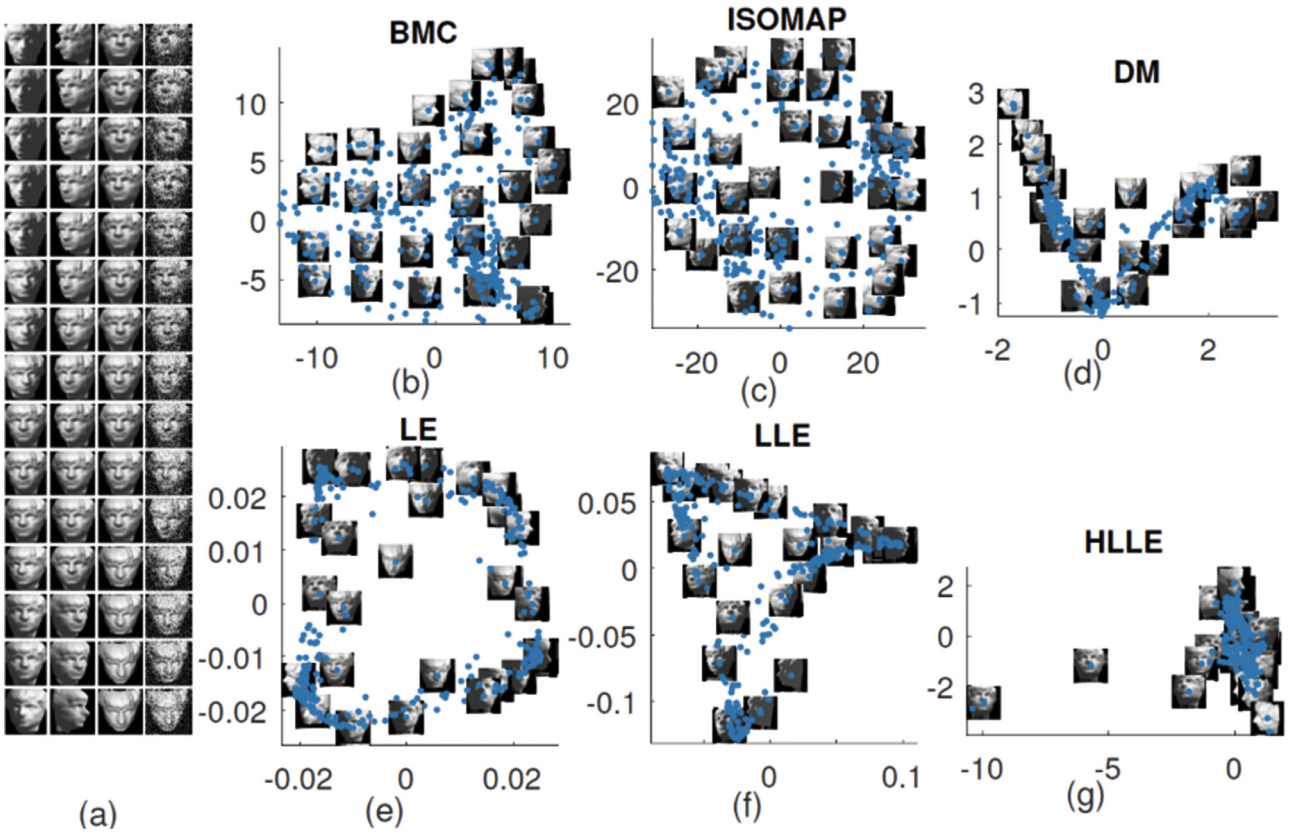


Fig. 6. 2-D embeddings of a image-set of 300 faces. (a) A sample of 60 face images of the image-set whereas columns from left to right indicate left-right light change, left-right pose change, up-down pose change, and a noisy version of images, respectively. This image-set is embedded in 2-D using (b) BMC with $r = 5$, $\rho = 1.04$, $\rho_1^{\xi} = \rho_1^{\eta} = 0.1$, and $\alpha_{ij}^{l} = 0.1$ and $\alpha_{ij}^{u} = 10$ for all ij; (c) ISOMAP with $\delta = 12$; (d) DM with $\gamma = 1$ and $t = 10$; (e) LE with $\delta = 12$ and $\gamma = 1$; (f) LLE with $\delta = 12$; and (g) HLLE with $\delta = 12$. The snapshots show the appearance of some faces on this manifold.

3.3. Face images

In this section, we embed a noisy sample of face images obtained from [23] and learn BMC's ability to preserve the underlying topology. This face image database consists of 600 face images of 64×64 dimensions having three variations: left-right orientation of light change; left-right pose change; and up-down pose change. Note that, each image of this database is a point in a 4096-D space.

We sample (uniform random sampling) 300 images from this database and contaminate them with a Gaussian noise of variance one as in Fig. 6(a). Similar to the handwritten digit example, we reshape each of these image into a row vector and concatenate all such 300 rows vertically to form the dataset. Here, we are not aware of geographical information about this dataset such as hollow regions or clusters like those in the semi-cylinder dataset in Section 3.1. Thus, we treat each point as a separate cluster, and compute D^l and D^u of the MDs using (2) and (3) with $\alpha_{ij}^{l} = .1$ and $\alpha_{ij}^{u} = 10$ for all ij in a similar manner as those in the semi-cylinder example. We also set the initial guesses to the same values as those in the semi-cylinder example. Since each image has three variables (2-D pose and 1-D orientation of light), the underlying manifold is 3-D which is incorporated by setting $r = 5$ according to Theorem 2.1. We run BMC with $\rho = 1.04$ (setting ρ is based on the same argument as that in the semi-cylinder example) and $p = 2$ for 800 iterations to produce a 2-D embedding (see Fig. 6(b)) of the dataset.

We run the DR methods ISOMAP as in Fig. 6(c), DM as in Fig. 6(d), LE as in Fig. 6(e), LLE as in Fig. 6(f), and HLLE as in Fig. 6(g), over this dataset with the same parameter values as those in the handwritten example and compute 2-D embeddings. We see in Fig. 6 that while HLLE performs a weak embedding (most of the images are projected onto a small area), the other methods demonstrate some level of good embedding.

In order to quantitatively analyze the embedding performance, we compute neighborhood preserving error [30], denoted by $\mathcal{E}_{\delta'}$. For that, first, we make an adjacency distance matrix $A^d = [A_{ij}^d]_{n\times n}$ of the dataset such that $A_{ij}^d = 1$, if the j-th point is one of δ' neighbors (an input parameter) of the i-th point; or $A_{ij}^d = 0$ otherwise. Note that, the parameter δ' is similar to δ of ISOMAP in Section 3. Similarly, we make an adjacency distance matrix $A^e = [A_{ij}^e]_{n\times n}$ of an embedding of interest. The neighborhood preserving error is defined as the percentage of normalized cumulative absolute dif-

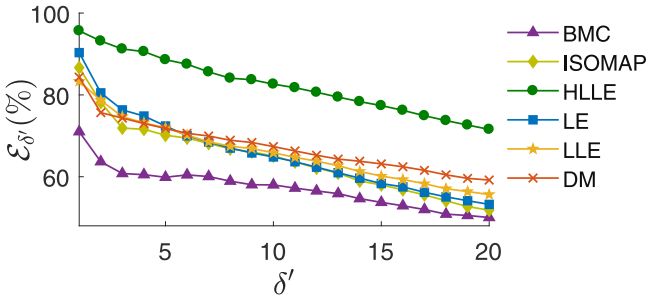


Fig. 7. Neighborhood preserving ability of BMC and the other DR methods versus different neighborhood sizes.

ference between the adjacency distance matrices A^d and A^e,

$$\mathcal{E}_{\delta'} = \frac{100}{\delta'(n-1)} \sum_{i,j=1}^{n} \left| A^d_{ij} - A^e_{ij} \right|, \tag{37}$$

where δ' is the size of the neighborhood of interest [30]. Fig. 7 shows $\mathcal{E}_{\delta'}$ versus $\delta' = 1, \ldots, 20$ for both BMC and the other DR methods. We observe in this figure that BMC preserves the neighborhood than that of the other DR methods for all the neighborhood sizes 1–20. Moreover, the neighborhood preserving ability of BMC is better than that of the other methods when the neighborhood size is small.

4. Conclusion

In this paper, we presented an NDR method using an MC scheme that 1) reveals a distance matrix for a partially observed matrix; 2) employs truncated rank; and 3) inputs the bounds for each entry of the partially observed matrix so that the recovered matrix should satisfy those bounds. This MC scheme inherits truncated nuclear norm regularization for truncated rank from [7] and inherits recovery of the full matrix using bounds from [16]. In order to assure that the recovered matrix is a distance matrix, we imposed that the recovered matrix is PSD using a novel approach as stated in Section 2.3. Fig. 4 justifies that BMC is robust of revealing an accurate low-rank approximation of a partially observed distance matrix that has the same rank as that of the manifold.

We observed in Section 3.2 that BMC attains the least clustering error among all the DR methods of interest. Thus, it justifies that BMC provides superior clustering when it embeds a cluster rich dataset like MNIST. However, the clustering error of BMC was as high as 16.33% since we performed a 2-D embedding without knowing the true dimensionality of the dataset. Thus, before performing this analysis, the true dimensionality of the manifold underlying the data should be revealed. In the future, we will run BMC with $r = 3, \ldots, m$ for some m and then analyze the SV spectrums of the recovered matrices to learn the true manifold dimensionality.

The face image example in Section 3.3 demonstrated BMC's capability of embedding a real-life dataset that is contaminated with a high level of noise. We observed that BMC preserves neighborhoods when it maps data onto an embedding better than the maps of the other DR methods of interest. Topological DR is a process of transforming high-dimensional data onto a low-dimensional embedding such that the data is homeomorphic[3] to the embedding. Since a homeomorphism can be explained as a continuous map of neighborhoods between data and embedding, neighboring points in the dataset should also be neighboring points in the embedding. Thus, the stronger the homeomorphism the lesser the neighborhood preserving error that we have seen from BMC in Fig. 7.

In the examples of this paper, we set wide gaps between lower and upper bounds. Given these bounds, BMC revealed a low-rank recovery distance matrix which showed superior embedding than that of the other DR methods of interest. However, due to wide bounds, BMC might reveal a low-rank distance matrix that is not close enough to the real distance matrix and encounter a considerable error. This is clearly observed by both the clustering error ($\mathcal{E}_c$) in Fig. 5 and the neighborhood preserving error ($\mathcal{E}_{\delta'}$) in Fig. 7. Although BMC is made to recover a low-rank recovery within bounds, we will be able to recover the distance matrix of interest by setting these bounds sufficiently narrow. For that, we plan to use narrow bounds by taking the curvature of the manifold into consideration. Specifically, Menger curvature [31], is used to estimate the point-wise curvatures of the observed points on the manifold, then these curvatures are interpolated over the missing portion (like holes), if any, to estimate the rest of the point-wise curvatures. Based on the curvature, we will compute close lower and upper bounds for the missing part of the distance matrix and then recover it using BMC.

BMC consists of some parameters and requires initial guesses for some variables. Setting the multiplier ρ and the initial guesses L_1, K_1, ζ_1, η_1, ρ_1^{ζ}, and ρ_1^{η} of the variables is determined upon efficient convergence of the BMC method. However, setting of the parameters r and p depends on the embedding that the user is interested. We are planning to perform a sensitivity analysis using Bayesian approach in the future to understand both the high influence parameters and better parameter values.

BMC reveals a PSD low-rank representation of a partially observed distance matrix whereas the recovered matrix satisfies user input bounds of the distances. However, it would be interesting to test if we could have recovered a low-rank representation L of the squared distance matrix D^2 along with a sparse representation S such that $D^2 = L + S$. This modification enables us to extract not only the low-rank potion of the distance matrix but also the sparse portion that will help us to study anomalies in the dataset. Detecting anomalies is helpful in learning bank frauds, structural defects, medical problems, cyber attacks, etc. In this new MC scheme, we could utilize the same truncated nuclear norm along with the PSD constraint, as in the BMC scheme, to ensure the low-rankness of L, and we could employ l_0 norm of S to extract a sparse matrix S from D^2 as presented in Wright et al. [15]. However, l_0 norm of S is non-convex and discontinuous that we could replace by the convex relaxation l_1 norm of S, [15]. This novel MC scheme is extended from BMC under two new modifications to the MC routing given in (8) as, 1) the objective function is $\underset{L,K,S}{\text{minimize}} \quad \|L\|_* - \text{tr}(U_r K V_r^T) + \beta \|S\|_1$, and 2) the second constraint is $D^l \leq K + S \leq D^u$, where the parameter β, $0 \leq \beta \leq 1$, controls the trade-off between L and S.

We believe that this is the first time a PSD low-rank truncated MC technique that inputs the partially observed matrix as bounds has been developed. Thus, we will be able to implement this technique for many applications in networking, financial market analysis, and collective motion. Specifically, we will utilize BMC technique to capture and characterize network topology of social networks such as Facebook, Arxiv, and Enron e-mail. We will also utilize BMC scheme to map network topology of 2-D and 3-D sensor networks from partial virtual coordinates to graph geodesics [1]. In addition, we are interested in DR of collective motion such as schools of fish and flocks of birds. Our recent work, [32,33], concludes that the transitions in collective motion can be represented as switchings of the underlying manifolds which we will study further in an efficient manner by utilizing BMC technique.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors would like to thank NSF XSEDE for allocating 50,000 CPU hours of computational time from the resource Jetstream under the request number DMS180007.

[3] A homeomorphism is defined as an equivalence relation and a one-to-one correspondence between points in two topological spaces τ_1 and τ_2 that is continuous in both directions $\tau_1 \to \tau_2$ and $\tau_2 \to \tau_1$.

References

[1] A.P. Jayasumana, R. Paffenroth, G. Mahindre, S. Ramasamy, K. Gajamannage, Network topology mapping from partial virtual coordinates and graph geodesics, IEEE/ACM Trans. Netw. 27 (6) (2019) 2405–2417, doi:10.1109/TNET.2019.2953921.

[2] K. Gajamannage, R. Paffenroth, Reconstruction of agents' corrupted trajectories of collective motion using low-rank matrix completion, in: Proceedings of the 2019 IEEE International Conference on Big Data, IEEE, 2019, pp. 2826–2834, doi:10.1109/BigData47090.2019.9006381.

[3] N. Bahadur, R. Paffenroth, K. Gajamannage, Dimenslon estlmatlon of equlty markets, in: Proceedings of the 2019 IEEE International Conference on Big Data, IEEE, 2019, pp. 5491–5498, doi:10.1109/BigData47090.2019.9006343.

[4] K. Gajamannage, R. Paffenroth, E.M. Bollt, A nonlinear dimensionality reduction framework using smooth geodesics, Pattern Recognit. 87 (2019) 226–236, doi:10.1016/j.patcog.2018.10.020.

[5] N. Komodakis, G. Tziritas, Image completion using global optimization, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, IEEE, 2006, pp. 442–449. doi:10.1109/CVPR.2006.141

[6] H. Steck, Training and testing of recommender systems on data missing not at random, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, New York, New York, USA, 2010, p. 713. doi:10.1145/1835804.1835895

[7] Y. Hu, D. Zhang, J. Ye, S. Member, X. Li, X. He, S. Member, Fast and accurate matrix completion via truncated nuclear norm regularization, IEEE Trans. Pattern Anal. Mach. Intell. 35 (9) (2013) 2117–2130.

[8] F. Nie, H. Huang, C. Ding, Low-rank matrix recovery via efficient schatten p-norm minimization, in: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI'12, AAAI Press, 2012, pp. 655–661.

[9] F. Nie, H. Wang, H. Huang, C. Ding, Joint schatten p-norm and lp-norm robust matrix completion for missing value recovery, Knowl. Inf. Syst. 42 (3) (2013) 525–544, doi:10.1007/s10115-013-0713-z.

[10] F. Nie, Z. Huo, H. Huang, Joint capped norms minimization for robust matrix recovery, in: IJCAI International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, California, 2017, pp. 2557–2563. doi:10.24963/ijcai.2017/356

[11] J.A. Lee, A. Lendasse, M. Verleysen, Nonlinear projection with curvilinear distances: isomap versus curvilinear distance analysis, Neurocomputing 57 (1–4) (2004) 49–76, doi:10.1016/j.neucom.2004.01.007.

[12] J.B. Tenenbaum, V. De Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323, doi:10.1126/science.290.5500.2319.

[13] K.Q. Weinberger, L.K. Saul, Unsupervised learning of image manifolds by semidefinite programming, Int. J. Comput. Vis. 70 (1) (2006) 77–90.

[14] S. Xiang, F. Nie, C. Zhang, C. Zhang, Nonlinear dimensionality reduction with local spline embedding, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1285–1298, doi:10.1109/TKDE.2008.204.

[15] J. Wright, Y. Peng, Y. Ma, A. Ganesh, S. Rao, Robust principal component analysis: exact recovery of corrupted low-rank matrices, Adv. Neural Inf. Process. Syst. (2009) 2080–2088, doi:10.1002/cpa.20132.

[16] R. Paffenroth, P. Du Toit, R. Nong, L. Scharf, A.P. Jayasumana, V. Bandara, Space-time signal processing for distributed pattern detection in sensor networks, IEEE J. Sel. Top. Signal Process. 7 (1) (2013) 38–49, doi:10.1109/JSTSP.2012.2237381.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2010) 1–122, doi:10.1561/2200000016.

[18] J.F. Cai, E.J. Candes, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim. 20 (4) (2008) 1956–1982, doi:10.1137/080738970.

[19] K.C. Toh, S. Yun, An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems, Pacific J. Optim. 6 (15) (2010) 615–640.

[20] J.C. Gower, Properties of euclidean and non-euclidean distance matrices, Linear Algebra Appl. 67 (1985) 81–97.

[21] H.I. Lee, J.W. Lee, Online update techniques for projection based robust principal component analysis, ICT Express 1 (2) (2015) 59–62, doi:10.1016/j.icte.2015.09.003.

[22] V.Y. Pan, Z.Q. Chen, Complexity of the matrix eigenproblem, in: Conference Proceedings of the Annual ACM Symposium on Theory of Computing, 1999, pp. 507–516, doi:10.1145/301250.301389.

[23] J.B. Tenenbaum, V. De Silva, J.C. Langford, Datasets for nonlinear dimensionality reduction, data for faces, 2016. http://web.mit.edu/cocosci/isomap/datasets.html.

[24] Y. LeCun, C. Cortes, C. Burges, The MNIST database of handwritten digits, 1998. http://yann.lecun.com/exdb/mnist/.

[25] S. Lafon, A.B. Lee, Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization, IEEE Trans. Pattern Anal. Mach. Intell. 28 (9) (2006) 1393–1403, doi:10.1109/TPAMI.2006.184.

[26] R.R. Coifman, S. Lafon, Diffusion maps, Appl. Comput. Harmon. Anal. 21 (1) (2006) 5–30, doi:10.1016/j.acha.2006.04.006.

[27] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: Proceedings of the Conference on Neural Information Processing Systems, volume 14, 2001, pp. 585–591.

[28] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326, doi:10.1126/science.290.5500.2323.

[29] S. Lloyd, Least squares quantization in PCM, IEEE Trans. Inf. Theory 28 (2) (1982) 129–137.

[30] K. Gajamannage, S. Butail, M. Porfiri, E.M. Bollt, Dimensionality reduction of collective motion by principal manifolds, Phys. D 291 (2015) 62–73, doi:10.1016/j.physd.2014.09.009.

[31] M. Melnikov, C. The, C. We, Menger curvature, in: Analytic capacity, rectifiability, Menger curvature and Cauchy integral, Vol. 2, Springer, 2002, pp. 29–54.

[32] K. Gajamannage, S. Butail, M. Porfiri, E.M. Bollt, Identifying manifolds underlying group motion in Vicsek agents, Eur. Phys. J. 224 (17–18) (2015) 3245–3256, doi:10.1140/epjst/e2015-50088-2.

[33] K. Gajamannage, E.M. Bollt, Detecting phase transitions in collective behavior using manifold's curvature, Math. Biosci. Eng. 14 (2) (2017) 437–453, doi:10.3934/mbe.2017027.



Kelum Gajamannage is an Assistant Professor in Mathematics in the Department of Mathematics and Statistics at Texas A&M University - Corpus Christi, USA. He was awarded his Ph.D. in Mathematics from Clarkson University, USA, and was awarded his MS in Applied Statistics and BS (with honor) in Mathematics from the University of Peradeniya, Sri Lanka. He was a Post-Doctoral Scholar for three years in the Department of Mathematical Sciences at Worcester Polytechnic Institute, USA and also worked as a Lecturer for three years in the Department of Science and Technology at Uva Wellassa University, Sri Lanka.



Randy Paffenroth graduated from Boston University with degrees in both mathematics and computer science and he was awarded his Ph.D. in Applied Mathematics from the University of Maryland in June of 1999. After attaining his Ph.D., Dr. Paffenroth spent seven years as a Staff Scientist in Applied and Computational Mathematics at the California Institute of Technology. In 2006, he joined Numerica Corporation where he held the position of Computational Scientist and Program Director. Dr. Paffenroth is currently an Associate Professor of Mathematical Sciences, Computer Science, and Data Science at Worcester Polytechnic Institute where his focus is on the WPI Data Science Program. His current technical interests include machine learning, signal processing, large scale data analytics, compressed sensing, and the interaction between mathematics, computer science and software engineering, with a focus on applications in cyber-defense.