Journal of Scientific Computing (2025) 103:8
https://doi.org/10.1007/s10915-025-02806-3

A Riemannian Optimization Approach to Clustering Problems

Wen Huang[1] · **Meng Wei**[2] · **Kyle A. Gallivan**[2] · **Paul Van Dooren**[3]

Received: 28 January 2024 / Revised: 2 December 2024 / Accepted: 16 January 2025

Abstract
This paper considers the optimization problem

$$\min_{X \in \mathcal{F}_v} f(X) + \lambda \|X\|_1,$$

where f is smooth, $\mathcal{F}_v = \{X \in \mathbb{R}^{n \times q} : X^T X = I_q, v \in \text{span}(X)\}$, and v is a given positive vector. The clustering models including but not limited to the models used by k-means, community detection, and normalized cut can be reformulated as such optimization problems. It is proven that the domain $\mathcal{F}_v$ forms a compact embedded submanifold of $\mathbb{R}^{n \times q}$ and optimization-related tools including a family of computationally efficient retractions and an orthonormal basis of any normal space of $\mathcal{F}_v$ are derived. A Riemannian proximal gradient method that allows an adaptive step size is proposed. The proposed Riemannian proximal gradient method solves its subproblem inexactly and still guarantees its global convergence. Numerical experiments on community detection in networks and normalized cut for image segmentation are used to demonstrate the performance of the proposed method.

Keywords Riemannian optimization · Manifold recognition · Clustering · Inexact proximal gradient method

1 Introduction

Optimization on Riemannian manifolds concerns optimizing a real-valued objective function defined on a Riemannian manifold. It has been of interest due to many important applications, e.g., image segmentation [60] and recognition [69], electrostatics and electronic structure calculation [34, 74], computer vision [40, 62], signal processing [43, 70, 72], numerical linear algebra [64], community detection [73], and machine learning [19].

Many Riemannian optimization methods for smooth objectives have been proposed and systemically analyzed, e.g., Riemannian trust-region Newton method [1, 5], Riemannian Broyden family method including BFGS method and its limited-memory version [39, 42,

✉ Wen Huang
wen.huang@xmu.edu.cn

1 School of Mathematical Sciences, Xiamen University, Xiamen, China

2 Department of Mathematics, Florida State University, Tallahassee, USA

3 Department of Mathematical Engineering, Université catholique de Louvain, Louvain-la-Neuve, Belgium

Published online: 12 February 2025

60], Riemannian trust-region symmetric rank-one update method and its limited-memory version [35, 36, 38], Riemannian Newton method and Riemannian non-linear conjugate gradient method [2, 63, 65, 82]. However, the work of Riemannian optimization for nonsmooth objective functions is relatively limited. Most work considers subgradient-based methods [8, 29, 30, 33, 81]. These methods either focus on geodesically convex objective functions or require solving a quadratic program to high accuracy, which limits the scope of applications. When the objective functions have structure, more efficient optimization algorithms can be developed. Recently, Chen et al. [18] considered nonsmooth problems on the Stiefel manifold that has the splittable form $\min_{X\in\mathcal{M}} f(X) + g(X)$, where the manifold $\mathcal{M}$ is the Stiefel manifold $\mathrm{St}(q, n) = \{X \in \mathbb{R}^{n\times q} : X^T X = I_q\}$, f is smooth, and g is nonsmooth but admits a simple proximal mapping. A proximal gradient method is proposed therein with global convergence and is applicable for solving large-scale problems. The proximal mapping is solved by a semi-smooth Newton algorithm. However, no convergence rate analysis is given. In [41], a different version of the proximal gradient method in [7, Section 10.2] for the splittable function is proposed without restricting the manifold to be the Stiefel manifold. A convergence rate analysis is also given. The proximal mapping in [41] involves an iterative algorithm that uses a semi-smooth Newton algorithm in every iteration and therefore can be less efficient than that in [18].[1] In [44], an inexact Riemannian proximal gradient method that does not require solving the Riemannian proximal mapping of [41] exactly is proposed. It relaxes the requirements of solving the proximal mapping while preserving the convergence properties. However, the theoretical results therein rely on the knowledge of the constant L of the L-retraction-smoothness of the function f, which may not be available. The present paper proposed a different inexact proximal gradient method that avoids this difficulty.

In this paper, we consider the nonsmooth optimization problems over a subset of the Stiefel manifold, i.e.,

$$\min_{X\in\mathcal{F}_v} f(X) + \lambda\|X\|_1, \tag{1.1}$$

where the gradient of f is Lipschitz continuous, $\|X\|_1$ denotes the 1-norm of X, i.e., $\|X\|_1 = \sum_{i=1}^{n}\sum_{j=1}^{q}|X_{ij}|$,

$$\mathcal{F}_v = \{X \in \mathbb{R}^{n\times q} : X^T X = I_q, v \in \mathrm{span}(X)\},$$

$v \in \mathbb{R}^n$ is a vector with all entries being positive, and $\mathrm{span}(X)$ denotes the columns space of X. Note that $q \le n$ since $X^T X = I_q$.

1.1 Applications

Problem (1.1) can be viewed as an alternative formulation of the problem

$$\min_{X\in\mathcal{A}_v^{(n,q)}} f(X) \tag{1.2}$$

under certain circumstances, where $\mathcal{A}_v^{(n,q)} = \{X \in \mathbb{R}^{n\times q} : X^T X = I_q, X \ge 0, v \in \mathrm{span}(X)\}$, $X \ge 0$ denotes that all entries of X are nonnegative, and v is a given positive vector. We next give a few important clustering problems that can be formulated in terms of (1.2) and then discuss the connections between Problem (1.2) and Problem (1.1).

[1] In some cases, the proximal mapping in [41] can be solved efficiently without resorting to the semi-smooth Newton algorithm, see [41, Section 5.2].

Example 1 (k-means model) The k-means algorithm [32, 52] was proposed to cluster vectors in $\mathbb{R}^d$. It is one of the most popular clustering algorithms due to its simplicity and efficiency. Given n points a_i in $\mathbb{R}^d$ and k initial estimations of the means of k groups, the k-means algorithm first assigns $a_i, i = 1, \ldots, n$ to their closest means and creates k groups. Then the k means are updated by computing the means of the new k groups. Such a process is repeated until the algorithm converges. The k-means algorithm can be shown to be a minimization algorithm for solving the optimization problem

$$\min_{X \in \mathcal{A}_{\mathbf{1}_n}^{(n,k)}} ||A - XX^T A||_F^2, \tag{1.3}$$

where $A = [a_1, a_2, \ldots, a_n]^T$, and $\mathbf{1}_n$ denotes the vector with all entries being one, see [14].

Example 2 (Community Detection) Real-world network systems often have a community structure, which is the division of network nodes into groups such that the network connections are denser within the groups and are sparser between the groups, see [57]. These groups are called communities, or modules. A variety of community detection algorithms have been developed in recent years [9, 54, 56, 58, 61, 77]. Among them, modularity optimization approaches have been shown to be highly effective in practical applications, see [27]. In [73], it is proven that in an ideal graph, the global minimizer of $f : \mathcal{A}_{\mathbf{1}_n}^{(n,q)} \to \mathbb{R} : X \mapsto -\text{trace}(X^T MX)$ is a matrix that represents the ground truth, where $M = A - A\mathbf{1}_n\mathbf{1}_n^T A/(\mathbf{1}_n^T A\mathbf{1}_n)$ is the modularity matrix, A is the adjacency matrix of the graph, and q is the number of communities. In the presence of noise, the community detection is still formulated as the optimization problem

$$\min_{X \in \mathcal{A}_{\mathbf{1}_n}} -\text{trace}(X^T MX), \tag{1.4}$$

under the assumption that the noise is not significant enough to change its minimizer.

Example 3 (Other Graph Partitioning Techniques) Besides community detection, other graph partitioning problems including general weighted graph cuts, such as ratio association, ratio cut, normalized cut, and Kernighan-Lin objective function, can be formulated as an optimization problem, as shown in [24],

$$\min_{Y^T DY = I_q, Y^T Y \text{ is diagonal}, \, Y \geq 0, \mathbf{1}_n \in \text{span}(Y)} -\text{trace}(Y^T DKDY), \tag{1.5}$$

where $K \in \mathbb{R}^{n \times n}$ is symmetric and $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix with all entries being positive. These graph partitioning problems have been used in many areas, such as circuit layout [16] and image segmentation [67]. Letting X denote $D^{1/2}Y$, it follows that the constraints in (1.5) yields $X^T X = I_q$, $X^T D^{-1} X$ is diagonal, $X \geq 0$, and $v \in \text{span}(X)$, where $v = \text{diag}(D^{1/2})$. Since $X^T X = I_q$ and $X \geq 0$ imply that each row of X has at most one nonzero entry, the constraint "$X^T D^{-1} X$ is diagonal" is redundant. It follows that $X \in \mathcal{A}_v^{(n,q)}$. Therefore, Problem (1.5) can be reformulated into

$$\min_{X \in \mathcal{A}_v^{(n,q)}} -\text{trace}(X^T D^{1/2} K D^{1/2} X), \tag{1.6}$$

which is in the form of (1.2).

Connections between Problem (1.1) *and Problem* (1.2) Problem (1.2) and Problem (1.1) are connected in such a way that the problems above can be solved with the latter using techniques developed in this paper. Problem (1.2) can be reformulated by replacing the non-negative

constraints $X \geq 0$ with a sparsity constraint $\|X\|_0 = n$ where $\|X\|_0$ corresponds to the total number of nonzero elements in X, which yields

$$\min_{X \in \mathcal{B}_v} f(X), \tag{1.7}$$

where $\mathcal{B}_v = \{X \in \mathbb{R}^{n\times q} : X^T X = I_q, \|X\|_0 = n, v \in \text{span}(X)\}$ and v is a positive vector. For the simplicity of notation, we use $\mathcal{A}_v$ to denote $\mathcal{A}_v^{(n,q)}$ when the superscript values of n and q are known from the context of the discussion. Problems (1.2) and (1.7) are essentially equivalent in the sense that their solutions are connected, as shown in Lemma 1.

Lemma 1 *Consider Problems (1.2) and (1.7) with the objective function f satisfying $f(X) = f(XD_i)$ for any i, where $D_i = \text{diag}(1, \ldots, 1, -1, 1, \ldots, 1)$, i.e., the i-th diagonal entry of D_i is -1. The following two statements hold:*

- *Let X be any matrix in $\mathcal{B}_v$. Then for any column of X, denoted by x_i, the signs of all nonzero entries in x_i are the same.*
- *Define a mapping $\vartheta : \mathbb{R}^{n\times q} \mapsto \mathbb{R}^{n\times q} : X \mapsto \hat{X} = XD_{j_1}D_{j_2}\ldots D_{j_s}$, where j_1, j_2, $\ldots$, j_s are the indices of the columns of X whose nonzero entries are all negative. Then X_* is a global minimizer of Problem (1.7) in the sense that $f(X_*) \leq f(Y), \forall Y \in \mathcal{B}_v$ if and only if $\vartheta(X_*)$ is a global minimizer of Problem (1.2) in the sense that $f(\vartheta(X_*)) \leq f(Z), \forall Z \in \mathcal{A}_v$.*

Proof Since $\|X\|_0 = n$ and $v \in \text{span}(X)$, each row of X has exactly one nonzero entry. Given any column of X, the nonzero entries in that column span the corresponding entries in v. Since all the entries in v have the same sign, the nonzero entries in any column of X must have the same sign. Therefore, the first statement holds.

Suppose that X_* is a global minimizer of (1.7) and that $\vartheta(X_*)$ is not a global minimizer of (1.2). Then there exists $Z \in \mathcal{A}_v$ such that $f(Z) < f(\vartheta(X_*))$. Since $\mathcal{A}_v \subset \mathcal{B}_v$ and $f(\vartheta(X_*)) = f(X_*)$, we have $f(Z) < f(\vartheta(X_*)) = f(X_*)$, which implies that X_* is not a global minimizer of (1.7) either. This is a contradiction. Therefore, if X_* is a global minimizer of (1.7), then $\vartheta(X_*)$ must also be a global minimizer of (1.2).

Suppose that $\vartheta(X_*)$ is a global minimizer of (1.2) and that X_* is not a global minimizer of (1.7). Then there exists $Z \in \mathcal{B}_v$ such that $f(Z) < f(X_*)$. It follows that $f(\vartheta(Z)) = f(Z) < f(X_*) = f(\vartheta(X_*))$. Since $\vartheta(Z)$ and $\vartheta(X_*)$ are in $\mathcal{A}_v$, $f(\vartheta(Z)) < f(\vartheta(X_*))$ implies that $\vartheta(X_*)$ is not a global minimizer of (1.2) either. This is a contradiction. Therefore, if $\vartheta(X_*)$ is a global minimizer of (1.2), then X_* must be a global minimizer of (1.7) thus proving the second statement.

□

Due to the constraints of $\mathcal{B}_v$, the sparsest matrix in $\mathcal{B}_v$ has n nonzero entries. We reformulate Problem (1.7) and use 1-norm penalization to promote the sparsity of X, which yields a continuous optimization problem in (1.1). Using 1-norm to promote sparsity on a manifold is not new and has been widely used for the Stiefel manifold, see e.g., [47, 75]. If the minimizer of (1.1), denoted by X_*, is sufficiently close to $\mathcal{B}_v$, then one can find the closest matrix in $\mathcal{B}_v$ by a mapping $P_{\mathcal{B}_v}(X_*)$, see Lemma 2. If the element in the i-th row j-th column of $P_{\mathcal{B}_v}(X_*)$ is not zero, then the i-th object is in the j-th cluster.

Lemma 2 *Let $v \in \mathbb{R}^n$ be a positive vector, W denote $\text{diag}(v)$, Y denote a matrix in $\mathcal{B}_v$, d_i denote the number of nonzero entries in i-th column of Y, $u_i \in \mathbb{R}^{d_i}$ denote the vector forming by the nonzero entries of the i-th column of Y, and $u \in \mathbb{R}^n$ denote $(u_1^T \ u_2^T \ \ldots \ u_q^T)^T$. If $X_* \in \mathcal{F}_v$ is sufficiently close to Y, then it holds that*

$$Y = P_{\mathcal{B}_v}(X_*),$$

where $P_{\mathcal{B}_v}(X_*) = WP_{\mathcal{B}_{\mathbf{1}_n}}(W^{-1}X_*)$, $P_{\mathcal{B}_v}(X_*) = \left(\frac{b_1}{\|b_1 \odot v\|} \ \ldots \ \frac{b_q}{\|b_q \odot v\|}\right)$, $\odot$ *denotes the Hadamard product,* $b_j \in \mathbb{R}^n$ *for* $j = 1, 2, \ldots, q$, *and*

$$(b_j)_i = \begin{cases} sign((X_*)_{ij}) & \text{if } (X_*)_{ij} \text{ has the largest magnitude in the } i\text{-th row;} \\ 0 & \text{otherwise.} \end{cases}$$

Proof Without loss of generality, assume that Y has the form

$$Y = \operatorname{diag}(u_1, u_2, \ldots, u_q) := \begin{pmatrix} u_1 & 0 & \ldots & 0 \\ 0 & u_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & u_q \end{pmatrix}.$$

Partition the vector v by $v = (v_1^T \ v_2^T \ \ldots v_q^T)^T$, where $v_i \in \mathbb{R}^{d_i}$. It follows that $u_i = s_i v_i / \|v_i\|, i = 1, 2, \ldots, q$, where s_i is either one or negative one. Therefore, $W^{-1}Y = \operatorname{diag}(\frac{s_1}{\|v_1\|}\mathbf{1}_{d_1}, \frac{s_2}{\|v_2\|}\mathbf{1}_{d_2}, \ldots, \frac{s_q}{\|v_q\|}\mathbf{1}_{d_q})$. If $W^{-1}X_*$ is sufficiently close to $W^{-1}Y$ in the sense that the location of the largest magnitude entry of each row does not change, then it holds that

$$W^{-1}Y = P_{\mathcal{B}_{\mathbf{1}_n}}(W^{-1}X_*),$$

which implies $Y = P_{\mathcal{B}_v}(X_*)$. □

In summary, Problems (1.1), (1.2) and (1.7) are closely connected and can be viewed as optimization models for solving the same applications.

1.2 Our Contribution

In this paper, we propose a new optimization model given in (1.1) to characterize clustering problems, including k-means model, community detection, normalized cut, and other graph partitioning techniques. It is proven here that the domain $\mathcal{F}_v$ forms an embedded submanifold of $\mathbb{R}^{n\times q}$. A family of computationally efficient retractions is developed. An orthonormal basis of any normal space of $\mathcal{F}_v$ is given. Such a basis yields a computationally efficient characterization of the normal space and is important in Riemannian proximal gradient methods. A Riemannian proximal gradient method that does not require solving the Riemannian proximal mapping exactly is developed, analyzed, and evaluated. Note that most existing Riemannian proximal gradient methods require solving their subproblems exactly [18, 41, 45]. Compared to the only existing inexact Riemannian proximal gradient method [44], the algorithm proposed in this paper allows adaptive step sizes rather than an unknown fixed step size and still guarantees global convergence. In the numerical experiments, the proposed model and optimization algorithm is shown to have performance superior to existing state-of-the-art algorithms in community detection and normalized cut problems.

1.3 Related Work

To the best of our knowledge, the literature does not consider Problems (1.1), (1.2), or (1.7) for generic functions f. They focus on some special formulations of the objective f. If the function f is given by

$$f(X) = -\operatorname{trace}(X^T M X) \tag{1.8}$$

with a positive semidefinite matrix M, then Problem (1.7) with $v = \mathbf{1}_n$ is a commonly-encountered objective function in the task of clustering. Spectral-type of clustering algorithms and k-means-based algorithms have been proposed [23, 78, 80]. For example, the spectral clustering algorithm in [80] first finds a basis of the eigenspace that corresponds to the q largest eigenvalues and then finds a matrix in $\mathcal{C} := \{X \in \mathbb{R}^{n\times q} : X^T X \text{ is diagonal}, X\mathbf{1}_q = \mathbf{1}_n\}$ that is closest to the eigenspace. The kernel k-means algorithm views the matrix M as a kernel matrix. It follows that the standard k-means algorithm can be used [23].

For a generic objective f, the closest formulation to Problem (1.2) is given in [46, 59]. The formulation therein is Problem (1.2) with the constraint $v \in \text{span}(X)$ being dropped, that is,

$$\min_{X^T X = I_q, X \geq 0} f(X). \tag{1.9}$$

The papers [46, 59] use different approaches to reformulate Problem (1.9). The former [46] develops a penalty method by penalizing the orthogonal constraints in X and keeping the nonnegative and multiple sphere constraints, and the latter [59] keeps the orthonormal constraints and penalizes the nonnegative constraints. If the feasible set $\mathcal{F}_v$ is a manifold, then Problem (1.1) can be optimized by Riemannian proximal gradient methods in [41, 44]. However, it is not considered in [41, 44] if the set $\mathcal{F}_v$ in Problem (1.1) is a manifold.

Related work given in [15] considers reformulating the k-means clustering problem as an optimization problem on the manifold $\mathcal{F}_{\mathbf{1}_n}$. Rather than promoting the sparsity, the authors propose to penalize the negativity of entries in X and reformulate the k-mean clustering problem as

$$\min_{X\in\mathcal{F}_{\mathbf{1}_n}} -\text{trace}(X^T M X) + \lambda \|X_-\|_{\text{F}}^2,$$

where X_- indicates the negative entries of X. Though the paper [15] states that $\mathcal{F}_{\mathbf{1}_n}$ is a manifold, but does not give a rigorous proof. Moreover, we consider a more general case for $\mathcal{F}_v$ with a positive vector v. The retraction proposed in [15] involves a computation of an n-by-n matrix exponential ($O(n^3)$ flops), which is unacceptable for large n. The retraction in the present paper (see Theorem 4) only requires $O(np^2)$ flops and can be used for large-scale problems.[2] In addition, this paper gives an orthonormal basis of the normal space of $\mathcal{F}_v$ which allows a computationally efficient characterization for the normal space. The well-known Riemannian steepest descent method is used in [15] while we propose a more sophisticated and effective inexact Riemannian proximal gradient method for the nonsmooth cost function in (1.1).

A preliminary version of this paper is given in [73], which focuses on the community detection problem [55] and therefore only considers $\mathcal{F}_{\mathbf{1}_n}$. In addition, the paper [73] uses iterations of the existing Riemannian proximal gradient method in [18] on the Stiefel manifold for the problem

$$\min_{x\in\text{St}(q,n)} -\text{trace}(X^T M X) + \lambda \|X\|_1,$$

and projects every iterate x_k onto $\mathcal{F}_{\mathbf{1}_n}$. Such a method, called the Riemannian projected proximal gradient method, is neither guaranteed to generate descent iterates in the sense of the function value nor guaranteed to convergence globally. The geometric structure of the constrained set is not explored either.

[2] Throughout this paper, the computational complexity is measured by flop counts. A flop is a floating point operation [28, Section 1.2.4].

An inexact Riemannian proximal gradient method has been proposed in [44] which is based on the Riemannian proximal gradient method [41]. The version in [44] is insightful from theoretical aspects, that is, theoretical conditions that guarantee local convergence rate are given. However, those results rely on a sufficiently large parameter $\tilde{L}$ in the Riemannian proximal mapping

$$\hat{\eta}_x \approx \arg\min_{\eta \in \mathrm{T}_x\mathcal{M}} \mathrm{trace}(\mathrm{grad}\, f(x)^T \eta) + \frac{\tilde{L}}{2}\|\eta\|^2 + g(R_x(\eta)), \tag{1.10}$$

and the use of a fixed step size, where R_x denotes a retraction on $\mathcal{M}$. In practice, a sufficiently large $\tilde{L}$ is usually unknown, and estimating $\tilde{L}$ requires extra work. The inexact Riemannian proximal gradient method proposed in the present paper avoids this problem, allows adaptive step size, guarantees global convergence, and therefore is preferable from the point of view of computational efficiency and provable robustness.

By following the discussions in [31, Section 6], a related set defined by $\mathfrak{F}_v = \{X \in \mathrm{Gr}(q, n) \mid [v] \subset X\}$ is recognized as a sub-Grassmannian and a subvariety of $\mathrm{Gr}(q, n)$, where $\mathrm{Gr}(q, n)$ denotes the Grassmann manifold, i.e., the set of q-dimensional linear subspaces in $\mathbb{R}^n$, and $[v]$ denotes the linear space spanned by v, i.e., $[v] = \{\alpha v \mid \alpha \in \mathbb{R}\}$. Moreover, $\mathfrak{F}_v$ is also a Schubert variety by [79, Definition 6 and Section 8]. Consider the natural projection $\pi : \mathrm{St}(q, n) \to \mathrm{Gr}(q, n)$ in [11, Example 9.19]. We can see that $\mathfrak{F}_v = \pi(\mathcal{F}_v)$. However, such a connection does not yield a straightforward derivation for showing that $\mathcal{F}_v$ is a manifold.

1.4 Organization

This paper is organized as follows. Section 2 defines the notation and presents some preliminaries. Section 3 proves that the set $\mathcal{F}_v$ is a manifold and the optimization-related geometry tools are also derived. Section 4 gives an inexact Riemannian proximal gradient method and its global convergence analysis. The numerical experiments are shown in Sect. 5. Finally, the conclusion and future work are stated in Sect. 6.

2 Notation and Preliminaries

Unless otherwise indicated, the Riemannian concepts of this paper follow from the standard literature, e.g., [2, 10] and the related notation follows from [2]. A Riemannian manifold $\mathcal{M}$ is a manifold endowed with a Riemannian metric $(\eta_x, \xi_x) \mapsto \langle \eta_x, \xi_x \rangle_x \in \mathbb{R}$, where η_x and ξ_x are tangent vectors in the tangent space of $\mathcal{M}$ at x. The induced norm in the tangent space at x is denoted by $\|\cdot\|_x$. Throughout this paper, unless otherwise indicated, we use the Euclidean metric, i.e., $\langle U, V \rangle = \mathrm{trace}(U^T V)$ and $\|U\| = \sqrt{\langle U, V \rangle} = \|U\|_{\mathrm{F}}$. Therefore, the subscript of $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ can be omitted. The tangent space of the manifold $\mathcal{M}$ at x is denoted by $\mathrm{T}_x\mathcal{M}$, and the tangent bundle, which is the disjoint union of all tangent spaces, is denoted by $\mathrm{T}\mathcal{M}$. An open ball on a tangent space is denoted by $\mathbb{B}(x, r) = \{\xi_x \in \mathrm{T}_x\mathcal{M} \mid \|\xi_x\| < r\}$. The Riemannian gradient of a function $h : \mathcal{M} \to \mathbb{R}$, denoted $\mathrm{grad} h(x)$, is the unique tangent vector satisfying: $\mathrm{D}h(x)[\eta_x] = \langle \eta_x, \mathrm{grad} h(x) \rangle_x, \forall \eta_x \in \mathrm{T}_x\mathcal{M}$, where $\mathrm{D}h(x)[\eta_x]$ denotes the directional derivative of h along the direction η_x.

A retraction, by definition, is a smooth (C^∞) mapping from the tangent bundle to the manifold such that

$$\text{(i) } R(0_x) = x \text{ for all } x \in \mathcal{M}, \text{ and} \tag{2.1}$$

$$\text{(ii)}\ \frac{d}{dt}R(t\eta_x)|_{t=0} = \eta_x \ \text{for all}\ \eta_x \in \mathrm{T}_x\mathcal{M} \tag{2.2}$$

where 0_x denotes the origin of $\mathrm{T}_x\mathcal{M}$. Moreover, R_x denotes the restriction of R to $\mathrm{T}_x\mathcal{M}$. The domain of R does not need to be the entire tangent bundle.

Given a manifold $\overline{\mathcal{M}}$, a subset $\mathcal{M}$ of $\overline{\mathcal{M}}$ can also form a special manifold, called an embedded submanifold of $\overline{\mathcal{M}}$. A characterization of embedded submanifolds is given in [11, Proposition 8.70]. For completeness, we state it as a definition in Definition 1. This characterization is used in Theorem 1 for proving the set $\mathcal{F}_v$ is an embedded submanifold of $\mathrm{St}(q,n)$.

Definition 1 (Embedded submanifolds of $\overline{\mathcal{M}}$) Let $\mathcal{M}$ be a subset of a manifold $\overline{\mathcal{M}}$. We say $\mathcal{M}$ is a (smooth) embedded submanifold of $\overline{\mathcal{M}}$ if either $\mathcal{M}$ is an open subset of $\overline{\mathcal{M}}$ or for a fixed integer $k \geq 1$ and for each $x \in \mathcal{M}$ there exists a neighbourhood $\mathcal{U}$ of x in $\overline{\mathcal{M}}$ and a smooth function $h : \mathcal{U} \to \mathbb{R}^k$ such that

(a) If y is in $\mathcal{U}$, then $h(y) = 0$ if and only if $y \in \mathcal{M}$; and
(b) $\mathrm{rank}\, \mathrm{D}h(x) = k$,

where D denotes the differential operator. Such a function h is called a local defining function for $\mathcal{M}$ at x.

If the manifold $\mathcal{M}$ is a compact embedded submanifold of $\mathbb{R}^n$, then by [12], there exist two positive constants M_1 and M_2 such that

$$\|R_x(\eta_x) - x\| \leq M_1\|\eta_x\| \tag{2.3}$$

$$\|R_x(\eta_x) - x - \eta_x\| \leq M_2\|\eta_x\|^2, \tag{2.4}$$

hold for any $x \in \mathcal{M}$ and $\eta_x \in \mathrm{T}_x\mathcal{M}$. A vector transport $\mathcal{T} : \mathrm{T}\mathcal{M} \oplus \mathrm{T}\mathcal{M} \to \mathrm{T}\mathcal{M} : (\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_x}\xi_x$ associated with a retraction R is a smooth mapping such that, for all (x, η_x) in the domain of R and all $\xi_x \in \mathrm{T}_x\mathcal{M}$, it holds that (i) $\mathcal{T}_{\eta_x}\xi_x \in \mathrm{T}_{R(\eta_x)}\mathcal{M}$, (ii) $\mathcal{T}_{0_x}$ is the identity operator $\mathrm{id} : \mathrm{T}_x\mathcal{M} \to \mathrm{T}_x\mathcal{M}$, and (iii) $\mathcal{T}_{\eta_x}$ is a linear map, where id denotes the identity operator. The vector transport by differential retraction $\mathcal{T}_R$ is defined by $\mathcal{T}_{R_{\eta_x}}\xi_x = \frac{d}{dt}R_x(\eta_x + t\xi_x)|_{t=0}$.

The Stiefel manifold $\mathrm{St}(q,n)$ is defined by $\mathrm{St}(q,n) = \{X \in \mathbb{R}^{n\times q} : X^TX = I_q\}$. The tangent space of $\mathrm{St}(q,n)$ at X is

$$\mathrm{T}_X\mathrm{St}(q,n) = \{X\Omega + X_\perp K : \Omega^T = -\Omega,\ K \in \mathbb{R}^{(n-q)\times q}\}, \tag{2.5}$$

where Ω is skew-symmetric and $X_\perp \in \mathbb{R}^{n\times(n-q)}$ is a matrix with orthonormal columns such that $\begin{pmatrix} X & X_\perp \end{pmatrix}$ is orthonormal.

0_n denotes a vector with length n and all entries zero and $0_{m\times n}$ denotes a m-by-n matrix with all entries zero. I_s denotes the s-by-s identity matrix. The subscript of 0 or I is omitted if its size is clear from the context. Given an n-by-n matrix M, e^M denotes the matrix exponential.

3 Manifold Structure of $\mathcal{F}_v$

Throughout this paper, the notation $\mathcal{F}_v$ always refers to the set with a positive vector v. In this section, we prove that the set $\mathcal{F}_v$ forms an embedded submanifold of $\mathbb{R}^{n\times q}$ and derive optimization-related tools. Theorem 1 show that $\mathcal{F}_v$ is an embedded submanifold of $\mathrm{St}(q,n)$.

Theorem 1 *The set $\mathcal{F}_v$ is an embedded submanifold of* $\mathrm{St}(q,n)$ *with dimension* $\dim(\mathrm{St}(q,n)) - (n-q) = nq - q(q+1)/2 - n + q$. *Furthermore, $\mathcal{F}_v$ is an embedded submanifold of $\mathbb{R}^{n\times q}$ with the same dimension and $\mathcal{F}_v$ is compact.*

Proof We verify that $\mathcal{F}_v$ is an embedded submanifold of $\mathrm{St}(q,n)$ by following Definition 1. For any $X \in \mathcal{F}_v$, let $X_\perp$ be a matrix such that $\begin{pmatrix} X & X_\perp \end{pmatrix}^T \begin{pmatrix} X & X_\perp \end{pmatrix} = I_n$. Therefore, by [26, (2.23)], we have that for any $V \in \mathrm{T}_X\mathrm{St}(q,n)$,

$$\mathrm{Exp}_X(V) = \begin{pmatrix} X & X_\perp \end{pmatrix} e^{\begin{pmatrix} \Omega & -K^T \\ K & 0 \end{pmatrix}} \begin{pmatrix} I_q \\ 0 \end{pmatrix}, \tag{3.1}$$

defines the exponential mapping with respect to the canonical metric, where e^M denotes the matrix exponential of M, $V = X\Omega + X_\perp K$, and the canonical metric is $\langle \eta_X, \xi_X \rangle_X = \mathrm{tr}(\eta_X^T(I_n - \frac{1}{2}XX^T)\xi_X)$ for $\eta_X, \xi_X \in \mathrm{T}_X\mathrm{St}(q,n)$. Since the matrix exponential of a skew-symmetric matrix is orthogonal, $\mathrm{Exp}_X(V)$ in (3.1) is an orthonormal matrix. By [25, Theorem 3.7], there exists a positive constant $\delta > 0$ such that Exp_X is a diffeomorphism in $\mathbb{B}(X,\delta)$. It follows that for any $Y \in \mathrm{Exp}_X(\mathbb{B}(X,\delta))$, the mapping $\mathrm{Exp}_X^{-1}(Y)$ is well-defined and $Y = \mathrm{Exp}_X(\mathrm{Exp}_X^{-1}(Y))$, i.e.,

$$Y = \begin{pmatrix} X & X_\perp \end{pmatrix} e^{\begin{pmatrix} X^T\mathrm{Exp}_X^{-1}(Y) & -\left(X_\perp^T\mathrm{Exp}_X^{-1}(Y)\right)^T \\ X_\perp^T\mathrm{Exp}_X^{-1}(Y) & 0 \end{pmatrix}} \begin{pmatrix} I_q \\ 0 \end{pmatrix}. \tag{3.2}$$

Define a function $\phi : \mathrm{Exp}_X(\mathbb{B}(x,\delta)) \to \mathbb{R}^{n\times(n-q)}$ by

$$\phi(Y) = \begin{pmatrix} X & X_\perp \end{pmatrix} e^{\begin{pmatrix} X^T\mathrm{Exp}_X^{-1}(Y) & -\left(X_\perp^T\mathrm{Exp}_X^{-1}(Y)\right)^T \\ X_\perp^T\mathrm{Exp}_X^{-1}(Y) & 0 \end{pmatrix}} \begin{pmatrix} 0 \\ I_{n-q} \end{pmatrix}.$$

It follows from (3.2) that

$$\phi(Y)^T Y = 0_{(n-q)\times q}. \tag{3.3}$$

Since Exp_X^{-1} is smooth in $\mathrm{Exp}_X(\mathbb{B}(X,\delta))$, ϕ is a smooth function in its domain. Furthermore, it follows from $\phi(X)^T X_\perp = I_{n-q}$ that there exists a constant $\tilde{\delta} > 0$ such that $\phi(Z)^T X_\perp$ is full rank, i.e.,

$$\mathrm{rank}\left(\phi(Z)^T X_\perp\right) = n - q, \tag{3.4}$$

for any $Z \in \mathrm{Exp}_X(\mathbb{B}(X,\tilde{\delta}))$. Let $\hat{\delta} = \min(\delta,\tilde{\delta})$ and $\mathcal{N}_X = \mathrm{Exp}_X(\mathbb{B}(X,\hat{\delta}))$. We now define a function h by $h : \mathcal{N}_X \to \mathbb{R}^{n-q} : Y \mapsto h(Y) = \phi(Y)^T v$. Next, we verify that the function h is a local defining function in the sense that $h^{-1}(0) = \mathcal{N}_X \cap \mathcal{F}_v$ and $\mathrm{D}h(Y) : \mathrm{T}_Y\mathrm{St}(q,n) \to \mathbb{R}^{n-q}$ is surjective for any $Y \in \mathcal{N}_X$.[3]

For any $Z \in h^{-1}(0)$, it holds that $\phi(Z)^T v = 0$ and $Z \in \mathrm{St}(q,n)$. Since $\phi(Z)^T v = 0$ implies $v \in \mathrm{span}(Z)$, we have $Z \in \mathcal{F}_v$, which means $h^{-1}(0) \subseteq \mathcal{N}_X \cap \mathcal{F}_v$. On the other hand, for any $Z \in \mathcal{N}_X \cap \mathcal{F}_v$, it is obvious that $h(Z) = 0$, which means $\mathcal{N}_X \cap \mathcal{F}_v \subseteq h^{-1}(0)$. Overall, the equation $h^{-1}(0) = \mathcal{N}_X \cap \mathcal{F}_v$ holds.

[3] Note that the local definition function only requires $\mathrm{D}h(Y)$ to be full rank at $Y = X$. Here, we prove a stronger result.

Let V denote $\mathrm{Exp}_X^{-1}(Y)$. For any $U \in \mathrm{T}_{\mathrm{Exp}_X(-V)}\mathrm{St}(q,n)$, let $\dot{V} = \mathcal{T}^{-1}_{\mathrm{Exp}_{-V}}(-U)$ and $W = \mathcal{T}_{\mathrm{Exp}_V}\dot{V}$, where $\mathcal{T}_{\mathrm{Exp}}$ denotes the vector transport by differentiating the exponential mapping (3.2). Note that $\dot{V}$ is well-defined since $-V \in \mathbb{B}(X, \hat{\delta})$. We have

$$\begin{aligned}
\mathrm{D}h(Y)[W] &= (\mathrm{D}\phi(Y)[W])^T v \\
&= \left(\begin{pmatrix} X & X_\perp \end{pmatrix} \mathrm{D}e^{\begin{pmatrix} X^T\mathrm{Exp}_X^{-1}(Y) & -\left(X_\perp^T\mathrm{Exp}_X^{-1}(Y)\right)^T \\ X_\perp^T\mathrm{Exp}_X^{-1}(Y) & 0 \end{pmatrix}} [W] \begin{pmatrix} 0 \\ I_{n-q} \end{pmatrix} \right)^T v \\
&= \begin{pmatrix} 0 & I_{n-q} \end{pmatrix} \left\{ \mathrm{D}\left(e^{\begin{pmatrix} -X^T\mathrm{Exp}_X^{-1}(Y) & \left(X_\perp^T\mathrm{Exp}_X^{-1}(Y)\right)^T \\ -X_\perp^T\mathrm{Exp}_X^{-1}(Y) & 0 \end{pmatrix}} \begin{pmatrix} I_q \\ 0 \end{pmatrix} \right)[W] \right\} \alpha,
\end{aligned}$$

where $\alpha = X^T v$, $(e^A)^T = e^{-A}$ for any skew symmetric matrix A, and $X_\perp v = 0$.

Define the functions

$$G(Y) = e^{\begin{pmatrix} -X^T\mathrm{Exp}_X^{-1}(Y) & \left(X_\perp^T\mathrm{Exp}_X^{-1}(Y)\right)^T \\ -X_\perp^T\mathrm{Exp}_X^{-1}(Y) & 0 \end{pmatrix}} \begin{pmatrix} I_p \\ 0 \end{pmatrix}, \text{ and}$$

$$H\begin{pmatrix} \Omega \\ K \end{pmatrix} = \mathrm{Exp}_X(X\Omega + X_\perp K).$$

Since $V, \dot{V} \in \mathrm{T}_X\mathrm{St}(q,n)$, there exist skew symmetric matrices $\Omega_V, \Omega_{\dot{V}}$ and matrices $K_V, K_{\dot{V}}$ such that $V = X\Omega_V + X_\perp K_V$ and $\dot{V} = X\Omega_{\dot{V}} + X_\perp K_{\dot{V}}$. By the chain rule $\mathrm{D}G \circ H\begin{pmatrix} \Omega_V \\ K_V \end{pmatrix}\left[\begin{pmatrix} \Omega_{\dot{V}} \\ K_{\dot{V}} \end{pmatrix}\right] = \mathrm{D}G\left(H\begin{pmatrix} \Omega_V \\ K_V \end{pmatrix}\right)\left[\mathrm{D}H\begin{pmatrix} \Omega_V \\ K_V \end{pmatrix}\left[\begin{pmatrix} \Omega_{\dot{V}} \\ K_{\dot{V}} \end{pmatrix}\right]\right]$ and $W = \mathcal{T}_{\mathrm{Exp}_V}\dot{V}$, we have that

$$\begin{aligned}
&\mathrm{D}e^{\begin{pmatrix} -X^T\mathrm{Exp}_X^{-1}(Y) & \left(X_\perp^T\mathrm{Exp}_X^{-1}(Y)\right)^T \\ -X_\perp^T\mathrm{Exp}_X^{-1}(Y) & 0 \end{pmatrix}} \begin{pmatrix} I_p \\ 0 \end{pmatrix}[\mathcal{T}_{\mathrm{Exp}_V}\dot{V}] \\
&= \mathrm{D}\left(e^{\begin{pmatrix} -\Omega_V & K_V^T \\ -K_V & 0 \end{pmatrix}} \begin{pmatrix} I_p \\ 0 \end{pmatrix} \right)\left[\begin{pmatrix} \Omega_{\dot{V}} \\ K_{\dot{V}} \end{pmatrix}\right].
\end{aligned}$$

It follows that

$$\begin{aligned}
\mathrm{D}h(Y)[W] &= -\begin{pmatrix} 0 & I_{n-q} \end{pmatrix} \mathrm{D}\left(e^{\begin{pmatrix} -\Omega_V & K_V^T \\ -K_V & 0 \end{pmatrix}} \begin{pmatrix} I_p \\ 0 \end{pmatrix} \right)\left[\begin{pmatrix} -\Omega_{\dot{V}} \\ -K_{\dot{V}} \end{pmatrix}\right]\alpha \\
&= \begin{pmatrix} 0 & I_{n-q} \end{pmatrix}\begin{pmatrix} X & X_\perp \end{pmatrix}^T \mathcal{T}_{\mathrm{Exp}_{-V}}(\dot{V}) = \begin{pmatrix} 0 & I_{n-q} \end{pmatrix}\begin{pmatrix} X & X_\perp \end{pmatrix}^T \mathcal{T}_{\mathrm{Exp}_{-V}}(\mathcal{T}^{-1}_{\mathrm{Exp}_{-V}}(-U)) \\
&= -\begin{pmatrix} 0 & I_{n-q} \end{pmatrix}\begin{pmatrix} X & X_\perp \end{pmatrix}^T U\alpha = -X_\perp^T U\alpha. \qquad (3.5)
\end{aligned}$$

Since U can be any tangent vector in $\mathrm{T}_{\mathrm{Exp}_X(-V)}\mathrm{St}(q,n)$ and $X_\perp^T\phi(\mathrm{Exp}_X(-V))$ is full rank by (3.4), the vector $-X_\perp^T U\alpha$ can be any one in $\mathbb{R}^{n-q}$. Therefore, $\mathrm{D}h(Y)$ is surjective and has full rank. Therefore, by [11, Definition 8.70], $\mathcal{F}_v$ is an embedded submanifold of $\mathrm{St}(q,n)$. Furthermore, by [11, Exercise 3.33], $\mathcal{F}_v$ is also an embedded submanifold of $\mathbb{R}^{n\times q}$.

Since $\mathcal{F}_v$ is a subset of $\mathrm{St}(q,n)$, it is a bounded set. Moreover, for any $Z\notin\mathcal{F}_v$, we have that either (i) $Z^TZ\neq I_q$ or (ii) $Z^TZ=I_q$ and $v\notin\mathrm{span}(Z)$. If $Z^TZ\neq I_q$, then consider the function $T(X)=\|X^TX-I_q\|^2$. We have $T(Z)>0$. Since $T(X)$ is a continuous function, there exists a neighborhood of Z such that for all Y in the neighborhood it holds that $T(Y)>0$, which implies $Y\notin\mathcal{F}_v$. If $Z^TZ=I_q$ and $v\notin\mathrm{span}(Z)$, then consider the function $\tilde{T}(X)=\|v-XX^Tv\|^2$. We have that $\tilde{T}(Z)=\|v-ZZ^Tv\|^2=\min_c\|v-Zc\|^2>0$ and $\tilde{T}(X)$ is a continuous function. Therefore, there exists a neighborhood of Z such that for all Y in the neighborhood it holds that $\tilde{T}(Y)>0$, which implies $Y\notin\mathcal{F}_v$. Overall, for any $Z\notin\mathcal{F}_v$, there exists a neighborhood of Z such that the intersection of the neighborhood and $\mathcal{F}_v$ is an empty set. It follows that $\mathcal{F}_v$ is a closed set. Thus, $\mathcal{F}_v$ is compact. □

An intuitive approach for proving $\mathcal{F}_v$ to be an embedded submanifold of $\mathrm{St}(q,n)$ is to consider the function $h:\mathrm{St}(q,n)\to\mathbb{R}^n:X\mapsto v-X^TXv$. We can see that $\mathcal{F}_v$ is the zero level set of h, i.e., $\mathcal{F}_v=h^{-1}(0)$. If one could show that h has a constant rank in a neighborhood of $\mathcal{F}_v$ in $\mathrm{St}(q,n)$, then it follows from [11, Proposition 8.77] that $\mathcal{F}_v$ is an embedded submanifold of $\mathrm{St}(q,n)$. Omitting the details, one can show that

$$(\mathrm{vec}\circ\mathrm{D}h(X)\circ\varpi)(\Omega,K)=-((v^TX)\otimes X_\perp+((v^TX_\perp)\otimes X^T)L)\mathrm{vec}(K),$$

where $\mathrm{vec}(M)$ denotes the vector obtained by stacking the columns of the matrix M, L is the perfect shuffle matrix such that $\mathrm{vec}(A^T)=L\mathrm{vec}(A),\forall A\in\mathbb{R}^{n\times(n-q)}$, $\varpi:\mathrm{skew}_q\times\mathbb{R}^{(n-q)\times q}\to\mathrm{T}_X\mathrm{St}(q,n):(\Omega,K)\mapsto X\Omega+X_\perp K$, and $\mathrm{skew}_q=\{\Omega\in\mathbb{R}^{q\times q}\mid\Omega^T=-\Omega\}$. However, the rank of $(v^TX)\otimes X_\perp+((v^TX_\perp)\otimes X^T)L$ may not have a constant rank $n-q$ in a neighborhood of $\mathcal{F}_v$.

Theorem 2 gives the tangent space at any $X\in\mathcal{F}_v$ and its perpendicular space with respect to the Euclidean metric.

Theorem 2 *Let $\mathcal{F}_v$ be the embedded submanifold of $\mathbb{R}^{n\times q}$. The tangent space of $\mathcal{F}_v$ at X is given by*

$$\mathrm{T}_X\mathcal{F}_v=\{X\Omega+X_\perp K:\Omega^T=-\Omega,K\in\mathbb{R}^{(n-q)\times q},KX^Tv=0\}$$

and the perpendicular space of $\mathrm{T}_X\mathcal{F}_v$ with respect to the Euclidean metric, called the normal space at X, is given by

$$\mathrm{N}_X\mathcal{F}=\{XS+X_\perp uv^TX:S=S^T,u\in\mathbb{R}^{n-q}\}.$$

Proof It follows from [11, Exercise 3.33] that $\mathrm{T}_X\mathcal{F}_v=\mathrm{ker}\mathrm{D}h(X)$. By (3.5), we have that for any $U\in\mathrm{T}_X\mathrm{St}(q,n)$, $\mathrm{D}h(X)[U]=-KX^Tv$, where $U=X\Omega+X_\perp K$ and Ω is any skew symmetric matrix and K is any n-by-$(n-q)$ matrix. Therefore, it holds that $\mathrm{ker}\mathrm{D}h(X)=\{X\Omega+X_\perp K:\Omega^T=-\Omega,K\in\mathbb{R}^{(n-q)\times q},KX^Tv=0\}$.

For any $V\in\mathrm{T}_X\mathcal{F}_v$ and $U\in\mathrm{N}_X\mathcal{F}_v$, it is easy to verify that $\mathrm{trace}(U^TV)=0$. In addition, $\dim(\mathrm{T}_X\mathcal{F}_v)+\dim(\mathrm{N}_X\mathcal{F}_v)=nq-n-q-q(q+1)/2+q(q+1)/2+n+q=nq=\dim(\mathbb{R}^{n\times q})$. Therefore, $\mathrm{N}_X\mathcal{F}_v=(\mathrm{T}_X\mathcal{F}_v)^\perp$ which implies $\mathrm{N}_X\mathcal{F}_v$ is the normal space of $\mathcal{F}_v$ at X. □

Theorem 3 *Given any $Z\in\mathbb{R}^{n\times q}$, the orthogonal projection to $\mathrm{N}_X\mathcal{F}_v$ is given by*

$$P_{\mathrm{N}_X}(Z)=X\frac{X^TZ+Z^TX}{2}+(I-XX^T)Z\hat{\alpha}\hat{\alpha}^T,$$

where $\hat{\alpha} = X^T v/\|X^T v\|$. *The orthogonal projection to* $\mathrm{T}_X\mathcal{F}_v$ *is therefore*

$$P_{\mathrm{T}_X}(Z) = X\frac{X^T Z - Z^T X}{2} + (I - XX^T)Z(I - \hat{\alpha}\hat{\alpha}^T).$$

Proof By observing the formats of $P_{\mathrm{N}_X}(Z)$ and $P_{\mathrm{T}_X}(Z)$, we have $P_{\mathrm{N}_X}(Z) \in \mathrm{N}_X\mathcal{F}_v$ and $P_{\mathrm{T}_X}(Z) \in \mathrm{T}_X\mathcal{F}_V$. Therefore, the result follows from $P_{\mathrm{N}_X}(Z) + P_{\mathrm{T}_X}(Z) = XX^T Z + (I - XX^T)Z = Z$. □

Given $X \in \mathrm{St}(q, n)$, the orthonormal projection from X to $\mathcal{F}_v$ with $v = \mathbf{1}_n$ has been derived in [73]. The orthonormal projection with any $v > 0$ can be derived similarly. We state the result without proof in Lemma 3.

Lemma 3 *For any* $X \in \mathrm{St}(q, n)$ *with* $X^T v \neq 0$, *the global minimizer of the problem* $P_{\mathcal{F}_v}(X) = \arg\min_{Y\in\mathcal{F}_v}\|X - Y\|_F^2$ *is given by* $Y_* = vq_*^T/\|v\|_2 + X(I - q_*q_*^T)$, *where* $q_* = X^T v/\|X^T v\|_2$.

One way to define a retraction on $\mathcal{F}_v$ is by orthogonal projection [3], i.e.,

$$R_X^{\mathrm{proj}}(V) = P_{\mathcal{F}_v}(X + V), \tag{3.6}$$

where $X \in \mathcal{F}$ and $V \in \mathrm{T}_X\mathcal{F}$. However, we do not have a closed-form solution of $P_{\mathcal{F}_v}(X+V)$ in general. A practical family of retractions is given in Theorem 4.

Theorem 4 *For any* $X \in \mathcal{F}_v$, *there exists a positive number* $\delta_X > 0$ *such that the mapping*

$$R_X : \mathbb{B}(x, \delta_X) \to \mathcal{F}_v : V \mapsto R_X(V) = P_{\mathcal{F}_v} \circ \tilde{R}_X \tag{3.7}$$

satisfies the two conditions of the retraction, i.e., e06)*and* (2.2), *where* $\tilde{R}$ *is any retraction on* $\mathrm{St}(q, n)$. *Moreover, if the retraction* $\tilde{R}$ *satisfies* $\mathrm{span}(\tilde{R}_X(V)) = \mathrm{span}(X + V)$, *then the domain of* $\tilde{R}$ *in* (3.7) *is the whole tangent bundle. Such retractions* $\tilde{R}$ *include the retraction by QR decomposition* [2, (4.8)] *and the retraction by polar decomposition* [2, (4.7)].

Proof Since X satisfies $X^T v \neq 0$, $\tilde{R}_X(0_X) = X$, and $\tilde{R}_X$ is smooth, there exists a positive $\delta_X > 0$ such that $P_{\mathcal{F}_v}(\tilde{R}_X(V))$ is well-defined for any $V \in \mathbb{B}(X, \delta_X)$. The smoothness of R follows from the smoothness of $\tilde{R}$ and $P_{\mathcal{F}_v}$. We have $R_X(0_X) = P_{\mathcal{F}_v}(\tilde{R}_X(0_X)) = P_{\mathcal{F}_v}(X) = X$, where the second equality follows from the property of the retraction $\tilde{R}$ and the last equation follows from the definition of the projection $P_{\mathcal{F}_v}$.

In addition, we have

$$\begin{aligned}\frac{d}{dt}R_X(tV)|_{t=0} =& \frac{d}{dt}\left(P_{\mathcal{F}_v} \circ \tilde{R}_X\right)(tV)|_{t=0} = \left(\mathrm{D}P_{\mathcal{F}_v}(\tilde{R}_X(tV))\left[\frac{d}{dt}\tilde{R}_X(tV)\right]\right)|_{t=0}\\ =& \mathrm{D}P_{\mathcal{F}_v}(X)[V] = \mathrm{D}R_X^{\mathrm{proj}}(0_X)[V] = V,\end{aligned}$$

where the second equality follows from the chain rule, the third equality follows from $\tilde{R}_X(0_V) = X$ and $\frac{d}{dt}\tilde{R}_X(tV)|_{t=0} = V$, and the last equality follows from the fact that (3.6) is a retraction.

For the second part of the result, we only need to verify that $(X + V)^T v \neq 0$ for all $V \in \mathrm{T}_X\mathcal{F}_v$. Let $\alpha = X^T v \neq 0$. By the form of the tangent space $\mathrm{T}_X\mathcal{F}_v$ in Theorem 2, we have $\alpha^T(X + V)^T v = \alpha^T(X + X\Omega + X_\perp K)^T v = \alpha^T\alpha + \alpha^T\Omega\alpha = \|\alpha\|_2^2 \neq 0$, which implies $(X + V)^T v \neq 0$. □

By Theorem 4, two retractions of $\mathcal{F}_v$ based on QR decomposition and polar decomposition are respectively given by

$$R_X^{\mathrm{qf}}(V) = vq_*^T/\|v\|_2 + \mathrm{qf}(X+V)(I - q_*q_*^T) \tag{3.8}$$

where $q_* = \mathrm{qf}(X+V)^T v/\|\mathrm{qf}(X+V)^T v\|_2$ and $\mathrm{qf}(X+V)$ denotes the Q factor of the QR decomposition of $X+V$ that, moreover, has positive diagonal entries in the R factor; and

$$R_X^{\mathrm{polar}}(V) = vq_*^T/\|v\|_2 + (X+V)(I+V^TV)^{-1/2}(I - q_*q_*^T), \tag{3.9}$$

where $q_* = (I+V^TV)^{-1/2}(X+V)^T v/\|(I+V^TV)^{-1/2}(X+V)^T v\|_2$. Since the dominant part of the computations in (3.8) and (3.9) are respectively the QR decomposition and polar decomposition, their computations both take $O(np^2)$ flops. The retraction proposed in [15, (14)] is computationally more expensive. Specifically, the retraction in [15] is given by

$$R_X(V) = \exp(B)\exp(A')X, \tag{3.10}$$

where $A = X^TV$, $A' = XAX^T$, and $B = VX^T - XV^T - 2A'$. The computation of (3.10) requires an evaluation of an exponential of an n-by-n matrix B and therefore can be computationally unacceptable when n is large.

The proposed proximal gradient method also relies on an orthonormal basis of the normal space of $\mathcal{F}_v$, which is given in Lemma 4.

Lemma 4 *The set*

$$\begin{aligned}\mathcal{B}_X =& \{X\mathbf{e}_i\mathbf{e}_i^T : i = 1,\cdots,q\} \cup \{\frac{1}{\sqrt{2}}X(\mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T) : i = 1,\cdots,q,\ j = i+1,\cdots,q\} \\ &\cup \{X_\perp\tilde{\mathbf{e}}_i\tilde{v}^TX, i = 1,\cdots,n-q\},\end{aligned} \tag{3.11}$$

defines an orthonormal basis of $N_X\mathcal{F}$ with respect to the Euclidean metric, where $x \in \mathcal{F}_v$, $(\mathbf{e}_1,\cdots,\mathbf{e}_q)$ is the canonical basis of $\mathbb{R}^q$, $(\tilde{\mathbf{e}}_1,\cdots,\tilde{\mathbf{e}}_{n-q})$ is the canonical basis of $\mathbb{R}^{n-q}$ and $\tilde{v} = v/\|v\|$.

Proof Let $T_{ij} = \frac{1}{\sqrt{2}}X(\mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T)$ if $i \neq j$, $T_{ii} = X\mathbf{e}_i\mathbf{e}_i^T$ and $\tilde{T}_i = X_\perp\tilde{\mathbf{e}}_i\tilde{v}^TX$. It is easy to verify that if $i_1 = i_2$ and $j_1 = j_2$, then $\mathrm{trace}(T_{i_1j_1}^T T_{i_2j_2}) = 1$, otherwise $\mathrm{trace}(T_{i_1j_1}^T T_{i_2j_2}) = 0$ for all i_1, j_1, i_2, j_2; and that $\mathrm{trace}(T_{ij}^T\tilde{T}_k) = 0$, for all i, j, k. We also have

$$\begin{aligned}\mathrm{trace}(\tilde{T}_i^T\tilde{T}_j) =& \mathrm{trace}\left(X^T\tilde{v}\tilde{\mathbf{e}}_i^TX_\perp^TX_\perp\tilde{\mathbf{e}}_j\tilde{v}^TX\right) = \mathrm{trace}\left(\tilde{\mathbf{e}}_i^T\tilde{\mathbf{e}}_j\tilde{v}^TXX^T\tilde{v}\right) \\ =& \mathrm{trace}\left(\tilde{\mathbf{e}}_i^T\tilde{\mathbf{e}}_j\tilde{v}^T\tilde{v}\right) = \tilde{\mathbf{e}}_i^T\tilde{\mathbf{e}}_j,\end{aligned}$$

where the second equality follows from $X_\perp^TX_\perp = I$ and the third equality follows from $\tilde{v} \in \mathrm{span}(X)$. □

Let $V_i, i = 1,\ldots q(q+1)/2+n-q$ denote the entries in the basis $\mathcal{B}_X$. Define a function by

$$B_X : \mathbb{R}^{q(q+1)/2+n-q} \to \mathbb{R}^{n\times q} : u \to B_Xu = \sum_{i=1}^{q(q+1)/2+n-q} u_iV_i \in N_x\mathcal{F}_v \tag{3.12}$$

and another function by

$$B_X^T : \mathbb{R}^{n\times p} \to \mathbb{R}^{q(q+1)/2+n-q} : V \to u, \tag{3.13}$$

where $u_i = \mathrm{trace}(V^T V_i)$. These two functions are used in the proximal gradient method given in Sect. 4.

Though the matrix $X_\perp$ is used in the orthonormal basis (3.11), one does not need to construct such a matrix and only needs to use two mappings $\alpha_X : \mathbb{R}^{n\times q} \to \mathbb{R}^{n\times q} : A \mapsto (X \ X_\perp)^T A$ and $\beta_X : \mathbb{R}^{n\times q} \to \mathbb{R}^{n\times q} : A \mapsto (X \ X_\perp)A$. These two mappings can be computed efficiently ($O(np^2)$), see details in [37, Algorithms 4 and 5]. It follows that the mappings B_X^T and B_X can be computed by $O(np^2)$ flops and the detailed implementations are stated in Algorithms 1 and 2.

Algorithm 1 Compute $B_X^T : \mathbb{R}^{n\times p} \to \mathbb{R}^{q(q+1)/2+n-q} : V \mapsto B_X^T(V)$

Input: $X \in \mathcal{F}_v$, $V \in \mathrm{N}_X\mathcal{F}_v$; a positive vector $v \in \mathbb{R}^n$; a function $\alpha_X : \mathbb{R}^{n\times p} \to \mathbb{R}^{n\times p} : A \mapsto (X \ X_\perp)^T A$.

1: $\begin{pmatrix} S \\ K \end{pmatrix} = \alpha_X(V)$, where $S \in \mathbb{R}^{p\times p}$ and $K \in \mathbb{R}^{(n-q)\times q}$;
2: $k = 1$;
3: **for** $j = 1, \ldots, p$ **do** $u(k) = S_{ii}$ and where S_{ii} denotes the i-th row, i-th column entry of the matrix S, and set $k \leftarrow k + 1$;
4: **end for**
5: **for** $j = 2, \ldots, p, i = 1, \ldots j-1$ **do** $u(k) = \sqrt{2}S_{ij}$, $u(k) = \sqrt{2}S_{ji}$, and $k \leftarrow k + 1$;
6: **end for**
7: $z = KX^T v\|v\|/\|vXX^T v\|$;
8: **for** $j = 1 : n - q$ **do** $u(k) = z_j$ and $k \leftarrow k + 1$;
9: **end for**
10: return u;

Algorithm 2 Compute $B_X : \mathbb{R}^{q(q+1)/2+n-q} \to \mathbb{R}^{n\times q} : u \mapsto B_X(u)$

Input: $X \in \mathcal{F}_v$, $u \in \mathbb{R}^{q(q+1)/2-n-q}$; a positive vector $v \in \mathbb{R}^n$; a function $\beta_X : \mathbb{R}^{n\times p} \to \mathbb{R}^{n\times p} : A \mapsto (X \ X_\perp)A$.

1: $k = 1$;
2: **for** $j = 1, \ldots, p$ **do** $S_{ii} = u(k)$ and $k \leftarrow k + 1$;
3: **end for**
4: **for** $j = 2, \ldots, p, i = 1, \ldots j-1$ **do** $S_{ij} = u(k)/\sqrt{2}$, $S_{ji} = u(k)/\sqrt{2}$, and $k \leftarrow k + 1$;
5: **end for**
6: **for** $j = 1 : n - q$ **do** $z_j = u(k)$ and $k \leftarrow k + 1$;
7: **end for**
8: return $\beta_X \begin{pmatrix} S \\ zv^T X/\|v\| \end{pmatrix}$;

4 A Manifold Proximal Gradient Method

For simplicity of notation, throughout this section, we use lowercase x to denote a point in the domain manifold $\mathcal{M} := \mathcal{F}_v$ and use $g_\lambda(x)$ to denote $\lambda\|x\|_1$. Therefore, Problem (1.1) becomes

$$\min_{x\in\mathcal{M}} F_\lambda(x) := f(x) + g_\lambda(x), \tag{4.1}$$

where the subscript λ is used to highlight the dependency on λ.

The proposed algorithm and convergence analysis rely on Assumption 1. Note that $g_\lambda(x) = \lambda\|x\|$ is a Lipschitz continuous function.

Assumption 1 The gradient of f is Lipschitz continuous on $\mathcal{M}$ with a Lipschitz constant L_f and the function g is Lipschitz continuous with Lipschitz constant L_g, where the Lipschitz continuity is defined in the sense of the Euclidean setting.

4.1 Algorithm Description

The manifold proximal gradient method proposed for Problem (4.1) is stated in Algorithm 3. It is based on the AManPG algorithm in [45] and for simplicity is referred to as I-AManPG throughout the remainder of the paper.

Algorithm 3 An Inexact Manifold Proximal Gradient Method (I-AManPG)

Input: Lipschitz constant L_f on ∇f, parameter $\mu > 0$ in the proximal mapping, line search parameter $\sigma \in (0, 1/(8\mu))$, positive integer N for safeguard, initial iterates Λ_y and Λ_z for the semi-smooth Newton algorithm;

$t_0 = 1$, $y_0 = x_0$, $z_0 = x_0$;
for $k = 0, \ldots$ **do**
  if $\mod(k, N) = 0$ **then** ▷ Invoke safeguard every N iterations
    Invoke Algorithm 4: $[z_{k+N}, x_k, y_k, t_k, \Lambda_z] = Alg4(z_k, x_k, y_k, t_k, F_\lambda(x_k), \Lambda_z)$;
  end if
  Approximately solve $\eta_{y_k} \approx \arg\min_{\eta \in \mathrm{T}_{y_k}\mathcal{M}} \langle \mathrm{grad} f(y_k), \eta\rangle + \frac{1}{2\mu}\|\eta\|^2 + g_\lambda(y_k + \eta)$ such that (4.13) holds;
  $x_{k+1} = R_{y_k}(\eta_{y_k})$;
  $t_{k+1} = \frac{\sqrt{4t_k^2+1}+1}{2}$;
  Compute $y_{k+1} = R_{x_{k+1}}\left(\frac{1-t_k}{t_{k+1}} P_{\mathrm{T}_{x_{k+1}}\mathcal{M}}(x_k - x_{k+1})\right)$;
end for

Algorithm 3 is a generalization of the accelerated proximal gradient method (FISTA) [6] to an embedded submanifold. The Euclidean version of FISTA consists of the following steps:

$$\begin{cases} \eta_{y_k} = \arg\min_{\eta \in \mathbb{R}^{n\times p}} \langle \nabla f(y_k), \eta\rangle + \frac{1}{2\mu}\|\eta\|^2 + g_\lambda(y_k + \eta) \\ x_{k+1} = y_k + \eta_{y_k} \\ t_{k+1} = \frac{\sqrt{4t_k^2+1}+1}{2} \\ y_{k+1} = x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k), \end{cases} \tag{4.2}$$

where μ is a positive constant. The subproblem in the first step of (4.2) is generalized to the Riemannian setting by

$$\arg\min_{\eta \in \mathrm{T}_{y_k}\mathcal{M}} \langle \mathrm{grad} f(y_k), \eta\rangle + \frac{1}{2\mu}\|\eta\|^2 + g_\lambda(y_k + \eta), \tag{4.3}$$

where η is required to be in the tangent space. Such a generalization has been used in [18].

It is well known that the FISTA algorithm does not converge globally for non-convex problems. Therefore, a modification is required in FISTA for global convergence. Here, we use the same approach as that in [45] by adding a safeguard of a proximal gradient step, see Algorithm 4. Specifically, let $x_{iN}, x_{iN+1}, \ldots, x_{iN+N}$ be a sequence of $N+1$ consecutive

Algorithm 4 Safeguard for Algorithm 1

Input: $(z_k, x_k, y_k, t_k, F_\lambda(x_k), \Lambda_z)$; shrinking parameter in line search $\nu \in (0,1)$, the maximum number of iterations for line search $N_{\max} > 0$;
Output: $[z_{k+N}, x_k, y_k, t_k, \Lambda_z]$;
1: Approximately solve $\eta_{z_k} \approx \arg\min_{\eta \in \mathrm{T}_{z_k}\mathcal{M}} \langle \mathrm{grad} f(z_k), \eta\rangle + \frac{1}{2\mu}\|\eta\|^2 + g_\lambda(z_k+\eta)$ such that (4.13) holds;
2: Set $\alpha = 1$, and $i_{\text{iter}} = 0$;
3: **while** $F_\lambda(R_{z_k}(\alpha\eta_{z_k})) > F_\lambda(z_k) - \sigma\alpha\|\eta_{z_k}\|^2$ and $i_{\text{iter}} < N_{\max}$ **do**
4: $\alpha = \nu\alpha$; $i_{\text{iter}} = i_{\text{iter}} + 1$;
5: **end while**
6: **if** $i_{\text{iter}} = N_{\max}$ **then**
7: Line search fails;
8: **end if**
9: **if** $F_\lambda(R_{z_k}(\alpha\eta_{z_k})) < F_\lambda(x_k)$ **then** ▷ Safeguard takes effect
10: $x_k = R_{z_k}(\alpha\eta_{z_k})$, $y_k = R_{z_k}(\alpha\eta_{z_k})$, and $t_k = 1$;
11: **else**
12: x_k, y_k and t_k keep unchanged;
13: **end if**
14: $z_{k+N} = x_k$; ▷ Update the compared iterate;

iterates generated by the Riemannian version of the FISTA, i.e., Steps 6 to 9 of Algorithm 3. We then compare the function value at x_{iN+N} to that at an iterate by a proximal gradient step, i.e., $R_{x_{iN}}(\alpha\eta_{x_{iN}})$, where α is an appropriate step size from Step 3 of Algorithm 4. If $F_\lambda(R_{x_{iN}}(\alpha\eta_{x_{iN}})) < F_\lambda(x_{iN+N})$, then the safeguard takes effect by restarting the Riemannian version of FISTA, see Step 9 of Algorithm 4.

Though Algorithm 3 is based on the AManPG in [45, Algorithm 1], they have an important difference that influences their efficiency. The difference is the accuracy of solving the subproblem

$$\arg\min_{\eta \in \mathrm{T}_{y_k}\mathcal{M}} \langle \mathrm{grad} f(y_k), \eta\rangle + \frac{1}{2\mu}\|\eta\|^2 + g_\lambda(y_k+\eta). \tag{4.4}$$

In [45], the authors require the subproblem to be solved exactly for convergence analysis, whereas we only require solving it with sufficient accuracy. The quantitive accuracy that guarantees global convergence is also given.

A proximal mapping in the Euclidean setting often admits a computationally cheap closed-form solution. However, in the Riemannian setting, the proximal mapping does not usually have a closed-form solution due to the existence of an extra linear constraint: $\eta \in \mathrm{T}_{y_k}\mathcal{M}$. The existing Riemannian proximal mappings in [18, 41, 44, 45] are solved by a semi-smooth Newton algorithm. In the theoretical analyses of [18, 41, 45] for global convergence, the Riemannian proximal mappings are assumed to be solved exactly. In [44], an inexact Riemannian proximal gradient (IRPG) is proposed. Though not solving the Riemannian proximal mapping exactly, IRPG assumes a sufficiently small μ in (4.3) which needs to be estimated in practice. Here, we also need the Riemannian proximal mapping to be solved approximately. It is shown later in Lemma 6 that if the Riemannian proximal mapping is solved with sufficient accuracy, then the resulting tangent vector is a descent direction, independent of the choice of μ.

If η_x^* is the exact solution of (4.3), then it satisfies

$$\eta_x^* = \arg\min_\eta \langle \xi_x, \eta\rangle + \frac{1}{2\mu}\langle \eta, \eta\rangle + g_\lambda(x+\eta) \quad \text{subject to} \quad \eta \in \mathrm{T}_x\mathcal{M}, \tag{4.5}$$

where we omit the subscript k, use x instead of y, and use

$$\xi_x = \operatorname{grad} f(x).$$

It follows that $\eta \in \mathrm{T}_x\mathcal{M}$ is equivalent to $B_x^T\eta = 0$, where B_x^T is a linear operator defined in (3.13). Therefore, the KKT condition for (4.5) is given by

$$\partial_\eta \mathcal{L}(\eta, \Lambda) = 0, \tag{4.6}$$

$$B_x^T\eta = 0, \tag{4.7}$$

where $\mathcal{L}(\eta, \Lambda)$ is the Lagrangian function defined by

$$\mathcal{L}(\eta, \Lambda) = \langle \xi_x, \eta\rangle + \frac{1}{2\mu}\langle \eta, \eta\rangle + g_\lambda(x+\eta) - \langle \Lambda, B_x^T\eta\rangle. \tag{4.8}$$

As shown in [18, (4.6)], Equation (4.6) yields

$$\eta = v(\Lambda) := \mathrm{Prox}_{\mu g}\left(x - \mu(\xi_x - B_x\Lambda)\right) - x, \tag{4.9}$$

where B_x is a linear operator defined in (3.12) and

$$\mathrm{Prox}_{\mu g}(z) = \arg\min_{v\in\mathbb{R}^{n\times p}} \frac{1}{2}\|v - z\|^2 + \mu g_\lambda(v) \tag{4.10}$$

denotes the proximal mapping. Substituting (4.9) into (4.7) yields that

$$\Psi(\Lambda) := B_x^T\left(\mathrm{Prox}_{\mu g}\left(x - \mu(\xi_x - B_x\Lambda)\right) - x\right) = 0, \tag{4.11}$$

which is a system of nonlinear equations with respect to Λ. Therefore, to solve (4.5), one can first find any root of (4.11) and substitute it back to (4.9) to obtain η_x^*.

Equation (4.11) can be solved efficiently by a semi-smooth Newton method. Analogous to the classical Newton method, the estimation of Λ is updated by $\Lambda_{k+1} = \Lambda_k + d_k$, where d_k is computed by solving a Newton equation, i.e.,

$$J_\Psi(\Lambda_k)[d] = -\Psi(\Lambda_k), \tag{4.12}$$

where $J_\Psi(\Lambda_k)$ is a generalized Jacobian of Ψ. Note that when $g_\lambda(x) = \lambda\|x\|_1$, it is well-known [7] that the solution to the proximal mapping (4.10) can be computed by thresholding each entry of z. Moreover, by the chain rule, we have

$$J_\Psi(\Lambda_k)[d] = B_x^T\left(\partial\mathrm{Prox}_{\mu g}\left(x - \mu(\xi_x - B_x\Lambda_k)\right) \odot (\mu B_x d)\right),$$

where $\partial\mathrm{Prox}_{\mu g}(\cdot)$ denotes the generalized Clarke subdifferential of $\mathrm{Prox}_{\mu g}(\cdot)$ and $\odot$ denotes the entrywise product of two matrices. Once again, when $g_\lambda(x) = \lambda\|x\|_1$ the generalized Clarke subdifferential of $\mathrm{Prox}_{\mu g}(\cdot)$ can also be computed in an entrywise manner [20, 51, 76]. Here, we do not require solving (4.11) exactly but only find a Λ such that $\|\Psi(\Lambda)\|$ is sufficiently small. See Lemma 5 for details.

4.2 Global Convergence Analysis

Lemma 5 states the key result used to prove the global convergence. In [18, Lemma 5.1], the inequality (4.14) is proven up to a coefficient under the assumption that the subproblem (4.3) is solved exactly. Here, it is shown that if the subproblem is solved accurately enough such that (4.13) holds, then we also have the inequality (4.14). The result (4.14) has not been given for the existing inexact Riemannian proximal gradient method [44] and is crucial to Algorithm 3.

Lemma 5 *Let* $\ell_x(\eta) = \langle \xi_x, \eta\rangle + \frac{1}{2\mu}\langle \eta, \eta\rangle + g_\lambda(x+\eta)$. *Let* ϵ *denote* $\Psi(\Lambda) = B_x^T v(\Lambda)$. *We then have*

$$g_\lambda(x) \geq \left\langle \xi_x, \hat{v}(\Lambda)\right\rangle + \frac{1}{2\mu}\|\hat{v}(\Lambda)\|^2 + g_\lambda(x+\hat{v}(\Lambda)) - (2L_g + \frac{1}{2\mu}\|\epsilon\|)\|\epsilon\|,$$

where $\hat{v}(\Lambda) = P_{\mathrm{T}_{x_k}\mathcal{M}}(\mathrm{Prox}_{\mu g}(x - \mu(\xi_x - B_x\Lambda)) - x)$. *Furthermore, if* ϵ *is sufficiently close to 0 in the sense that*

$$\|\Psi(\Lambda)\| = \|\epsilon\| \leq \sqrt{4\mu^2 L_g^2 + \|\hat{v}(\Lambda)\|^2/2} - 2\mu L_g, \tag{4.13}$$

then it holds that

$$\ell_x(\alpha\hat{v}(\Lambda)) - \ell_x(0) \leq -\left(\frac{\alpha(1-2\alpha)}{4\mu}\right)\|\hat{v}(\Lambda)\|^2, \quad \forall \alpha \in [0,1]. \tag{4.14}$$

Proof Consider the optimization problem

$$\min_{B_x^T\eta = \epsilon} \ell_x(\eta). \tag{4.15}$$

Problem 4.15 is obtained by slightly shifting the hyperplane constraints of the subproblem in (4.5). Next, we will show that if the shift ϵ is sufficiently small, then the resulting vector $\hat{v}(\Lambda)$ shares the same property (4.14) of the minimizer of (4.5). Its KKT condition is given by

$$\partial_\eta \mathcal{L}(\eta, \Lambda) = 0, \qquad B_x^T\eta = \epsilon,$$

which is satisfied by $v(\Lambda)$ defined in (4.9). Therefore, $v(\Lambda)$ is the minimizer of $\ell_x(\eta)$ over the set $\mathcal{S} = \{v : B_x^T v = \epsilon\}$, i.e.,

$$v(\Lambda) = \arg\min\nolimits_{v\in\mathcal{S}} \ell_x(\eta) = \langle \xi_x, \eta\rangle + \frac{1}{2\mu}\langle \eta, \eta\rangle + g_\lambda(x+\eta). \tag{4.16}$$

Define the vector $v_0 = B_x\epsilon$. It can be easily verified that $B_x^T v_0 = B_x^T B_x\epsilon = \epsilon$. Therefore, it holds that $v_0 \in \mathcal{S}$. By $\frac{1}{\mu}$-strong convexity of ℓ_x, we have

$$\ell_x(v_0) \geq \ell_x(v(\Lambda)) + \langle \partial\ell_x(v(\Lambda)), v_0 - v(\Lambda)\rangle + \frac{1}{2\mu}\|v_0 - v(\Lambda)\|^2. \tag{4.17}$$

From the optimality condition of Problem (4.15), we have that $0 \in P_{\mathrm{T}_\eta\mathcal{S}}\partial\ell_x(v(\Lambda))$. Since $\mathrm{T}_\eta\mathcal{S} = \{u : B_x^T u = 0\}$ and $B_x^T(v_0 - v(\Lambda)) = \epsilon - \epsilon = 0$, it holds that $v_0 - v(\Lambda) \in \mathrm{T}_\eta\mathcal{S}$. Therefore, we have

$$0 \in \langle \partial\ell_x(v(\Lambda)), v_0 - v(\Lambda)\rangle. \tag{4.18}$$

It follows from (4.17) and (4.18) that

$$\ell_x(B_x\epsilon) \geq \ell_x(v(\Lambda)) + \frac{1}{2\mu}\|v(\Lambda) - B_x\epsilon\|^2. \tag{4.19}$$

Substituting the definition of ℓ_x into inequality (4.19) and noting $\epsilon = B_x^T v(\Lambda)$, we have that

$$\begin{aligned}&\frac{1}{2\mu}\|B_x B_x^T v(\Lambda)\|^2 + g_\lambda(x + B_x B_x^T v(\Lambda))\\ &\quad\geq \left\langle \xi_x, v(\Lambda) - B_x B_x^T v(\Lambda)\right\rangle + \frac{1}{2\mu}\|v(\Lambda)\|^2 + \quad g_\lambda(x+v(\Lambda)) + \frac{1}{2\mu}\|v(\Lambda) - B_x B_x^T v(\Lambda)\|^2.\end{aligned} \tag{4.20}$$

It follows that

$$\begin{aligned} g_\lambda(x) &\geq \langle \xi_x, P_{\mathrm{T}_x\mathcal{M}} v(\Lambda)\rangle + \frac{1}{2\mu}\|P_{\mathrm{T}_x\mathcal{M}} v(\Lambda)\|^2 + g_\lambda(x + v(\Lambda)) + g_\lambda(x) \\ &\quad - g_\lambda(x + B_x\epsilon) - \frac{1}{2\mu}\|B_x B_x^T v(\Lambda)\|^2 \\ &\geq \langle \xi_x, \hat{v}(\Lambda)\rangle + \frac{1}{2\mu}\|\hat{v}(\Lambda)\|^2 + g_\lambda(x + \hat{v}(\Lambda)) + g_\lambda(x + v(\Lambda)) \\ &\quad - g_\lambda(x + (I - B_x B_x^T)v(\Lambda)) + g_\lambda(x) - g_\lambda(x + B_x\epsilon) - \frac{1}{2\mu}\|B_x\epsilon\|^2 \\ &\geq \langle \xi_x, \hat{v}(\Lambda)\rangle + \frac{1}{2\mu}\|\hat{v}(\Lambda)\|^2 + g_\lambda(x + \hat{v}(\Lambda)) - |g_\lambda(x + v(\Lambda)) - g_\lambda(x + v(\Lambda) - B_x\epsilon)| \\ &\quad - |g_\lambda(x) - g_\lambda(x + B_x\epsilon)| - \frac{1}{2\mu}\|B_x\epsilon\|^2 \\ &\geq \langle \xi_x, \hat{v}(\Lambda)\rangle + \frac{1}{2\mu}\|\hat{v}(\Lambda)\|^2 + g_\lambda(x + \hat{v}(\Lambda)) - (2L_g + \frac{1}{2\mu}\|B_x\epsilon\|)\|B_x\epsilon\|, \end{aligned} \tag{4.21}$$

where the first inequality follows from (4.20) and $\|v(\Lambda)\|^2 \geq 0$, the second inequality follows from $\hat{v}(\Lambda) = P_{\mathrm{T}_x\mathcal{M}} v(\Lambda) = (I - B_x B_x^T)v(\Lambda)$, and the fourth inequality follows from the Lipschitz continuity of g with Lipschitz constant L_g. This completes the proof for the first result.

Since g is convex, we have

$$\begin{aligned} g_\lambda(x + \alpha\hat{v}(\Lambda)) - g_\lambda(x) &= g_\lambda(\alpha(x + \hat{v}(\Lambda)) + (1 - \alpha)x) - g_\lambda(x) \\ &\leq \alpha(g_\lambda(x + \hat{v}(\Lambda)) - g_\lambda(x)). \end{aligned} \tag{4.22}$$

Combining (4.21) and (4.22) yields

$$\begin{aligned} \ell_x(\alpha\hat{v}(\Lambda)) - \ell_x(0) &= \langle \xi_x, \alpha\hat{v}(\Lambda)\rangle + \frac{1}{2\mu}\|\alpha\hat{v}(\Lambda)\|^2 + g_\lambda(x + \alpha\hat{v}(\Lambda)) - g_\lambda(x) \\ &\leq \alpha\left(\langle \xi_x, \hat{v}(\Lambda)\rangle + \frac{\alpha}{2\mu}\|\hat{v}(\Lambda)\|^2 + g_\lambda(x + \hat{v}(\Lambda)) - g_\lambda(x)\right) \\ &\leq \alpha\left(\frac{\alpha}{2\mu}\|\hat{v}(\Lambda)\|^2 - \frac{1}{2\mu}\|\hat{v}(\Lambda)\|^2 + (2L_g + \frac{1}{2\mu}\|B_x\epsilon\|)\|B_x\epsilon\|\right). \end{aligned} \tag{4.23}$$

By $\|\epsilon\| = \|B_x\epsilon\| \leq \sqrt{4\mu^2 L_g^2 + \|\hat{v}(\Lambda)\|^2/2} - 2\mu L_g$, we have

$$(2L_g + \frac{1}{2\mu}\|B_x\epsilon\|)\|B_x\epsilon\| \leq \frac{1}{4\mu}\|\hat{v}(\Lambda)\|^2. \tag{4.24}$$

The second result follows from (4.23) and (4.24). Finally, (4.13) follows from the definition of $\Psi(\Lambda)$. □

Lemma 6 implies that η_{z_k} is a descent direction and the while loop in Step 3 of Algorithm 4 terminates in a finite number of iterations. Given (4.14) in Lemma 5, the proof of Lemma 6 follows the same steps as that of [18, Lemma 5.2] and is therefore omitted.

Lemma 6 *Suppose Assumption 1 holds. Then for any $\mu > 0$, there exists a constant $\bar{\alpha} \in (0, 1]$ such that for any $0 < \alpha < \bar{\alpha}$, Step 3 of Algorithm 4 is satisfied, and the sequence $\{z_k\}$ generated by Algorithm 3 satisfies*

$$F_\lambda(R_{z_k}(\alpha\eta_{z_k})) - F_\lambda(z_k) \leq -\frac{\alpha}{8\mu}\|\eta_{z_k}\|^2.$$

Moreover, the step size $\alpha > \rho\bar{\alpha}$ for all k.

Though the subproblem is solved inexactly, a zero search direction given by $\hat{v}(\Lambda)$ with condition (4.13) implies that the current iterate x is a stationary point, which coincides with [18, Lemma 5.3].

Lemma 7 *If $\eta_x = \hat{v}(\Lambda) = 0$, then x is a stationary point of Problem (4.1).*

Proof If $\eta_x = \hat{v}(\Lambda) = 0$, then by (4.13), we have that $\Psi(\Lambda) = 0$, which implies that the subproblem (4.5) is solved exactly and $\eta_x^* = 0$. By [18, Lemma 5.3], x is a stationary point of Problem 4.1. □

The main convergence result is given in Theorem 5. The proof follows the spirit of [45, Theorem 1]. Here, we only highlight their differences. Note that the convergence result is non-asymptotic.

Theorem 5 *Suppose Assumption 1 holds, then any accumulation point of the sequence $\{z_0, z_N, z_{2N}, \ldots, z_{iN}, \ldots\}$ generated by Algorithm 3 is a stationary point, i.e., if z_* is an accumulation point of the above sequence, then $0 \in P_{\mathrm{T}_{z_*}\mathcal{M}}\partial F_\lambda(z_*)$.*

Proof Since the subscript of z_k in Algorithm 3 is a multiple of N, we use $\{\tilde{z}_i\}$ to denote $\{z_k\}$, where $\tilde{z}_i = z_{iN}$. Let $(\eta_{\tilde{z}_i}, \Lambda_{\tilde{z}_i})$ denote the output of the Semi-smooth Newton algorithm when the input is $(\tilde{z}_i, \Lambda_{\tilde{z}_{i-1}}, \mathrm{grad}\, f(\tilde{z}_i))$, i.e., the input and output of Step 1 of Algorithm 4.

By the safeguard in Algorithm 4 and Lemma 6, we have $F_\lambda(\tilde{z}_{i+1}) - F_\lambda(\tilde{z}_i) \le -\frac{\rho\bar{\alpha}}{8\mu}\|\eta_{\tilde{z}_k}\|^2$. Since F_λ is continuous and $\mathcal{F}$ is compact, the function F_λ is bounded from below. It follows that

$$\infty > \sum_{i=0}^{\infty} F_\lambda(\tilde{z}_i) - F_\lambda(\tilde{z}_{i+1}) \ge \frac{\rho\bar{\alpha}}{8\mu}\|\eta_{\tilde{z}_k}\|^2, \tag{4.25}$$

which implies

$$\lim_{i\to\infty} \|\eta_{\tilde{z}_k}\| = 0. \tag{4.26}$$

By (4.26), (4.13), and $\eta_{\tilde{z}_k} = \hat{v}(\Lambda_{\tilde{z}_i})$, we have

$$\lim_{i\to\infty} \|\Psi(\Lambda_{\tilde{z}_i})\| = 0. \tag{4.27}$$

Since $\hat{v}(\Lambda_{\tilde{z}_i}) = P_{\mathrm{T}_{\tilde{z}_i}\mathcal{M}} v(\Lambda_{\tilde{z}_i}) = v(\Lambda_{\tilde{z}_i}) - B_{\tilde{z}_i}\Psi(\Lambda_{\tilde{z}_i})$, we have

$$\|v(\Lambda_{\tilde{z}_i})\| = \|\hat{v}(\Lambda_{\tilde{z}_i}) + B_{\tilde{z}_i}\Psi(\Lambda_{\tilde{z}_i})\| \le \|\hat{v}(\Lambda_{\tilde{z}_i})\| + \|B_{\tilde{z}_i}\Psi(\Lambda_{\tilde{z}_i})\| = \|\eta_{\tilde{z}_k}\| + \|\Psi(\Lambda_{\tilde{z}_i})\|. \tag{4.28}$$

Combining (4.26), (4.27) and (4.28) yields

$$\lim_{i\to\infty} \|v(\Lambda_{\tilde{z}_i})\| = 0. \tag{4.29}$$

By (4.16), we have

$$v(\Lambda_{\tilde{z}_i}) = \arg\min_{\eta\in\mathcal{S}_{\tilde{z}_i}} \ell_x(\eta) = \langle \mathrm{grad}\, f(\tilde{z}_i), \eta\rangle + \frac{1}{2\mu}\langle \eta, \eta\rangle + g_\lambda(\tilde{z}_i + \eta), \tag{4.30}$$

where $\mathcal{S}_{\tilde{z}_i} = \{v : B_{\tilde{z}_i}^T v = B_{\tilde{z}_i}^T v(\Lambda_{\tilde{z}_i})\}$. Using (4.29) and (4.30) and following the steps in the proof of [45, Theorem 1], we have that any accumulation point of $\{\tilde{z}_i\}$ is a stationary point. □

5 Numerical Experiments

In this section, the performance of the proposed method, Algorithm 3, is compared to other state-of-the-art methods using problems from community detection in networks and normalized cut for image segmentation.

5.1 Parameter Setting and Testing Environment

The subproblem in Algorithm 3 is solved by the regularized semi-smooth Newton algorithm in [76]. Let E-AManPG denote Algorithm 3 with the condition that $\|\Psi(\Lambda)\| \leq 10^{-10}$. Therefore, we can view E-AManPG is essentially an exact one since E-AManPG solves the Riemannian proximal mapping to high accuracy.

The parameters L_f and λ are problem-dependent and specified later. The parameters in Algorithm 3 are set to be $\mu = 1/L_f$, $\sigma = 10^{-4}$, $\nu = 0.5$, $N = 5$, $\Lambda_y = \Lambda_z = 0$ and the parameter in Algorithm 4 is set to be $N_{\max} = 5$.

Unless otherwise indicated, I-AManPG and E-AManPG stop if the value of $\|\eta_{z_k}\|$ reduces at least by a factor of 10^3. Inspired by Lemma 2, the last iterate is projected to the set $\mathcal{B}_v$ by the mapping $P_{\mathcal{B}_v}(X)$.

I-AManPG and E-AManPG are implemented in Matlab R2019b. All the experiments are performed on an Apple Mac platform with 1.4 GHz Quad-Core Intel Core i5. The amount of system memory is 8GB. The CPU has 4 cores, and all 8 computational threads are used. The codes for reproducing the numerical experiments in this paper is available at https://www.math.fsu.edu/~whuang2/papers/AROACP.htm.

5.2 Community Detection

In this section, we evaluate the performance of community detection by optimizing the formulation

$$\min_{X \in \mathcal{F}_{\mathbf{1}_n}} -\text{trace}(X^T M X) + \lambda \|X\|_1, \tag{5.1}$$

with I-AManPG algorithm.

5.2.1 Data Sets

Results are presented for solving community detection problems on synthetic LFR benchmark networks [50]. Specifically, LFR benchmark networks assume that the distributions of degree and community size are power laws with exponents τ_1 and τ_2 respectively. Each node shares a fraction $1 - \mu_{\text{LFR}}$ of its edges with the other nodes of its community and a fraction μ_{LFR} with nodes of the other communities, where $0 \leq \mu_{\text{LFR}} \leq 1$ is the mixing parameter. A software package to generate the benchmark networks is available at https://www.santofortunato.net/resources.

5.2.2 Comparison of I-AManPG and E-AManPG on LFR Benchmark Networks

The first set of experiments compares the efficiency and effectiveness of I-AManPG and E-AManPG to demonstrate the utility of the inexact form of the algorithm. Throughout this

Table 1 Comparison of the efficiency of I-AManPG and E-AManPG

	I-A	E-A	I-A	E-A	I-A	E-A	I-A	E-A
(N, n_c)	(500, 10)		(1000, 10)		(5000, 10)		(10000, 10)	
iter	64	52	50	59	63	58	55	55
SSNiter	28	212	13	248	34	311	52	330
nf	143	115	112	131	140	128	123	122
ng	83	65	64	73	81	72	71	68
nR	142	114	111	130	139	127	122	121
nSG	4	14	2	15	4	13	3	10
F	-67.0	-67.0	-149	-149	-284	-284	-251	-251
$\frac{\|\eta_{z_k}\|}{\|\eta_{z_0}\|}$	7.0_{-4}	5.7_{-4}	5.5_{-4}	5.1_{-4}	6.3_{-4}	5.8_{-4}	5.2_{-4}	6.9_{-4}
time	0.15	0.31	0.17	0.75	0.84	3.03	1.54	5.19

section, the parameters τ_1, τ_2, and μ_{LFR} are set to -2, -1, and 0.1 respectively. Four sets of other parameters are used to define a range of networks. They are as follows:

- $N = 500$, $d_{ave} = 10$, $d_{max} = 20$, $N_c = 50$, $n_c = 10$;
- $N = 1000$, $d_{ave} = 20$, $d_{max} = 40$, $N_c = 100$, $n_c = 10$;
- $N = 5000$, $d_{ave} = 40$, $d_{max} = 80$, $N_c = 500$, $n_c = 10$;
- $N = 10000$, $d_{ave} = 40$, $d_{max} = 80$, $N_c = 1000$, $n_c = 10$;

where N denotes the number of nodes, d_{ave} denotes the average node degree, d_{max} denotes the maximum node degree, N_c denotes the number of nodes that all communities have, and n_c denotes the number of communities. The balancing parameter λ in (5.1) is set to 0.3. This value roughly balances the two terms of the cost function for the LFR benchmark networks defined by the parameters set above.

Table 1 contains the data to compare I-AManPG and E-AManPG for LFR benchmark networks. For each of the four LFR parameter sets, 10 random networks were generated. The computational time for each network was computed as the average computational time of 10 runs and the computational time is in seconds. The quantities reported in the table are the averages over the 10 networks from each parameter set. I-A denotes I-AManPG and E-A denotes E-AManPG. The LFR parameter set used for each pair of columns is identified by values (N, n_c). The labels iter, SSNiter, nf, ng, nR, nSG, F, $\frac{\|\eta_{z_k}\|}{\|\eta_{z_0}\|}$, and time, respectively, denote the number of iterations in AManPG, the number of iterations in semi-smooth Newton method, the number of function evaluations, the number of gradient evaluations, the number of retraction evaluations, the number of safeguards (Step 9) that are taken, the function value at the final iterate, the reduction of the norm of search directions $\frac{\|\eta_{z_k}\|}{\|\eta_{z_0}\|}$, and the computational time in seconds. The subscript k denotes a scale of 10^k.

As shown in Table 1, I-AManPG and E-AManPG find the same solutions in the sense that their function values are the same up to three significant digits. In fact, though not reported in the table, in our experimental setting, both I-AManPG and E-AManPG always converge to the same solution which always represents the ground truth partition. So both are equally effective. In addition, I-AManPG requires less work when solving the Riemannian proximal mapping in the sense that the number of semi-smooth Newton iterations in each outer iteration is small compared to E-AManPG. Moreover, less accuracy for solving the Riemannian proximal mapping does not influence the number of outer iterations significantly.

Therefore, I-AManPG is more efficient than E-AManPG in terms of computational time. For the rest of the experiments, we use I-AManPG as the representative method.

5.2.3 Comparison of the Effectiveness of I-AManPG and State-of-the-Art Community Detection Methods

In this section, community detection by the optimization model (5.1) using I-AManPG is compared to three state-of-the-art methods: Danon et al.'s algorithm [21], the Louvain method [9] and Newman's spectral optimization method [56].

These algorithms all aim to maximize the modularity $Q = \frac{1}{2m} trace(X^T M X)$ over the set of indicator matrices, where m is the number of edges, and an indicator matrix X is defined by $X \geq 0$, $X^T X$ is diagonal, and each row of X has a single entry with value one. Each indicator matrix specifies a partitioning of the nodes into communities. Danon et al.'s algorithm is an agglomerative method that is a variant of Newman's fast greedy method [54] which starts with each node as a singleton community. Intra-community edges are added one-by-one by choosing the edge such that the modified partition gives the maximum increase of modularity with respect to the previous configuration. The Louvain method is an agglomerative method where each iteration comprises two phases. The first phase creates intermediate-communities by merging pairs of nodes such that the modularity increases. The first phase terminates when no such pair merging increases modularity. In the second phase, a smaller graph, called the reduced graph, is created where each node in this graph represents an intermediate-community. Newman's spectral optimization method is a divisive method that computes the eigenvectors of the modularity matrix M corresponding to the largest positive eigenvalues. The nodes are grouped into two parts based on the signs of the component of the eigenvectors. The process is then repeated for each of the parts until splitting a given community of nodes makes a zero or negative contribution to the total modularity.

To make fair comparisons, we use publicly-available Matlab implementations of these algorithms. The codes for Danon et al.'s algorithm, Louvain's algorithm, and Newman's spectral method are respectively from [13, 49, 66]. Note that the codes for Newman's spectral method in [13] do not embed the fine-tuning stage and use a different stopping criteria as in paper [56]. We modified the stopping criteria in the codes such that it has the same stopping criteria as in paper [56]. The codes in [13] use dense matrix computations, and we modified them with significantly more efficient sparse computations. In this way, the computational efficiency was improved to the point where it produced times that were reasonable to include in these comparisons.

To compare the effectiveness of the four methods, we consider three quality measurements: normalized mutual information (NMI) [22], adjusted mutual information (AMI) [71], and purity [53]. NMI is a similarity measure between two partitions that represents their normalized mutual entropy. AMI further corrects the measure for randomness by adopting a hypergeometric model of randomness. We refer interested readers to [22, (2)] and [71, Section 4.1] for the definitions. Both NMI and AMI take on values between 0 and 1. Values closer to 1 indicate greater consistency between the partitions.

Given two partitions X and Y of N nodes, the purity is given by $\text{purity}(X, Y) = \frac{1}{N} \sum_k \max_j |X_k \cap Y_j|$, where X_k denotes the set of nodes in k-th community of partition X, and likewise for Y_j, and $|X_k \cap Y_j|$ denotes the number of nodes in $X_k \cap Y_j$. The value of purity is also between 0 and 1. The closer it is to one, the better the two partitions are nested. In our numerical experiments, the ground truth is known and therefore the computed partition is compared to the ground truth. When the two partitionings do not have the same number of communities good nesting of the partition with the larger number of communities in the

partition with the smaller number is an indication that further division or agglomeration could yield a closer approximation of ground truth. Such a pair of partitions is therefore preferred to a pair with lower purity. Since purity is not symmetric, we take X to be the partition with the larger number of communities.

For LFR benchmark networks used in the comparisons in this section, the parameters τ_1, τ_2, N, d_{ave}, d_{max}, N_c, and n_c are respectively set to -2, -1, 1000, 20, 40, 50, and 20. The value of λ in (5.1) is 0.3, as in the previous set of experiments. The empirical results with multiple values of μ_{LFR} are reported in Table 2. As before, each result is an average over 10 randomly selected LFR benchmark networks.

The input parameter for I-AManPG determining the number of communities to be produced is q and q_c is the number of communities computed by each method. The "force_q" label denotes the versions with modified termination criteria so that q_c is as close to q_{true} as the methods allow.

From the results in Table 2, we observe that when $\mu_{\text{LFR}} = 0$, I-AManPG yields NMI = AMI = purity = 1, the same modularity value and the same assignment to $q_{true} = 20$ strongly connected communities. The Louvain method also has the same results with ground-truth communities while Danon et al.'s algorithm and Newman's spectral algorithm can get results that are very close to the ground-truth communities. Specifically, they can detect exactly ground-truth communities for 9 of 10 random LFR graphs. When μ_{LFR} takes 0.1 to 0.4, I-AManPG and Louvain algorithm can detect the exact ground-truth communities. When $\mu_{\text{LFR}} = 0.5, 0.6$, I-AManPG can get results very close to ground-truth partitions and the results are competitive results with the Louvain algorithm, but with less time. When $\mu_{\text{LFR}} = 0.7, 0.8$, the results for all of these four algorithms are far away from the ground-truth partitions because the community structures in these cases are not strong. Danon's algorithm and Newman spectral algorithm detect relatively inaccurate communities and relatively small qualifying external or internal measurements, i.e., NMI, AMI, purity, and modularity for all noisy cases. From the computational times, we can see that I-AManPG requires relatively less time than the others. It is worth noting that the number of edges m does not change much and only the distribution of edges changes a lot as the mixing parameter increases. So, the computational time for Danon's algorithm and Newman's spectral method do not change much as the mixing parameter increases because the computational time of these two algorithms depends more on m rather than on the distribution of edges.

As the mixing parameter increases, the difficulty level of detecting the correct number of communities increases as well. I-AManPG requires the desired number of communities as an input parameter value, q, and the choice of an initial q and the development of a dynamic adaptation strategy are key ongoing research tasks for I-AManPG. Since the experiments in Table 2 use $q = q_{true}$ for I-AManPG and the other methods that are not "forced" are given no indication of q_{true}, experiments where I-AManPG uses $q \neq q_{true}$ probe the quality of the $q \neq q_{true}$ communities produced by I-AManPG compared to ground truth. For each value of the input parameter $q = 10, 17, \ldots, 23$ and mixing parameter $\mu_{\text{LFR}} = 0, 0.1, \ldots, 0.8$, I-AManPG was applied to 10 randomly generated LFR benchmark networks. The results are shown in Table 3.

Consider the results of different algorithms for the cases with the same computed number of communities. From the results in Tables 2 and 3, we observe that NMI, AMI, modularity and purity of I-AManPG are larger than the results of Danon's algorithm for $q = 18$ and $\mu = 0.2$, Newman's spectral algorithm for $q = 21$ and $\mu = 0.2$ and Newman's spectral algorithm for $q = 19$ and $\mu = 0.3$. NMI, AMI, modularity and purity of I-AManPG are competitive with the results of Louvain's algorithm for $q = 19$ and $\mu = 0.6$ and I-AManPG requires less time.

Table 2 Compare the effectiveness of I-AManPG to other state-of-the-art methods

		μ_{LFR}								
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Danon	NMI	0.9998	0.9891	0.9394	0.8504	0.7331	0.5808	0.3781	0.1412	0.0548
	AMI	0.9998	0.9870	0.9166	0.7922	0.6399	0.4736	0.2878	0.0935	0.0215
	Mod	0.9496	0.8436	0.7225	0.5920	0.4687	0.3452	0.2458	0.1892	0.1814
	purity	0.9999	0.9938	0.9739	0.9414	0.9058	0.8344	0.6886	0.4269	0.3095
	time	2.8304	2.8597	2.8199	2.7648	2.7287	2.8318	2.7739	2.7625	2.7318
	q_c	20	20	18	15	11	9	7	7	8
Danon_force_q	NMI	0.9998	0.9889	0.9397	0.8494	0.7309	0.5794	0.3939	0.1778	0.0954
	AMI	0.9998	0.9870	0.9201	0.7962	0.6415	0.4727	0.2950	0.1035	0.0282
	Mod	0.9496	0.8433	0.7209	0.5868	0.4615	0.3394	0.2431	0.1872	0.1794
	purity	0.9999	0.9935	0.9714	0.9349	0.8954	0.8198	0.6778	0.4201	0.3065
	time	2.8353	2.9606	2.8222	2.8068	2.8055	2.8959	2.8183	2.8459	2.7432
	q_c	20	20	20	20	20	20	20	20	20
Louvain	NMI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9987	**0.9805**	0.2862	0.0784
	AMI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9974	**0.9652**	0.2249	0.0358
	Mod	**0.9497**	**0.8499**	**0.7503**	**0.6500**	**0.5499**	0.4496	**0.3477**	**0.2098**	**0.1967**
	purity	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9999**	**0.9950**	0.4734	0.2660
	time	0.5444	0.7291	1.3703	1.8963	2.6797	3.3418	4.5768	9.2669	8.8130
	q_c	20	20	20	20	20	20	19	11	11

Table 2 continued

		μ_{LFR}								
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Louvain_force_q	NMI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9987	**0.9805**	0.2981	0.0847
	AMI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9974	**0.9652**	0.2382	0.0392
	Mod	**0.9497**	**0.8499**	**0.7503**	**0.6500**	**0.5499**	0.4496	**0.3477**	**0.2098**	**0.1967**
	purity	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9999**	**0.9950**	0.4642	0.2418
	time	**0.5440**	0.7478	1.0333	1.2042	1.7002	2.0761	2.7676	5.4529	5.5061
	q_c	20	20	20	20	20	20	19	12	12
Newman_Eig Sparse	NMI	0.9988	0.7225	0.7132	0.6760	0.5498	0.3912	0.2807	0.1340	0.0497
	AMI	0.9985	0.6521	0.6396	0.6122	0.4704	0.3071	0.2098	0.0907	0.0235
	Mod	0.9482	0.5493	0.5057	0.4157	0.3051	0.2379	0.1917	0.1578	0.1461
	purity	0.9994	0.7828	0.8134	0.7460	0.6723	0.6366	0.5654	0.4367	0.3594
	time	0.6346	**0.4391**	**0.4197**	**0.4522**	**0.4140**	**0.3333**	**0.3565**	**0.3130**	**0.3026**
	q_c	20	24	21	19	15	9	7	6	6
Newman_Eig_force_q Sparse	NMI	0.9988	0.6831	0.6787	0.6674	0.5498	0.3912	0.2807	0.1340	0.0497
	AMI	0.9985	0.5996	0.5991	0.6026	0.4704	0.3071	0.2098	0.0907	0.0235
	Mod	0.9482	0.4747	0.4463	0.4004	0.3051	0.2379	0.1917	0.1578	0.1461
	purity	0.9994	0.7998	0.8001	0.7358	0.6723	0.6366	0.5654	0.4367	**0.3594**
	time	0.6458	0.4666	0.4373	0.4528	0.4231	0.3418	0.3656	0.3213	0.3114
	q_c	20	18	17	18	15	9	7	6	6
I-AManPG	NMI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9998**	0.9600	**0.4517**	**0.1294**
	AMI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9998**	0.9539	**0.4037**	**0.0563**
	Mod	**0.9497**	**0.8499**	**0.7503**	**0.6500**	**0.5499**	**0.4498**	0.3416	0.1735	0.1113
	purity	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9999**	0.9679	**0.5605**	0.3044
	time	0.6357	0.4693	0.5870	0.9494	0.6749	0.4720	1.0332	1.6307	1.6757
	q	20	20	20	20	20	20	20	20	20

Table 3 Testing the effectiveness of I-AManPG for input parameter values $q \neq q_{true}$

		μ_{LFR}								
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
I-AManPG	NMI	0.7178	0.7178	0.7178	0.7176	0.7137	0.6966	0.6230	0.2957	0.0795
	AMI	0.5452	0.5452	0.5452	0.5451	0.5428	0.5327	0.4856	0.2202	0.0324
	Mod	0.6924	0.6198	0.5474	0.4733	0.4005	0.3251	0.2531	0.1534	0.1007
	purity	**1.0000**	**1.0000**	**1.0000**	0.9999	0.9971	0.9842	0.9243	**0.6363**	**0.5118**
	time	**0.1939**	**0.2175**	**0.5020**	**0.3373**	**0.2593**	0.4983	**0.3297**	**0.7154**	**0.6616**
	q	10	10	10	10	10	10	10	10	10
I-AManPG	NMI	0.9531	0.9523	0.9531	0.9523	0.9531	0.9493	0.9121	0.4132	0.1124
	AMI	0.9049	0.9034	0.9049	0.9034	0.9049	0.8999	0.8675	0.3535	0.0466
	Mod	0.9245	0.8266	0.7306	0.6324	0.5363	0.4367	0.3362	0.1724	0.1100
	purity	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9985	0.9704	0.5765	0.3498
	time	1.0201	0.7729	0.6772	0.6584	0.7162	0.7193	1.0411	1.7427	1.5559
	q	17	17	17	17	17	17	17	17	17
I-AManPG	NMI	0.9717	0.9722	0.9665	0.9726	0.9717	0.9709	0.9411	0.4298	0.1183
	AMI	0.9415	0.9424	0.9368	0.9433	0.9415	0.9417	0.9131	0.3733	0.0498
	Mod	0.9362	0.8382	0.7367	0.6423	0.5427	0.4442	0.3424	0.1721	0.1089
	purity	**1.0000**	**1.0000**	0.9947	**1.0000**	**1.0000**	0.9989	0.9773	0.5746	0.3379
	time	1.0509	0.9585	0.7564	0.7069	0.8014	0.7777	1.1587	2.0853	1.9374
	q	18	18	18	18	18	18	18	18	18

Table 3 continued

		μ_{LFR}								
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
I-AManPG	NMI	0.9883	0.9883	0.9845	0.9883	0.9796	0.9828	0.9520	0.4339	0.1264
	AMI	0.9753	0.9753	0.9716	0.9753	0.9667	0.9699	0.9365	0.3803	0.0557
	Mod	0.9453	0.8460	0.7446	0.6474	0.5445	0.4468	**0.3427**	0.1728	0.1111
	purity	**1.0000**	**1.0000**	0.9962	**1.0000**	0.9914	0.9949	0.9714	0.5634	0.3203
	time	0.9389	0.9057	0.8584	0.9534	0.7417	1.0226	1.0085	2.1798	1.8913
	q	19	19	19	19	19	19	19	19	19
I-AManPG	NMI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9998**	0.9600	0.4517	0.1294
	AMI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9998**	0.9539	0.4037	0.0563
	Mod	**0.9497**	**0.8499**	**0.7503**	**0.6500**	**0.5499**	**0.4498**	0.3416	**0.1735**	0.1113
	purity	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9999**	0.9679	0.5605	0.3044
	time	0.6577	0.5052	0.6423	0.9890	0.7112	**0.4951**	1.0755	1.7648	1.7855
	q	20	20	20	20	20	20	20	20	20

Table 3 continued

		μ_{LFR}								
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
I-AManPG	NMI	0.9944	0.9946	0.9946	0.9945	0.9945	0.9927	0.9620	0.4541	0.1322
	AMI	0.9881	0.9885	0.9885	0.9881	0.9883	0.9863	**0.9547**	0.4056	0.0566
	Mod	0.9303	0.8338	0.7370	0.6380	0.5402	0.4422	0.3384	0.1731	0.1107
	purity	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9985	0.9671	0.5498	0.1751
	time	1.6041	1.2957	1.4677	1.1984	2.4890	1.3366	1.0639	1.9795	2.0597
	q	21	21	21	21	21	21	21	21	21
I-AManPG	NMI	0.9890	0.9890	0.9894	0.9890	0.9887	0.9868	**0.9649**	**0.4649**	0.1352
	AMI	0.9764	0.9765	0.9773	0.9765	0.9763	0.9742	0.9500	**0.4159**	0.0574
	Mod	0.9124	0.8161	0.7239	0.6263	0.5315	0.4349	0.3358	0.1728	**0.1123**
	purity	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9996	0.9981	0.9831	0.5642	0.1770
	time	1.7685	1.6543	1.4107	0.8671	1.7232	1.4291	0.9652	2.4757	2.5771
	q	22	22	22	22	22	22	22	22	22
I-AManPG	NMI	0.9835	0.9838	0.9842	0.9834	0.9836	0.9805	0.9532	0.4622	**0.1402**
	AMI	0.9649	0.9654	0.9663	0.9647	0.9655	0.9618	0.9317	0.4105	**0.0606**
	Mod	0.8937	0.8009	0.7114	0.6148	0.5233	0.4278	0.3288	0.1710	0.1120
	purity	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9997	0.9975	**0.9784**	0.5662	0.1798
	time	1.9044	1.7584	1.6944	1.2244	1.8282	2.0833	1.3244	2.9445	2.5133
	q	23	23	23	23	23	23	23	23	23

Table 3 shows that for any particular μ_{LFR}, as the value of the input parameter q moves away from $q_{true} = 20$ the modularity decreases, while the NMI and AMI achieved I-AManPG move away from desirable values close to 1. This does not mean that the partitions are not good relative to the ground-truth partition. For $\mu_{\text{LFR}} = 0$ and 0.1 the community partitions for $q = 17, 18, 19, 20, 21, 22$ are perfectly nested, i.e., the extra communities of partition of $q+1$ are refinements of the partition of q by splitting without crossing the ideal community boundaries. Since q_{true} is in this set, this says that for values of μ_{LFR} that imply strong community structure I-AManPG produces communities that respect the affinities of the ground-truth partition. For these two values of μ_{LFR}, when $q = 10$, the farthest from q_{true} in the set, the I-AManPG partition is perfectly nested relative to the next partition, i.e., $q = 17$, and the ground-truth partition. When μ_{LFR} has values from 0.2 to 0.6, each partitioning for $q = 10, 17, 18, 19, 21, 22, 23$ is well-nested with partitioning of q_{true} and the associated purity values are very close to 1. For $\mu_{\text{LFR}} = 0.7$ and 0.8, the community structure is not strong in the ground-truth LFR networks and therefore purity would be expected to degrade. These results provide promising evidence for the possibility of development of a dynamic adaptation strategy for I-AManPG. Since the sparsification to project a Stiefel element to an assignment applies, in general, to a dense $N \times q$ matrix, storage and computation can become excessive when a large number of communities must be produced. Effective nesting means this can be avoided efficiently as is done with other divisive projection-based algorithms.

5.3 Normalized Cut

Normalized cut has been widely used for image segmentation. Its optimization formulation is given by

$$\min_{X \in \mathcal{A}_v} f_{\text{NC}}(X) = -\text{trace}(X^T D^{-1/2} W D^{-1/2} X), \tag{5.2}$$

which has been described in the context of (1.6). This problem assumes graph-based data that is represented by an appropriate matrix characterizing the relationships between the basic data elements from the application problem. In the case of gray image segmentation, the matrix $W \in \mathbb{R}^{mn \times mn}$ is an affinity matrix of an m by n pixels gray image, $D \in \mathbb{R}^{mn \times mn}$ is a diagonal matrix with $D_{ii} = \sum_{j=1}^{mn} W_{ij}$, and $v = \text{diag}(D^{1/2})$. Here, we use the approach in [68] to choose W and D.

Problem (5.2) can be optimized by the weighted kernel k-means algorithm, see e.g., [24, Algorithm 1]. Note that Problem (5.2) has many low-quality local minimizers and descent optimization algorithms usually are not able to escape from them. Thus, initialization plays an important role in finding an acceptable solution. Let U be the $n \times q$ matrix of the q leading eigenvector of the matrix $D^{-1/2} W D^{-1/2}$. If X is only required to be orthonormal, then U is a global minimizer of (5.2). Since U is unlikely to be in $\mathcal{A}_v$, one approach is to find a matrix in $\mathcal{A}_v$ that is close to U. Different notions of closeness yield different methods. Next, we introduce four initialization methods, including the proposed one based on AManPG.

First, Bach and Jordan [4] seek to find a matrix $Y \in \mathcal{A}_v$ that minimizes

$$\|UU^T - YY^T\|_F. \tag{5.3}$$

In other words, the difference between U and Y is measured by the orthogonal projection matrix. The weighted kernel k-means is suggested to solve (5.3) see [4, Figure 1]. However, similar to (5.2), the kernel k-means for (5.3) may also get stuck in a local minimizer. We use k-means++ in Matlab for the initialization of the kernel k-means for (5.3).

Second, Shi and Malik [67] propose to find an indicator matrix that is closest to U up to a rotation. Specifically, let $\tilde{U}$ denote the matrix formed by normalizing all rows of U. The task is to find an indicator matrix Z and a q-by-q orthonormal matrix Q that minimize

$$\|Z - \tilde{U}Q\|_F.$$

Shi and Malik [67] use an alternating minimization algorithm to find Z and Q. Note that this approach neither guarantees to find the global optimum nor uses the weight vector v. Therefore, this approach may not find a satisfactory solution. Here, we use the C and Matlab hybrid implementation from [68].

Third, Karypis and Kumar [48] developed METIS, a fast, multi-level graph partitioning algorithm that produces equally-sized clusters. It is shown to be an effective method for the kernel k-means initialization. Note that METIS does not aim to minimize the objective (5.2). We use the C implementation from http://glaros.dtc.umn.edu/gkhome/metis/metis/download with the Matlab interface from https://github.com/dgleich/metismex.

Fourth, we propose to initialize the weighted kernel k-means algorithm by I-AManPG. Specifically, Problem (5.2) can be reformulated as

$$\min_{X \in \mathcal{F}_v} -\text{trace}(X^T D^{-1/2} W D^{-1/2} X) + \lambda \|X\|_1, \tag{5.4}$$

which can be optimized by I-AManPG. We further propose to gradually increase λ rather than choosing a fixed value of λ since increasing λ tends to give better solutions in our experiments.[4] The clusters are specified by $P_{\mathcal{B}_v}(X_*)$, as described in Sect. 5.1. Such clusters are then used as initializations for the weighted kernel k-means algorithm.

The four initialization methods are denoted, respectively, by BJ, SM, ME, and AM. Their combinations with the weighted kernel k-means algorithms are denoted, respectively, by BJ-k, SM-k, ME-k, and AM-k. The implementation of the weighted kernel k-means algorithm is modified from [17].[5] The test images are from [68] and the built-in images in Matlab. We further resize them to have 160-by-160 pixels as shown in Fig. 1.

An average of the negative function values $-f_{\text{NC}}$ of 10 random runs are reported in Fig. 2. We do not report the computational time since the implementations of these methods use different languages and their computational time cannot be rigorously compared. The qualities of these methods are compared based on the objective function value f_{NC}. As shown in the figure, METIS initializations are not preferred since they do not aim to minimize f_{NC}. Though SM, SM-k, BJ, BJ-k are competitive to AM and AM-k in many cases, they do not perform well in certain images, such as ME and ME-k for the football image with 3 clusters, and BJ and BJ-k for the tape image with 3 clusters. AManPG-based methods are clearly most robust in the sense of minimizing the function f_{NC} over $\mathcal{A}_v$. The values of $-f_{\text{NC}}$ by AM-k are often the highest ones. Even if they are not, they are still close to the highest ones. The empirical evidence supports the expectations given in the motivation discussion above that I-AManPG is competitive with or superior to initialization strategies in the current literature.

6 Conclusions and Future Work

We proposed an optimization model for clustering problems. The domain $\mathcal{F}_v$ was proven to be an embedded submanifold and its geometry structures were derived. An inexact Riemannian

[4] The λ in I-AManPG increases by 0.01, 0.04, and 0.2.

[5] The implementation in [17] is for unweighted kernel k-means. We modified it for weighted kernel k-means.

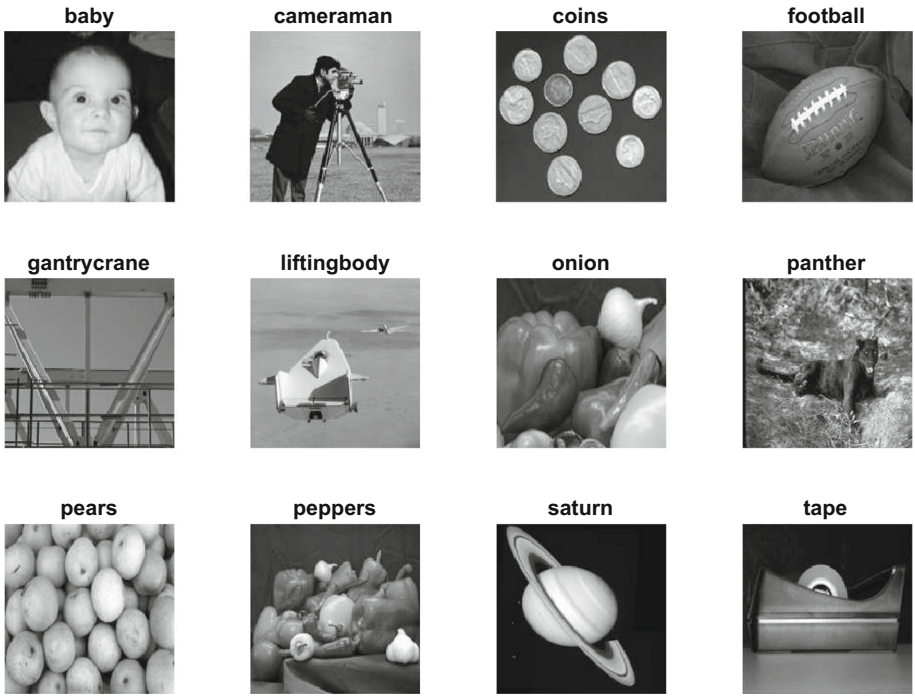

Fig. 1 The test images

proximal gradient method was proposed and its global convergence proved. It was shown empirically that the proposed optimization model was more effective than the state-of-the-art methods in community detection and normalized cut for image segmentation.

Future work will address a more comprehensive analysis of the choice of the parameters λ and q. The current method requires an estimation of the number of clusters q. A critical future task is to develop a strategy to dynamically update the number of clusters thereby enabling more efficient computation for problems with a large number of communities.

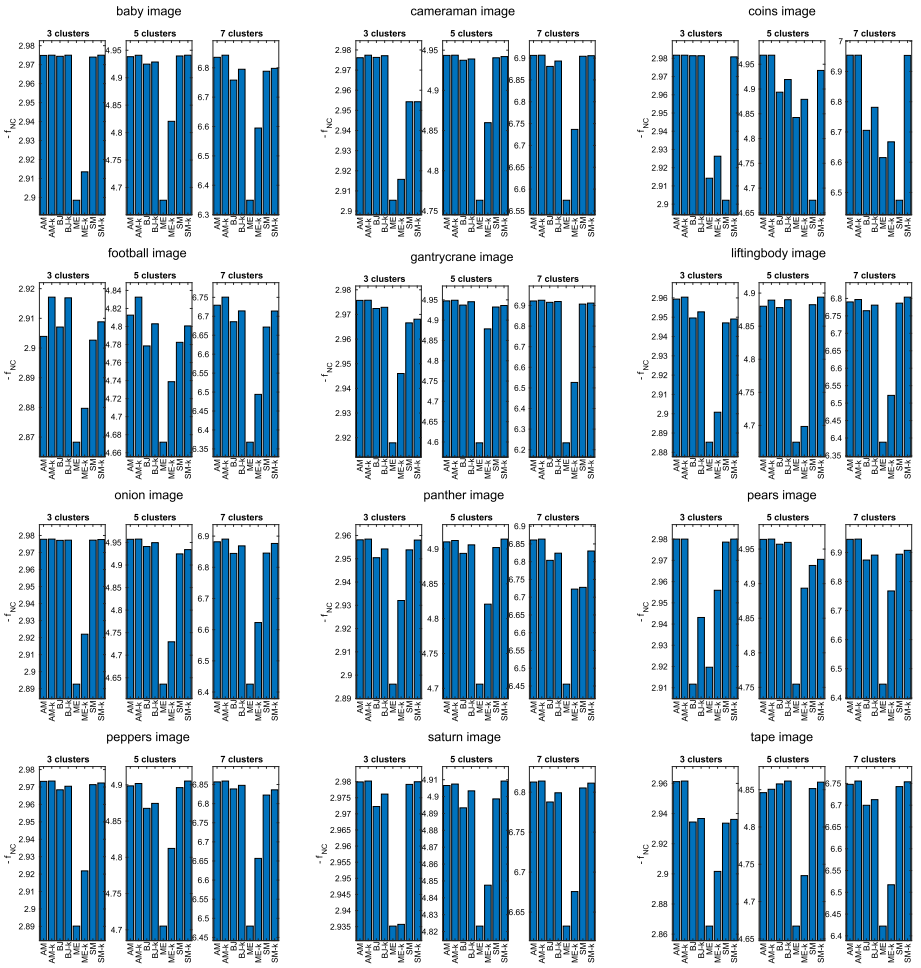

Fig. 2 An average of 10 random runs is reported. y-axis represents the function values. Multiple numbers of clusters are tested

Acknowledgements Wen Huang was partially supported by the National Natural Science Foundation of China (No. 12001455 and No. 12371311), the National Natural Science Foundation of Fujian Province (No. 2023J06004), the Fundamental Research Funds for the Central Universities (No. 20720240151), Xiaomi Young Talents Program. Kyle A. Gallivan was partially supported by National Science Foundation Grant CIBR1934157.

Funding This work was supported by the National Natural Science Foundation of China (No. 12371311), the Natural Science Foundation of Fujian Province (No. 2023J06004), the Fundamental Research Funds for the Central Universities (No. 20720240151), and Xiaomi Young Talents Program.

Data Availability The datasets generated during and analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Absil, P.-A., Baker, C.G., Gallivan, K.A.: Trust-region methods on Riemannian manifolds. Found. Comput. Math. **7**(3), 303–330 (2007)
2. Absil, P.-A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2008)
3. Absil, P.-A., Malick, J.: Projection-like retractions on matrix manifolds. SIAM J. Optim. **22**(1), 135–158 (2012)
4. Bach, F.R., Jordan M.I. Learning spectral clustering. In: Proceedings of the 16th International Conference on Neural Information Processing Systems, NIPS'03, pp. 305–312, Cambridge, MA, US. MIT Press (2003)
5. Baker, C.G.: Riemannian manifold trust-region methods with applications to eigenproblems. PhD thesis, Florida State University, School of Computational Science (2008)
6. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009). https://doi.org/10.1137/080716542
7. Beck, A.: First-Order Methods in Optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA (2017)
8. Bento, G.C., Ferreira, O.P., Melo, J.G.: Iteration-complexity of gradient, subgradient and proximal point methods on Riemannian manifolds. J. Optim. Theory Appl. **173**(2), 548–562 (2017)
9. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theory Exp. **2008**(10), P10008 (2008)
10. Boothby, W.M.: An Introduction to Differentiable Manifolds and Riemannian Geometry, 2nd edn. Academic Press, Cambridge (1986)
11. Boumal, N.: An introduction to optimization on smooth manifolds (2022)
12. Boumal, N., Absil, P.A., Cartis, C.: Global rates of convergence for nonconvex optimization on manifolds. IMA J. Numer. Anal. **39**(1), 1–33 (2018)
13. Bounova, G.: Matlab Tools for Network Analysis (2009). http://strategic.mit.edu/downloads.php?page=matlab_networks
14. Boutsidis, C., Drineas, P., Mahoney, M.W.: Unsupervised feature selection for the k-means clustering problem. In: Advances in Neural Information Processing Systems, pp. 153–161 (2009)
15. Carson, T., Mixon, D.G., Villar, S.: Manifold optimization for k-means clustering. In: 2017 International Conference on Sampling Theory and Applications (SampTA), pp. 73–77. IEEE (2017)
16. Chan, P.K., Schlag, F.: Spectral k-way ratio-cut partitioning and clustering. IEEE Trans. Comput. Aided Design Integr. Circuits Syst. **13**(9), 1088–1096 (1994)
17. Chen, M.: Pattern recognition and machine learning toolbox (2021)
18. Chen, S., Ma, S., So, A.M.-C., Zhang, T.: Proximal gradient method for nonsmooth optimization over the Stiefel manifold. SIAM J. Optim. **30**(1), 210–239 (2020)
19. Cherian, A., Sra, S.: Riemannian dictionary learning and sparse coding for positive definite matrices. IEEE Trans. Neural Netw. Learn. Syst. **28**(12), 2859–2871 (2017)
20. Clarke, F.H.: Optimization and nonsmooth analysis. Classics in Applied Mathematics of SIAM (1990)
21. Danon, L., Diaz-Guilera, A., Arenas, A.: The effect of size heterogeneity on community identification in complex networks. J. Stat. Mech. Theory Exp. **2006**(11), P11010 (2006)
22. Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. J. Stat. Mech. Theory Exp. **2005**(09), P09008 (2005)
23. Dhillon, I.S., Guan, Y., Kulis, B.: Kernel k-means, spectral clustering and normalized cuts. In: Proceedings of the 10th Association for Computing Machinery(ACM) Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) International Conference on Knowledge Discovery and Data Mining (2004)
24. Dhillon, I., Guan, Y., Kulis, B.: A unified view of kernel k-means. spectral clustering and graph cuts. Technical report, Department of Computer Sciences, University of Texas at Austin (2005)
25. do Carmo, M.P.: Riemannian Geometry. Mathematics: Theory & Applications (1992)
26. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. **20**(2), 303–353 (1998). https://doi.org/10.1137/S0895479895290954
27. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3–5), 75–174 (2010)
28. Golub, G.H., Van Loan, C.F.: Matrix computations, 3rd edn. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press (1996)
29. Grohs, P., Hosseini, S.: ϵ-subgradient algorithms for locally lipschitz functions on Riemannian manifolds. Adv. Comput. Math. (2015). https://doi.org/10.1007/s10444-015-9426-z
30. Grohs, P., Hosseini, S.: Nonsmooth trust region algorithms for locally Lipschitz functions on Riemannian manifolds. IMA J. Numer. Anal. (2015). https://doi.org/10.1093/imanum/drv043

31. Harris, J.: Algebraic Geometry: A First Course, vol. 133. Springer Science & Business Media, Berlin (1992)
32. Hartigan, J.A., Wong, M.A.: A K-means clustering algorithm. Appl. Stat. **28**(1), 100–108 (1979)
33. Hosseini, S., Huang, W., Yousefpour, R.: Line search algorithms for locally Lipschitz functions on Riemannian manifolds. SIAM J. Optim. **28**(1), 596–619 (2018)
34. Jiang, H., Jiang, B., Lin, L., Wen, Z., Yuan, Y.: Structured quasi-newton methods for optimization with orthogonality constraints. SIAM J. Sci. Comput. **41**(4), A2239–A2269 (2019)
35. Huang, W.: Optimization algorithms on Riemannian manifolds with applications. PhD thesis, Florida State University, Department of Mathematics (2013)
36. Huang, W., Absil, P.-A., Gallivan, K.A.: A Riemannian symmetric rank-one trust-region method. Math. Program. **150**(2), 179–216 (2015)
37. Huang, W., Absil, P.-A., Gallivan, K.A.: Intrinsic representation of tangent vectors and vector transport on matrix manifolds. Numer. Math. **136**(2), 523–543 (2017)
38. Huang, W., Gallivan, K.A.: A limited-memory Riemannian symmetric rank-one trust-region method with a restart strategy. J. Sci. Comput. **93**(1), 1 (2022)
39. Huang, W., Gallivan, K.A., Absil, P.-A.: A Broyden class of quasi-Newton methods for Riemannian optimization. SIAM J. Optim. **25**(3), 1660–1685 (2015)
40. Huang, W., Gallivan, K.A., Srivastava, A., Absil, P.-A.: Riemannian optimization for elastic shape analysis. In: Proceedings of the 21st Internaltional Symposium on Mathematical Theory of Networks and Systems (MTNS 2014) (2014)
41. Huang, W., Wei, K.: Riemannian proximal gradient methods. Mathematical Programming (2021). https://doi.org/10.1007/s10107-021-01632-3
42. Huang, W., Absil, P.A., Gallivan, K.A.: A Riemannian BFGS method without differentiated retraction for nonconvex optimization problems. SIAM J. Optim. **28**(1), 470–495 (2018)
43. Huang, W., Hand, P.: Blind deconvolution by a steepest descent algorithm on a quotient manifold. SIAM J. Imaging Sci. **11**(4), 2757–2785 (2018)
44. Huang, W., Wei, K.: An Inexact Riemannian Proximal Gradient Method (2021)
45. Huang, W., Wei, K.: An extension of fast iterative shrinkage-thresholding algorithm to Riemannian optimization for sparse principal component analysis. Numer. Linear Algebra Appl. **29**, e2409 (2021)
46. Jiang, B., Meng, X., Wen, Z., Chen, X.: An exact penalty approach for optimization with nonnegative orthogonality constraints (2019). CoRR, ArXiv:1907.12424
47. Jolliffe, I.T., Trendafilov, N.T., Uddin, M.: A modified principal component technique based on the Lasso. J. Comput. Graph. Stat. **12**(3), 531–547 (2003)
48. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. **20**(1), 359–392 (1998)
49. Kehagias, A.: Community Detection Toolbox (2021). https://www.mathworks.com/matlabcentral/fileexchange/45867-community-detection-toolbox
50. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E **78**(4), 046110 (2008)
51. Li, X., Sun, D., Toh, K.-C.: A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems. SIAM J. Optim. **28**(1), 433–458 (2018)
52. Macqueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability (1967)
53. Manning, C.D.: Prabhakar Raghavan. utze, introduction to information retrieval (2008)
54. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Phys. Rev. E **69**(6), 066133 (2004)
55. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E **74**(3), 036104 (2006)
56. Newman, M.E.J.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. **103**(23), 8577–8582 (2006)
57. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004)
58. Newman, M.E.J., Leicht, E.A.: Mixture models and exploratory analysis in networks. Proc. Natl. Acad. Sci. **104**(23), 9564–9569 (2007)
59. Qian, Y., Pan, S., Xiao, L.: Exact penalty methods for minimizing a smooth function over the nonnegative orthogonal set (2021)
60. Ring, W., Wirth, B.: Optimization methods on Riemannian manifolds and their application to shape space. SIAM J. Optim. **22**(2), 596–627 (2012). https://doi.org/10.1137/11082885X
61. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. Proc. Natl. Acad. Sci. **105**(4), 1118–1123 (2008)

62. Samir, C., Huang, W.: Coordinate descent optimization for one-to-one correspondence and supervised classification of 3D shapes. Appl. Math. Comput. **388**, 125539 (2021)
63. Sato, H.: A Dai–Yuan-type Riemannian conjugate gradient method with the weak Wolfe conditions. Comput. Optim. Appl. **64**(1), 101–118 (2016)
64. Sato, H., Iwai, T.: A Riemannian optimization approach to the matrix singular value decomposition. SIAM J. Optim. **23**(1), 188–212 (2013)
65. Sato, H., Iwai, T.: A new, globally convergent Riemannian conjugate gradient method. Optimization **64**(4), 1011–1031 (2015)
66. Scherrer, A.: Matlab Version for Louvain's Algorithm (2008). https://perso.uclouvain.be/vincent.blondel/research/louvain.html
67. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
68. Shi, J., Cour, T., Yu, S.: Normalized Cut Segmentation Code (2004)
69. Turaga, P., Veeraraghavan, A., Srivastava, A., Chellappa, R.: Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. IEEE Trans. Pattern Anal. Mach. Intell. **33**(11), 2273–86 (2011). https://doi.org/10.1109/TPAMI.2011.52
70. Vandereycken, B.: Low-rank matrix completion by Riemannian optimization–extended version. SIAM J. Optim. **23**(2), 1214–1236 (2013)
71. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J. Mach. Learn. Res. **11**, 2837–2854 (2010)
72. Wei, K., Cai, J.-F., Chan, T.F., Leung, S.: Guarantees of Riemannian Optimization for Low Rank Matrix Completion (1) (2016)
73. Wei, M., Huang, W., Gallivan, K.A., Van Dooren, P.: community detection by a Riemannian projected proximal gradient method. In: Proceedings of the 24th Internaltional Symposium on Mathematical Theory of Networks and Systems (2021)
74. Wen, Z., Yin, W.: A feasible method for optimization with orthogonality constraints. Mathematical Programming (2012). https://doi.org/10.1007/s10107-012-0584-1
75. Xiao, G., Bai, Z.-J., Ching, W.-K.: A columnwise update algorithm for sparse stochastic matrix factorization. SIAM J. Matrix Anal. Appl. **43**(4), 1712–1735 (2022)
76. Xiao, X., Li, Y., Wen, Z., Zhang, L.: A regularized semi-smooth newton method with projection steps for composite convex programs. J. Sci. Comput. **76**(1), 364–389 (2018)
77. Yang, L., Cao, X., He, D., Wang, C., Wang, X., Zhang, W.: Modularity based community detection with deep learning. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), vol. 16, pp. 2252–2258 (2016)
78. Ye, J., Zhao, Z., Wu, M.: Discriminative k-means for clustering. In: Lyu, H., Sha, N., Qin, S., Yan, M., Xie, Y., Wang, R. (eds.) Advances in Neural Information Processing Systems, vol. 20. Curran Associates, Inc., New York (2008)
79. Ye, K., Lim, L.-H.: Schubert varieties and distances between subspaces of different dimensions. SIAM J. Matrix Anal. Appl. **37**(3), 1176–1197 (2016)
80. Yu, S.X., Shi, J.: Multiclass spectral clustering. In: Proceedings Ninth IEEE International Conference on Computer Vision, vol. 1, pp. 313–319 (2003)
81. Zhang, H., Sra, S.: First-order methods for geodesically convex optimization. In: Conference on Learning Theory (2016)
82. Zhu, X.: A Riemannian conjugate gradient method for optimization on the Stiefel manifold. Comput. Optim. Appl. **67**(1), 73–110 (2017)