Community Detection by a Riemannian Projected Proximal Gradient Method ⋆

Meng Wei * **Wen Huang** **,1 **Kyle A. Gallivan** *
Paul Van Dooren ***

* Department of Mathematics, Florida State University, 208 Love
Building, 1017 Academic Way, Tallahassee, FL 32306-4510, USA
(e-mail: mwei@math.fsu.edu, gallivan@math.fsu.edu)
** School of Mathematical Sciences, Xiamen University, Xiamen,
Fujian, P.R.China, 361005 (e-mail: wen.huang@xmu.edu.cn)
*** Institute of Information and Communication Technologies,
Electronics, and Applied Mathematics (ICTEAM), Catholic University
of Louvain (UCL), 4 Av. G. Lemaitre, 1348 Louvain-la-Neuve,
Belgium   (e-mail: paul.vandooren@uclouvain.be)

Abstract: Community detection plays an important role in understanding and exploiting the structure of complex systems. Many algorithms have been developed for community detection using modularity maximization or other techniques. In this paper, we formulate the community detection problem as a constrained nonsmooth optimization problem on the compact Stiefel manifold. A Riemannian projected proximal gradient method is proposed and used to solve the problem. Numerical experimental results on synthetic benchmarks and real-world networks show that our algorithm is effective and outperforms several state-of-art algorithms.

Keywords: Community Detection, Modularity Matrix, Riemannian Optimization, Projected Proximal Gradient

1. INTRODUCTION

Describing and analyzing complex systems in mathematical models is a challenging problem. Networks are a natural representation for many kinds of complex systems, where networks are sets of nodes or vertices joined together in pairs by links or edges. There are several types of networks. For example, Facebook is a large social network, where more than one billion people are connected via virtual acquaintanceship. Another common example is the internet, the physical network of computers, routers, and modems which are linked via cables or wireless signals. Many other examples come from biology, physics, engineering, computer science, ecology, economics, marketing, etc.

Real-world networked 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 (Newman and Girvan, 2004). These groups are called communities, or modules. Detecting community structure in a network is a powerful tool for understanding and exploiting the structure of networks, and it has various practical applications (Girvan and Newman, 2002).

A variety of community detection algorithms have been developed in recent years, such as the GN algorithm (Newman, 2004), the spectral modularity maximization algorithm (Newman, 2006b), the Louvain method (Blondel et al., 2008), the Infomap algorithm (Rosvall and Bergstrom, 2008), statistical inference (Newman and Leicht, 2007), deep learning (Yang et al., 2016). Modularity optimization approaches have been shown to be highly effective in practical applications (Fortunato, 2010). Recently, optimization over Riemannian manifolds has drawn much attention because of its application in many different fields. Almost all of the manifold optimization methods require computing the derivatives of the objective function and do not apply to the case where the objective function is nonsmooth. In (Chen et al., 2020), the authors proposed a Riemannian proximal gradient method called ManPG for a class of nonsmooth nonconvex optimization problems over a Stiefel manifold

$$\min F(X) := f(X) + g(X), \tag{1}$$
$$s.t.\ X \in \mathcal{M} := St(q, n) = \{X : X \in \mathbb{R}^{n \times q}, X^T X = I_q\},$$

where I_q denotes the $q \times q$ identity matrix ($q < n$), f is smooth, possibly nonconvex, and its gradient ∇f is Lipschitz continuous, g is convex, possibly nonsmooth, and is Lipschitz continuous and the proximal mapping of g is easy to find.

In (Huang and Wei, 2019), the authors extended the fast iterative shrinkage-thresholding (FISTA) algorithm (Beck

and Teboulle, 2009) to solve (1), and the accelerated Riemannian manifold proximal gradient algorithm performed better than ManPG. In (Huang and Wei, 2021), they developed and analyzed a generalization of the proximal gradient methods with and without acceleration for nonsmooth Riemannian optimization problems.

In this paper, we propose the accelerated Riemannian manifold projected proximal gradient (ARPPG) method for community detection, and we solve the community detection problem using a constrained nonsmooth optimization problem over a Stiefel manifold.

The paper is organized as follows. In Section 2, we define assignment matrices and show that the ideal graph assignment is a global maximal solution of the modularity function. In Section 3, we show the connection between the modularity matrix and the Stiefel manifold, and then transform the community detection problem to the constrained Stiefel optimization problem. Because the constraint defines a feasible set that is a subset of the Stiefel manifold, we must apply a projection to the proximal result. This leads to the accelerated Riemannian manifold projected proximal gradient (ARPPG) algorithm. Extensive numerical experiments on synthetic and real world networks are described in Section 4. Finally, conclusions and future work are stated in Section 5.

2. DERIVATION OF GLOBAL MAXIMUM OVER ASSIGNMENT MATRICES

2.1 Assignment matrices

We will denote a q dimensional vector with all entries being 1 by $\mathbf{1}_q$ and denote the $q \times q$ permutation matrices by P_q.

A matrix in the set of assignment matrices, $\mathcal{A}_{n,q}$, is defined as

Definition 1. The matrix $X \in \{0,1\}^{n \times q}$, with $n \geq q$, is an assignment matrix if it satisfies

  (i) $X\mathbf{1}_q = \mathbf{1}_n$,
  (ii) $X^T X = \mathbf{diag}(n_1, \cdots, n_q)$ where $n_i = ||Xe_i||_1$.

X is said to be in canonical ordering if the rows are permuted so that

$$X = \begin{pmatrix} \mathbf{1}_{n_1} & & & \\ & \mathbf{1}_{n_2} & & \\ & & \ddots & \\ & & & \mathbf{1}_{n_q} \end{pmatrix}.$$

Of course, the column ordering is not unique for the canonical form, i.e., XP_q is the same community assignment but with a different correspondence between the sets and the columns of the assignment matrix. For essential uniqueness, the additional constraint of $n_1 \geq n_2 \geq \ldots \geq n_q$ can be imposed. The columns are orthogonal, but not orthonormal, and X has exactly n nonzero elements all of which have the value of 1. As a result, X defines a partitioning of the indices $1, \ldots, n$ into q disjoint sets.

2.2 The Modularity Cost Function

From (Newman, 2006a), the scalar cost function $f(X)$ called modularity (up to a scalar $\frac{1}{2m}$) can be written as

a quadratic function over $n \times q$ matrices defined by the matrix

$$M = A - \frac{A\mathbf{1}_n\mathbf{1}_n^T A}{2m}, \quad f(X) = \text{trace}(X^T M X),$$

where A is the adjacency matrix of the graph, M is the modularity matrix, m is the number of edges, n is the number of vertices in the graph and the total degree of the graph is $2m = \mathbf{1}_n^T A\mathbf{1}_n$.

The value of $f(X)$ is invariant under permutations on the columns of the assignment matrix X, i.e., $f(XP_q) = \text{trace}(P_q^T X^T M X P_q)$. So there are multiple optimal ways of specifying the same community assignment.

2.3 Maximal of the Modularity Function on Ideal Graphs

We consider in this section so-called ideal graphs. An ideal graph is a graph where the communities are cliques and there are no edges between the cliques.

When A is an ideal graph with q communities we know it can be written (Marchand, 2017)

$$A = \tilde{Z}_* \tilde{Z}_*^T$$

where $\tilde{Z}_* \in \mathcal{A}_{n,q}$ is not necessarily in canonical form and there exists a row permutation P so that

$$PAP^T = A_P = Z_* Z_*^T$$

where A_P is block diagonal with diagonal blocks $\mathbf{1}_{n_i}\mathbf{1}_{n_i}^T = z_i z_i^T$ for $1 \leq i \leq q$ and

$$Z_* = \begin{pmatrix} \mathbf{1}_{n_1} & & & \\ & \mathbf{1}_{n_2} & & \\ & & \ddots & \\ & & & \mathbf{1}_{n_q} \end{pmatrix} = \begin{pmatrix} z_1 & & & \\ & z_2 & & \\ & & \ddots & \\ & & & z_q \end{pmatrix}$$

is in canonical form.

The corresponding modularity matrices for an ideal A and the corresponding block diagonal A_P are given by

$$\begin{aligned} M = M^T &= A - \frac{A\mathbf{1}_n\mathbf{1}_n^T A}{2m} \\ &= \tilde{Z}_* \tilde{Z}_*^T - \frac{\tilde{Z}_* \tilde{Z}_*^T \mathbf{1}_n \mathbf{1}_n^T \tilde{Z}_* \tilde{Z}_*^T}{2m} \\ &= \tilde{Z}_*(I_q - \frac{\tilde{s}\tilde{s}^T}{2m})\tilde{Z}_*^T, \end{aligned}$$

and

$$M_P = Z_*(I_q - \frac{ss^T}{2m})Z_*^T,$$

where $s = Z_*^T\mathbf{1}_n = (n_1 \ \ldots \ n_q)^T$, and $2m = \mathbf{1}_n^T A_P\mathbf{1}_n = s^T s = \sum_{i=1}^q n_i^2$.

The cost function $f(X)$ is invariant under reorderings of A, so we can analyze any row ordering of Z_* denoted generically as Z below. The following result for the value of $f(Z)$, i.e., the cost function at the assignment matrix that generates the ideal matrix A, follows directly from the definitions.

Lemma 2. If $A = ZZ^T$ for $Z \in \mathcal{A}_{n,q}$ then

$$f(Z) = \sum_{i=1}^q n_i^2 - \frac{\sum_{i=1}^q n_i^4}{\sum_{i=1}^q n_i^2}.$$

We show that the value $f(X)$ for any $X \in \mathcal{A}_{n,q}$ is bounded above by $f(Z)$ in Theorem 5. The following lemmas are easily proven and are useful in proving the main result.

Lemma 3. If $A = ZZ^T$ for $Z \in \mathcal{A}_{n,q}$ then for any $X \in \mathcal{A}_{n,q}$

$$f(X) = \text{trace}(X^T Z(I_q - \frac{ss^T}{2m})Z^T X) \leq \sum_{i=1}^{q} \gamma_i v_i^T X X^T v_i,$$

where $s = (n_1 \ \dots \ n_q)^T$, $v_i = Ze_i$,

$$2m = \sum_{i=1}^{q} n_i^2, \ \gamma_i := 1 - \frac{n_i^2}{2m},$$

where $0 \leq \gamma_i < 1$.

Lemma 4. Given $Z \in \mathcal{A}_{n,q}$, any $X \in \mathcal{A}_{n,q}$ satisfies

$$v_i^T X X^T v_i \leq v_i^T Z Z^T v_i, \quad 1 \leq i \leq q$$

where $v_i = Ze_i$. Equality holds only when $X = ZP_q$, i.e., a column permutation of Z.

The desired result is stated as Theorem 5.

Theorem 5. If $A = ZZ^T$ for $Z \in \mathcal{A}_{n,q}$ is an ideal adjacency matrix then for any $X \in \mathcal{A}_{n,q}$

$$f(X) \leq f(Z),$$

where $f(X) = \text{trace}(X^T Z(I_q - \frac{ss^T}{2m})Z^T X)$, $s = (n_1 \ \dots \ n_q)^T$, $2m = \sum_{i=1}^{q} n_i^2$.

Proof. The series of lemmas above yields

$$f(X) \leq \sum_{i=1}^{q} \left(1 - \frac{n_i^2}{2m}\right) v_i^T Z Z^T v_i.$$

Note that

$$\sum_{i=1}^{q} \left(1 - \frac{n_i^2}{2m}\right) v_i^T Z Z^T v_i$$
$$= \sum_{i=1}^{q} v_i^T Z Z^T v_i - \sum_{i=1}^{q} \frac{n_i^2}{2m} v_i^T Z Z^T v_i$$
$$= \sum_{i=1}^{q} n_i^2 - \frac{\sum_{i=1}^{q} n_i^4}{2m}$$
$$= \sum_{i=1}^{q} n_i^2 - \frac{\sum_{i=1}^{q} n_i^4}{\sum_{i=1}^{q} n_i^2} = f(Z).$$

Theorem 5 shows that the ideal graph assignment is a global maximum of the modularity function over $\mathcal{A}_{n,q}$.

3. STIEFEL MANIFOLD ALGORITHMS FOR COMMUNITY DETECTION

The algorithms discussed here assume the cost function

$$f(X) = \text{trace}(X^T M X),$$
$$where \ \ M = A - \frac{A\mathbf{1}_n \mathbf{1}_n^T A}{\mathbf{1}_n^T A \mathbf{1}_n}.$$

3.1 The connection between the modularity matrix and the Stiefel manifold $St(q, n)$

Lemma 6. Let $Z \in \mathcal{A}_{n,q}$ and define $M = A - \frac{A\mathbf{1}\mathbf{1}^T A}{\mathbf{1}^T A\mathbf{1}}$. If A is the adjacency matrix of an ideal graph, then $A = ZZ^T$ and

$$\mathcal{R}(A) = \mathcal{R}(Z) = (\mathcal{R}(M) \oplus^\perp \mathcal{R}(\mathbf{1}_n)), \quad (2)$$

$$\mathcal{N}(M) = (\mathcal{N}(Z^T) \oplus^\perp \mathcal{R}(\mathbf{1}_n)) = (\mathcal{N}(A) \oplus^\perp \mathcal{R}(\mathbf{1}_n)), \quad (3)$$

where $\mathcal{R}(A)$ denotes the range of A, $\mathcal{N}(A)$ denotes the null space of A, and $\oplus^\perp$ denotes the direct sum of two perpendicular spaces.

Proof. Note that the symmetry of A and M implies $\mathcal{N}(M) = \mathcal{R}(M)^\perp$ and $\mathcal{N}(A) = \mathcal{R}(A)^\perp$. Therefore (2) and (3) are equivalent. It follows from the definition of M that

$$M = Z(I_q - \frac{ss^T}{s^T s})Z^T = Z(I_q - \frac{ss^T}{s^T s})^2 Z^T$$
$$= [Z(I_q - \frac{ss^T}{s^T s})][(I_q - \frac{ss^T}{s^T s})Z^T].$$

This implies that

$$\mathcal{R}(M) = \mathcal{R}(Z(I_q - \frac{ss^T}{s^T s})), \quad \mathcal{N}(M) = \mathcal{N}((I_q - \frac{ss^T}{s^T s})Z^T),$$

and we also have

$$\mathcal{R}(A) = \mathcal{R}(Z), \quad \mathcal{N}(A) = \mathcal{N}(Z^T).$$

Since the projector $(I_q - \frac{ss^T}{s^T s})$ has rank $n-1$, it follows that the ranges and null spaces of A and M have dimensions that can only differ by 1 at most. Now consider the vector $\mathbf{1}_n$. Since $s = Z^T \mathbf{1}_n$, we have $M\mathbf{1}_n = 0$ and since M is symmetric, $\mathbf{1}_n$ is orthogonal to $\mathcal{R}(M)$. Since $\mathbf{1}_n = Z\mathbf{1}_q$, we have $\mathbf{1}_n \in \mathcal{R}(Z) = \mathcal{R}(A)$. Together, these two properties prove (2), and hence also (3).

Since $rank(M) = q - 1$ and $M = M^T$, we have the eigendecomposition

$$M = X_* \Gamma X_*^T,$$

where

$$X_* \in St(q-1, n), \ \ \Gamma = diag(\gamma_1, \dots, \gamma_{q-1}), \ \ \gamma_i \neq 0.$$

It then follows by Lemma 6 that $\left[X_* \ \frac{\mathbf{1}_n}{\sqrt{n}}\right] \in St(q, n)$ since $\mathcal{R}(X_*)$ is a subspace of $\mathcal{R}(M)$.

3.2 An Important Basis for an Ideal $\mathcal{R}(A)$

For the modularity matrix, the relationship between A and M is one of deflation of range that allows the characterization of the part of $\mathcal{R}(A) = \mathcal{R}(Z)$ that is removed when considering $\mathcal{R}(M)$ as shown in (2).

Therefore, we can now get the anticipated result of $\mathcal{R}(Z) = \mathcal{R}\left(\left[X_* \ \frac{\mathbf{1}_n}{\sqrt{n}}\right]\right)$.

3.3 A Constrained Stiefel Optimization Problem

Multiple Extrema: Note that if a space $\mathcal{B}$ of dimension q has a basis that is an assignment matrix then it has $q!$ such bases all of which are of the form ZP_q where Z is any assignment matrix basis and $P \in \{0, 1\}^{q \times q}$ is a permutation matrix. All of these matrices have exactly n nonzero elements which is the minimum count possible for bases of the space. If the columns of such a matrix, Z, are normalized in Euclidean 2-norm length then an element of $St(q, n)$ is produced with n_i elements in column i all with the value $1/\sqrt{n_i}$ with $\sum_{i=1}^{q} n_i = n$. These are the global minima of

$$\min_{X \in St(q,n), \mathcal{R}(X)=\mathcal{B}} \|X\|_1,$$

where the l_1 norm is defined as $\|X\|_1 = \sum_{ij}\|X_{ij}\|$ imposing the sparsity of X.

In practical numerical computation, even on ideal matrices and certainly on problems for which noise perturbs A and Z from ideal, some projection is needed to take a matrix in $St(q,n)$ to the "nearest" matrix in $\mathcal{A}_{n,q}$.

A Constrained Stiefel Optimization Problem: The constrained Stiefel optimization problem used to perform community detection is

$$X_* = \underset{X \in St(q,n),\, \mathbf{1}_n \in \mathcal{R}(X)}{\operatorname{argmax}} \operatorname{trace}(X^T M X) - \lambda\|X\|_1, \quad (4)$$

where $\lambda > 0$ is a tuning parameter controlling the balance between variance and sparsity. The approach to compute X_* is given in Algorithm 1.

Algorithm 1 Algorithm for the Constrained Stiefel Optimization Problem

1: *Step 1:* Compute $Y_* \in St(q-1,n)$ where

$$Y_* = \underset{X \in St(q-1,n)}{\operatorname{argmax}} \operatorname{trace}(X^T M X).$$

and set the initial guess for Step 2 as

$$X_0 = \left[Y_* \quad \frac{\mathbf{1}_n}{\sqrt{n}}\right].$$

2: *Step 2:* Compute $X_* \in St(q,n), \mathbf{1}_n \in \mathcal{R}(X)$ where

$$X_* = \underset{X \in St(q,n),\, \mathbf{1}_n \in \mathcal{R}(X)}{\operatorname{argmax}} \operatorname{trace}(X^T M X) - \lambda\|X\|_1,$$

with X_0 as the initial guess.

3: *Step 3:* Get the assignment matrix $\hat{X}_*$ by setting the element with the largest magnitude in each row of X_* as 1, and the others as 0 when X_* is sufficiently sparse. Assess the assignment matrix $\hat{X}_*$ and determine whether it is acceptable as a solution to the community detection problem or if the parameter λ should be updated. If λ is updated then return to Step 2.

Step 1 can be computed using any trace maximization algorithm. Our code uses RNewton in ROPTLIB (Huang et al., 2018). A projection is needed to define a Riemannian projected proximal gradient algorithm to solve this problem in Step 2. In fact, this projection can be used for any line search based algorithm where $Y_k = R(\alpha D_k)$ for a Riemannian retraction R must be feasible. The projection used in the proposed algorithm is described below.

In Step 3, we use the idea of continuation to choose the parameter λ that defines the cost function. We can get the optimal X_1^* after setting the initial λ_0 and X_0. We then increase λ_0 and use X_1^* as the initial matrix to get X_2^*. We continue this procedure until the cost function $\operatorname{trace}(X^T M X) - \lambda\|X\|_1$ does not improve anymore.

Step 2 is the main part of the algorithm, and it is inspired by (Huang and Wei, 2019). In (Huang and Wei, 2019), the authors generalized the FISTA from the Euclidean space to the Riemannian setting and considered the general nonconvex optimization problem

$$\min_{X \in \mathcal{M}} F(X) = f(X) + g(X), \quad (5)$$

where $\mathcal{M} \subset \mathbb{R}^{n \times q}$ is a Riemannian submanifold, $f : \mathbb{R}^{n \times q} \to \mathbb{R}$ is L-continuously differentiable (may be non-

convex) and g is continuous and convex but may not be differentiable.

The optimization problem (4) is a special case of problem (5), where $f(X) = \operatorname{trace}(X^T M X)$ is L-continuously differentiable and $g(X) = -\lambda\|X\|_1$ is continuous, convex, but not differentiable. However, there is an essential difference between (5) and (4) in that there is a constraint $\mathbf{1}_n \in \mathcal{R}(X)$ that defines a feasible set $\mathcal{F} \subset St(q,n)$. The accelerated Riemannian manifold proximal gradient method (Huang and Wei, 2019) is modified to define the accelerated Riemannian manifold projected proximal gradient (ARPPG) method by adding the projection (7) derived in the next section. The details of ARPPG are in Algorithm 2.

Algorithm 2 Accelerated Riemannian Manifold Projected Proximal Gradient Method(ARPPG)

Input: Lipschitz constant L on ∇f, parameter $\mu \in (0, 1/L)$ in the proximal mapping, line search parameter $\sigma \in (0,1)$, shrinking parameter in line search $\beta \in (0,1)$, positive integer N for safeguard;

1: $t_0 = 1, y_0 = x_0, z_0 = x_0; \lambda = \lambda_0$
2: **for** $k = 0, \dots$ **do**
3:     **if** $\operatorname{mod}(k, N) = 0$ **then**   ▷ Invoke safeguard every N iterations
4:         Invoke Algorithm 3: $[z_{k+N}, x_k, y_k, t_k] = Algo3(z_k, x_k, y_k, t_k, F(x_k));$
5:     **end if**
6:     Compute

$$\eta_{y_k} = \underset{\eta \in T_{y_k}\mathcal{M}}{\operatorname{argmin}} \langle \operatorname{grad} f(y_k), \eta \rangle + \frac{1}{2\mu}\|\eta\|_F^2 + g(y_k + \eta);$$

7:     $x_{k+1} = R_{y_k}(\eta_{y_k});$
8:     $x_{k+1} = proj(x_{k+1});$
9:     $t_{k+1} = \frac{\sqrt{4t_k^2 + 1} + 1}{2};$
10:    Compute

$$y_{k+1} = R_{x_{k+1}}\left(\frac{1 - t_k}{t_{k+1}} R_{x_{k+1}}^{-1}(x_k)\right);$$

11:    Compute $y_{k+1} = proj(y_{k+1}).$
12: **end for**
13: $X_* = x_{k+1}$

There are several retractions that can be constructed for the Stiefel manifold. Algorithm 2 uses the efficient retraction based on the singular value decomposition (SVD):

$$[Q, R] = \operatorname{qr}(X + \eta_X), \; [U, S, V] = \operatorname{svd}(R),$$
$$R_X(\eta_X) = Q(UV^T),$$

where qr and svd mean computing the compact QR decomposition and SVD of a matrix, respectively. $R_X^{-1}(Y) = YS - X$, where S is the solution of the Lyapunov equation $(X^T Y)S + S(Y^T X) = 2I_q$.

The Projection: Given $X \in St(q,n)$, the task is to find a $Y \in St(q,n)$ with $\mathbf{1}_n \in \mathcal{R}(Y)$ that minimizes $\|X - Y\|_F^2$. Letting $f(Y; X) = \operatorname{trace}(X^T Y)$ denote a cost function parameterized by X, the problem can be formulated in an equivalent form by noting

$$\min_{Y \in \mathcal{F}} \|X - Y\|_F^2 \leftrightarrow \max_{Y \in \mathcal{F}} f(Y; X)$$

Algorithm 3 Safeguard for Algorithm ARPPG

Input: $[z_k, x_k, y_k, t_k, F(x_k)]$;
Output: $[z_{k+N}, x_k, y_k, t_k]$;

1: Compute

$$\eta_{z_k} = \underset{\eta \in T_{z_k}\mathcal{M}}{\operatorname{argmin}} \langle \operatorname{grad} f(z_k), \eta \rangle + \frac{1}{2\mu}||\eta||_F^2 + g(z_k + \eta);$$

2: Set $\alpha = 1$;
3: **while** $F(proj(R_{z_k}(\alpha\eta_{z_k}))) > F(z_k) - \sigma\alpha||\eta_{z_k}||_F^2$ **do**
4:     $\alpha = \beta\alpha$;
5: **end while**
6: **if** $F(proj(R_{z_k}(\alpha\eta_{z_k}))) < F(x_k)$ **then**   ▷ Safeguard takes effect
7:     $x_k = R_{z_k}(\alpha\eta_{z_k})$, $y_k = R_{z_k}(\alpha\eta_{z_k})$, and $t_k = 1$;
8:     $x_k = proj(x_k)$, $y_k = proj(y_k)$;
9: **else**
10:     x_k, y_k and t_k keep unchanged;
11: **end if**
12: $z_{k+N} = x_k$;   ▷ Update the compared iterate

where $\mathcal{F} = \{Y \in St(q, n), \ \mathbf{1}_n \in \mathcal{R}(Y)\}$. The maximum value of $f(Y; X)$ is q and is achieved when $X \in \mathcal{F}$ and the problem is invariant with respect to $Q \in \mathcal{O}(q)$ where $\mathcal{O}(q)$ is the orthogonal group consisting of q-by-q orthogonal matrices, i.e.,

$$f(Y; X) = f(YQ; XQ).$$

Note that the cost function changes for this invariance. In general, $f(Y; X) \neq f(YQ; X)$.

For an element of the feasible set $\mathcal{F}$, there must exist $Q \in \mathcal{O}(q)$ such that $\hat{Y} = YQ = [\tilde{1}_n \ \tilde{Y}]$ with $\tilde{1}_n = \mathbf{1}_n/\sqrt{n}$, $\tilde{Y} \in St(q-1, n)$ and $\mathcal{R}(\tilde{Y}) \perp \tilde{1}_n$. There are, of course, many such $\hat{Y}$ possible. This can be seen from

$$Q = [q_1 \ Q_\perp], \ \hat{Y} = [\tilde{1}_n \ \tilde{Y}] = [Yq_1 \ YQ_\perp].$$

Given Y, the vector q_1 is uniquely defined but $Q_\perp$ is any orthonormal completion of q_1 and $\tilde{Y} = YQ_\perp$ varies with the choice of $Q_\perp$. This can be used to parameterize the cost function over $\mathcal{F}$ to give an alternative form of the optimization problem defining the projection and reveal a constructive form of the solution Y_*.

The two forms of the optimization problem are

$$Y_* = \underset{Y \in \mathcal{F}}{\operatorname{argmax}} \operatorname{trace}(X^T Y),$$
$$where \ \mathcal{F} = \{Y \in St(q, n), \ \mathbf{1}_n \in \mathcal{R}(Y)\};$$

$$(\hat{Y}_*, Q_*) = \underset{\hat{Y} \in \mathcal{G}, \ Q \in \mathcal{O}(q)}{\operatorname{argmax}} \operatorname{trace}(Q^T X^T \hat{Y}),$$
$$\mathcal{G} = \{\hat{Y} = [\tilde{1}_n \ \tilde{Y}] \mid \tilde{Y} \in St(q-1, n), \ \mathbf{1}_n \perp \mathcal{R}(\tilde{Y})\}.$$

The second form can be solved analytically and a solution for the first form recovered easily. The cost function for the second form can be expanded as

$$\operatorname{trace}(Q^T X^T \hat{Y}) = q_1^T X^T \tilde{1}_n + \operatorname{trace}(Q_\perp^T X^T \tilde{Y}).$$

The first term of the sum in the cost function is independent of the second term while the second term is essentially determined by the choice of q_1. For any q_1 and orthonormal completion $Q_\perp$, the maximum value of q_1 for the second term is achieved by $\tilde{Y} = XQ_\perp$.

Given this optimal choice of $\tilde{Y}$ parameterized by Q, the problem then becomes finding the optimal q_* for

$$\max_{q_1 \in St(1,n)} q_1^T X^T \tilde{1}_n$$

and $Q_* = [q_* \ Q_\perp^*]$ where $Q_\perp^*$ is any orthonormal completion of q_*. This has a maximum value of 1 if and only if $\tilde{1}_n \in \mathcal{R}(X)$. Otherwise it is maximized by

$$q_* = \frac{X^T \tilde{1}_n}{\|X^T \tilde{1}_n\|_2}.$$

There are several maximizers given by

$$q_* = \frac{X^T \tilde{1}_n}{\|X^T \tilde{1}_n\|_2}$$

$Q_\perp^* \in St(q-1, n)$ is any orthonormal completion of q_*

$$\tilde{Y}_* = XQ_\perp^*, \ \hat{Y}_* = [\tilde{1}_n \ \tilde{Y}_*].$$

Finally, Y_*, the maximizer for the original parameterized form of $f(Y; X)$ can be determined from $\hat{Y}_*$

$$Y_* = \hat{Y}_* Q_*^T = [\tilde{1}_n \ \tilde{Y}_*][q_* \ Q_\perp^*]^T = \tilde{1}_n q_*^T + XQ_\perp^*(Q_\perp^*)^T.$$

This form shows that the choice of $Q_\perp^*$, i.e., the basis for $\mathcal{R}^\perp(q_*)$, that determines $\hat{Y}_*$ does not result in multiple Y_* since the projector $Q_\perp^*(Q_\perp^*)^T$ is invariant.

Therefore, a computationally efficient form of the unique solution is given by

$$Y_* = \underset{Y \in \mathcal{F}}{\operatorname{argmax}} f(Y; X) = \tilde{1}_n q_*^T + XQ_\perp^*(Q_\perp^*)^T \quad (6)$$

$$= \tilde{1}_n q_*^T + X(I - q_* q_*^T), \quad (7)$$

$$q_* = \frac{X^T \tilde{1}_n}{\|X^T \tilde{1}_n\|_2}. \quad (8)$$

4. NUMERICAL EXPERIMENTS

4.1 Empirical Evaluation Techniques

ARPPG was evaluated using a family of synthetic benchmark networks and real-world networks by comparing its performance to that of three state-of-the-art algorithms for community detection: the GN algorithm (Newman, 2004), the Infomap algorithm (Rosvall and Bergstrom, 2008) and the Louvain method (Blondel et al., 2008). GN and Louvain methods were applied to maximizing the modularity $Q = \frac{1}{2m}\operatorname{trace}(X^T M X)$, where X is an assignment matrix that specifies a partitioning of the nodes into communities. Even though the Infomap method was not designed to maximize the modularity, it is one of the best performing methods out of the framework, see (Lancichinetti et al., 2011). So, we also compare our algorithm with the Infomap method. ARPPG maximized the cost function defined earlier based on modularity and a sparsity penalty term.

The assignments of nodes to communities produced by each algorithm for a given problem were compared using their modularity values. However, since the modularity used here is one of many cost functions in the literature that heuristically define preferred assignments, a metric independent of the cost function was used to assess the quality of the assignments. A ground truth assignment of nodes to communities is associated with each benchmark

graph. Given the ground truth, normalized mutual information (NMI) (Danon et al., 2005) and adjusted mutual information (AMI) (Vinh et al., 2010) were used to compare the quality of the communities. The values of NMI and AMI are in $[0, 1]$ with larger values indicating higher similarity.

AMI is introduced in addition to NMI to correct the measures for randomness by specifying a model by which random partitions are generated. AMI is defined as

$$AMI(X,Y) = \frac{I(X,Y) - E\{I(X,Y)\}}{\max\{H(X), H(Y)\} - E\{I(X,Y)\}},$$

where

$$E\{I(X,Y)\} = \sum_{u,v} \sum_{n_{uv}=(a_u+b_v-N)^+}^{\min(a_u,b_v)} \frac{n_{uv}}{N} \log\left(\frac{N \cdot n_{uv}}{a_u b_v}\right) \times$$

$$\frac{a_u! b_v! (N-a_u)! (N-b_v)!}{N! n_{uv}! (a_u - n_{uv})! (b_v - n_{uv})! (N - a_u - b_v + n_{uv})!}$$

by adopting a hypergeometric model of randomness, where $(a_u + b_v - N)^+$ denotes $\max(1, a_u + b_v - N)$, and $a_u = \sum_v n_{uv}$ and $b_v = \sum_u n_{uv}$.

The synthetic benchmarks have clearly defined ground truth based on intracommunity connectivity graphs that are strongly connected but not necessarily completely connected as in our ideal case defined above. The members of the family of networks are defined by a parameter that makes the network have an increasingly ill-defined community structure. As a result, any reasonable algorithm should detect community structure accurately when it is well-defined and the discrimination ability of the algorithm is tested as the definition degrades. Additionally, we must consider the robustness of the combinatorial algorithms relative to their runtime choices, e.g., the particular random walks used in Infomap or the order and manner in which one-node moves are considered in the Louvain method. Similarly, ARPPG and other algorithms based on optimization over a continuous domain are dependent on their initial conditions or other strategies to avoid finding an unacceptable local maximum.

For a network representing real-world relationships there can be ground truth based on a clear definition of the entities that define the nodes, empirical observations such as observed social behavior, or classifications based on opinions of human observers who may or may not be experts in a discipline related to the information. As a result, different cost functions may characterize the desired ground truth with different levels of accuracy. The use of the geometry, the sparsity constraint and continuation is an attempt to improve the robustness and aid in the selection of parameters such as the number of communities and the penalty parameter.

4.2 Synthetic Benchmarks

The generalized LFR benchmark graphs (Lancichinetti and Fortunato, 2009) were used as the synthetic network benchmarks. These subsume the well-known benchmark proposed by Girvan and Newman (Girvan and Newman, 2002) and are more challenging for community detection algorithms. For the LFR benchmarks we used in this

paper, the networks have $N = 1000$ nodes, the average node degree is 20, the maximum node degree is 50, the communities have between 20 and 100 nodes, the exponent of the degree power law distribution is -2, and the exponent of the community size power law distribution is -1. The numbers of communities for the LFR benchmarks are around 20.

In the construction of the benchmark graphs, the mixing parameter μ determines the difficulty level to detect communities. Setting $\mu = 0$, gives a graph defining a ground truth where the communities are strongly connected components and there are no edges between the communities. This is more challenging than the ideal ground truth of communities that are cliques used to motivate the optimization problem. For any value of $\mu > 0$, the graph also has an associated ground truth but the mixing causes the community structure to be less clearly defined.

For the LFR benchmarks we used in this paper, the networks have $N = 1000$ nodes, the average node degree is 20, the maximum node degree is 50, the communities have between 20 and 100 nodes, the exponent of the degree power law distribution is -2, and the exponent of the community size power law distribution is -1. The numbers of communities for the LFR benchmarks are around 20.

Results for the LFR networks: For the LFR benchmark with $\mu = 0$, as expected and required, all four algorithms have $NMI = 1$, the same modularity value and the same assignment to $q_{true} = 24$ strongly connected communities. ARPPG requires the desired number of communities as a parameter value and in this case it was taken as $q = q_{true} = 24$. The choice of an initial q and the development of a dynamic adaptation strategy are key ongoing tasks for ARPPG. There is promising evidence that it is possible. For $\mu = 0$ and ARPPG run with $q = 25$ and $q = 26$, i.e., near q_{true}, the modularity decreases as q increases. The final values of NMI for $q = 25$ and $q = 26$ change only slightly 0.99 and 0.98 respectively. Of course this information is not available for the algorithm to use, but it is due to the fact that the partitioning for $q = 25$ and $q = 26$ are nested in the partitioning for $q = q_{true} = 24$, i.e., the extra communities are refinements of the 24 by splitting without crossing the ideal community boundaries. Any nodes that are not in the same community in the ideal partitioning remain in different communities in the refined partitions. This information can be detected by the algorithm and used to guide adjustment of q while revealing a hierarchical structure relevant to discussion of resolution limits (Fortunato and Barthelemy, 2007) and alternative cost functions, e.g., the constant Potts model (Traag et al., 2011).

The algorithms were also tested with multiple nonzero values of μ. The values of NMI and modularity are shown in Table 1 where ARPPG uses $q = q_{true}$ determined by the network for each value of μ. All four methods determine the ground truth community assignments for the networks with $\mu \leq 0.3$. For $\mu = 0.4$ and $\mu = 0.5$ three methods determine the associated ground truths and one comes very close: GN with $NMI = 0.99$, $AMI = 0.99$ and ARPPG with $NMI = 0.99$, $AMI = 0.99$ respectively.

Table 1. Performance on LFR Bechmark Networks

Methods	Measurements	The mixing parameter μ							
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
GN	NMI	1	1	1	0.9972	0.8694	0.6679	0.4932	0.4886
	AMI	1	1	1	0.9962	0.7202	0.2539	0.0142	0.0031
	Modularity	0.8254	0.7268	0.6283	0.5280	0.3579	0.1230	0.0393	0.0329
Infomap	NMI	1	1	1	1	1	0	0	0
	AMI	1	1	1	1	1	0	0	0
	Modularity	0.8254	0.7268	0.6283	0.5288	0.4440	0	0	0
Louvain	NMI	1	1	1	1	1	0.9527	0.2192	0.0677
	AMI	1	1	1	1	1	0.9107	0.1748	0.0267
	Modularity	0.8254	0.7268	0.6283	0.5288	0.4440	0.3390	0.2093	0.1921
ARPPG	NMI	1	1	1	1	0.9935	0.8811	0.3422	0.0967
	AMI	1	1	1	1	0.9927	0.8651	0.3014	0.0473
	Modularity	0.8254	0.7268	0.6283	0.5288	0.4427	0.3239	0.1712	0.1355

ARPPG using $q \neq q_{true}$ for $\mu \leq 0.4$ demonstrates trends like those for $\mu = 0$ upon which a q adaptation strategy might be built. As q increases from q_{true}, NMI, AMI and modularity decrease at a rate that increases as μ increases. The partitions are nested, then only slightly not nested with one or two nodes crossing communities of the ground truth assignment, and finally with a significant loss of nesting.

For the noisy cases in Table 1, GN degrades quickly while ARPPG and the Louvain method degrade more slowly. Infomap achieves an $NMI = 1$, $AMI = 1$ until $\mu = 0.5$ then drops to near 0. The performance of Infomap and the Louvain method are sensitive to their runtime decisions, e.g., the Infomap performance here uses the heuristic available in the publicly distributed code of running the method multiple times and choosing the "best" result. ARPPG, on the other hand, with its continuation strategy and initial condition selection using RNewton was seen to be remarkably robust even in the noisy situations.

4.3 Real World Networks

Three widely used real-world networks were used to assess the performance of ARPPG. The first is an American college football network (Girvan and Newman, 2002), in which the nodes represent football teams, and an edge exists between the nodes if there is a match between two teams. The ground truth community assignment is given by the membership in the same athletic conference, i.e., indisputable observations. The second is Zachary's karate club network (Zachary, 1977), which is an undirected social network of friendship between 34 members of a karate club at a university. Edges connect individuals who were observed to interact outside the activities of the karate club. The ground truth is based on the splitting of the membership into 2 new disjoint karate clubs. However, there is a second ground truth based of 4 communities of 2 disjoint social groups within each of the 2 new clubs. The 2 community ground truth is defined by indisputable observation, the 4 community ground truth is based on less precise social interaction data. The third is the Polbooks network (Newman, 2006b) of books about US politics published around the time of the 2004 presidential election and sold by the online bookseller Amazon.com. Edges between books represent frequent co-purchasing of books by the same buyers. The ground truth is determined by the subjective classification of the books by a non-expert hu-

man observer. Given the difference in the level of certainty becoming increasingly debatable in these benchmarks, it is expected that detecting communities should be more difficult and dependent on cost function selection and algorithm tuning for each of the three in turn.

Table 2 summarizes the performance on the real-world networks. Note that overall modularity values for the community assignments produced are significantly lower than those for the synthetic networks and the different assignments produced all have similar modularity values with significantly different quality as measured by NMI and AMI. This is most pronounced for the opinion-based ground truth of the Polbooks network as expected. For the football network, ARPPG using $q = q_{true}$ produces an assignment close to the ground truth. Infomap produces the same 12 community assignment but requires multiple runs, some of which produce significantly different assignments. GN and the Louvain method do not get the correct number of communities despite achieving a value of modularity close to that from the other algorithms. ARPPG run with $q \neq q_{true}$ exhibits the same trends on modularity and nesting discussed for the synthetic networks as desired.

For the karate club network, only ARPPG with $q = q_{true} = 2$ produces the ground truth with 2 communities. When ARPPG is run with $q \neq q_{true}$ it exhibits the desired nesting trends and, in particular, for $q = 4$ it produces the second ground truth known for the network. (The NMI and AMI in the table is not 1 for that case because it is the 4 community ground truth compared to the 2 community ground truth.) The Louvain algorithm produces different 4 community assignments depending on the order of traversal of the nodes. The 4 community ground truth is one of them but the one in the table are not quite the same as is seen from the NMI and AMI differing from that of ARPPG. Infomap produces different community assignments with varying numbers of communities in different runs. The result in the table is the best one. As expected, the Polbooks network is the most difficult. Modularity does not predict well the quality of the assignment measured by NMI and AMI. Even ARPPG with $q = q_{true}$ does not produce an assignment as close to ground truth as it does for the other two networks. The fact that modularity does not clearly indicate the ground truth is also seen in the trends for ARPPG with $q \neq q_{true}$. Nesting is not observed and the best modularity is observed for $q = 5 \neq q_{true} = 3$.

Table 2. Performance on Real-World Networks (the best performance is in bold), where n is the number of nodes, m is the number of edges, q_{true} is the number of ground truth communities and numbers in parentheses are the numbers of communities detected. For ARPPG the numbers in parentheses are also the values used for ARPPG's parameter q.

Datasets	n	m	q_{true}	Measurements	GN	Infomap	Louvain	ARPPG			
Football	115	613	12	NMI	0.879(10)	**0.924**(12)	0.890(10)	**0.924**(12)	0.911(13)	0.912(14)	0.882(10)
				AMI	0.802(10)	**0.898**(12)	0.821(10)	**0.898**(12)	0.861(13)	0.848(14)	0.813(10)
				Modularity	0.600(10)	0.601(12)	**0.605**(10)	0.601(12)	0.581(13)	0.566(14)	0.596(10)
Karate	34	78	2	NMI	0.580(5)	0.700(3)	0.587(4)	**1.000**(2)	0.811(3)	0.687(4)	0.542(5)
				AMI	0.402(5)	0.579(3)	0.425(4)	**1.000**(2)	0.672(3)	0.505(4)	0.364(5)
				Modularity	0.401(5)	0.402(3)	0.419(4)	0.372(2)	0.373(3)	**0.420**(4)	0.382(5)
Polbooks	105	441	3	NMI	0.559(5)	0.494(6)	0.537(5)	**0.565**(3)	0.503(4)	0.465(5)	0.439(6)
				AMI	0.488(5)	0.390(6)	0.458(5)	**0.535**(3)	0.424(4)	0.362(5)	0.323(6)
				Modularity	0.517(5)	0.523(6)	**0.527**(5)	0.508(3)	0.504(4)	0.510(5)	0.505(6)

5. CONCLUSION

This paper proposes a new Riemannian projected proximal gradient method applied to modularity with a convex nonsmooth sparsity penalty term for community detection. Numerical results show that ARPPG is competitive with state-of-the-art algorithms in terms of quality of assignment and robustness. Observations of performance as algorithm parameters vary provide leading evidence that a parameter adaptation strategy and an efficient implementation are feasible.

REFERENCES

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1), 183–202.

Blondel, V.D., Guillaume, J.L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008.

Chen, S., Ma, S., Man-Cho So, A., and Zhang, T. (2020). Proximal gradient method for nonsmooth optimization over the Stiefel manifold. *SIAM Journal on Optimization*, 30(1), 210–239.

Danon, L., Diaz-Guilera, A., Duch, J., and Arenas, A. (2005). Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09), P09008.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3-5), 75–174.

Fortunato, S. and Barthelemy, M. (2007). Resolution limit in community detection. *Proceedings of the national academy of sciences*, 104(1), 36–41.

Girvan, M. and Newman, M.E. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12), 7821–7826.

Huang, W. and Wei, K. (2021). Riemannian proximal gradient methods. *Mathematical Programming*. doi: 10.1007/s10107-021-01632-3.

Huang, W., Absil, P.A., Gallivan, K.A., and Hand, P. (2018). ROPTLIB: an object-oriented C++ library for optimization on Riemannian manifolds. *ACM Transactions on Mathematical Software (TOMS)*, 44(4), 43.

Huang, W. and Wei, K. (2019). Extending FISTA to Riemannian optimization for sparse PCA. *arXiv preprint arXiv:1909.05485*.

Lancichinetti, A. and Fortunato, S. (2009). Community detection algorithms: a comparative analysis. *Physical Review E*, 80(5), 056117.

Lancichinetti, A., Radicchi, F., Ramasco, J.J., and Fortunato, S. (2011). Finding statistically significant communities in networks. *PloS one*, 6(4), e18961.

Marchand, M.S. (2017). *Low-rank Riemannian optimization approach to the role extraction problem*. Ph.D. thesis, The Florida State University.

Newman, M.E. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6), 066133.

Newman, M.E. (2006a). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3), 036104.

Newman, M.E. (2006b). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23), 8577–8582.

Newman, M.E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69(2), 026113.

Newman, M.E. and Leicht, E.A. (2007). Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, 104(23), 9564–9569.

Rosvall, M. and Bergstrom, C.T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4), 1118–1123.

Traag, V.A., Van Dooren, P., and Nesterov, Y. (2011). Narrow scope for resolution-limit-free community detection. *Physical Review E*, 84(1), 016114.

Vinh, N.X., Epps, J., and Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11, 2837–2854.

Yang, L., Cao, X., He, D., Wang, C., Wang, X., and Zhang, W. (2016). Modularity based community detection with deep learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, volume 16, 2252–2258.

Zachary, W.W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4), 452–473.