

A Riemannian Optimization Approach for Role Model Extraction

Melissa Marchand¹, Wen Huang², Arnaud Browet², Paul Van Dooren² and Kyle A. Gallivan¹

Abstract—The ability to compute meaningful clusters of nodes is important in the analysis of large networks. A particular approach to this problem is the use of role models of a graph. For large networks, the algorithms must be specifically designed to extract role models while maintaining efficiency in storage and computations. Browet et al. have investigated the computation of role models for both moderately sized and large networks. They proposed an efficient iteration on low-rank matrices to compute an approximation to the required pairwise node similarity measure at MTNS 2014. In this paper, we summarize a new approach to compute an approximation to the pairwise node similarity measure for large networks based on Riemannian optimization. A comparison of our optimization approach with that of Browet et al. shows that our approach computes the same approximate solution in significantly less time.

I. INTRODUCTION

Many complex systems can be represented as network structures, e.g., human interactions, food webs, and gene interactions. Recent works have focused on the extraction of clusters to comprehend large networks and obtain relevant statistical properties. In recent years, researchers have proposed various measures and algorithms to identify community structures, i.e., subgroups of densely connected nodes [9], [14], [15]. However, this structural distribution of nodes in a graph is not always representative, for example, bipartite networks or cycle graphs do not contain communities, even if they might be heavily structured. Also, less attention has been turned toward discovering more general structures, known as role extraction or block modeling [8], [16].

In previous work, Browet et al. solved the role extraction problem by assuming different roles in a network represent groups of nodes that have similar flow patterns [5], [6], [7]. Thus, communities are roles where each node in a role mainly interacts with other nodes within the same role. However, many other role interactions can be defined, such as, a leader-follower model on social network interactions or a block cycle model for food webs. To represent these flow patterns, Browet et al. defined a pairwise node similarity measure to compare the neighborhood patterns of every node, where the

measure is high for any pair of nodes sharing analogous flow properties. They proposed a two-stage role extraction algorithm. The first stage computes a measure of node-to-node similarity using a matrix iterative scheme and the second stage extracts roles using a fast community detection algorithm. For large networks, a modification of the two-stage algorithm was developed to save on computation. The iterative scheme to compute the matrix with elements that measured pairwise node similarity was modified to include projection onto low-rank matrices to converge to a low-rank approximation of the similarity matrix. The modifications resulted in a noticeable improvement for large networks.

In recent years, there has been significant advances in solving optimization problems with constraints that can be characterized as a Riemannian manifold. Improvements in convergence theory and the characterization of the computational demands and mathematical structure of fundamental operations such as vector transport and retraction have produced a systematic theory for families of algorithms and well-organized computational library implementations (see for example [3], [10], [11], [12], and [17]). In this paper, we summarize a new approach to compute a low-rank approximation to the pairwise node similarity matrix for large networks based on Riemannian optimization. This is done by developing a cost function over the Riemannian manifold of symmetric positive semidefinite matrices of fixed-rank and, after deriving the required Riemannian objects, applying Riemannian optimization algorithms appropriate for large networks. A comparison of our optimization approach with that of Browet et al. shows that our approach computes the same approximate solution in significantly less time.

The paper is organized as follows. In Section II both the original and low-rank approaches of Browet et al. are described and associated convergence results stated. A cost function is derived from the low-rank matrix iteration in Section III. The manifold framework is discussed and the Riemannian objects required to apply Riemannian optimization algorithms are stated in Section IV. In Section V, numerical experiments using role structure examples like those used in [6] are used to demonstrate the improvement due to the Riemannian approach to the pairwise node similarity computation. Finally, conclusions and future work are presented in Section VI. Details of this work will be presented in a forthcoming dissertation.

II. LOW-RANK APPROXIMATION OF SIMILARITY MATRIX BY BROWET ET AL.

Given a weighted and directed graph $G_A(V, E)$, where V is the set of vertices and E is the set of edges and its

*This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office. It was also partially supported by the U.S. National Science Foundation under grant NSF 1262476.

¹ Florida State University - Department of Mathematics, Florida State University, 208 Love Building, 1017 Academic Way, Tallahassee, FL 32306-4510, Corresponding author: mmarchan@math.fsu.edu

²Intstitute of Information and Communication Technologies, Electronics, and Applied Mathematics (ICTEAM), Catholic University of Louvain (UCL), 4 Av. G. Lemaitre, 1348 Louvain-la-Neuve, Belgium

adjacency matrix $A \in \mathbb{R}^{n \times n}$ is defined as $A_{i,j} \neq 0$ if $(i, j) \in E$ for $i, j \in V$, the pairwise node similarity measure $S \in \mathbb{R}^{n \times n}$, presented in [5], is defined as

$$S_{k+1} = \Gamma_A [I + \beta^2 S_k] \quad (1)$$

where

$$\Gamma_A : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n} : \Gamma_A[X] = AXA^T + A^T X A, \quad (2)$$

I is the identity matrix, and $\beta \in \mathbb{R}$. The similarity score $S_{i,j}$ is the number of common target nodes from node i to node j with the same neighborhood patterns. Note that the similarity matrix S is a symmetric positive semidefinite matrix and that the parameter β can be chosen to increase the weight of long neighborhood paths while guaranteeing convergence of the sequence S_k in (1) [5] [6].

The iteration of Browet et al. to compute S is given by (with $S_0 = 0$),

$$S_{k+1} = S_1 + \beta^2 \Gamma_A [S_k] \quad (3)$$

where

$$S_1 = AA^T + A^T A.$$

Browet et al. showed that if β is chosen such that

$$\beta^2 \leq \frac{1}{\rho(A \otimes A + (A \otimes A)^T)},$$

where $\rho(\cdot)$ denotes the spectral radius, then the sequence converges to the fixed point solution S^* given by

$$\text{vec}(S^*) = [I - \beta^2 (A \otimes A + (A \otimes A)^T)]^{-1} \text{vec}(S_1), \quad (4)$$

where $\text{vec}(S)$ denotes the vectorization of the matrix S formed by stacking the columns of S into one column vector. A less computationally expensive bound to ensure convergence is given by

$$\beta^2 \leq \frac{1}{\rho((A + A^T)^2)}.$$

However, even if the sequence is guaranteed to converge, its computational cost to compute the fixed point solution may be unacceptably high. Thus, Browet et al. defined a low-rank approximation of the similarity matrix S_k . For $S_k^{(r)} = X_k X_k^T$, where $X_k \in \mathbb{R}_*^{n \times r}$ is a full-column rank matrix of size $n \times r$, the low-rank similarity approximation scheme is defined as

$$\begin{aligned} S_1^{(r)} &= \Pi^{(r)} [[A \mid A^T] [[A \mid A^T]^T]] = X_1 X_1^T \\ S_{k+1}^{(r)} &= \Pi^{(r)} \left[S_1^{(r)} + \beta^2 \Gamma \left[S_k^{(r)} \right] \right] = X_{k+1} X_{k+1}^T \end{aligned} \quad (5)$$

where $\Pi^{(r)}[\cdot]$ is an orthogonal projector onto the dominant subspace of dimension at most r , which is computed as a truncated SVD on the R_k -factor (i.e., $R_k = U_k \Sigma_k V_k$) of the QR factorization of Y_k defined as

$$Y_k = [X_1 \mid \beta A X_k \mid \beta A^T X_k].$$

The updated factor X_{k+1} is computed as

$$X_{k+1} = Q_k U_k \Sigma_k.$$

The stopping criterion for the low-rank iteration of Browet et al. is

$$\|R \begin{pmatrix} I_r & 0 \\ 0 & -I_r \end{pmatrix} R^T\|_F \leq \epsilon \quad (6)$$

where $R \in \mathbb{R}^{2r \times 2r}$ is from the QR factorization of $[X_{k+1} \mid X_k]$.

Browet et al. proved, using perturbation theory, that the low-rank iterative scheme (5) converges locally to a fixed point $S^{(r)}$ if the spectral gap at the r^{th} singular value is sufficiently large [5].

III. COST FUNCTION DERIVATION

The first step in developing a Riemannian optimization approach to replace the low-rank algorithm above is to write it in a form that is related to an iteration that is more clearly related to optimization and then to deduce a cost function. Observe that (5) can be written as

$$\begin{aligned} S_{k+1}^{(r)} &= \Pi^{(r)} \left[S_k^{(r)} + \left(S_1^{(r)} - S_k^{(r)} + \beta^2 \Gamma \left[S_k^{(r)} \right] \right) \right] \\ &= \Pi^{(r)} \left[S_k^{(r)} + \nabla f \left(S_k^{(r)} \right) \right]. \end{aligned}$$

where

$$\begin{aligned} f(S) &= \text{trace} \left(S^T \left(S_1^{(r)} - \frac{1}{2} S + \beta^2 A S A^T \right) \right) \\ &= \left\langle S, S_1^{(r)} - \frac{1}{2} S + \beta^2 A S A^T \right\rangle_F, \end{aligned} \quad (7)$$

$\text{trace}(\cdot)$ denotes the trace operator, i.e., for matrix $A \in \mathbb{R}^{n \times n}$, $\text{trace}(A) = \sum_{i=1}^n a_{i,i}$, and $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius inner product. One can verify this by considering the directional derivative of f along $Z \in \mathbb{R}^{n \times n}$, that is,

$$\begin{aligned} Df(S)[Z] &= \text{trace} \left(S_1^{(r)T} Z - \frac{1}{2} (Z^T S + S^T Z) \right. \\ &\quad \left. + \beta^2 (A Z A^T S^T + A S A^T Z^T) \right) \\ &= \text{trace} \left(Z^T (S_1^{(r)} - S + \beta^2 (A^T S A + A S A^T)) \right) \\ &= \left\langle Z, S_1^{(r)} - S + \beta^2 \Gamma[S] \right\rangle_F. \end{aligned}$$

Thus, the gradient of f is, as desired,

$$\nabla f(S) = S_1^{(r)} - S + \beta^2 \Gamma[S]. \quad (8)$$

If the similarity matrix S is not full-rank, i.e., $S = X X^T$ where $X \in \mathbb{R}^{n \times r}$ and $\text{rank}(X) = r$ then S can be defined viewed as an element of the symmetric positive semidefinite fixed-rank manifold, $\mathcal{S}_+(n, r)$, where $\mathcal{S}_+(n, r)$ is defined by

$$\begin{aligned} \mathcal{S}_+(n, r) &:= \{ S \in \mathbb{R}^{n \times n} \mid S = S^T \succeq 0, \text{rank}(S) = r \} \\ &:= \{ X X^T \mid X \in \mathbb{R}^{n \times r}, \text{rank}(X) = r \}. \end{aligned} \quad (9)$$

Finally, given the cost function $f : \mathcal{S}_+(n, r) \rightarrow \mathbb{R} : S \mapsto f(S)$, we define the function $F : \mathbb{R}^{n \times r} \rightarrow \mathbb{R} : X \mapsto F(X)$ where $F(X) = f(X X^T)$ and the optimization problem

$$\max_{X \in \mathbb{R}^{n \times r}} F(X). \quad (10)$$

is considered for the approximation of the pairwise similarity matrix.

IV. RIEMANNIAN OPTIMIZATION ON SYMMETRIC POSITIVE SEMIDEFINITE FIXED-RANK MANIFOLD

Observe that (10) is invariant by right-multiplication of X by orthogonal matrices O of size $r \times r$. Thus, we need to remove the degeneracy of the critical points. To do this, we will define the equivalence class as

$$[X] = \{XO \mid O \in \mathcal{O}_r\} \quad (11)$$

where \mathcal{O}_r is the orthogonal group, i.e., $\mathcal{O}_r := \{O \in \mathbb{R}^{r \times r} \mid O^T O = I_r\}$. The set of all equivalence classes is defined as

$$\mathbb{R}_*^{n \times r} / \mathcal{O}_r := \{[X] \mid X \in \mathbb{R}_*^{n \times r}\} \quad (12)$$

where $\mathbb{R}_*^{n \times r}$ is the noncompact Stiefel manifold, i.e., the set of full-rank matrices in $\mathbb{R}^{n \times r}$. Absil et al. showed in [1] that the quotient $\mathbb{R}_*^{n \times r} / \mathcal{O}_r$ is a quotient manifold. Therefore, (10) can be rewritten on the quotient manifold $\mathbb{R}_*^{n \times r} / \mathcal{O}_r$, i.e.,

$$\max_{[X] \in \mathbb{R}_*^{n \times r} / \mathcal{O}_r} F_r([X]) \quad (13)$$

where the function $F_r : \mathbb{R}_*^{n \times r} / \mathcal{O}_r \rightarrow \mathbb{R} : [X] \mapsto F_r([X]) = F(X)$. To use known Riemannian optimization methods (e.g., Riemannian steepest descent, limited-memory Riemannian Broyden-Fletcher-Goldfarb-Shanno (BFGS)) to solve (13), a few important mathematical objects are required. They are defined below.

A. Notation

Given a manifold \mathcal{M} , $T_X \mathcal{M}$ denotes the tangent space space of \mathcal{M} at $X \in \mathcal{M}$. For the quotient manifold, \mathcal{M} / \sim , the tangent space decomposes into the vertical space, denoted $\mathcal{V}_X \mathcal{M} / \sim$, and the horizontal space, denoted $\mathcal{H}_X \mathcal{M} / \sim$ where $X \in \mathcal{M}$ is the representative element for the element of the quotient space $\tilde{X} \in \mathcal{M} / \sim$. The Riemannian metric, which is the smooth varying inner product between two elements on the tangent space $T_X \mathcal{M}$, is denoted

$$g_X(\eta, \xi) = \langle \eta, \xi \rangle_X$$

for $\eta, \xi \in T_X \mathcal{M}$. The flat \flat notation is used as ξ^\flat which denotes a function from $T_X \mathcal{M}$ to \mathbb{R} , which is $\xi^\flat \eta = g_X(\xi, \eta)$, for all $\eta \in T_X \mathcal{M}$.

Lastly, given a function $f(X)$ on \mathcal{M} , $\text{grad} f(X)$ denotes the Riemannian gradient of f at X .

B. Riemannian Objects

The derivations of the Riemannian objects for the symmetric positive semidefinite fixed-rank manifold, $\mathcal{S}_+(n, r)$, are given in [13]. However, due to the size of the networks used for the role model problem, we would like to use the intrinsic dimension representation discussed in [10, Section 9.5]. That is, a tangent vector $\eta \in T_X \mathcal{M}$ can be represented by a d -dimensional vector, denoted v , of coordinates in a given basis B_X of $T_X \mathcal{M}$. If the columns of B_X forms an orthonormal basis of $T_X \mathcal{M}$, then many operations are inexpensive to compute [10, Section 9.5]. However, the basis derived from the horizontal space given in [13] is not an orthonormal basis with respect to the metric also in [13]. Thus, we will use an

alternative Riemannian metric for $\mathcal{S}_+(n, r)$, which will give us an alternative horizontal space.

An alternative Riemannian metric on the quotient manifold $\mathbb{R}_*^{n \times r} / \mathcal{O}_r$ is

$$g_X(\eta, \xi) = \text{trace}((X^T X) \eta^T \xi) \quad (14)$$

for all $\eta, \xi \in \mathcal{H}_X \mathbb{R}_*^{n \times r} / \mathcal{O}_r$ where the horizontal space is defined as

$$\mathcal{H}_X \mathbb{R}_*^{n \times r} / \mathcal{O}_r := \{XS + X_\perp K \mid S = S^T, S \in \mathbb{R}^{r \times r}, K \in \mathbb{R}^{(n-r) \times r}\} \quad (15)$$

and the vertical space is defined as

$$\mathcal{V}_X \mathbb{R}_*^{n \times r} / \mathcal{O}_r := \{X\Omega \mid \Omega^T = -\Omega, \Omega \in \mathbb{R}^{r \times r}\}. \quad (16)$$

Then, an orthonormal basis of (15) with respect to (14) is

$$\begin{aligned} & \{XL^{-T} e_i e_i^T L^{-1}, i = 1, \dots, r\} \cup \left\{ \frac{1}{\sqrt{2}} XL^{-T} \right. \\ & (e_i e_j^T - e_j e_i^T) L^{-1}, i = 1, \dots, r, j = i + 1, \dots, r\} \\ & \left. \cup \{X_\perp \tilde{e}_i e_i^T L^{-1}, i = 1, \dots, n - r, j = 1, \dots, r\}, \right. \end{aligned}$$

where (e_1, \dots, e_r) is the canonical basis of \mathbb{R}^r , $(\tilde{e}_1, \dots, \tilde{e}_{(n-r)})$ is the canonical basis of \mathbb{R}^{n-r} , and $X^T X = LL^T$ is the Cholesky decomposition. The dimension of the manifold is

$$\dim \mathcal{S}_+(n, r) = nr - \frac{1}{2}r(r-1) \quad (17)$$

where n is the size of the network and r is the rank.

Since Riemannian algorithms need a retraction and, for some, a vector transport, these objects must be defined for $\mathcal{S}_+(n, r)$. The retraction used is represented by the mapping $R_X : \mathcal{H}_X \mathbb{R}_*^{n \times r} / \mathcal{O}_r \rightarrow \mathbb{R}_*^{n \times r}$, where it is defined as

$$R_X(\eta) = X + \eta. \quad (18)$$

The vector transport by parallelization is defined by

$$\mathcal{T} = B_Y B_X^\flat \quad (19)$$

where B_Y and B_X are orthonormal bases of $\mathcal{H}_Y \mathbb{R}_*^{n \times r} / \mathcal{O}_r$ and $\mathcal{H}_X \mathbb{R}_*^{n \times r} / \mathcal{O}_r$, respectively, i.e., $B_X^\flat B_X = I$ for all X where I is the identity matrix [11]. Then, the d -dimensional representation of (19) for $v = B_X^\flat \eta$ is

$$\begin{aligned} \mathcal{T}^d v &= B_Y^\flat \mathcal{T} \eta = B_Y^\flat B_Y B_X^\flat \eta \\ &= (B_Y^\flat B_Y) (B_X^\flat B_X) v = v \end{aligned} \quad (20)$$

where \mathcal{T}^d is a d -dimensional representation of the vector transport [10, Section 9.5].

Since B_X forms an orthonormal basis of $\mathcal{H}_X \mathbb{R}_*^{n \times r} / \mathcal{O}_r$, then the Riemannian metric (14) reduces to the Euclidean metric for the intrinsic representations, i.e.,

$$\hat{g}_X(\eta, \xi) = v^T u, \quad (21)$$

where $\eta = B_X v$ and $\xi = B_X u \in \mathcal{H}_X \mathbb{R}_*^{n \times r} / \mathcal{O}_r$ [10, Section 9.5].

C. Riemannian Gradient

Recall the definition of Riemannian gradient [2, Section 3.6]: Given a smooth scalar field f on a Riemannian manifold \mathcal{M} , the gradient of f at X , denoted by $\text{grad}f(X)$, is defined as the unique element of $T_X\mathcal{M}$ that satisfies

$$Df(X)[\eta] = g_X(\text{grad}f(X), \eta) \quad (22)$$

for all $\eta \in T_X\mathcal{M}$ where $g_X(\cdot, \cdot)$ denotes the Riemannian metric. Consider the directional derivative of F along any $\eta \in \mathcal{H}_X \mathbb{R}_*^{n \times r} / \mathcal{O}_r$, i.e.,

$$\begin{aligned} DF(X)[\eta] &= \text{trace} \left(\eta X^T S_1^{(r)} + X \eta^T S_1^{(r)} - \frac{1}{2} (\eta X^T X X^T \right. \\ &\quad + X \eta^T X X^T + X X^T \eta X^T + X X^T X \eta^T) \\ &\quad + \beta^2 (\eta X^T A X X^T A^T + X \eta^T A X X^T A^T \\ &\quad \left. + X X^T A \eta X^T A^T + X X^T A X \eta^T A^T) \right) \\ &= \text{trace} \left((\eta X^T + X \eta^T)^T S_1^{(r)} - (\eta X^T + X \eta^T)^T X X^T \right. \\ &\quad \left. + \beta^2 (\eta X^T + X \eta^T)^T (A X X^T A^T + A^T X X^T A) \right) \\ &= \text{trace} \left((\eta X^T + X \eta^T)^T (S_1^{(r)} + X X^T + \beta^2 \Gamma [X X^T]) \right) \\ &= \text{trace} \left((\eta X^T + X \eta^T)^T \text{grad}f(X X^T) \right) \\ &= \text{trace} (X \eta^T \text{grad}f(X X^T)) + \text{trace} (\eta X^T \text{grad}f(X X^T)) \\ &= \text{trace} (\eta^T \text{grad}f(X X^T) X) + \text{trace} (\eta^T \text{grad}f(X X^T)^T X). \end{aligned}$$

Since $\text{grad}f(X X^T) = \text{grad}f(X X^T)^T$, then the directional derivative of F along any $\eta \in \mathcal{H}_X \mathbb{R}_*^{n \times r} / \mathcal{O}_r$ is

$$DF(X)[\eta] = \text{trace} (\eta^T 2 \text{grad}f(X X^T) X). \quad (23)$$

Using the definition of the Riemannian gradient (22), (23) can be rewritten to look like the Riemannian metric (14), i.e.,

$$\begin{aligned} DF(X)[\eta] &= \text{trace} (\eta^T 2 \text{grad}f(X X^T) X) \\ &= \text{trace} (\eta^T 2 \text{grad}f(X X^T) X (X^T X)^{-1} (X^T X)) \\ &= \text{trace} ((X^T X) \eta^T 2 \text{grad}f(X X^T) X (X^T X)^{-1}). \end{aligned}$$

Thus, the Riemannian gradient of (13) at X is

$$\text{grad}F_r(X) = 2 \text{grad}f(X X^T) X (X^T X)^{-1}. \quad (24)$$

V. NUMERICAL EXPERIMENTS

All codes were written in Matlab and all experiments are performed in Matlab R2015a on a 64 bit Mac platform with 2.5 GHz and 4 GB of memory.

The Riemannian optimization methods used in the experiments are steepest descent (RSD) [2] and limited-memory BFGS (LRBFGS) [12]. Both algorithms are line search algorithms and use the back tracking line search to find an Armijo point [2, Definition 4.2.2]. The initial stepsize is taken to be 1, the coefficient c_1 in the Armijo condition is 10^{-4} , and the ratio ρ for the decreasing the stepsize is 0.25.

The stopping criterion for both optimization methods requires the norm of the final gradient over the initial gradient to be less than 10^{-6} , and for the algorithm of Browet et al. $\epsilon = 10^{-6}$ in (6). The maximum number of iterations for

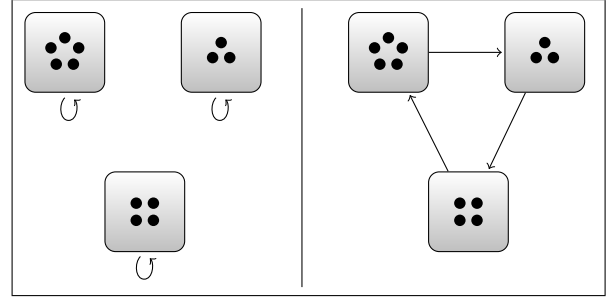


Fig. 1. Two different types of role graphs. Left: Community structure. Right: Block cycle.

the algorithm of Browet et al. and outer iterations for the optimization algorithms was 2000. The relative error was computed for each fixed-rank r , i.e.,

$$\frac{\|S_{approx}^{(r)} - S^*\|_F}{\|S^*\|_F} \quad (25)$$

where $S_{approx}^{(r)}$ is the approximate similarity matrix computed from either the algorithm of Browet et al. or our optimization approach and S^* is the fixed point solution given by (4). Table I summarizes the notation used when stating the experimental results.

As in [6], we computed the similarity measure of Erdős-Rényi random graphs containing a block structure (see Figure 1). To build our graphs, first we chose a directed role graph $G_B(V_B, E_B)$, i.e., each node in G_B defines a role that we want to identify. The first role graph we considered corresponds to a community structure where the nodes in a role interact mainly with other nodes in the same role. The second role graph corresponds to a block cycle where each node in a role interacts mainly with nodes in the following role in the cycle.

Given the role graph G_B , we built our random graph $G_A(V_A, E_A)$ in the same manner used in Browet et al. [6] where each node in G_A has a corresponding role in G_B . Edges were added to E_A according to two probability parameters p_{in} and p_{out} . For every pair of nodes $i, j \in V_A$, an edge $(i, j) \in E_A$ was added with probability p_{in} if there is an edge between the corresponding roles in G_B . If the edges does not exist between the corresponding roles in G_B , then the edge is added with probability p_{out} . This created adjacency matrices A that were numerically low-rank (i.e., the numerical rank

TABLE I
NOTATION FOR REPORTING EXPERIMENTAL RESULTS.

$iter$	number of iterations
nf	number of function evaluations
ng	number of gradient evaluations
nV	number of vector transport operations
nR	number of retraction evaluations
f_f	final function value
gf_0	Riemannian metric value of the initial gradient
gf_f	Riemannian metric value of the final gradient
$rel\ error$	relative error defined by (25)
$norm\ diff$	stopping criterion (6)
t	time (seconds)

of A for both role graphs is 3). Experimental results for $(p_{in}, p_{out}) = (0.9, 0.1)$ and $(p_{in}, p_{out}) = (0.7, 0.3)$ for both examples are displayed below.

Table II displays the results for the Erdős-Rényi random graph with community structure for $p_{in} = 0.9$ and $p_{out} = 0.1$. From the table, we can see that the relative error for all three methods was the same as r increased and that after $r = 3$ the relative error changed very little. In comparing the time, the Riemannian methods were faster than the algorithm of Browet et al. for each fixed-rank r . Comparing the number of iterations it took for the methods to converge to a solution,

the algorithm of Browet et al. was consistent in taking 34 iterations to converge to an approximate solution while LRFBFGS took between 6 and 7 iterations to converge and RSD took 18 iterations to converge. Thus, the optimization algorithms were faster at converging to the same solution as the algorithm of Browet et al. in iterations and time. The efficiency of the LRFBFGS method can be attributed to the fewer function evaluations, gradient evaluations, and number of retractions. Also, even though LRFBFGS has vector transport computations and RSD does not, the vector transport is by parallelization, and due to the intrinsic representation of our

TABLE II

RESULTS FOR ERDŐS-RÉNYI GRAPH WITH COMMUNITY STRUCTURE FOR $(p_{in}, p_{out}) = (0.9, 0.1)$. THE SUBSCRIPT ν INDICATES A SCALE OF 10^ν .

	r	1	2	3	4	5	10	15
RSD	<i>iter</i>	18	18	18	18	18	18	18
	<i>nf</i>	37	37	37	37	37	37	37
	<i>ng</i>	19	19	19	19	19	19	19
	<i>nV</i>	0	0	0	0	0	0	0
	<i>nR</i>	36	36	36	36	36	36	36
	<i>f_f</i>	-1.94 ₁₁	-2.31 ₁₁	-2.41 ₁₁	-2.41 ₁₁	-2.41 ₁₁	-2.41 ₁₁	-2.41 ₁₁
	<i>gf_f</i>	2.04 ₋₁	2.04 ₋₁	2.04 ₋₁	2.04 ₋₁	2.04 ₋₁	2.04 ₋₁	2.04 ₋₁
	<i>rel error</i>	1.10	6.48 ₋₁	2.14 ₋₁	2.13 ₋₁	2.13 ₋₁	2.10 ₋₁	2.06 ₋₁
	<i>gf_f/gf₀</i>	4.89 ₋₇	4.72 ₋₇	4.71 ₋₇	4.71 ₋₇	4.71 ₋₇	4.71 ₋₇	4.71 ₋₇
	<i>t (sec)</i>	8.61 ₋₂	1.26 ₋₁	1.50 ₋₁	2.20 ₋₁	2.26 ₋₁	2.26 ₋₁	3.95 ₋₁
LRFBFGS	<i>iter</i>	6	6	7	7	7	7	7
	<i>nf</i>	8	8	9	9	9	9	9
	<i>ng</i>	7	7	8	8	8	8	8
	<i>nV</i>	44	44	52	52	52	52	52
	<i>nR</i>	7	7	8	8	8	8	8
	<i>f_f</i>	-1.94 ₁₁	-2.31 ₁₁	-2.41 ₁₁	-2.41 ₁₁	-2.41 ₁₁	-2.41 ₁₁	-2.41 ₁₁
	<i>gf_f</i>	1.45 ₋₂	1.73 ₋₁	1.25 ₋₁	1.26 ₋₁	1.26 ₋₁	1.75 ₋₁	2.24 ₋₁
	<i>rel error</i>	1.10	6.48 ₋₁	2.14 ₋₁	2.13 ₋₁	2.13 ₋₁	2.10 ₋₁	2.06 ₋₁
	<i>gf_f/gf₀</i>	3.48 ₋₈	4.01 ₋₇	2.89 ₋₇	2.92 ₋₇	2.92 ₋₇	4.06 ₋₇	5.18 ₋₇
	<i>t (sec)</i>	8.82 ₋₂	5.05 ₋₂	6.27 ₋₂	7.76 ₋₂	9.19 ₋₂	9.17 ₋₂	1.28 ₋₁
Browet's LR Alg	<i>iter</i>	34	34	34	34	34	34	34
	<i>rel error</i>	1.10	6.48 ₋₁	2.14 ₋₁	2.13 ₋₁	2.13 ₋₁	2.10 ₋₁	2.06 ₋₁
	<i>norm diff</i>	8.21 ₋₇	8.19 ₋₇	8.23 ₋₇	8.17 ₋₇	8.26 ₋₇	8.22 ₋₇	8.24 ₋₇
	<i>t (sec)</i>	2.27 ₋₁	2.59 ₋₁	2.93 ₋₁	3.08 ₋₁	5.36 ₋₁	6.23 ₋₁	7.30 ₋₁

TABLE III

RESULTS FOR ERDŐS-RÉNYI GRAPH WITH COMMUNITY STRUCTURE FOR $(p_{in}, p_{out}) = (0.7, 0.3)$. THE SUBSCRIPT ν INDICATES A SCALE OF 10^ν .

	r	1	2	3	4	5	10	15
RSD	<i>iter</i>	18	18	18	18	18	18	18
	<i>nf</i>	37	37	37	37	37	37	37
	<i>ng</i>	19	19	19	19	19	19	19
	<i>nV</i>	0	0	0	0	0	0	0
	<i>nR</i>	36	36	36	36	36	36	36
	<i>f_f</i>	-2.86 ₁₁	-2.88 ₁₁	-2.89 ₁₁	-2.89 ₁₁	-2.89 ₁₁	-2.89 ₁₁	-2.89 ₁₁
	<i>gf_f</i>	2.47 ₋₁	2.47 ₋₁	2.47 ₋₁	2.47 ₋₁	2.47 ₋₁	2.47 ₋₁	2.47 ₋₁
	<i>rel error</i>	7.11 ₋₁	6.13 ₋₁	5.45 ₋₁	5.43 ₋₁	5.41 ₋₁	5.31 ₋₁	5.21 ₋₁
	<i>gf_f/gf₀</i>	4.89 ₋₇	4.89 ₋₇	4.89 ₋₇	4.89 ₋₇	4.89 ₋₇	4.89 ₋₇	4.89 ₋₇
	<i>t (sec)</i>	8.80 ₋₂	1.33 ₋₁	1.56 ₋₁	1.59 ₋₁	2.25 ₋₁	2.45 ₋₁	3.20 ₋₁
LRFBFGS	<i>iter</i>	6	7	7	7	7	7	7
	<i>nf</i>	8	14	9	9	9	9	9
	<i>ng</i>	7	8	8	8	8	8	8
	<i>nV</i>	44	52	52	52	52	52	52
	<i>nR</i>	7	13	8	8	8	8	8
	<i>f_f</i>	-2.86 ₁₁	-2.88 ₁₁	-2.89 ₁₁	-2.89 ₁₁	-2.89 ₁₁	-2.89 ₁₁	-2.89 ₁₁
	<i>gf_f</i>	1.62 ₋₂	3.94 ₋₂	4.52 ₋₂	1.57 ₋₁	1.70 ₋₁	2.91 ₋₁	4.29 ₋₁
	<i>rel error</i>	7.11 ₋₁	6.13 ₋₁	5.45 ₋₁	5.43 ₋₁	5.41 ₋₁	5.31 ₋₁	5.21 ₋₁
	<i>gf_f/gf₀</i>	3.20 ₋₈	7.80 ₋₈	8.93 ₋₈	3.11 ₋₇	3.37 ₋₇	5.76 ₋₇	8.48 ₋₇
	<i>t (sec)</i>	8.87 ₋₂	9.03 ₋₂	6.79 ₋₂	7.40 ₋₂	7.72 ₋₂	1.33 ₋₁	2.04 ₋₁
Browet's LR Alg	<i>iter</i>	34	34	34	34	34	35	35
	<i>rel error</i>	7.11 ₋₁	6.13 ₋₁	5.45 ₋₁	5.43 ₋₁	5.41 ₋₁	5.31 ₋₁	5.21 ₋₁
	<i>norm diff</i>	9.95 ₋₇	9.99 ₋₇	9.97 ₋₇	9.93 ₋₇	9.96 ₋₇	4.55 ₋₇	4.53 ₋₇
	<i>t (sec)</i>	2.81 ₋₁	2.64 ₋₁	3.01 ₋₁	3.26 ₋₁	3.28 ₋₁	6.31 ₋₁	8.53 ₋₁

tangent vectors, it just happens to be the identity. Hence, the vector transport does not slow down the computational time of LRBFGS. After we approximated our similarity measures, we implemented the fast community detection algorithm by Browet et al. described in [7] to extract the roles. The roles extracted from the similarity matrix approximations of the algorithm of Browet et al. were the same as the roles extracted from the similarity matrix approximations of the Riemannian methods.

In Table III, the probability parameters are $p_{in} = 0.7$ and $p_{out} = 0.3$. For these parameters, the blocks in the adjacency

matrix have become less distinct and we can see that there is less of a distinction in the relative error across the fixed-ranks. Also, as before, LRBFGS took the fewest number of iterations and converged to a solution in the smallest time.

The results for the Erdős-Rényi random graph with block cycle structure for $p_{in} = 0.9$ and $p_{out} = 0.1$ are in Table IV. For this example, the methods were all able to detect the numerical rank 3 as there is little change in the relative error after $r = 3$. However, as the blocks become less distinct, the change in the relative error across r becomes less distinct (see Table V). Also, as in the previous example, LRBFGS

TABLE IV

RESULTS FOR ERDŐS-RÉNYI GRAPH WITH BLOCK CYCLE STRUCTURE FOR $(p_{in}, p_{out}) = (0.9, 0.1)$. THE SUBSCRIPT ν INDICATES A SCALE OF 10^ν .

	r	1	2	3	4	5	10	15
RSD	<i>iter</i>	19	19	18	18	18	18	18
	<i>nf</i>	39	39	37	37	37	37	37
	<i>ng</i>	20	20	19	19	19	19	19
	<i>nV</i>	0	0	0	0	0	0	0
	<i>nR</i>	38	38	36	36	36	36	36
	<i>f_f</i>	-1.34 ₁₁	-1.60 ₁₁	-1.88 ₁₁	-1.88 ₁₁	-1.88 ₁₁	-1.88 ₁₁	-1.88 ₁₁
	<i>gf_f</i>	1.81 ₋₁	2.83 ₋₁	1.72 ₋₁	1.72 ₋₁	1.72 ₋₁	1.72 ₋₁	1.72 ₋₁
	<i>rel error</i>	1.08	6.41 ₋₁	2.17 ₋₁	2.16 ₋₁	2.16 ₋₁	2.12 ₋₁	2.09 ₋₁
	<i>gf_f/gf₀</i>	5.33 ₋₇	8.19 ₋₇	4.60 ₋₇	4.60 ₋₇	4.60 ₋₇	4.60 ₋₇	4.60 ₋₇
	<i>t (sec)</i>	8.59 ₋₂	1.33 ₋₁	1.46 ₋₁	1.66 ₋₁	2.35 ₋₁	2.30 ₋₁	3.34 ₋₁
LRBFGS	<i>iter</i>	6	10	9	9	9	9	9
	<i>nf</i>	8	12	11	11	11	11	11
	<i>ng</i>	7	11	10	10	10	10	10
	<i>nV</i>	44	76	68	68	68	68	68
	<i>nR</i>	7	11	10	10	10	10	10
	<i>f_f</i>	-1.34 ₁₁	-1.60 ₁₁	-1.88 ₁₁	-1.88 ₁₁	-1.88 ₁₁	-1.88 ₁₁	-1.88 ₁₁
	<i>gf_f</i>	2.54 ₋₁	1.13 ₋₁	2.72 ₋₁	2.72 ₋₁	2.71 ₋₁	2.69 ₋₁	2.66 ₋₁
	<i>rel error</i>	1.08	6.41 ₋₁	2.17 ₋₁	2.16 ₋₁	2.16 ₋₁	2.12 ₋₁	2.09 ₋₁
	<i>gf_f/gf₀</i>	7.48 ₋₇	3.27 ₋₇	7.27 ₋₇	7.26 ₋₇	7.26 ₋₇	7.21 ₋₇	7.13 ₋₇
	<i>t (sec)</i>	8.65 ₋₂	8.15 ₋₂	8.92 ₋₂	8.27 ₋₂	1.16 ₋₁	4.62 ₋₁	1.80 ₋₁
Browet's LR Alg	<i>iter</i>	34	34	34	34	34	34	34
	<i>rel error</i>	1.08	6.41 ₋₁	2.17 ₋₁	2.16 ₋₁	2.16 ₋₁	2.12 ₋₁	2.09 ₋₁
	<i>norm diff</i>	6.77 ₋₇	6.94 ₋₇	6.93 ₋₇	6.94 ₋₇	6.94 ₋₇	6.88 ₋₇	6.99 ₋₇
	<i>t (sec)</i>	1.68 ₋₁	2.55 ₋₁	3.45 ₋₁	3.19 ₋₁	3.82 ₋₁	6.45 ₋₁	8.51 ₋₁

TABLE V

RESULTS FOR ERDŐS-RÉNYI GRAPH WITH BLOCK CYCLE STRUCTURE FOR $(p_{in}, p_{out}) = (0.7, 0.3)$. THE SUBSCRIPT ν INDICATES A SCALE OF 10^ν .

	r	1	2	3	4	5	10	15
RSD	<i>iter</i>	18	18	18	18	18	18	18
	<i>nf</i>	37	37	37	37	37	37	37
	<i>ng</i>	19	19	19	19	19	19	19
	<i>nV</i>	0	0	0	0	0	0	0
	<i>nR</i>	36	36	36	36	36	36	36
	<i>f_f</i>	-2.63 ₁₁	-2.64 ₁₁	-2.65 ₁₁	-2.65 ₁₁	-2.65 ₁₁	-2.66 ₁₁	-2.66 ₁₁
	<i>gf_f</i>	2.48 ₋₁	2.48 ₋₁	2.37 ₋₁	2.37 ₋₁	2.38 ₋₁	2.38 ₋₁	2.38 ₋₁
	<i>rel error</i>	7.28 ₋₁	6.35 ₋₁	5.60 ₋₁	5.58 ₋₁	5.56 ₋₁	5.45 ₋₁	5.36 ₋₁
	<i>gf_f/gf₀</i>	5.12 ₋₇	5.11 ₋₇	4.90 ₋₇	4.90 ₋₇	4.90 ₋₇	4.90 ₋₇	4.90 ₋₇
	<i>t (sec)</i>	7.98 ₋₂	1.25 ₋₁	1.47 ₋₁	2.17 ₋₁	2.19 ₋₁	2.25 ₋₁	3.31 ₋₁
LRBFGS	<i>iter</i>	6	9	8	8	8	8	8
	<i>nf</i>	8	11	10	10	10	10	10
	<i>ng</i>	7	10	9	9	9	9	9
	<i>nV</i>	44	68	60	60	60	60	60
	<i>nR</i>	7	10	9	9	9	9	9
	<i>f_f</i>	-2.63 ₁₁	-2.64 ₁₁	-2.65 ₁₁	-2.65 ₁₁	-2.65 ₁₁	-2.66 ₁₁	-2.66 ₁₁
	<i>gf_f</i>	2.49 ₋₂	6.80 ₋₂	2.39 ₋₁	2.39 ₋₁	2.57 ₋₁	2.98 ₋₁	3.38 ₋₁
	<i>rel error</i>	7.28 ₋₁	6.35 ₋₁	5.60 ₋₁	5.58 ₋₁	5.56 ₋₁	5.45 ₋₁	5.36 ₋₁
	<i>gf_f/gf₀</i>	5.14 ₋₈	1.40 ₋₇	4.94 ₋₇	4.94 ₋₇	5.31 ₋₇	6.14 ₋₇	6.96 ₋₇
	<i>t (sec)</i>	3.37 ₋₂	8.26 ₋₂	1.05 ₋₁	7.64 ₋₂	8.58 ₋₂	1.21 ₋₁	1.37 ₋₁
Browet's LR Alg	<i>iter</i>	34	34	34	34	34	34	34
	<i>rel error</i>	7.28 ₋₁	6.35 ₋₁	5.60 ₋₁	5.58 ₋₁	5.56 ₋₁	5.45 ₋₁	5.36 ₋₁
	<i>norm diff</i>	9.53 ₋₇	9.55 ₋₇	9.53 ₋₇	9.57 ₋₇	9.60 ₋₇	9.59 ₋₇	9.61 ₋₇
	<i>t (sec)</i>	1.84 ₋₁	2.65 ₋₁	2.95 ₋₁	3.12 ₋₁	3.32 ₋₁	5.78 ₋₁	6.94 ₋₁

reached an approximate solution faster and in fewer iterations than RSD and the algorithm of Browet et al.

VI. CONCLUSION

In this paper, a fixed-rank Riemannian optimization approach to approximate the pairwise node similarity measure used in role extraction was summarized. When applied to Erős-Rényi random graphs containing block structure amenable to the identification of roles, the Riemannian approach computed low-rank similarity approximations very close to those produced by the low-rank iterative algorithm of Browet et al. The Riemannian approach computed the approximations in significantly less time. Future work includes applying the pairwise node similarity measure and our approach to networks from applications such as food webs and gene interactions. For these networks, the number of roles is unknown; hence, the numerical rank of the similarity matrix is unknown. This implies the need to include the choice of rank in the free parameters. Algorithms and convergence theory has been developed recently for such problems in Zhou's dissertation [17]. One of the applications considered in that dissertation is the evaluation of similarity between two graphs using the metric by Blondel et al [4]. A more sophisticated algorithm including rank adaptation for the pairwise node similarity in role model extraction should be possible and a key to an effective and efficient practical role model extraction algorithm.

REFERENCES

- [1] P.-A. Absil, M. Ishteva, L. De Lathauwer and S. Van Huffel, "A Geometric Newton Method for Oja's Vector Field", *Neural Computation*, vol. 21, no. 5, pp. 1415-1433, 2009.
- [2] P.-A. Absil, R. Mahony, R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.
- [3] C. G. Baker. "Riemannian manifold trust-region methods with applications to eigenproblems". PhD dissertation, Florida State University, 2008.
- [4] V. Blondel, A. Gajardo, M. Heymans, P. Senellart, and P. Van Dooren. "A measure of similarity between graph vertices: Applications to synonym extraction and web searching". *SIAM Review*, vol. 46, no. 4, pp. 647-666, 2004.
- [5] A. Browet, "Algorithms for community and role detection in networks," Ph.D. dissertation, Université Catholique de Louvain, Department of Mathematical Engineering, 2014.
- [6] A. Browet and P. Van Dooren, "Low-rank Similarity Measure for Role Model Extraction," In *Proceedings of the 21st International Symposium on Mathematical Theory of Networks and Systems*, pp. 1412 – 1418, 2014.
- [7] A. Browet, P.-A. Absil, and P. Van Dooren, "Fast community detection using local neighbourhood search," *ArXiv e-prints*, 2013.
- [8] T. P. Cason, "Role Extraction in Networks," Ph.D. dissertation, Université Catholique de Louvain, Department of Mathematical Engineering, 2012.
- [9] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75-174, 2010.
- [10] W. Huang, "Optimization algorithms on Riemannian manifolds with applications," Ph.D. dissertation, Florida State University, Department of Mathematics, 2013.
- [11] W. Huang, P.-A. Absil and K. A. Gallivan, "A Riemannian symmetric rank-one trust-region method," *Mathematical Programming*, vol. 150, pp. 179-216, 2015.
- [12] W. Huang, K. A. Gallivan and P.-A. Absil, "A Broyden class of quasi-Newton methods for Riemannian optimization," *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1660-1685, 2015.
- [13] M. Journée, F. Bach, P.-A. Absil and R. Sepulchre, "Low-Rank Optimization on the Cone of Positive Semidefinite Matrices," *SIAM Journal on Optimization*, vol 20, no. 5, pp. 2327-2351, 2010.
- [14] M. A. Porter, J.-P. Onnela and P. J. Mucha, "Communities in networks," *Notices of the AMS*, vol. 56, no. 9, pp. 1082-1097, 2009.
- [15] V. Traag, *Algorithms and Dynamical Models for Communities and Reputation in Social Networks*, Springer Theses, Heidelberg, 2014.
- [16] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, Cambridge University Press, 1994.
- [17] G. Zhou, "Rank-constrained Optimization: A Riemannian manifolds Approach," Ph.D. dissertation, Florida State University, Department of Mathematics, 2015.