Operations Research Letters 44 (2016) 255–259

Contents lists available at ScienceDirect

Operations Research Letters

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

Approximating the minimum rank of a graph via alternating projection

Franklin H.J. Kenter

Department of Computational and Applied Mathematics, Rice University, United States

ARTICLE INFO

Article history:
Received 9 July 2014
Received in revised form
31 January 2016
Accepted 1 February 2016
Available online 8 February 2016

Keywords:
Minimum rank of a graph
Alternating projections
Heuristics
Zero-forcing

ABSTRACT

The minimum rank problem asks to find the minimum rank over all matrices with a given pattern associated with a graph. This problem is NP-hard, and there is no known approximation method. Further, this problem has no straightforward convex relaxation. In this article, a numerical algorithm is given to heuristically approximate the minimum rank using alternating projections. The effectiveness of this algorithm is demonstrated by comparing its results to a related parameter: the zero-forcing number. Using these methods, numerical evidence for the minimum rank graph complement conjecture is provided.

1. Introduction

Seeking the minimum rank of a matrix satisfying given constraints is a well-studied problem with applications in signal processing [11,24], data analysis [9], topological embeddings [10], as well as economics [22]. In general, these problems are NP-hard. However, certain cases, such as the affine minimum rank problem, can be solved exactly or approximately with reliable probabilistic methods and convex optimization [20,27,18].

The problem presented here is to minimize the rank over all symmetric matrices with a specified sparsity pattern given by a graph. This sparsity pattern requires certain entries to be zero and others to be nonzero; the requirement that certain entries be nonzero makes the restriction non-affine and non-convex. Further, these restrictions have no convex relaxation because the zero-matrix lies within the closure of the corresponding set of matrices. Hence, recently developed techniques cannot apply [20,27,18].

The biggest obstacle to study of the minimum rank problem for graphs is that not only is there no known effective method to compute the minimum rank of a graph, but also, there are no efficient approximation algorithms either. Our main contribution is the development of an algorithm to approximate the minimum rank for larger graphs. Computationally, the minimum rank problem for graphs is NP-hard [21], and exact computation, in general, has remained elusive for anything but very small graphs. Recent results have classified all graphs on n vertices with minimum rank 0, 1, 2, $n-2$, $n-1$, and n [2]. Additionally, the minimum rank for certain families of graphs has been determined exactly [3,6,18]. However, calculating, or even effectively estimating, the minimum rank of a graph, in general, still remains open.

The minimum rank problem of a graph has a wide range of additional variants and applications. First, the minimum rank of a graph is a relaxation of the *Colin de Verdière graph invariant* which serves as a strong connection between Schrödinger operators, topological embeddings of graphs, and matrix analysis. In fact, the Colin de Verdière graph invariant gives a spectral characterization for planar graphs [10,25]. In addition, the *zero-forcing number* of a graph introduced in [2] has been used to bound the minimum rank of a graph from below. We will use the zero-forcing number as a comparison of our analysis; details are given in the next section. While the zero-forcing number has yielded several interesting results in accurately computing the minimum rank for certain families of graphs [21], the zero-forcing number itself is also NP-hard [1]. The zero-forcing number is actually adapted from a previous concept known as *power domination* which serves to optimally place power monitoring units within a power network [4], and its study and application continues to this day [14,28]. Further, recently the study of zero-forcing has been extended to determine the controllability of quantum [16,23] and dynamical systems [19].

We present an iterative algorithm which alternates between minimizing the rank of a matrix and fitting the matrix to the given sparsity pattern. This idea is an extension of *the method of alternating projections* pioneered by Von Neumann [26] which has since been perfected by others [7,8] and also extended to manifolds [17]. Details are given in the next section.

E-mail address: franklin.kenter@gmail.com.

http://dx.doi.org/10.1016/j.orl.2016.02.001
0167-6377/

Empirically, this algorithm is effective at accurately computing the minimum rank. We demonstrate the effectiveness of our algorithm by comparing our results to a combinatorial parameter, the zero-forcing number, which bounds the minimum rank from below. Our computational results show that our algorithm is generally close to the theoretical minimum as given by the zero-forcing number. In fact, these results demonstrate that the zero-forcing number is generally a good heuristic approximation for the minimum rank of a graph when the number of vertices is less than 23. This algorithm is efficient for graphs up to 100 vertices. Finally, using our algorithm, we provide numerical evidence for an outstanding conjecture regarding the minimum rank of a graph and its complement. We provide a mathematical foundation that the algorithm does converge when the initial point is sufficiently close to a solution under mild conditions.

This paper is organized as follows. In the next section, we give the preliminaries and definitions. Within Section 3, we present our algorithm and demonstrate its convergence in Section 4. In Section 5, we discuss the computational results of the algorithm and compare them to that of zero-forcing, and in the next section we give numerical evidence to the minimum rank conjecture. Finally, in Section 7, we discuss our conclusions and future work.

2. Preliminaries

Let $G = (V, E)$ be a graph with n vertices and m edges. For our purposes, we use simple graphs with no loops or multiedges. The *adjacency matrix of* G, denoted $\mathbf{A}$, is the $n \times n$ symmetric matrix indexed by the vertices where $\mathbf{A}_{ij} = 1$ whenever $\{i, j\}$ is an edge and $= 0$ otherwise. A weighted adjacency matrix of G is a symmetric $n \times n$ matrix, $\mathbf{W}$, such that for $i \neq j$, $\mathbf{W}_{ij} \neq 0$ whenever $\{i, j\}$ is an edge and $= 0$ otherwise, and the diagonal entries may take any values which may be different from one another. Note that in our case, we specifically disallow zero weights but allow negative weights. A matrix that is a weighted adjacency matrix of G is said be a *representation of* G.

Given a graph G, the minimum rank problem seeks to minimize rank($\mathbf{W}$) over all weighted adjacency matrices of G. The optimum value is called the *minimum rank of* G, denoted $mr(G)$. We will use $amr(G)$ for the output value of our approximation algorithm.

It will be necessary to formulate this problem in terms of manifolds. For our purposes, we assume all manifolds to be smooth and differentiable. We consider the space of $n \times n$ symmetric matrices, $\mathbb{R}^{n\times n}_{sym}$. For the graph G, we define the manifold $\mathcal{S}(G)$ to be the affine subspace in $\mathbb{R}^{n\times n}_{sym}$ consisting of all representations of G. In particular, $\mathcal{S}(G)$ is an affine subspace with select subspaces of lower dimension removed. We will let $\mathcal{S}^*(G)$ denote the affine subspace corresponding to the topological closure of $\mathcal{S}(G)$; that is, $\mathcal{S}^*(G)$ is the set of all representations of G of all subgraphs of G. For $k \leq n$, let $\mathcal{R}_n(k)$ be the manifold of all real-symmetric $n \times n$ matrices of rank at most k. In which case, the minimum rank problem can be formulated as follows:

$$mr(G) := \min k \quad \text{s.t. } \mathcal{S}(G) \cap \mathcal{R}_n(k) \neq \emptyset.$$

Additionally, two manifolds $\mathcal{M}, \mathcal{N}$ within $\mathbb{R}^{n\times n}_{sym}$ intersect transversely at $x \in \mathcal{M} \cap \mathcal{N}$ if the tangent spaces, $T_{\mathcal{M}}, T_{\mathcal{N}}$, have $T_{\mathcal{M}} + T_{\mathcal{N}} = \mathbb{R}^{n\times n}_{sym}$.

The main tool we use in our algorithm is *the method of alternating projections* pioneered by von Neumann [26]:

Proposition 1 (*von Neumann, See for Example [7]*)**.** *Let S and T be closed subspaces of a Hilbert space, $\mathcal{H}$, and let $\mathbf{P}_S$ and $\mathbf{P}_T$ be the orthogonal projection operators onto S and T respectively. Then, for any point $h \in \mathcal{H}$,*

$$\lim_{\ell\to\infty} (\mathbf{P}_S\mathbf{P}_T)^\ell h \in S \cap T \quad \text{and} \quad \lim_{\ell\to\infty} (\mathbf{P}_T\mathbf{P}_S)^\ell h \in S \cap T.$$

That is, one can find an intersection point of two subspaces by alternating projections onto the two subspaces. This concept has since been perfected by others [7,8] and also extended to manifolds [17]. In particular, given a metric space $\mathcal{S}$ with distance metric d and two compact sets $S, T \subset \mathcal{S}$, an *orthogonal projection from* S to T is a function $P : S \to T$ such that for any $s \in S$, $P(s) = \arg\min_{t\in T} d(s, t)$. It should be noted that the minimum exists by compactness, and also, an orthogonal projection from one set to another is not necessarily unique.

Let $\mathbf{M} = \mathbf{Q}\Lambda\mathbf{Q}^T$ be the real orthonormal diagonalization of a real symmetric matrix $\mathbf{M}$ where Λ is a diagonal matrix of eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ with $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$ and $\mathbf{Q}$ is a matrix of (orthogonal) eigenvectors in an order corresponding to the eigenvalues in Λ. A *k-rank approximation* of $\mathbf{M}$ is $\mathbf{M}_k = \mathbf{Q}\Lambda_k\mathbf{Q}^T$ where Λ_k is a diagonal matrix with entries $\lambda_1, \lambda_2, \ldots, \lambda_k, 0, \ldots 0$. Note that a k-rank approximation is unique (up to a sign on the columns of $\mathbf{Q}$) whenever $\mathbf{M}$ has distinct eigenvalues.

We will use $\mathbf{I}$ to denote the identity matrix. Also, for two matrices with the same dimensions, we let $\mathbf{A} \circ \mathbf{M}$ denote the entry-wise (or "Hadamard") product. That is, $(\mathbf{A} \circ \mathbf{M})_{ij} = \mathbf{A}_{ij} \cdot \mathbf{M}_{ij}$.

For matrix norms we will focus upon the *Frobenius norm*, denoted, $\|\mathbf{M}\|_F$, which is given by: $\|\mathbf{M}\|_F^2 = \sum_{i,j} \mathbf{M}_{i,j}^2$. Alternatively, $\|\mathbf{M}\|_F^2 = \sum_k \sigma_k^2$ where σ_k are the singular values of $\mathbf{M}$. For more information, see [15].

In the later part of our paper, we compare our results with the zero-forcing number of a graph. The zero-forcing process originally given in [2] is defined as follows. Start with a set of vertices $S \subset V(G)$ as colored and all other vertices uncolored. If a colored vertex s has all but one of its neighbors colored, then the uncolored neighbor, t, changes from uncolored to colored. In which case, we say *s forces t*. This forcing process continues until all of the vertices of the graph become colored or no more vertices can force. If an initial set S eventually forces the entire graph under this forcing process, S is called a *forcing set*. The *zero-forcing number*, $Z(G)$, is the size of the smallest forcing set of the graph.

The importance of $Z(G)$ in relation to the minimum rank is the following:

Proposition 2 (*AIM Special Work Group [2]*)**.**

$$mr(G) \geq n - Z(G).$$

Later in our paper, we will utilize the previous proposition in order to evaluate our algorithm for the minimum rank, thereby providing a spread of possible values for $mr(G)$.

3. The algorithm

The main idea of the algorithm is to alternate between setting $\mathbf{B} \leftarrow \mathbf{B} \circ (\mathbf{A} + \mathbf{I})$ (zeroing out entries) and setting $\mathbf{B} \leftarrow \mathbf{B}_k$ (taking the k-rank approximation of B) sufficiently many times. Hence, after each step $\mathbf{B}$ is either a k-rank matrix *or* a representation of G (or a subgraph). Note, that the step $\mathbf{B} \leftarrow \mathbf{B} \circ (\mathbf{A} + \mathbf{I})$ does not turn the matrix $\mathbf{B}$ into a representation of G, as it only zeros out entries required to be zero. Rather, it only guarantees that $\mathbf{B}$ is a representation of a subgraph of G (which could be G itself). Hence, if the algorithm converges, it should converge to a matrix that is both rank k and a representation of G or a subgraph of G. Afterwards, it is checked if $\mathbf{B} \in \mathcal{S}(G)$ (that is, if $\mathbf{B}$ is a matrix representation of G). If $\mathbf{B} \in \mathcal{S}(G)$, then $\mathbf{B}$ is a k-rank representation of G, and hence, the algorithm will lower k in order to find a representation with lower rank. Otherwise, the algorithm with raise k in order to find a representation at all. By using a bisection method with regard to k, the algorithm will return the least possible k for which it can find a representation. The main benefit of the algorithm is that not only will it return the minimum rank k it finds, but