

# Disjunctive Branch-and-Bound for Certifiably Optimal Low-Rank Matrix Completion

Dimitris Bertsimas

Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, USA.

ORCID: 0000-0002-1985-1003

dbertsim@mit.edu

Ryan Cory-Wright

Department of Analytics, Marketing and Operations, Imperial Business School, London, UK.

ORCID: 0000-0002-4485-0619

r.cory-wright@imperial.ac.uk

Sean Lo

Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, USA.

ORCID: 0000-0001-8456-6471

seanlo@mit.edu

Jean Pauphilet

Management Science & Operations, London Business School, London, UK.

ORCID: 0000-0001-6352-0984

jpauphilet@london.edu

---

**Abstract.** Low-rank matrix completion consists of computing a matrix of minimal complexity that recovers a given set of observations as accurately as possible. Unfortunately, existing methods for matrix completion are heuristics that, while highly scalable and often identifying high-quality solutions, do not provide an instance-wise certificate of optimality. We reexamine matrix completion with an optimality-oriented eye. We reformulate low-rank matrix completion problems as convex problems over the non-convex set of projection matrices and implement a disjunctive branch-and-bound scheme that solves them to certifiable optimality. Further, we derive a novel and often near-exact class of convex relaxations by decomposing a low-rank matrix as a sum of rank-one matrices and incentivizing that two-by-two minors in each rank-one matrix have determinant zero. In numerical experiments, our new convex relaxations decrease the optimality gap by two orders of magnitude compared to existing attempts, and our disjunctive branch-and-bound scheme solves  $n \times m$  rank- $k$  matrix completion problems to certifiable optimality or near optimality in hours for  $\max\{m, n\} \leq 2500$  and  $k \leq 5$ . Moreover, this reduction in the training error translates into an average 2%–50% reduction in the test set error compared with alternating minimization-based methods.

**Key words:** Low-rank matrix completion; branch-and-bound; eigenvector branching; matrix perspective relaxation; semidefinite programming.

---

## 1. Introduction

This paper proposes a branch-and-bound scheme for solving low-rank matrix completion to provable optimality. Given observations  $A_{i,j} : (i, j) \in \mathcal{I} \subseteq [n] \times [m]$  from a matrix  $A \in \mathbb{R}^{n \times m}$ , we seek a low-

rank matrix  $X \in \mathbb{R}^{n \times m}$  which approximates the observed entries of  $A$ . This admits the formulation:

$$\min_{X \in \mathbb{R}^{n \times m}} \frac{1}{2\gamma} \|X\|_F^2 + \frac{1}{2} \sum_{(i,j) \in I} (X_{i,j} - A_{i,j})^2 \quad \text{s.t.} \quad \text{Rank}(X) \leq k, \quad (1)$$

where the hyperparameter  $k$  bounds the rank of  $X$  and  $\gamma$  controls the degree of regularization. The objective in (1) involves a squared Frobenius norm penalty as done in Keshavan and Montanari (2010), Sun and Zhang (2012). In the literature, such a regularization has been justified to improve out-of-sample prediction and avoid overfitting in high-noise regimes (Keshavan and Montanari 2010) or to make the objective strongly convex in  $X$  (Sun and Zhang 2012). We provide a more detailed discussion on regularization for matrix completion problems in Section EC.1.

Problem (1) has received a great deal of attention since the Netflix Competition (Bell and Koren 2007), and heuristic methods now find high-quality solutions for very large-scale instances. Unfortunately, low-rank problems cannot be modeled as mixed-integer conic optimization problems (see Lubin et al. 2022) and thus are out of scope for traditional combinatorial optimization algorithms. For this reason, developing a certifiably optimal method for low-rank matrix completion would be theoretically meaningful. Moreover, from a practical perspective, Bertsimas et al. (2022) have shown that solving (1) via a provably optimal method gives a smaller out-of-sample mean squared error (MSE) than the MSE obtained via a heuristic. However, to the best of our knowledge, no method solves Problem (1) to provable optimality for  $n, m$  beyond 50 or  $k$  beyond 1 (Naldi 2016, Bertsimas et al. 2022).

In this paper, we design a custom branch-and-bound method that solves Problem (1) to provable optimality using eigenvector disjunctions as branching regions. Numerically, we assess the scalability of our approach for instances with  $\max\{m, n\}$  up to 2500 and  $k$  up to 5. We also show numerically that this approach obtains more accurate solutions than state-of-the-art methods like alternating minimization implementations of the method of Burer and Monteiro (2003), by up to 50% for some small-scale instances with many local optima.

### 1.1. Literature Review

We propose a branch-and-bound algorithm that solves Problem (1) to provable optimality at scale. To develop this algorithm, we require three ingredients: a strategy for recursively partitioning the solution space; a technique for generating high-quality convex relaxations on each partition; and a local optimization strategy for quickly obtaining good solutions in a given partition. To put our contribution into context, we now review all three aspects of the relevant literature and refer to Udell et al. (2016), Bertsimas et al. (2022) for overviews of low-rank optimization.

*Branching* A variety of spatial branch-and-bound schemes have been proposed for generic non-convex quadratically constrained quadratic optimization problems such as (1), since the work of McCormick (1976). Unfortunately, these solvers currently cannot obtain high-quality solutions for problems with more than fifty variables (see Kronqvist et al. 2019, for a benchmark and references). Recently, Bertsimas et al. (2022) reformulated (1) via projection matrices and solved problems with  $50 \times 50$  matrices and  $k = 1$  using off-the-shelf solvers that use McCormick branching regions. Problem-specific branching strategies have been investigated for other non-convex QCQPs, such as two-trust-region problems (Anstreicher 2022) or optimal power flow problems (Kocuk et al. 2018). Our branching strategy, like that of Anstreicher (2022), is based on the idea of computing eigenvectors for non-convex semidefinite constraints that are violated at a relaxed solution, as proposed by Saxena et al. (2010). However, while Saxena et al. (2010) derive cutting planes, we take a different approach and use eigenvectors to develop a branching scheme.

*Relaxations* A number of works have proposed solving Problem (1) via their convex relaxations, originating with Fazel (2002), who proposed replacing the rank minimization objective with a trace term for positive semidefinite factor analysis and matrix completion problems (see also Candès and Recht 2009, for an extension to the asymmetric case). More recently, Bertsimas et al. (2023) proposed a general procedure for obtaining strong bounds to low-rank problems (see also Kim et al. 2022, Li and Xie 2025, for related attempts). Namely, they combined the projection matrix reformulation of Bertsimas et al. (2022) with a matrix analog of perspective functions (c.f. Ebadian et al. 2011) and obtained a new class of convex relaxations for low-rank problems.

*High-quality heuristics* The most popular approach for finding good solutions to Problem (1) is to enforce the rank constraint via the decomposition  $X = UV$ , where the matrices  $U$  and  $V$  have  $k$  columns and  $k$  rows respectively, and to iteratively optimize with respect to  $U$  and  $V$  (Burer and Monteiro 2003). However, the Burer-Monteiro approach is only guaranteed to converge to a local solution and several works have investigated conditions under which convergence to a global optimum is guaranteed for matrix completion (Jain et al. 2013, Hardt 2014, Ge et al. 2016, Ma and Fattahi 2023) or generic semidefinite optimization problems (Boumal et al. 2016, Bhojanapalli et al. 2016, Cifuentes 2021). These guarantees require assumptions about the data-generating process that may be hard to verify on a particular instance (Bandeira et al. 2013) and hold only probabilistically (with high probability). In contrast, a branch-and-bound algorithm provides, for any instance, a feasible solution with an instance-wise certificate of near-optimality (i.e., an optimality gap).

## 1.2. Contributions

We develop a spatial branch-and-bound scheme which solves medium-sized instances of Problem (1) to certifiable (near) optimality.

The key contributions of the paper are threefold. First, we derive an eigenvector-based branching scheme that recursively partitions the feasible region and strengthens the matrix perspective relaxations of Problem (1) more effectively than via McCormick relaxations (see Proposition 1). This approach bears similarity to the eigenvector disjunctive cuts proposed in Saxena et al. (2010); however, instead of a cutting-plane approach, we use our eigenvector disjunctions to construct a spatial branching scheme. Second, we implement this branching strategy in a complete branch-and-bound algorithm. Third, by combining an old characterization of rank via determinant minors, which is new to the matrix completion literature, with the Shor relaxation, we derive new and tighter convex relaxations for matrix completion problems, from which we derive valid inequalities which can strengthen the semidefinite relaxation solved at branch-and-bound nodes.

Our approach improves the scalability of certifiably optimal methods for Problem (1) compared to the state-of-the-art. In particular, we solve instances of Problem (1) with  $\max\{m, n\}$  in the thousands and  $k$  up to 5 to provable optimality in minutes or hours, while previous attempts such as Bertsimas et al. (2022) solve problems where  $n = m = 50$  and  $k = 1$  in hours, but return optimality gaps larger than 100% after hours when  $n \geq 60$  or  $k > 1$ . This is because our approach involves a custom branching scheme that supports imposing semidefinite constraints at the root node and refining relaxations via eigenvector branching, while Bertsimas et al. (2022) used a commercial solver which does not support semidefinite relaxations at the root node and refines relaxations using weaker McCormick disjunctions.

## 1.3. Structure

The rest of the paper is structured as follows:

In Section 2, we derive Problem (1)'s matrix perspective relaxation and propose a disjunctive branching scheme for improving it. With our disjunctive inequalities, we separate an optimal solution to a relaxation from its feasible region via a single eigenvector disjunction into  $2^k$  subregions. In contrast, using McCormick disjunctions, we prove in Section 2.2 that disjuncting over less than  $2^{n-4}$  subregions never separates a solution from a McCormick relaxation.

In Section 3, we design a spatial branch-and-bound algorithm that converges to a certifiably optimal solution of Problem (1). We also discuss different aspects of our algorithmic implementation,

including node selection, branching strategy, and an alternating minimization algorithm to obtain high-quality feasible solutions.

In Section 4, we leverage a characterization of the rank of a matrix via its determinant minors to develop a novel convex relaxation for low-rank problems, and derive new valid inequalities. These valid inequalities can be used to tighten the semidefinite relaxations solved at each node in our branch-and-bound scheme.

In Section 5, we investigate the performance of our branch-and-bound scheme via a suite of numerical experiments. We identify that our new convex relaxation and valid inequalities reduce (sometimes substantially) the optimality gap at the root node and that our branch-and-bound scheme solves instances of Problem (1) to certifiable (near) optimality when  $\max\{m, n\} = 2500$  in minutes or hours. We also verify that running our branch-and-bound method with a time limit of minutes or hours yields matrices with an out-of-sample MSE up to 50% lower than that of state-of-the-art heuristics, such as the method of Burer and Monteiro (2003) and its recent refinements.

#### 1.4. Notations

We let non-boldface characters, lowercase bold-faced characters, and uppercase bold-faced characters (e.g.,  $b, \mathbf{x}, \mathbf{A}$ ) denote scalars, vectors, and matrices, respectively. Calligraphic uppercase characters such as  $\mathcal{Z}$  denote sets. We let  $[n]$  denote the running set of indices  $\{1, \dots, n\}$ . We let  $\mathbf{e}$  denote the vector of ones,  $\mathbf{0}$  the vector of all zeros,  $\mathbf{e}_j$  the  $j$ -th vector of the canonical basis, and  $\mathbb{I}$  the identity matrix. We let  $\mathcal{S}^n$  denote the set of  $n \times n$  symmetric matrices, and  $\mathcal{S}_+^n$  denote  $n \times n$  positive semidefinite matrices. We let  $\mathcal{Y}_n^k := \{ \mathbf{Y} \in \mathcal{S}^n : \mathbf{Y}^2 = \mathbf{Y}, \text{tr}(\mathbf{Y}) \leq k \}$  denote the set of  $n \times n$  orthogonal projection matrices with rank at most  $k$ . Its convex hull is  $\text{Conv}(\mathcal{Y}_n^k) = \{ \mathbf{Y} \in \mathcal{S}^n : \mathbf{0} \preceq \mathbf{Y} \preceq \mathbb{I}, \text{tr}(\mathbf{Y}) \leq k \}$ .

## 2. Mixed-Projection Formulations, Relaxations, and Disjunctions

In this section, we derive Problem (1)'s semidefinite relaxation in Section 2.1, and refine its relaxation via eigenvector disjunctions. We also establish that our proposed strategy, which we refer to as 'eigenvector disjunctions' or 'eigenvector branching', allows us to separate an optimal solution to the original semidefinite relaxation by branching on a single set of eigenvector disjunctions. In comparison, we justify the poor performance of traditional disjunctions based on McCormick inequalities by proving in Section 2.2 that McCormick disjunctions with less than  $2^{n-4}$  regions cannot improve the semidefinite relaxation.

## 2.1. Mixed-Projection Formulations and Relaxations

Motivated by Bertsimas et al. (2022, 2023), we introduce a trace- $k$  projection matrix  $Y \in \mathcal{Y}_n^k$  to model the rank of  $X$  via the bilinear constraint  $X = YX$ . Hence, we can replace the rank constraint on  $X$  by a linear constraint on  $Y$ :  $\text{tr}(Y) \leq k$ . By Bertsimas et al. (2023, Theorem 1), we enforce the bilinear constraint implicitly via the domain of a matrix perspective function and write (1) as:

$$\min_{Y \in \mathcal{Y}_n^k} \min_{\substack{X \in \mathbb{R}^{n \times m} \\ \Theta \in \mathcal{S}^m}} \frac{1}{2\gamma} \text{tr}(\Theta) + \frac{1}{2} \sum_{(i,j) \in \mathcal{I}} (X_{i,j} - A_{i,j})^2 \quad \text{s.t.} \quad \begin{pmatrix} Y & X \\ X^\top & \Theta \end{pmatrix} \succeq \mathbf{0}. \quad (2)$$

By relaxing  $\mathcal{Y}_n^k$  to its convex hull,  $\text{Conv}(\mathcal{Y}_n^k) = \{Y \in \mathcal{S}_+^n : Y \preceq \mathbb{I}, \text{tr}(Y) \leq k\}$ , we immediately have the following semidefinite relaxation (see also Bertsimas et al. 2022, Lemma 4):

$$\min_{\substack{Y \in \text{Conv}(\mathcal{Y}_n^k) \\ U \in \mathbb{R}^{n \times k}}} \min_{\substack{X \in \mathbb{R}^{n \times m} \\ \Theta \in \mathcal{S}^m}} \frac{1}{2\gamma} \text{tr}(\Theta) + \frac{1}{2} \sum_{(i,j) \in \mathcal{I}} (X_{i,j} - A_{i,j})^2 \quad \text{s.t.} \quad \begin{pmatrix} Y & X \\ X^\top & \Theta \end{pmatrix} \succeq \mathbf{0}, Y \succeq UU^\top, \quad (3)$$

where we introduce the redundant matrix variable  $U \in \mathbb{R}^{n \times k}$  in order to obtain a tractable set of disjunctions in the next section. Note that if  $Y$  is a rank- $k$  projection matrix then we have that  $Y = UU^\top$  for some  $U \in \mathbb{R}^{n \times k}$ . In our relaxation, we impose the semidefinite constraint  $Y \succeq UU^\top$ , with the idea to enforce the reverse inequality via the disjunctions we are about to derive.

When we solve (3), one of two situations occurs. Either we obtain a solution  $\hat{Y}$  with binary eigenvalues and the relaxation is tight, or  $\hat{Y}$  has strictly fractional eigenvalues. In the latter case, we improve the relaxation by separating  $\hat{Y}$  from the feasible region using disjunctive programming techniques, as we propose in the rest of this section.

## 2.2. Failure of McCormick Disjunctions

To further justify the need for a new type of disjunction, we emphasize that most commercial spatial branch-and-bound codes for non-convex quadratic optimization problems rely on McCormick disjunctions (McCormick 1976). However, these disjunctions can be ineffective, as shown in a different context by Khajavirad (2023). In this section, we support theoretically the ineffectiveness of McCormick disjunctions: For our matrix completion problem, we show that the value of the relaxation (3) is left unimproved after  $2^{n-4}$  McCormick disjunctions. Our theoretical result echoes the poor numerical performance of McCormick disjunctions we observe in Section 5.

Formally, for each entry of  $\mathbf{U}$ ,  $U_{i,j}$ , we start with box constraints  $U_{i,j} \in [-1, 1]$  and recursively partition the region into smaller boxes  $U_{i,j} \in [\underline{U}_{i,j}, \bar{U}_{i,j}]$ . We also introduce variables  $V_{i,j_1,j_2}$  to model the product  $U_{i,j_1}U_{i,j_2}$ . The convex hull of the set  $\left\{ (v, x, y) \mid v = xy, (x, y) \in [\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}] \right\}$ , denoted  $\mathcal{M}(\underline{x}, \bar{x}, \underline{y}, \bar{y})$ , is given in McCormick (1976) as:

$$\left\{ (v, x, y) \mid \begin{array}{l} (x, y) \in [\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}], \\ \max\{\underline{x}y + \underline{y}x - \underline{x}\underline{y}, \bar{x}\bar{y} + \bar{y}\bar{x} - \bar{x}\bar{y}\} \leq v \leq \min\{\underline{y}x + \bar{x}\bar{y} - \bar{x}\underline{y}, \bar{y}x + \underline{x}\underline{y} - \underline{x}\bar{y}\} \end{array} \right\}.$$

For our matrix completion problem, this yields a disjunction over relaxations of the form:

$$\begin{aligned} \min_{\substack{\mathbf{Y} \in \text{Conv}(\mathcal{Y}_n^k), \\ \mathbf{U} \in \mathbb{R}^{n \times k}, \\ \mathbf{V} \in \mathbb{R}^{n \times k \times k}}} \min_{\mathbf{X} \in \mathbb{R}^{n \times m}, \Theta \in \mathcal{S}^m} \frac{1}{2\gamma} \text{tr}(\Theta) + \frac{1}{2} \sum_{(i,j) \in \mathcal{I}} (X_{i,j} - A_{i,j})^2 \quad \text{s.t.} \quad \begin{pmatrix} \mathbf{Y} & \mathbf{X} \\ \mathbf{X}^\top & \Theta \end{pmatrix} \succeq \mathbf{0}, \mathbf{Y} \succeq \mathbf{U}\mathbf{U}^\top, \\ \sum_{i=1}^n V_{i,j,j} = 1 \quad \forall j \in [k], \sum_{i=1}^n V_{i,j_1,j_2} = 0 \quad \forall j_1, j_2 \in [k] : j_1 \neq j_2, \\ (V_{i,j_1,j_2}, U_{i,j_1}, U_{i,j_2}) \in \mathcal{M}(\underline{U}_{i,j_1}, \bar{U}_{i,j_1}, \underline{U}_{i,j_2}, \bar{U}_{i,j_2}), \quad \forall i \in [n], \forall j_1, j_2 \in [k]. \end{aligned} \quad (4) \end{aligned}$$

We now prove that a large family of McCormick disjunctions cannot improve on our root node relaxation (proof deferred to Appendix EC.2):

**PROPOSITION 1.** *Consider Problem (4) and suppose that for every column  $j$  in  $\mathbf{U}$  we have a disjunction  $\cup_i [\underline{U}_{i,j}^t, \bar{U}_{i,j}^t]$  for each  $i$  in some index set  $\mathcal{I}(j) \subseteq [n]$ , but  $[\underline{U}_{i',j}, \bar{U}_{i',j}] = [-1, 1]$  for all other indices  $i' \notin \mathcal{I}(j)$ . Further assume that  $|\mathcal{I}(j_1)| \leq n - 1$  and  $|\mathcal{I}(j_1) \cup \mathcal{I}(j_2)| \leq n - 4$  for any  $j_1, j_2 \in [k], j_1 \neq j_2$ . Then, Problem (4) possesses the same optimal objective value as Problem (3).*

Proposition 1 reveals that, with McCormick disjunctions, we must expand a number of nodes strictly greater than  $2^{n-4}$  before improving upon the root node relaxation—indeed,  $|\mathcal{I}(j_1) \cup \mathcal{I}(j_2)| \leq |\cup_{j \in [k]} \mathcal{I}(j)|$  and  $|\cup_{j \in [k]} \mathcal{I}(j)|$  is lower than the base-2 logarithm of the number of nodes expanded. In addition, the condition of Proposition 1 holds if  $|\mathcal{I}(j)| \leq (n - 4)/2, \forall j \in [k]$  so we can even construct a disjunction with  $2^{k(n-4)/2}$  nodes expanded, yet that does not improve the root node relaxation. This behavior will contrast with our eigenvector disjunction, which separates the solution to the root node relaxation with a single cut and often improves (numerically) our root node relaxation immediately. This observation complements a body of work demonstrating that McCormick relaxations are often dominated by other relaxations in both theory and practice (Kocuk et al. 2018, Khajavirad 2023). Moreover, it challenges the ongoing practice in commercial non-convex solvers, which use McCormick disjunctions.

### 2.3. Improving Relaxations via Eigenvector Branching

We generalize the lifted approach proposed by Saxena et al. (2010) for mixed-integer QCQO to optimization over rank- $k$  projection matrices, where  $k > 1$  is allowed. While Saxena et al. (2010) use their disjunctions to derive disjunctive cuts and design a cutting-plane approach, we construct a branching scheme based on these disjunctions.

For  $Y$  to be a rank- $k$  projection matrix, it should satisfy  $\hat{Y} \preceq \hat{U}\hat{U}^\top$  in an optimal solution  $(\hat{Y}, \hat{U}, \hat{X}, \hat{\Theta})$  of the semidefinite relaxation (3). Owing to the constraint  $Y \succeq UU^\top$ , this would imply that  $\hat{Y} = \hat{U}\hat{U}^\top$ , and thus  $\hat{Y}$  is a rank- $k$  matrix. Since  $\text{Span}(\hat{X}) \subseteq \text{Span}(\hat{Y})$  (Bertsimas et al. 2023), we can conclude that  $\hat{X}$  solves the original rank-constrained problem (1). Otherwise, we have  $\hat{Y} \not\preceq \hat{U}\hat{U}^\top$ , i.e., there exists an  $\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 = 1$  such that  $\|\hat{U}^\top \mathbf{x}\|_2^2 < \mathbf{x}^\top \hat{Y} \mathbf{x}$ . Therefore, we would like to impose the (non-convex) inequality  $\mathbf{x}^\top Y \mathbf{x} \leq \|\mathbf{U}^\top \mathbf{x}\|_2^2$  via a disjunction.

Observe that, for any feasible  $U$ ,  $\|\mathbf{U}^\top \mathbf{x}\|_2^2 \leq 1$ . First, we decompose the square-norm term  $\|\mathbf{U}^\top \mathbf{x}\|_2^2 = \sum_{j \in [k]} (\mathbf{e}_j^\top \mathbf{U}^\top \mathbf{x})^2$ . Then we bound each term by a piecewise-linear upper approximation. For any  $u_0 \in (-1, 1)$ , the function  $u \in [-1, 1] \mapsto u^2$  has the following piecewise upper approximation with breakpoint  $u_0$ :

$$u^2 \leq \begin{cases} -u + uu_0 + u_0 & \text{if } u \in [-1, u_0], \\ u + uu_0 - u_0 & \text{if } u \in [u_0, 1]. \end{cases} \quad (5)$$

Using  $u_0 = \mathbf{e}_j^\top \hat{U}^\top \mathbf{x}$  yields the following upper bound:

$$(\mathbf{e}_j^\top \mathbf{U}^\top \mathbf{x})^2 \leq \begin{cases} \mathbf{x}^\top \hat{U} \mathbf{U}^\top \mathbf{x} + \mathbf{e}_j^\top (\hat{U} - \mathbf{U})^\top \mathbf{x} & \text{if } \mathbf{e}_j^\top \mathbf{U}^\top \mathbf{x} \in [-1, \mathbf{e}_j^\top \hat{U}^\top \mathbf{x}], \\ \mathbf{x}^\top \hat{U} \mathbf{U}^\top \mathbf{x} + \mathbf{e}_j^\top (\mathbf{U} - \hat{U})^\top \mathbf{x} & \text{if } \mathbf{e}_j^\top \mathbf{U}^\top \mathbf{x} \in [\mathbf{e}_j^\top \hat{U}^\top \mathbf{x}, 1]. \end{cases}$$

and the following disjunction over  $2^k$  regions:

$$\bigvee_{L \subseteq [k]} \left\{ Y \left| \begin{array}{l} \mathbf{e}_j^\top \mathbf{U}^\top \mathbf{x} \in [-1, \mathbf{e}_j^\top \hat{U}^\top \mathbf{x}] \quad \forall j \in L, \\ \mathbf{e}_j^\top \mathbf{U}^\top \mathbf{x} \in [\mathbf{e}_j^\top \hat{U}^\top \mathbf{x}, 1] \quad \forall j \in [k] \setminus L, \\ \mathbf{x}^\top Y \mathbf{x} \leq \mathbf{x}^\top \hat{U} \mathbf{U}^\top \mathbf{x} + \sum_{j \in L} \mathbf{e}_j^\top (\hat{U} - \mathbf{U})^\top \mathbf{x} + \sum_{j \in [k] \setminus L} \mathbf{e}_j^\top (\mathbf{U} - \hat{U})^\top \mathbf{x} \end{array} \right. \right\}. \quad (6)$$

As a result, we can strengthen our convex relaxation (3) by following this disjunction and optimizing over the  $2^k$  resulting convex problems. We formalize this result in the following proposition:

**PROPOSITION 2.** *Let  $(\hat{Y}, \hat{U}, \hat{X}, \hat{\Theta})$  be an optimal solution to (3) so that  $\|\hat{U}^\top \mathbf{x}\|_2^2 < \mathbf{x}^\top \hat{Y} \mathbf{x}$  for some vector  $\mathbf{x} \in \mathbb{R}^n$  such that  $\|\mathbf{x}\|_2 = 1$ . Then, any solution to (3) with  $Y = UU^\top$  satisfies (6), but  $(\hat{Y}, \hat{U}, \hat{X}, \hat{\Theta})$  does not satisfy (6).*

*Proof of Proposition 2* Let  $U, Y$  be matrices so that  $UU^\top = Y$ . We have  $\mathbf{x}^\top (Y - UU^\top) \mathbf{x} = 0$  for any vector  $\mathbf{x}$ , which implies the disjunction is equivalent to requiring that, for each  $j$ , either  $e_j^\top U^\top \mathbf{x} \in [-1, e_j^\top \hat{U}^\top \mathbf{x}]$  or  $e_j^\top U^\top \mathbf{x} \in [e_j^\top \hat{U}^\top \mathbf{x}, 1]$ , which is trivially true.

On the other hand, at the point  $U = \hat{U}$ , our approximation of the constraint  $\mathbf{x}^\top Y \mathbf{x} \leq \|U^\top \mathbf{x}\|_2^2$  is tight so our disjunction requires  $\mathbf{x}^\top (\hat{U} \hat{U}^\top - \hat{Y}) \mathbf{x} \geq 0$ , which contradicts  $\mathbf{x}^\top (\hat{U} \hat{U}^\top - \hat{Y}) \mathbf{x} < 0$ .  $\square$

Proposition 2 reveals that the disjunction (6) in Problem (3) separates the optimal solution to (3) whenever it is infeasible for the original problem (1).

**REMARK 1.** For tractability considerations, we work on a lifted version of the semidefinite relaxation (3), which involves additional variables  $U \in \mathbb{R}^{n \times k}$  such that  $Y = UU^\top$ . Instead, we could derive a set of disjunctions on  $Y$  alone by manipulating the constraint  $Y^2 = Y$ . Unfortunately, while simpler to derive, this results in a set of  $2^n$ , rather than  $2^k$ , disjunctions, which is problematic as  $k \ll n$  in practical low-rank matrix completion problems. That being said, working on a lifted formulation may create symmetry issues: if  $(Y, U, X, \Theta)$  is a solution to the relaxation, then so is  $(Y, U\Pi, X, \Theta)$  for any  $k$ -by- $k$  permutation matrix  $\Pi$ . Hence, Problem (3)'s lower bound may not actually improve until we apply a disjunction for each permutation of the columns of  $U$ , which is likely computationally prohibitive in practice. We alleviate these issues by exploring branching strategies and symmetry-breaking constraints in Section 3.

**REMARK 2.** We obtain our  $2^k$  disjunction by constructing a piecewise linear upper approximation of  $u \in [-1, 1] \mapsto u^2$  with two pieces, i.e., a single breakpoint  $u_0$ . We can construct tighter approximations by considering more pieces/breakpoints. For example, with the two breakpoints  $\{-|u_0|, +|u_0|\}$ , we obtain the following 3-piece linear approximation

$$u^2 \leq \begin{cases} -u - u|u_0| - |u_0| & \text{if } u \in [-1, -|u_0|], \\ u_0^2 & \text{if } u \in [-|u_0|, |u_0|], \\ u + u|u_0| - |u_0| & \text{if } u \in [|u_0|, 1], \end{cases} \quad (7)$$

and with the three breakpoints  $\{-|u_0|, 0, |u_0|\}$ ,

$$u^2 \leq \begin{cases} -u - u|u_0| - |u_0| & \text{if } u \in [-1, -|u_0|], \\ -u|u_0| & \text{if } u \in [-|u_0|, 0], \\ u|u_0| & \text{if } u \in [0, |u_0|], \\ u + u|u_0| - |u_0| & \text{if } u \in [|u_0|, 1]. \end{cases} \quad (8)$$

In general, introducing more pieces results in a stronger but more expensive to compute disjunctive bound. Therefore, there is a trade-off between the quality of the piecewise linear upper approximation and the number of convex optimization problems that need to be solved to compute the corresponding bound, which we explore numerically in Section 5.

### 3. A Custom Branch-and-Bound Algorithm

In this section, we propose a nonlinear branch-and-bound framework to obtain  $\epsilon$ -optimal and  $\epsilon$ -feasible solutions to Problem (2) by recursively solving its semidefinite relaxation (3), partitioning the feasible region via the eigenvector disjunctions derived in Section 2. We first provide a high-level description of our approach and relevant pseudocode, before giving an overview of our branch-and-bound design decisions. Finally, we discuss how alternating minimization can be applied in our branch-and-bound tree to discover high-quality incumbent solutions.

#### 3.1. Pseudocode and Convergence Guarantees

At the root node, we solve the matrix perspective relaxation (3). Upon solving the relaxation, we obtain a solution  $(\hat{Y}, \hat{U})$  and compute the smallest eigenvector of  $\hat{U}\hat{U}^\top - \hat{Y}$ , denoted  $x$ . If  $x^\top (\hat{U}\hat{U}^\top - \hat{Y})x \geq -\epsilon$ , then  $\hat{Y}$  is a projection matrix up to tolerance  $\epsilon$  (since  $\hat{Y} \succeq \hat{U}\hat{U}^\top$ ) and solves (2) to optimality and  $\epsilon$ -feasibility. Otherwise, we have  $x^\top (\hat{U}\hat{U}^\top - \hat{Y})x < -\epsilon$  and we apply a disjunction as in Equation (6). Each of the  $2^k$  pieces of the disjunction defines a new node or subproblem for our branch-and-bound algorithm, which is appended to the list of open nodes.

We proceed recursively by selecting a subproblem from the list of open nodes, solving the corresponding convex relaxation, and imposing a new disjunction which creates more nodes, until we identify an  $\epsilon$ -feasible solution  $(\hat{Y}, \hat{U})$ .

We fathom (or prune) a node in two situations. First, if  $\lambda_{\min}(\hat{U}\hat{U}^\top - \hat{Y}) \geq -\epsilon$  then the relaxed solution is an  $\epsilon$ -feasible solution which provides a global upper bound on (2)'s optimal value and we need not impose additional disjunctions at this node. As we establish in Theorem 1, this is guaranteed to occur at a sufficiently large tree depth for any  $\epsilon > 0$ . Second, if the value of the node

relaxation is within  $\epsilon$  of our global upper bound, then no solution with the constraints inherited from this node can improve upon an incumbent solution by more than  $\epsilon$ .

We describe our branch-and-bound scheme in Algorithm 1. In Algorithm 1, we describe a subproblem as a set of constraints  $\mathcal{Q}$  and a depth  $t$ . The constraints in  $\mathcal{Q}$  stem from the eigenvector cuts described in Section 2 and are parameterized by the solution at which they were obtained,  $(\hat{U}, \hat{Y})$ , the most negative eigenvector  $\mathbf{x}$ , and a vector  $\mathbf{z} \in [q]^k$  encoding which one of the  $q^k$  regions we are considering. For conciseness, we denote by  $\bar{f}_{i,q}(\cdot; u_0)$  the  $i$ -th piece of the piecewise linear upper-approximation with  $q$  pieces of  $u \mapsto u^2$ , obtained from the breakpoint  $u_0$ .

To verify that Algorithm 1 converges, we prove in Theorem 1 that a breadth-first node selection strategy which iteratively minimizes (3) and imposes a disjunctive cut of the form (6), according to a most negative eigenvector of  $\mathbf{U}\mathbf{U}^\top - \mathbf{Y}$ , an optimal solution to (3) at the current iterate, eventually identifies an  $\epsilon$ -feasible solution (where  $\lambda_{\min}(\mathbf{U}\mathbf{U}^\top - \mathbf{Y}) > -\epsilon$ ) within a finite number of iterations for any  $\epsilon > 0$ . The proof of Theorem 1 reveals that any node sufficiently deep in our search tree which contains a feasible solution to the semidefinite relaxation (intersected with appropriate constraints from our disjunctions) gives an  $\epsilon$ -feasible solution to the original problem. Therefore, due to the enumerative nature of branch-and-bound, this result verifies that our approach converges for any search and node selection strategy, possibly with a different number of pieces in each disjunction.

**THEOREM 1.** *Let  $(\mathbf{Y}_L, \mathbf{U}_L)$  denote a solution generated by the  $L$ -th iterate of the procedure:*

- *For each  $\ell \in \mathbb{N}$ , set  $(\mathbf{Y}_\ell, \mathbf{U}_\ell)$  according to the optimal solution of (3), possibly with disjunctive cuts of the form (6).*

- *If  $\lambda_{\min}(\mathbf{U}_\ell \mathbf{U}_\ell^\top - \mathbf{Y}_\ell) \geq -\epsilon$  then terminate.*

- *Else, impose the disjunctive cut (6) in (3) with the most negative eigenvector  $\mathbf{x}_\ell$  of  $(\mathbf{U}_\ell \mathbf{U}_\ell^\top - \mathbf{Y}_\ell)$ :*

$$\mathbf{x}_\ell \in \arg \min_{\mathbf{x} \in \mathbb{R}^k: \|\mathbf{x}\|_2=1} \langle \mathbf{x} \mathbf{x}^\top, \mathbf{U}_\ell \mathbf{U}_\ell^\top - \mathbf{Y}_\ell \rangle.$$

*For any  $\epsilon > 0$  there exists an  $L \in \mathbb{N}$  so that  $\lambda_{\min}(\mathbf{U}_L \mathbf{U}_L^\top - \mathbf{Y}_L) \geq -\epsilon$  and  $\mathbf{Y}_L$  is an optimal,  $\epsilon$ -feasible solution to (2). Moreover, suppose we set  $\epsilon \rightarrow 0$ . Then, any limit point of  $\{\mathbf{Y}_\ell\}_{\ell=1}^\infty$  defines an optimal solution to (2).*

Note that an ‘‘optimal,  $\epsilon$ -feasible’’ solution refers to a solution with objective value at least as good as any solution to (2), which is  $\epsilon$ -feasible. This is also known as a ‘‘superoptimal’’ solution.

*Proof of Theorem 1* In this proof, for the  $\ell$ -th iterate  $(\mathbf{Y}_\ell, \mathbf{U}_\ell)$ , we denote  $\mathbf{U}_\ell^t$  the  $t$ -th column of  $\mathbf{U}_\ell$ . Suppose that at the  $L$ -th iterate this procedure has not converged. Then, since  $(\mathbf{Y}_L, \mathbf{U}_L)$  satisfies the disjunction (6) for any  $\mathbf{x}_\ell$  such that  $\ell < L$ , we have that

$$\sum_{t=1}^k \langle \mathbf{x}_\ell^\top \mathbf{U}_\ell^t, \mathbf{x}_\ell^\top \mathbf{U}_L^t \rangle + |(\mathbf{U}_\ell^t - \mathbf{U}_L^t)^\top \mathbf{x}_\ell| \geq \langle \mathbf{x}_\ell \mathbf{x}_\ell^\top, \mathbf{Y}_L \rangle. \quad (9)$$

But  $(\mathbf{Y}_\ell, \mathbf{U}_\ell)$  was not  $\epsilon$ -feasible, and thus we have that

$$\langle \mathbf{x}_\ell \mathbf{x}_\ell^\top, \mathbf{Y}_\ell \rangle - \sum_{t=1}^k \langle \mathbf{x}_\ell^\top \mathbf{U}_\ell^t, \mathbf{x}_\ell^\top \mathbf{U}_\ell^t \rangle > \epsilon. \quad (10)$$

Adding these inequalities then reveals that

$$\langle \mathbf{x}_\ell \mathbf{x}_\ell^\top, \mathbf{Y}_\ell - \mathbf{Y}_L \rangle + \sum_{t=1}^k \langle \mathbf{x}_\ell^\top \mathbf{U}_\ell^t, \mathbf{x}_\ell^\top (\mathbf{U}_L^t - \mathbf{U}_\ell^t) \rangle + \sum_{t=1}^k |\mathbf{x}_\ell^\top (\mathbf{U}_\ell^t - \mathbf{U}_L^t)| > \epsilon. \quad (11)$$

Next, since  $|\mathbf{x}_\ell^\top \mathbf{U}_\ell^t| \leq 1$  by construction, using this identity and taking absolute values allows us to conclude that

$$|\langle \mathbf{x}_\ell \mathbf{x}_\ell^\top, \mathbf{Y}_\ell - \mathbf{Y}_L \rangle| + 2 \sum_{t=1}^k |\mathbf{x}_\ell^\top (\mathbf{U}_\ell^t - \mathbf{U}_L^t)| > \epsilon. \quad (12)$$

Moreover, by applying the Cauchy-Schwarz inequality to both terms in this inequality we obtain  $\langle \mathbf{x}_\ell \mathbf{x}_\ell^\top, \mathbf{Y}_\ell - \mathbf{Y}_L \rangle \leq \|\mathbf{Y}_L - \mathbf{Y}_\ell\|_F$  and  $\sum_{t=1}^k |\mathbf{x}_\ell^\top (\mathbf{U}_\ell^t - \mathbf{U}_L^t)| \leq \sum_{t=1}^k \|\mathbf{U}_\ell^t - \mathbf{U}_L^t\|_2 = \|\mathbf{U}_\ell - \mathbf{U}_L\|_{2,1}$ , where  $\|\mathbf{X}\|_{2,1} := \sum_{j \in [k]} \left( \sum_{i \in [n]} X_{i,j}^2 \right)^{1/2}$  denotes the  $\|\cdot\|_{2,1}$  norm. Further, by norm equivalence we have  $\|\mathbf{U}_\ell - \mathbf{U}_L\|_{2,1} \leq \sqrt{k} \|\mathbf{U}_\ell - \mathbf{U}_L\|_F$ . Combining these results yields

$$\|\mathbf{Y}_\ell - \mathbf{Y}_L\|_F + 2\sqrt{k} \|\mathbf{U}_\ell - \mathbf{U}_L\|_F > \epsilon. \quad (13)$$

That is, with respect to the norm  $n_k : (\mathbf{Y}, \mathbf{U}) \mapsto \|\mathbf{Y}\|_F + 2\sqrt{k} \|\mathbf{U}\|_F$ , our procedure never visits any ball of radius  $\frac{\epsilon}{2}$  twice. Moreover, the set of feasible  $(\mathbf{Y}, \mathbf{U})$  is bounded via  $\mathbb{I} \succeq \mathbf{Y} \succeq \mathbf{U}\mathbf{U}^\top$ ,  $\text{tr}(\mathbf{Y}) \leq k$ . Therefore, we have that  $\|\mathbf{Y}\|_F \leq \sqrt{k}$  and  $\|\mathbf{U}\|_F \leq \sqrt{k}$ . Thus,  $n_k(\mathbf{Y}, \mathbf{U}) \leq \sqrt{k} + 2k$  and there are finitely many non-overlapping balls of radius  $\frac{\epsilon}{2}$  which contain a point in the feasible region. Therefore, for any  $\epsilon > 0$ , there exists some  $L \in \mathbb{N}$  such that our procedure converges within  $L$  iterations.

Finally, let  $(\mathbf{Y}^\star, \mathbf{U}^\star)$  be a limit point of the sequence  $\{(\mathbf{Y}_\ell, \mathbf{U}_\ell)\}_{\ell=1}^\infty$ , and  $(\mathbf{X}^\star, \mathbf{\Theta}^\star)$  be the additional variables in the associated optimal solution. In the limit, we must have  $\mathbf{Y}^\star = \mathbf{U}^\star \mathbf{U}^{\star\top}$  so  $\text{rank}(\mathbf{Y}^\star) \leq \text{rank}(\mathbf{U}^\star) \leq k$ . Let us define  $\mathbf{P}$ , the orthogonal projection onto  $\text{span}(\mathbf{Y}^\star)$ . The matrix  $\mathbf{P} \in \mathcal{Y}_n^k$  because  $\text{rank}(\mathbf{P}) = \text{rank}(\mathbf{Y}^\star) \leq k$ . Observe that  $(\mathbf{Y}^\star, \mathbf{U}^\star, \mathbf{X}^\star, \mathbf{\Theta}^\star)$  solves a relaxation of (2), so it suffices to show that  $(\mathbf{P}, \mathbf{U}^\star, \mathbf{X}^\star, \mathbf{\Theta}^\star)$  is feasible for (2) and attains the same objective value. Feasibility follows from the fact that  $\mathbb{I} \succeq \mathbf{Y}^\star$  so  $\mathbf{P} \succeq \mathbf{Y}^\star$ . Objective value is obviously unchanged as it depends on  $\mathbf{X}^\star$  and  $\mathbf{\Theta}^\star$  only.  $\square$

---

**Algorithm 1** Branch-and-bound for low-rank matrix completion Problem (1)

---

- 1: Initialize  $Q = \{ (\emptyset, 0) \}$ ,  $Z_{\text{lower}} = -\infty$ ,  $Z_{\text{upper}} = +\infty$ ;
- 2: **while**  $Q$  is non-empty and  $Z_{\text{upper}} - Z_{\text{lower}} > \epsilon$  **do**
- 3:     Retrieve a problem  $(\mathcal{D}, t)$  from the queue:  $Q \leftarrow Q \setminus (\mathcal{D}, t)$ ;
- 4:     Solve the following problem, yielding  $(\hat{\Theta}_{(t+1)}, \hat{Y}_{(t+1)}, \hat{X}_{(t+1)}, \hat{U}_{(t+1)})$  with objective  $Z$ :

$$\begin{aligned}
& \min_{\substack{\Theta \in \mathcal{S}^m \\ Y \in \mathcal{S}^n \\ X \in \mathbb{R}^{n \times m} \\ U \in \mathbb{R}^{n \times k}}} \frac{1}{2} \sum_{(i,j) \in \mathcal{I}} (X_{i,j} - A_{i,j})^2 + \frac{1}{2\gamma} \text{tr}(\Theta) \\
& \text{s.t.} \quad \begin{pmatrix} Y & X \\ X^\top & \Theta \end{pmatrix} \succeq \mathbf{0}, \quad \mathbf{0} \preceq Y \preceq \mathbb{I}_n, \quad \text{tr}(Y) \leq k, \quad \begin{pmatrix} Y & U \\ U^\top & \mathbb{I}_k \end{pmatrix} \succeq \mathbf{0}, \\
& \quad \left. \begin{aligned} & U_j^\top x \in [b_{z_j}, b_{z_j+1}] \quad \forall j \in [k] \\ & \langle Y, xx^\top \rangle \leq \sum_{j=1}^k \bar{f}_{z_j; q}(U_j^\top x; \hat{U}_j^\top x) \end{aligned} \right\} \forall (\hat{U}, \hat{Y}, x, z) \in \mathcal{D}
\end{aligned} \tag{14}$$

- 5:     **if**  $\hat{U}_{(t+1)} \hat{U}_{(t+1)}^\top - \hat{Y}_{(t+1)} \succeq \mathbf{0}$  **then**      $\triangleright$  If relaxed solution is feasible for Problem (2)
  - 6:         **if**  $Z < Z_{\text{upper}}$  **then**
  - 7:              $Z_{\text{upper}} \leftarrow Z$  and  $(\Theta_{\text{opt}}, Y_{\text{opt}}, X_{\text{opt}}, U_{\text{opt}}) \leftarrow (\hat{\Theta}_{(t+1)}, \hat{Y}_{(t+1)}, \hat{X}_{(t+1)}, \hat{U}_{(t+1)})$
  - 8:         **end if**
  - 9:     **else if**  $Z < Z_{\text{upper}}$  **then**      $\triangleright$  If node relaxation is not greater than best solution found
  - 10:         Compute (unit-length)  $x_{(t+1)}$  s.t.  $x_{(t+1)}^\top (\hat{U}_{(t+1)} \hat{U}_{(t+1)}^\top - \hat{Y}_{(t+1)}) x_{(t+1)} < -\epsilon$ ;
  - 11:         **for each**  $z \in [q]^k$  **do**      $\triangleright$  Generate  $q^k$  subproblems and add them to the queue
  - 12:              $\mathcal{D}_z := \mathcal{D} \cup \{ (\hat{U}_{(t+1)}, \hat{Y}_{(t+1)}, x_{(t+1)}, z) \}$
  - 13:              $Q \leftarrow Q \cup \{ (\mathcal{D}_z, t+1) \}$
  - 14:         **end for**
  - 15:         Update  $Z_{\text{lower}}$  as the minimum of the lower bounds of all unexplored nodes.
  - 16:     **end if**
  - 17: **end while**
  - 18: **return**  $(\Theta_{\text{opt}}, Y_{\text{opt}}, X_{\text{opt}}, U_{\text{opt}})$
- 

In our branch-and-bound code, we also impose symmetry-breaking constraints on  $U$  throughout the tree, by noting that  $U$  represents a set of  $k$  orthonormal vectors in  $\mathbb{R}^n$ . We do so via:

$$U_{ij} \geq 0, \quad \forall j \in [k], \forall i \in \{n-k+j, n-k+j+1, \dots, n\} \tag{15}$$

We now expound on implementation details of Algorithm 1 pertaining to node selection, branching strategy, and incumbent selection via alternating minimization.

### 3.2. Branch-and-bound Design Decisions

We briefly review several design decisions in our branch-and-bound scheme in this section, and refer the interested reader to Section EC.3 for a more detailed discussion. We evaluate these design decisions empirically in Section 5.3.

*Node selection (EC.3.1):* To decide which node to explore next (line 4 in Algorithm 1), we consider depth-first search (where nodes are selected in a last-in-first-out manner), breadth-first search (where nodes are selected in a first-in-first-out manner), and best-first search (where the node with the lowest remaining lower bound is selected). We find in our numerical experiments that the best-first strategy performs best empirically.

*Branching strategy (EC.3.2):* As discussed in Remark 2, our eigenvector branching for a rank- $k$  matrix generates  $q^k$  child nodes per parent node when using a piecewise linear approximation of  $u \mapsto u^2$  with  $q$  pieces. Accordingly, another design decision is the number of pieces  $q \geq 2$  which should be used.

### 3.3. Incumbent Selection via Alternating Minimization

To generate high-quality feasible solutions, we apply an alternating minimization heuristic at each node. Following Burer and Monteiro (2003), Burer and Vandembussche (2006), we decompose  $\mathbf{X} = \mathbf{U}\mathbf{V}$ , start with an initial solution  $\mathbf{U}_{\text{initial}}$ , and iteratively optimize with respect to  $\mathbf{U}$  and  $\mathbf{V}$ . We describe the procedure for our matrix completion problem in Algorithm 2.

Compared with the traditional Burer-Monteiro approach, we identify the  $\mathbf{U}$ -factor in the decomposition  $\mathbf{X} = \mathbf{U}\mathbf{V}$  with the  $\mathbf{U}$  variable in our semidefinite relaxation. In particular, we want the columns of  $\mathbf{U}$  to be orthogonal so we impose the necessary constraints:  $\|\mathbf{U}_i\|_2^2 \leq 1, \forall i \in [k]$  and  $\|\mathbf{U}_i \pm \mathbf{U}_j\|_2^2 \leq 2, \forall i, j \in [k], i \neq j$ . At the root node, we impose no additional constraints. At subsequent nodes, we impose the linear constraints on  $\mathbf{U}$  involved in the node relaxation (14), i.e.,  $\mathbf{U} \in \mathcal{P}_U = \{\mathbf{U} : \mathbf{U}_j^\top \mathbf{x} \in [b_{z_j}, b_{z_j+1}], \forall j \in [k], \forall (\mathbf{x}, \mathbf{z}) \in \mathcal{D}\}$ . For the initial solution  $\mathbf{U}_{\text{initial}}$ , we do the following:

- At the root node, we define  $\mathbf{X}_{\text{initial}}$  by  $(\mathbf{X}_{\text{initial}})_{i,j} = A_{i,j}$  if  $(i, j) \in \mathcal{I}$ , and 0 otherwise and define  $\mathbf{U}_{\text{initial}} \in \mathbb{R}^{n \times k}$  as the  $\mathbf{U}$ -factor of the rank- $k$  truncated SVD of  $\mathbf{X}_{\text{initial}}$ .
- At child nodes, letting  $(\mathbf{Y}_R, \mathbf{U}_R, \mathbf{X}_R, \mathbf{\Theta}_R)$  be the solution to the relaxation (14), we define  $\mathbf{U}_{\text{initial}}$  as the  $\mathbf{U}$ -factor of a compact SVD of  $\mathbf{Y}_R$ .

---

**Algorithm 2** Alternating minimization for Problem (3)

---

1: **function** ALTERNATINGMINIMIZATION( $U_{\text{initial}}$ )

2:     Initialize with  $\hat{U}^0 \leftarrow U_{\text{initial}}$ ,  $t \leftarrow 0$

3:     **while** not converged **do**

4:         Update  $(\hat{V}^{t+1}, \hat{U}^{t+1})$  as follows, and increment  $t$ :

$$\hat{V}^{t+1} = \arg \min_{V \in \mathbb{R}^{k \times m}} \frac{1}{2} \sum_{(i,j) \in \mathcal{I}} \left( (\hat{U}^t V)_{i,j} - A_{i,j} \right)^2 + \frac{1}{2\gamma} \|\hat{U}^t V\|_F^2 \quad (16)$$

$$\hat{U}^{t+1} = \arg \min_{\substack{U \in \mathbb{R}^{n \times k}, U \in \mathcal{P}_U, \\ \|U_i\|_2 \leq 1 \ \forall i \in [k], \|U_i \pm U_j\|_2 \leq 2 \ \forall i < j \in [k]}} \frac{1}{2} \sum_{(i,j) \in \mathcal{I}} \left( (U \hat{V}^{t+1})_{i,j} - A_{i,j} \right)^2 + \frac{1}{2\gamma} \|U \hat{V}^{t+1}\|_F^2 \quad (17)$$

5:     **end while**

6:     Let  $X_{\text{altmin}} \leftarrow \hat{U}_{\text{end}} \hat{V}_{\text{end}}$ , and let  $U_{\text{altmin}} \leftarrow$  compact SVD of  $X_{\text{altmin}}$

7:     **return**  $U_{\text{altmin}}$

8: **end function**

---

For tractability considerations, we may not run this alternating minimization procedure at every node. Instead, we randomly decide to run Algorithm 2, with a probability of selection that decreases with the tree depth. In Section 5, we investigate the trade-off between the improved upper bounds from implementing alternating minimization at certain leaf nodes against its time cost.

#### 4. Convex Relaxations and Valid Inequalities

In this section, we use a different characterization of low-rank matrices in terms of their determinant minors to derive a stronger convex relaxation for low-rank matrix completion problems. We also use this relaxation to derive valid inequalities that strengthen (3). These relaxations can be used as a drop-in replacement for the semidefinite relaxation (14) solved in Algorithm 1, and we demonstrate their effectiveness numerically in Section 5.2.

Our starting point is the observation that the rank of  $X$  is fully characterized by the following well-known lemma (as proven in, e.g., Horn and Johnson 1985):

**LEMMA 1.** *Let  $X$  be a matrix. Then, the rank of  $X$  is at most  $k$  if and only if all  $(k+1) \times (k+1)$  minors of  $X$  have determinant 0.*

For our low-rank matrix completion problems, we decompose  $X$  into a sum of rank-one matrices, apply Lemma 1's characterization to each rank-one matrix individually, and combine this characterization with our matrix perspective relaxation to obtain stronger relaxations.

#### 4.1. Convex Relaxations

We construct our new convex relaxation via the following steps. First, we decompose  $\mathbf{X} = \sum_{t=1}^k \mathbf{X}^t$ , as a sum of rank-one matrices  $\mathbf{X}^t$ . Second, for each  $t \in [k]$ , we introduce a matrix to model the outer product of each vectorized two-by-two minor of  $\mathbf{X}^t$  with itself, in particular letting  $W_{i,j}^t$  model  $(X_{i,j}^t)^2$  in all moment matrices. Third, we use that each two-by-two minor should have zero determinant to eliminate some moment variables —by introducing the same variable  $V_{(i_1,i_2),(j_1,j_2)}^{3,t}$  on each entry of the anti-diagonal of the moment matrix. Fourth, we model the term  $X_{i,j}^2 = \left(\sum_{t=1}^k X_{i,j}^t\right)^2 = \sum_{t \in [k]} (X_{i,j}^t)^2 + 2 \sum_{t' < t} X_{i,j}^t X_{i,j}^{t'}$  in the objective by also performing a relaxation on the moment matrix generated by the vector  $\left(1 \ X_{i,j}^1 \ X_{i,j}^2 \ \dots \ X_{i,j}^k\right)$  and selecting the appropriate off-diagonal terms to model each product  $X_{i,j}^{t_1} X_{i,j}^{t_2}$ . Finally, we replace  $X_{i,j}^2$  with  $W_{i,j}$  in the objective where applicable and link  $\Theta_{i,j}$  with the appropriate terms in  $\mathbf{W}$  to impose appropriate objective pressure in our relaxation. We have the following result:

**PROPOSITION 3.** *The following semidefinite relaxation is at least as strong as the matrix perspective relaxation for matrix completion (3) where  $k \in [\min(n, m)]$ :*

$$\begin{aligned}
& \min_{\substack{\mathbf{X}^t, \mathbf{W}, \mathbf{W}^t \in \mathbb{R}^{n \times m} \ \forall t \in [k], \\ \mathbf{Y} \in \text{Conv}(\mathcal{Y}_n^k), \ \Theta \in \mathcal{S}_+^m, \\ \mathbf{V}^{t,t}, \mathbf{H}^{t,t'}, \ \forall t, t' \in [k]}} \frac{1}{2\gamma} \text{tr}(\Theta) + \frac{1}{2} \sum_{(i,j) \in \mathcal{I}} \left( A_{i,j}^2 + W_{i,j} - 2 \sum_{t \in [k]} X_{i,j}^t A_{i,j} \right) \quad (18) \\
& \text{s.t.} \quad \begin{pmatrix} \mathbf{Y} & \sum_{t \in [k]} \mathbf{X}^t \\ \sum_{t \in [k]} \mathbf{X}^{t \top} & \Theta \end{pmatrix} \succeq \mathbf{0}, \\
& \quad \begin{pmatrix} 1 & X_{i_1, j_1}^t & X_{i_1, j_2}^t & X_{i_2, j_1}^t & X_{i_2, j_2}^t \\ X_{i_1, j_1}^t & W_{i_1, j_1}^t & V_{i_1, (j_1, j_2)}^{1,t} & V_{(i_1, i_2), j_1}^{2,t} & V_{(i_1, i_2), (j_1, j_2)}^{3,t} \\ X_{i_1, j_2}^t & V_{i_1, (j_1, j_2)}^{1,t} & W_{i_1, j_2}^t & V_{(i_1, i_2), (j_1, j_2)}^{3,t} & V_{(i_1, i_2), j_2}^{2,t} \\ X_{i_2, j_1}^t & V_{(i_1, i_2), j_1}^{2,t} & V_{(i_1, i_2), (j_1, j_2)}^{3,t} & W_{i_2, j_1}^t & V_{i_2, (j_1, j_2)}^{1,t} \\ X_{i_2, j_2}^t & V_{(i_1, i_2), (j_1, j_2)}^{3,t} & V_{(i_1, i_2), j_2}^{2,t} & V_{i_2, (j_1, j_2)}^{1,t} & W_{i_2, j_2}^t \end{pmatrix} \succeq \mathbf{0}, \quad \begin{aligned} & \forall i_1 < i_2 \in [n], \\ & \forall j_1 < j_2 \in [m], \\ & \forall t \in [k], \end{aligned} \\
& \quad \begin{pmatrix} 1 & X_{i,j}^1 & \dots & X_{i,j}^k \\ X_{i,j}^1 & W_{i,j}^1 & \dots & H_{i,j}^{1,k} \\ \vdots & \vdots & \ddots & \vdots \\ X_{i,j}^k & H_{i,j}^{1,k} & \dots & W_{i,j}^k \end{pmatrix} \succeq \mathbf{0}, \quad \forall i \in [n], j \in [m], \\
& \quad \Theta_{j,j} = \sum_{i \in [n]} W_{i,j}, \quad W_{i,j} = \sum_{t \in [k]} \left( W_{i,j}^t + \sum_{t' \in [k]: t' \neq t} H_{i,j}^{t',t} \right) \quad \forall i \in [n], j \in [m].
\end{aligned}$$

*Proof of Proposition 3* This follows immediately from the fact that any feasible solution to (18) is feasible in (3) with objective value no larger, since  $W_{i,j} \geq X_{i,j}^2$ .  $\square$

## 4.2. Partial Convex Relaxations

Despite its strength, our new relaxation is also computationally prohibitive to solve at scale. Accordingly, we use it to derive valid inequalities that we further impose on (3) to develop more scalable (yet less tight) semidefinite relaxations.

To develop our relaxation, we begin with the following problem, which is equivalent to (3) but also includes each slice  $\mathbf{X}^t$  and variables  $W_{i,j}$  to model  $X_{i,j}^2$  for each  $(i, j)$ :

$$\begin{aligned}
 \min_{\substack{\mathbf{X}^t, \mathbf{W}, \mathbf{W}^t \in \mathbb{R}^{n \times m} \quad \forall t \in [k], \\ \mathbf{Y} \in \text{Conv}(\mathcal{Y}_n^k), \quad \Theta \in \mathcal{S}_+^m, \mathbf{V}}} & \quad \frac{1}{2\gamma} \text{tr}(\Theta) + \frac{1}{2} \sum_{(i,j) \in \mathcal{I}} \left( A_{i,j}^2 + W_{i,j} - 2 \sum_{t \in [k]} X_{i,j}^t A_{i,j} \right) \\
 \text{s.t.} & \quad \begin{pmatrix} \mathbf{Y} & \sum_{t \in [k]} \mathbf{X}^t \\ \sum_{t \in [k]} \mathbf{X}^{t\top} & \Theta \end{pmatrix} \succeq \mathbf{0}, \\
 & \quad \Theta_{j,j} = \sum_{i \in [n]} W_{i,j}, \quad \forall j \in [m], \quad \left( \sum_{t \in [k]} X_{i,j}^t \right)^2 \leq W_{i,j}, \quad \forall i \in [n], j \in [m].
 \end{aligned} \tag{19}
 \end{aligned}$$

Given a solution to this relaxation, we select a subset of the  $2 \times 2$  minors of  $\mathbf{X}$  according to a pre-specified criterion, e.g., all  $2 \times 2$  minors where all four entries  $(i_1, i_2, j_1, j_2)$  are observed, and a random sample of  $2 \times 2$  minors where at least three entries are observed. Next, for each such minor  $(i_1, i_2, j_1, j_2)$ , we introduce appropriately indexed variables  $\mathbf{V}, \mathbf{W}^t, \mathbf{H}$ , omit the constraint linking

$X_{i,j}^t$  and  $W_{i,j}$ , and impose the following constraints:

$$\begin{pmatrix} 1 & X_{i_1,j_1}^t & X_{i_1,j_2}^t & X_{i_2,j_1}^t & X_{i_2,j_2}^t \\ X_{i_1,j_1}^t & W_{i_1,j_1}^t & V_{i_1,(j_1,j_2)}^{1,t} & V_{(i_1,i_2),j_1}^{2,t} & V_{(i_1,i_2),(j_1,j_2)}^{3,t} \\ X_{i_1,j_2}^t & V_{i_1,(j_1,j_2)}^{1,t} & W_{i_1,j_2}^t & V_{(i_1,i_2),(j_1,j_2)}^{3,t} & V_{(i_1,i_2),j_2}^{2,t} \\ X_{i_2,j_1}^t & V_{(i_1,i_2),j_1}^{2,t} & V_{(i_1,i_2),(j_1,j_2)}^{3,t} & W_{i_2,j_1}^t & V_{i_2,(j_1,j_2)}^{1,t} \\ X_{i_2,j_2}^t & V_{(i_1,i_2),(j_1,j_2)}^{3,t} & V_{(i_1,i_2),j_2}^{2,t} & V_{i_2,(j_1,j_2)}^{1,t} & W_{i_2,j_2}^t \end{pmatrix} \succeq \mathbf{0}, \quad \forall t \in [k],$$

$$\begin{pmatrix} 1 & X_{i,j}^1 & X_{i,j}^2 & \dots & X_{i,j}^k \\ X_{i,j}^1 & W_{i,j}^1 & H_{i,j}^{1,2} & \dots & H_{i,j}^{1,k} \\ X_{i,j}^2 & H_{i,j}^{1,2} & W_{i,j}^2 & \dots & H_{i,j}^{2,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ X_{i,j}^k & H_{i,j}^{1,k} & H_{i,j}^{2,k} & \dots & W_{i,j}^k \end{pmatrix} \succeq \mathbf{0}, \quad \forall i \in \{i_1, i_2\}, j \in \{j_1, j_2\},$$

$$W_{i,j} = \sum_{t \in [k]} \left( W_{i,j}^t + \sum_{t' \in [k]: t' \neq t} H_{i,j}^{t',t} \right), \quad \forall i \in \{i_1, i_2\}, j \in \{j_1, j_2\}.$$

All in all, we impose  $k$   $5 \times 5$  PSD constraints and at most  $4(k+1) \times (k+1)$  PSD constraints for each  $2 \times 2$  submatrix we aim to model, which allows us to control the computational complexity of our relaxation via the number of cuts at each problem size. In particular, the computational complexity of modeling a Shor relaxation of a given  $2 \times 2$  minor scales independently of  $n, m$ .

## 5. Numerical Experiments

In this section, we evaluate the numerical performance of our branch-and-bound scheme, implemented in Julia version 1.9 using Mosek version 10.0 to solve all semidefinite optimization problems. All experiments were conducted on MIT's Supercloud cluster (Reuther et al. 2018), which hosts Intel Xeon Platinum 8260 processors. To perform our experiments, we generate synthetic instances of matrix completion problems, as described in Section 5.1.

We evaluate the effectiveness of the valid inequalities from Section 4 in Section 5.2. Next, in Section 5.3, we benchmark the performance of different node selection, branching, breakpoint, and incumbent generation strategies for our algorithm. Finally, we evaluate the scalability of our branch-and-bound algorithm in Section 5.4, in terms of its ability to identify an optimal solution, and find

feasible solutions that outperform a popular alternating minimization strategy. To bridge the gap between theory and practice, we make our branch-and-bound implementation available at `github.com/sean-lo/OptimalMatrixCompletion.jl` and our experiments at `github.com/sean-lo/OLRMC_experiments`.

### 5.1. Generation of Synthetic Instances

We compute a matrix of observations,  $\mathbf{A}_{\text{full}} \in \mathbb{R}^{n \times m}$ , from a low-rank model:  $\mathbf{A}_{\text{full}} = \mathbf{UV} + \epsilon \mathbf{Z}$ , where the entries of  $\mathbf{U} \in \mathbb{R}^{n \times k}$ ,  $\mathbf{V} \in \mathbb{R}^{k \times m}$ , and  $\mathbf{Z} \in \mathbb{R}^{n \times m}$  are drawn independently from a standard normal distribution, and  $\epsilon \geq 0$  models the degree of noise. We fix  $\epsilon = 0.1$ . We then sample a random subset  $\mathcal{I} \subseteq [n] \times [m]$  of predefined size, which contains at least one entry in each row and column of the matrix (see also Candès and Recht 2009, Section 1.1.2). To do so, if the target size  $|\mathcal{I}|$  is large enough, we iteratively add random entries until this property is satisfied, which will happen with fewer than  $|\mathcal{I}|$  draws with high probability. In regimes where  $|\mathcal{I}|$  is small, i.e., close to  $k(n+m)$ , we use a random permutation matrix to sample  $k(n+m)$  entries, one in each row and column, directly. We then independently sample the remaining entries of  $\mathcal{I}$  to reach the desired size.

For simplicity, in our experiments, we set  $m = n$  and only vary the dimension of the problem,  $n$ , and the number of observed entries,  $|\mathcal{I}|$ . Also, to allow for a better comparison across instances of varying size  $n$ , we generate one large  $N \times N$  matrix  $\mathbf{A}_{\text{full}}$  and consider its top-left  $n$ -by- $n$  submatrices for various values of  $n$ , which creates correlations between the instances generated for different values of  $n$ —the instances generated for a given size  $n$ , however, remain independent since they come from different matrices  $\mathbf{A}_{\text{full}}$  obtained with different random seeds.

We remark that this data generation process is standard in the literature and generates matrices from the same distribution as those considered by many other authors (e.g., Candès and Recht 2009, Bertsimas et al. 2022).

### 5.2. Root Node: Strengthened Relaxations

In this section, we evaluate the benefit of the valid inequalities presented in Section 4.

As argued in Section 4.2, we can strengthen the matrix perspective relaxation (3) by imposing additional semidefinite constraints (Shor LMIs) on all  $2 \times 2$  minors of the slices of  $\mathbf{X}$ ,  $\mathbf{X}^t$ ,  $t \in [k]$ . To minimize the impact on tractability, however, we do not impose these constraints for all minors. Instead, we consider two-by-two minors with all four entries present in  $\mathcal{I}$  (denoted  $\mathcal{M}_4$ ), or with at least three entries in  $\mathcal{I}$  ( $\mathcal{M}_3$ ). We compare the original relaxation (3) with (19) strengthened with Shor LMIs for all minors in  $\mathcal{M}_4$ , all minors in  $\mathcal{M}_4$  and a (random) half of the minors in  $\mathcal{M}_3$ , and

all minors in  $\mathcal{M}_4$  and  $\mathcal{M}_3$ , using two metrics: optimality gap at the root node (the upper bound is obtained via alternating minimization) and computational time for solving the relaxation.

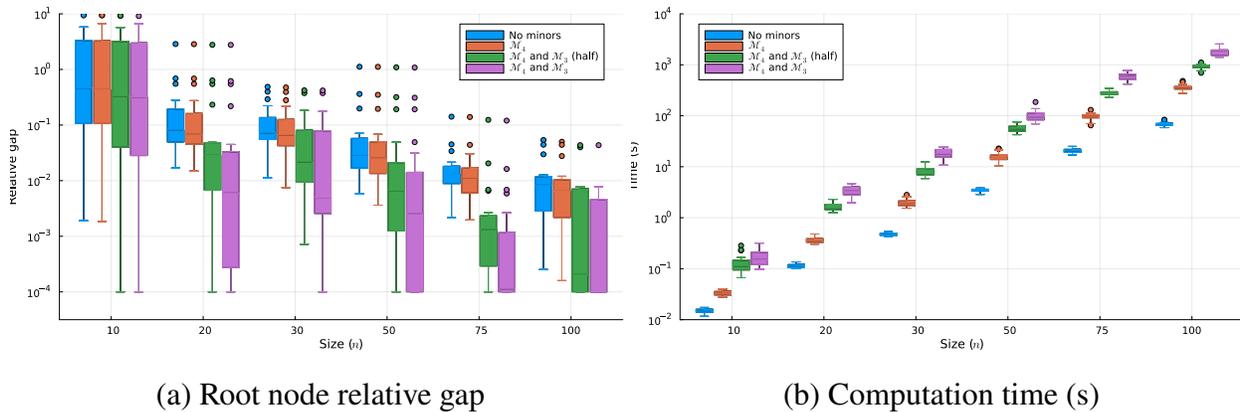
We consider rank-one matrix completion where  $n \in \{10, 20, 30, 50, 75, 100\}$ , and  $|\mathcal{I}| = 2n \log(n)$ . Figures 1a and 1b represent the optimality gap at the root node and computational time respectively of using these strengthened relaxations. We observe that, while adding Shor LMIs for minors in  $\mathcal{M}_4$  only has a small impact on the optimality gap, considering all minors in  $\mathcal{M}_4$  and  $\mathcal{M}_3$  significantly reduce the optimality gaps. For example, for  $n \geq 50$ , it reduces the median optimality gap by nearly two orders of magnitude (from  $10^{-2}$  down to  $10^{-4}$ ). This comes with a corresponding one-order-of-magnitude increase in computation time. Introducing Shor LMIs for all minors in  $\mathcal{M}_4$  and half of the minors in  $\mathcal{M}_3$  partially alleviates the computational burden while retaining some benefits in relaxation tightness, providing us with greater flexibility to control the optimality-tractability tradeoff. Detailed results are reported in Tables EC.1–EC.2 in EC.4.1.

We observe that the relative strength of our relaxation crucially depends on the proportion of missing entries, especially as the rank  $k$  increases. Figures 2a–2b show the objective value of our relaxation with Shor LMIs for minors in  $\mathcal{M}_4$ , relative to the relaxation without any minors, for different values of  $|\mathcal{I}|$ . We consider  $|\mathcal{I}| = Cn$  (resp.  $|\mathcal{I}| = Cn^2$ ) for rank-one (resp. rank-two) instances in Figure 2a (resp. Figure 2b). In both cases, we observe that the relative benefit from adding Shor LMIs to the root node relaxation increases as  $C$  increases. We also observe that the number of entries required to observe an improvement from the Shor LMIs is higher in the rank-two than in the rank-one case—notably in its scaling with respect to  $n$ . A larger number of observed entries, however, might be computationally more expensive (because there are more Shor LMIs to add) and not available in practice.

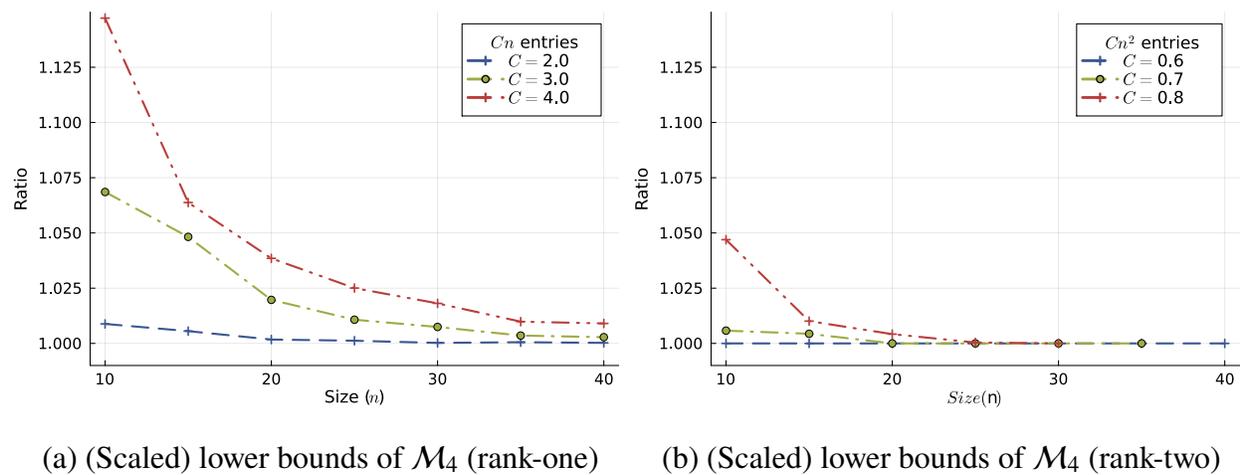
### 5.3. Branch-and-bound Design Decisions

In this section, we benchmark the efficacy of different algorithmic design options for our branch-and-bound algorithm, including whether to use an eigenvector branching scheme (as developed in Section 2) or a McCormick-based branching scheme (as described in Section 2.2); the node expansion strategy (breadth-first, best-first, or depth-first); and whether alternating minimization, as described in Section 3.3, should be run at child nodes in the tree.

Tables 1–2 report the final optimality gaps and total computational time of our branch-and-bound scheme with different configurations as we vary  $n$ . We impose two termination criteria for these experiments: a relative optimality gap of  $10^{-4}$  and a time limit of one hour. Eigenvector disjunctions achieve optimality gaps about an order of magnitude smaller than McCormick disjunctions. Running



**Figure 1** Root node relative gap (a) and computation time (b) at the root node for rank-one  $n$ -by- $n$  matrix completion problems with  $2n \log_{10}(n)$  filled entries, in a regime with low regularization ( $\gamma = 80.0$ ).



**Figure 2** (Scaled) lower bounds of  $\mathcal{M}_4$  relaxation relative to the relaxation with no Shor LMIs, with  $Cn$  or  $Cn^2$  entries, with low regularization ( $\gamma = 80.0$ ).

alternating minimization at child nodes also improves the average optimality gap by around an order of magnitude, which provides evidence that the Burer-Monteiro alternating minimization method is not optimal in practice when run from the root node, and can be improved via branch-and-bound. Moreover, the best-first node selection strategy, which comprises selecting the unexpanded node with the smallest lower bound at each iteration, outperforms both breadth-first and depth-first search. This phenomenon is consistent across  $p$  and  $\gamma$ ; see Tables EC.3–EC.8. Accordingly, we use best-first search with eigenvector disjunctions and alternating minimization run at child nodes throughout the rest of the paper, unless explicitly indicated otherwise.

We remark that in preliminary numerical experiments, we also considered solving matrix completion problems via the multi-tree branch-and-cut approach proposed in Bertsimas et al. (2022),

$n$	Alternating minimization	With McCormick disjunctions			With eigenvector disjunctions		
		Best-first	Breadth-first	Depth-first	Best-first	Breadth-first	Depth-first
10	✗	$6.63 \times 10^{-1}$	$6.82 \times 10^{-1}$	$6.84 \times 10^{-1}$	$5.72 \times 10^{-1}$	$5.61 \times 10^{-1}$	$2.65 \times 10^{-1}$
10	✓	$1.26 \times 10^{-2}$	$2.18 \times 10^{-2}$	$1.63 \times 10^{-1}$	$3.57 \times 10^{-3}$	$1.01 \times 10^{-2}$	$5.29 \times 10^{-2}$
20	✗	$8.24 \times 10^{-2}$	$8.24 \times 10^{-2}$	$8.24 \times 10^{-2}$	$6.44 \times 10^{-2}$	$6.75 \times 10^{-2}$	$8.78 \times 10^{-2}$
20	✓	$9.44 \times 10^{-3}$	$1.19 \times 10^{-2}$	$1.19 \times 10^{-2}$	$1.71 \times 10^{-3}$	$3.36 \times 10^{-3}$	$9.52 \times 10^{-3}$
30	✗	$4.88 \times 10^{-2}$	$4.88 \times 10^{-2}$	$4.88 \times 10^{-2}$	$4.15 \times 10^{-2}$	$4.23 \times 10^{-2}$	$4.79 \times 10^{-2}$
30	✓	$1.36 \times 10^{-2}$	$1.71 \times 10^{-2}$	$1.71 \times 10^{-2}$	$1.71 \times 10^{-3}$	$2.88 \times 10^{-3}$	$1.12 \times 10^{-2}$
40	✗	$4.92 \times 10^{-2}$	$4.92 \times 10^{-2}$	$4.92 \times 10^{-2}$	$4.44 \times 10^{-2}$	$4.49 \times 10^{-2}$	$2.14 \times 10^{-2}$
40	✓	$2.03 \times 10^{-2}$	$2.03 \times 10^{-2}$	$2.03 \times 10^{-2}$	$6.53 \times 10^{-4}$	$9.50 \times 10^{-4}$	$1.10 \times 10^{-2}$
50	✗	$1.50 \times 10^{-2}$	$1.50 \times 10^{-2}$	$1.50 \times 10^{-2}$	$2.68 \times 10^{-2}$	$2.81 \times 10^{-2}$	$2.97 \times 10^{-2}$
50	✓	$6.06 \times 10^{-3}$	$6.06 \times 10^{-3}$	$6.09 \times 10^{-3}$	$2.22 \times 10^{-4}$	$3.36 \times 10^{-4}$	$8.83 \times 10^{-3}$

**Table 1** Final optimality gap across rank-one matrix completion problems with  $|\mathcal{I}| = pn \log_{10}(n)$  filled entries, averaged over 20 instances per row ( $p = 2.0$ ,  $\gamma = 20.0$ ).

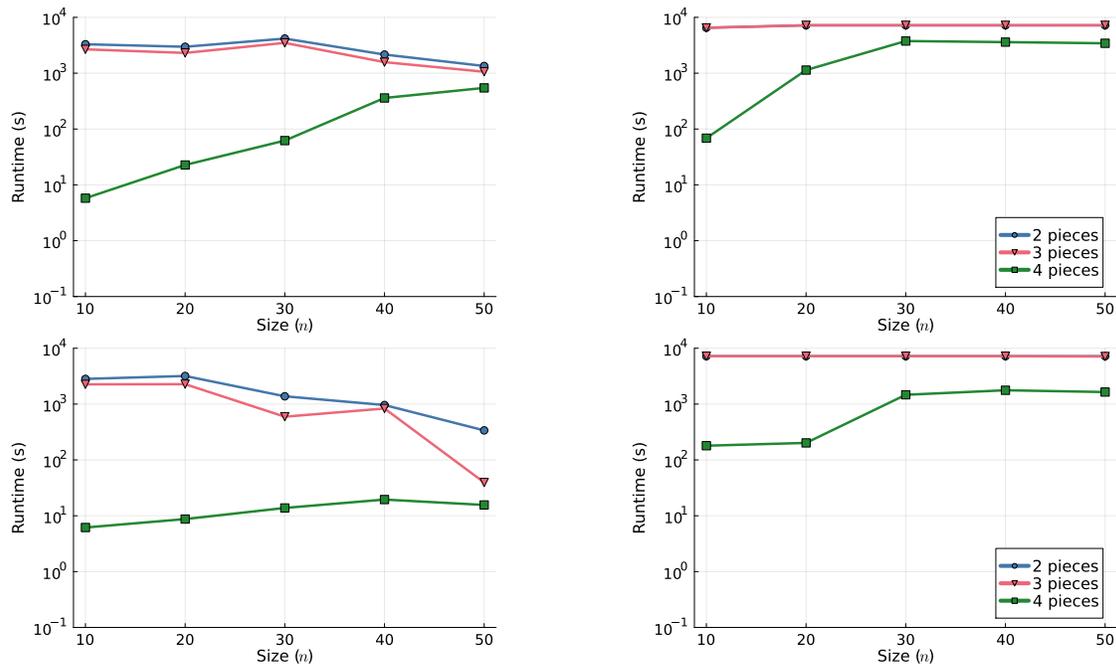
$n$	Alternating minimization	With McCormick disjunctions			With eigenvector disjunctions		
		Best-first	Breadth-first	Depth-first	Best-first	Breadth-first	Depth-first
10	✗	3061.5	3078.2	3240.0	2448.9	2546.7	3240.0
10	✓	1453.9	1603.7	3061.8	1675.8	1974.4	2824.6
20	✗	3060.1	3060.1	3060.1	1939.4	2254.9	2894.2
20	✓	2540.1	2700.1	2700.1	1539.7	1862.5	2370.4
30	✗	3600.3	3600.4	3600.2	3084.1	3155.5	3600.3
30	✓	3060.4	3060.4	3060.3	2285.9	2363.7	3060.4
40	✗	2881.3	2881.1	2881.0	1818.1	1827.6	2881.4
40	✓	2701.1	2701.2	2701.2	1203.4	1290.2	2701.0
50	✗	2703.0	2702.9	2702.8	1749.5	2400.8	2703.7
50	✓	2343.4	2343.1	2343.1	959.32	1299.7	2437.2

**Table 2** Total computational time (s) across rank-one matrix completion problems with  $|\mathcal{I}| = pn \log_{10}(n)$  filled entries, averaged over 20 instances per row ( $p = 2.0$ ,  $\gamma = 20.0$ ).

and directly with Gurobi's non-convex QCQP solver. Unfortunately, neither approach was competitive with either the McCormick disjunction or the eigenvector disjunction approach, likely because Gurobi does not allow semidefinite constraints to be imposed, and the root node relaxation without semidefinite constraints is often quite weak. Indeed, for instances where  $n = 50$  and  $k > 1$ , neither approach produced a better lower bound than the matrix perspective relaxation.

We also investigate the effect of the number of pieces in our disjunction,  $q$ , on the time needed to achieve a  $10^{-4}$  optimality gap for rank-one matrix completion problems, in Figure 3. We do not observe a significant difference between using  $q = 2$  or  $q = 3$  pieces. However, we find that implementing a disjunctive scheme with  $q = 4$  pieces allows our branch-and-bound strategy to

converge orders of magnitude faster, across all values of  $p$  and  $\gamma$ . We suspect this occurs because four-piece disjunctions include zero as a breakpoint, which breaks some symmetry. However, we also observe that this advantage vanishes as  $n$  increases; Figure EC.1 confirms this on larger instances.

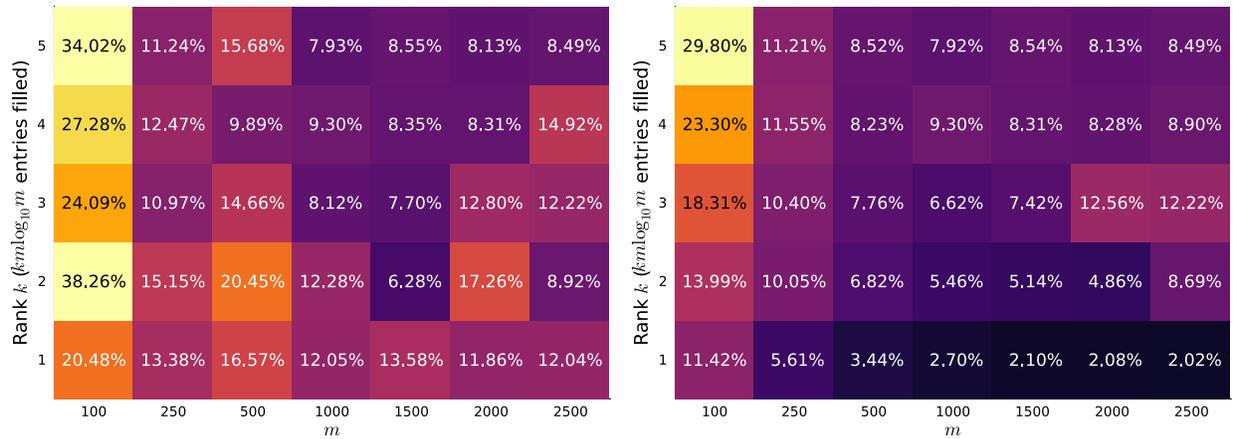


**Figure 3** Comparison of time taken to optimality (relative gap  $10^{-4}$ ) for rank-one matrix completion problems with  $pn \log_{10}(n)$  filled entries, over different numbers of pieces  $q \in \{2, 3, 4\}$  in the upper-approximation. We set  $p = 2.0$  (less entries) in the top plots, and  $p = 3.0$  (more entries) in the bottom plots. We set  $\gamma = 20.0$  (more regularization) in the left plots, and  $\gamma = 80.0$  (less regularization) in the right plots.

#### 5.4. Scalability Experiments

Table 2 reveals that the strongest implementation of our algorithm solves rank-one matrix completion problems where  $m = n = 50$  in minutes. Accordingly, we now investigate the scalability of our algorithm with this configuration and its ability to obtain higher-quality low-rank matrices than heuristics. We apply our algorithm to rank- $k$  matrix completion problems on rectangular instances of size  $50 \times m$ , with  $km \log_{10}(m)$  observed entries,  $k \in \{1, 2, 3, 4, 5\}$ , and a time limit of three hours.

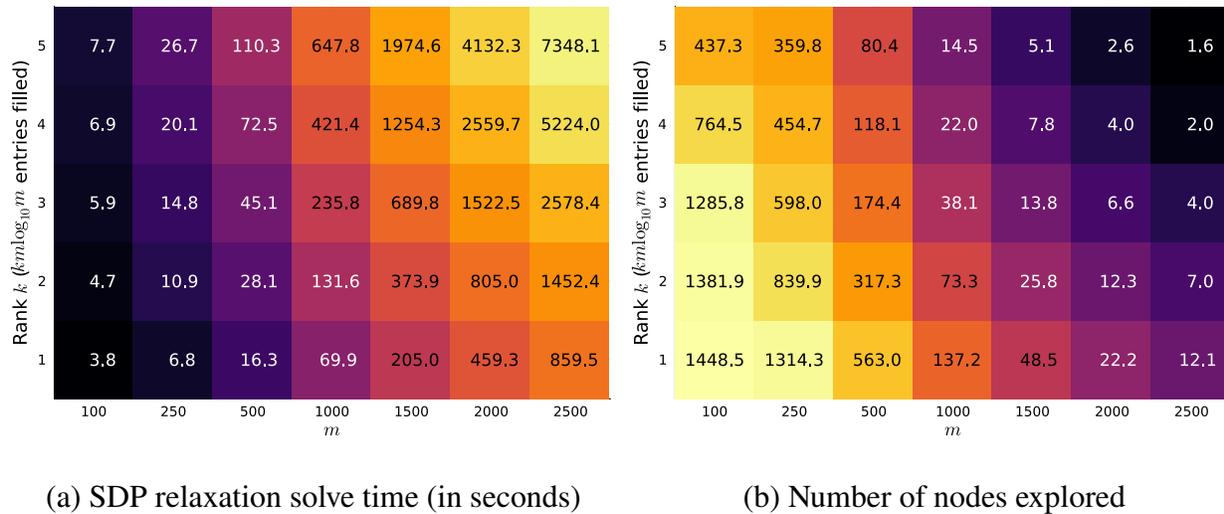
Figure 4 depicts the relative gap between the root node relaxation and the best incumbent solution at the root node (left panel), and after applying branch-and-bound for three hours (right panel). Comparing the left and right panels, we observe that Algorithm 1 is effective at reducing the optimality gap across all  $(m, k)$  values, and is especially effective for  $k = 1$ .



**Figure 4** Comparison of relative optimality gap at root node (left) and after running branch-and-bound for three hours (right) for rank- $k$  matrix completion problems of dimension  $50 \times m$ , with  $km \log_{10}(m)$  filled entries, varying  $m$  and  $k$ , with  $\gamma = 120.0$ , averaged over 10 random instances.

As the rank  $k$  increases (and to a lesser extent, as  $m$  increases), the difference between the gap at the root node (left panel) and at termination (right panel) is less acute. However, this observation does not necessarily imply that our algorithm is less effective as  $k$  and  $m$  grow, for at least two reasons. First, the root node gap decreases with  $m$  and  $k$ , providing less opportunity for branch-and-bound to improve; for example, for  $k = 2$ , the gap at the root node decreases from 38% when  $m = 100$  to 9% when  $m = 2500$ . Second, our branch-and-bound algorithm is terminated after a fixed time limit (three hours in all our experiments). The time needed to solve each SDP relaxation, however, increases with  $k$  (and, to a lesser extent, with  $m$ ), as reported on the left panel of Figure 5. As a result, our branch-and-bound algorithm can explore a smaller number of nodes within the time limit when  $k$  or  $m$  is larger (see right panel of Figure 5). Consequently, the observation that the improvement over the root node gap is less acute when  $k$  and  $m$  increase could also be an artifact of using the same time limit for all instance sizes. This limitation is hard to overcome in practice: stopping the algorithm after a fixed number of nodes explored would suffer from the same limitation, and it is hard to know how to increase time limit (or node limit) with  $k$  and  $m$  to allow for a ‘fair’ comparison across  $(m, k)$  values. For example, it is quite surprising to see that for  $k = 1, 2$  and  $m \geq 2000$ , exploring 10–20 nodes only can already bring significant gap improvements.

We further demonstrate that the solutions found by alternating minimization over branch-and-bound subregions are also superior in terms of downstream out-of-sample error. Figure 6 (and Table EC.9) compares the solution found via alternating minimization at the root node against the



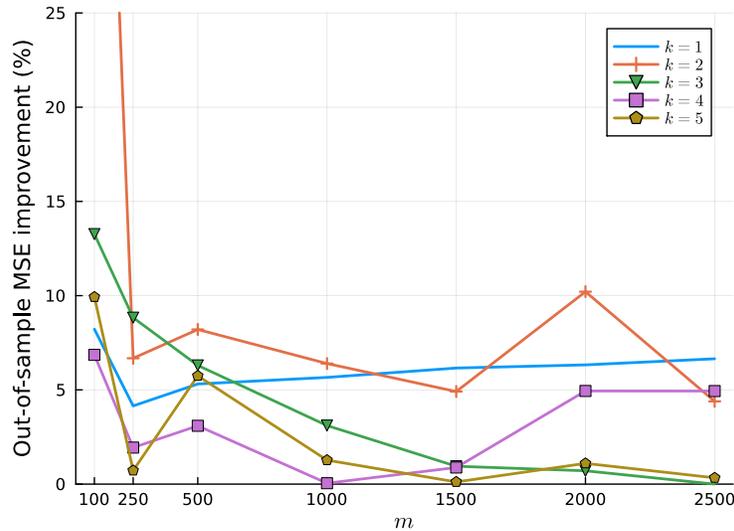
**Figure 5** SDP relaxation solve time in seconds (left) and number of branch-and-bound nodes explored (right) for rank- $k$  matrix completion problems of dimension  $50 \times m$ , with  $km \log_{10}(m)$  filled entries, varying  $m$  and  $k$ , with  $\gamma = 120.0$ , averaged over 10 random instances.

best solution found by our branch-and-bound scheme after branching for three hours, as measured by the percentage improvement in out-of-sample mean squared error (Figure EC.2 shows the before-and-after values of the out-of-sample MSE). We observe that the improved solutions found by branch-and-bound translate to improvements in MSE across the board. The improvements are consistently in the 1-10% range, and the largest improvements are when  $m$  is small and Burer-Monteiro at the root node yields poor-quality solutions. In short, branch-and-bound improves the test set performance of our imputed matrices.

### 5.5. Summary of Findings from Numerical Experiments

The main findings from our numerical experiments are as follows:

- Section 5.2 demonstrates that our valid inequalities for matrix completion problems strengthen, often significantly, the semidefinite relaxation (3), and indeed routinely improve the root node gap by an order of magnitude or more in the rank-one case. However, their efficiency depends on the number of observed entries, especially at larger ranks. Future work could investigate more scalable semidefinite relaxations that are applicable when  $k > 1$ .
- Section 5.3 investigates the impact of design choices in our branch-and-bound scheme, and demonstrates that eigenvector-based disjunctions obtain optimality gaps over an order of magnitude smaller than McCormick-based ones in the same amount of computational time.
- Section 5.4 demonstrates the scalability of our branch-and-bound scheme for matrix completion problems. While scalability depends on the rank, data regime and extent of regularization, we



**Figure 6** Percentage improvement in out-of-sample MSE for rank- $k$   $50 \times m$  matrix completion problems, with  $km \log_{10}(m)$  filled entries, varying  $m$  and  $k$ , with  $\gamma = 120.0$ , averaged over 10 random instances.

show that we can solve rectangular  $50 \times m$  instances with  $m$  as large as 2500 in hours. One necessary limitation of branch-and-bound is memory; our method requires  $\mathcal{O}(K \min\{m, n\}^2)$  memory, where  $K$  is the number of nodes explored.

- We observe that, at the information-theoretic threshold identified by Candès and Recht (2009) with  $\mathcal{O}(nk \log n)$  entries observed, our branch-and-bound scheme obtains low-rank matrices with an out-of-sample predictive power 1%–50% better than solutions obtained via a Burer-Monteiro heuristic, depending on the rank and the dimensionality of the problem.

## 6. Conclusion

In this paper, we propose a new branch-and-bound scheme for solving low-rank matrix completion problems to certifiable optimality. The framework considers matrix perspective relaxations and recursively partitions their feasible regions using eigenvector disjunctions. We provide theoretical and empirical evidence to justify the superiority of eigenvector disjunctions compared with the widely used McCormick disjunctions. We further strengthen the semidefinite relaxations via valid inequalities. On rank-one matrix completion problems, we numerically find that these inequalities routinely improve the root node gap by an order of magnitude.

Altogether, our approach scales to matrix completion problems on  $n \times m$  matrices with  $\max\{m, n\} = 2500$  and rank up to 5 with optimality guarantees. Compared with the existing and popular scalable heuristics (which, unfortunately, do not provide any instance-wise optimality certificates), our branch-and-bound scheme obtains low-rank matrices with an out-of-sample

reconstruction error 1%–50% lower, depending on the rank and the dimensionality of the problem, illustrating the practical value of certifiably optimal methods for solving matrix completion problems.

## Acknowledgments

Ryan Cory-Wright gratefully acknowledges the MIT-IBM Research Lab for hosting him while part of this work was conducted.

## References

- Anstreicher KM (2022) Solving two-trust-region subproblems using semidefinite optimization with eigenvector branching. *Journal of Optimization Theory and Applications* 1–17.
- Bandeira AS, Dobriban E, Mixon DG, Sawin WF (2013) Certifying the restricted isometry property is hard. *IEEE Transactions on Information Theory* 59(6):3448–3450.
- Bell RM, Koren Y (2007) Lessons from the Netflix prize challenge. Technical report, AT&T Bell Laboratories.
- Bertsimas D, Cory-Wright R, Pauphilet J (2022) Mixed-projection conic optimization: A new paradigm for modeling rank constraints. *Operations Research* 70(6):3321–3344.
- Bertsimas D, Cory-Wright R, Pauphilet J (2023) A new perspective on low-rank optimization. *Mathematical Programming* 202:47–92.
- Bhojanapalli S, Neyshabur B, Srebro N (2016) Global optimality of local search for low rank matrix recovery. *Advances in Neural Information Processing Systems* 29.
- Boumal N, Voroninski V, Bandeira A (2016) The non-convex Burer-Monteiro approach works on smooth semidefinite programs. *Advances in Neural Information Processing Systems* 29.
- Burer S, Monteiro RD (2003) A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming* 95(2):329–357.
- Burer S, Vandenbussche D (2006) Solving Lift-and-Project Relaxations of Binary Integer Programs. *SIAM Journal on Optimization* 16(3):726–750.
- Candès EJ, Plan Y (2010) Matrix completion with noise. *Proceedings of the IEEE* 98(6):925–936.
- Candès EJ, Recht B (2009) Exact matrix completion via convex optimization. *Foundations of Computational Mathematics* 9(6):717–772.
- Cifuentes D (2021) On the Burer–Monteiro method for general semidefinite programs. *Optimization Letters* 15(6):2299–2309.
- Ebadian A, Nikoufar I, Eshaghi Gordji M (2011) Perspectives of matrix convex functions. *Proceedings of the National Academy of Sciences* 108(18):7313–7314.
- Fazel M (2002) *Matrix Rank Minimization with Applications*. Ph.D. thesis, Stanford University.

- Ge R, Lee JD, Ma T (2016) Matrix completion has no spurious local minimum. *Advances in Neural Information Processing Systems* 29.
- Hardt M (2014) Understanding alternating minimization for matrix completion. *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, 651–660 (IEEE).
- Horn RA, Johnson CR (1985) *Matrix Analysis* (Cambridge University Press, New York).
- Ibaraki T (1976) Theoretical comparisons of search strategies in branch-and-bound algorithms. *International Journal of Computer & Information Sciences* 5:315–344.
- Jain P, Netrapalli P, Sanghavi S (2013) Low-rank matrix completion using alternating minimization. *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, 665–674.
- Keshavan RH, Montanari A (2010) Regularization for matrix completion. *2010 IEEE International Symposium on Information Theory*, 1503–1507 (IEEE).
- Khajavirad A (2023) On the strength of recursive McCormick relaxations for binary polynomial optimization. *Operations Research Letters* 51(2):146–152.
- Kim J, Tawarmalani M, Richard JPP (2022) Convexification of permutation-invariant sets and an application to sparse principal component analysis. *Mathematics of Operations Research* 47(4):2547–2584.
- Kocuk B, Dey SS, Sun XA (2018) Matrix minor reformulation and SOCP-based spatial branch-and-cut method for the AC optimal power flow problem. *Mathematical Programming Computation* 10(4):557–596.
- Kronqvist J, Bernal DE, Lundell A, Grossmann IE (2019) A review and comparison of solvers for convex MINLP. *Optimization and Engineering* 20(2):397–455.
- Lawler EL, Wood DE (1966) Branch-and-bound methods: A survey. *Operations Research* 14(4):699–719.
- Li Y, Xie W (2025) On the partial convexification for low-rank spectral optimization: rank bounds and algorithms. *Mathematical Programming* 1–58.
- Lubin M, Vielma JP, Zadik I (2022) Mixed-integer convex representability. *Mathematics of Operations Research* 47(1):720–749.
- Ma J, Fattahi S (2023) On the optimization landscape of Burer-Monteiro factorization: When do global solutions correspond to ground truth? *arXiv preprint arXiv:2302.10963* .
- Mazumder R, Saldana D, Weng H (2020) Matrix completion with nonconvex regularization: Spectral operators and scalable algorithms. *Statistics and Computing* 30(4):1113–1138.
- McCormick GP (1976) Computability of global solutions to factorable nonconvex programs: Part I—convex underestimating problems. *Mathematical Programming* 10(1):147–175.
- Mnih A, Salakhutdinov RR (2007) Probabilistic matrix factorization. *Advances in Neural Information Processing Systems* 20.
- Naldi S (2016) Solving rank-constrained semidefinite programs in exact arithmetic. *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, 357–364.

- Reuther A, Kepner J, Byun C, Samsi S, Arcand W, Bestor D, Bergeron B, Gadepally V, Houle M, Hubbell M, Jones M, Klein A, Milechin L, Mullen J, Prout A, Rosa A, Yee C, Michaleas P (2018) Interactive supercomputing on 40,000 cores for machine learning and data analysis. *2018 IEEE High Performance Extreme Computing Conference (HPEC)*, 1–6 (IEEE).
- Saxena A, Bonami P, Lee J (2010) Convex relaxations of non-convex mixed integer quadratically constrained programs: extended formulations. *Mathematical Programming* 124(1):383–411.
- Srebro N, Rennie J, Jaakkola T (2004) Maximum-margin matrix factorization. *Advances in Neural Information Processing Systems* 17.
- Sun T, Zhang CH (2012) Calibrated elastic regularization in matrix completion. *Advances in Neural Information Processing Systems* 25.
- Udell M, Horn C, Zadeh R, Boyd S (2016) Generalized low rank models. *Foundations and Trends® in Machine Learning* 9(1):1–118.
- Zhou Y, Wilkinson D, Schreiber R, Pan R (2008) Large-scale parallel collaborative filtering for the Netflix prize. *International Conference on Algorithmic Applications in Management*, 337–348 (Springer).

## Electronic Companion

### EC.1. Regularization for Matrix Completion

In this section, we provide additional motivation for the presence of the Frobenius regularization  $\frac{1}{2\gamma}\|\mathbf{X}\|_F^2$  in the objective of our matrix completion problem.

In Problem (1), we introduce an additional penalty on the squared Frobenius norm of  $\mathbf{X}$ , in line with previous work on regularized matrix completion. For example, several matrix completion formulations introduce a low-rank decomposition of  $\mathbf{X}$ ,  $\mathbf{X} = \mathbf{UV}$  with  $\mathbf{U} \in \mathbb{R}^{n \times k}$ ,  $\mathbf{V} \in \mathbb{R}^{k \times m}$  together with a penalty on the squared Frobenius norm of the factors  $\mathbf{U}$  and  $\mathbf{V}$  (see, e.g., Srebro et al. 2004, Mnih and Salakhutdinov 2007, Zhou et al. 2008). This regularization on the factors  $\mathbf{U}$  and  $\mathbf{V}$  can be related to a nuclear norm regularization on  $\mathbf{X}$  (Srebro et al. 2004, Lemma 1)—note that the nuclear norm is often used as a convex surrogate for rank (see, e.g., Candès and Recht 2009, Candès and Plan 2010, Mazumder et al. 2020) but, in these works, this regularization term is introduced in addition to an explicit low-rank decomposition. Motivated by the ElasticNet regularization for sparse linear regression, Sun and Zhang (2012) consider a matrix completion problem with a penalty on the nuclear norm and on the squared Frobenius norm of  $\mathbf{X}$ . Similarly, and closer to our formulation, Keshavan and Montanari (2010) consider a matrix completion problem with an explicit rank requirement and a squared Frobenius penalty on  $\mathbf{X}$ .

Furthermore, we can connect the optimal objective value of the regularized and the unregularized matrix completion problems:

**PROPOSITION EC.1.** *Let  $v(\gamma)$  denote the optimal objective value of Problem (1), with minimizer  $\mathbf{X}^*(\gamma)$  and let  $v^*$  denote the optimal objective value of the unregularized problem with minimizer  $\mathbf{X}^*$ . We have the following inequality*

$$v(\gamma) - \frac{1}{2\gamma}\|\mathbf{X}^*\|_F^2 \leq v^* \leq v(\gamma) \tag{EC.1}$$

Hence, for a fixed input data  $\mathbf{A}$ , we have  $v(\gamma) \rightarrow v^*$  as  $\gamma \rightarrow \infty$ . This shows that the regularized objective can approximate the nominal objective to arbitrary precision, by taking  $\gamma \rightarrow \infty$ .

*Proof of Proposition EC.1* To show the first inequality, we use the fact that  $\mathbf{X}^\star$  is feasible for the minimization problem (1) to write

$$v(\gamma) \leq \frac{1}{2\gamma} \|\mathbf{X}^\star\|_F^2 + \frac{1}{2} \sum_{(i,j) \in \mathcal{I}} (X_{i,j}^\star - A_{i,j})^2 = \frac{1}{2\gamma} \|\mathbf{X}^\star\|_F^2 + v^\star.$$

For the second inequality, we use the fact that  $\mathbf{X}^\star(\gamma)$  is feasible for the unregularized problem and write

$$v^\star \leq \frac{1}{2} \sum_{(i,j) \in \mathcal{I}} (X^\star(\gamma)_{i,j} - A_{i,j})^2 \leq \frac{1}{2\gamma} \|\mathbf{X}^\star(\gamma)\|_F^2 + \frac{1}{2} \sum_{(i,j) \in \mathcal{I}} (X^\star(\gamma)_{i,j} - A_{i,j})^2 = v(\gamma). \quad \square$$

## EC.2. Proof of Proposition 1

*Proof of Proposition 1* It suffices to show that every feasible solution to Problem (3) can be mapped to a feasible solution to (4) with the same objective value. Therefore, let  $(\mathbf{Y}, \mathbf{U})$  be a feasible solution to Problem (3). For  $(i, j) \in [n] \times [k]$ ,  $U_{i,j}$  satisfies a constraint  $U_{i,j} \in [\underline{U}_{i,j}, \bar{U}_{i,j}]$ . If  $i \in \mathcal{I}(j)$ , this corresponds to a branch of the McCormick disjunction (and we omit the index  $t$  of that particular branch for concision), while if  $i \notin \mathcal{I}(j)$ , it is simply  $U_{i,j} \in [-1, 1]$  (i.e.,  $\underline{U}_{i,j} = -1$  and  $\bar{U}_{i,j} = 1$ ). Then, we need to construct a  $\mathbf{V} \in \mathbb{R}^{n \times k \times k}$  such that  $(\mathbf{Y}, \mathbf{U}, \mathbf{V})$  satisfies the constraints in (4) to establish the result.

Take  $j \in [k]$ . For any  $i \in [n]$ , we need  $(V_{i,j,j}, U_{i,j}, U_{i,j}) \in \mathcal{M}(\underline{U}_{i,j}, \bar{U}_{i,j}, \underline{U}_{i,j}, \bar{U}_{i,j})$ . We set  $V_{i,j,j} = U_{i,j}^2$  for any  $i \in \mathcal{I}(j)$  (which clearly satisfies the McCormick constraint). If  $i \notin \mathcal{I}(j)$ , the McCormick constraint  $(V_{i,j,j}, U_{i,j}, U_{i,j}) \in \mathcal{M}(-1, 1, -1, 1)$  reduces to  $V_{i,j,j} \in [2|U_{i,j}| - 1, 1]$ . We distinguish two cases—note that one of these must hold since  $|\mathcal{I}(j)| \leq n - 1$  and hence there exists some  $i \notin \mathcal{I}(j)$

*Case 1: Suppose that  $\sum_{i \notin \mathcal{I}(j)} U_{i,j}^2 > 0$ .* Then, first, we observe that the constraint  $\mathbf{I} \succeq \mathbf{U}\mathbf{U}^\top$  implies  $\sum_{i \in [n]} U_{i,j}^2 \leq 1$ . Thus, define  $\alpha \geq 1$  such that  $\sum_{i \in \mathcal{I}(j)} U_{i,j}^2 + \alpha \sum_{i \notin \mathcal{I}(j)} U_{i,j}^2 = 1$ . We set  $V_{i,j,j} = \alpha U_{i,j}^2$  for any  $i \notin \mathcal{I}(j)$ . We have  $V_{i,j,j} \geq U_{i,j}^2 \geq 2|U_{i,j}| - 1$  because  $\alpha \geq 1$ , and  $V_{i,j,j} \leq 1$ .

*Case 2: If  $\sum_{i \notin \mathcal{I}(j)} U_{i,j}^2 = 0$*  (hence,  $U_{i,j} = 0, \forall i \notin \mathcal{I}(j)$ ), denote  $\delta = 1 - \sum_{i \in \mathcal{I}(j)} U_{i,j}^2 \in [0, 1]$  and set  $V_{i,j,j} = \delta / (n - |\mathcal{I}(j)|) \in [0, 1] \subseteq [2|U_{i,j}| - 1, 1]$ .

In both cases, we have that  $V_{i,j,j} \in [2|U_{i,j}| - 1, 1]$  as well as  $\sum_{i=1}^n V_{i,j,j} = 1$ .

Similarly, consider  $j_1, j_2 \in [k]$  with  $j_1 \neq j_2$  and set  $V_{i,j_1,j_2} = U_{i,j_1} U_{i,j_2}$  for any  $i \in \mathcal{I}(j_1) \cup \mathcal{I}(j_2)$ . For these indices  $i$ ,  $(V_{i,j_1,j_2}, U_{i,j_1}, U_{i,j_2}) \in \mathcal{M}(\underline{U}_{i,j_1}, \bar{U}_{i,j_1}, \underline{U}_{i,j_2}, \bar{U}_{i,j_2})$  by definition. For  $i \notin \mathcal{I}(j_1) \cup$

$\mathcal{I}(j_2)$ —for concision, we will denote  $\mathcal{F} := [n] \setminus (\mathcal{I}(j_1) \cup \mathcal{I}(j_2))$ —we need to find a value  $V_{i,j_1,j_2}$  such that  $V_{i,j_1,j_2} \in [|U_{i,j_1} + U_{i,j_2}| - 1, 1 - |U_{i,j_1} - U_{i,j_2}|]$  and  $\sum_{i \in [n]} V_{i,j_1,j_2} = \sum_{i \in \mathcal{I}(j_1) \cup \mathcal{I}(j_2)} U_{i,j_1} U_{i,j_2} + \sum_{i \in \mathcal{F}} V_{i,j_1,j_2} = 0$ . Equivalently, we need to show that  $-\sum_{i \in \mathcal{I}(j_1) \cup \mathcal{I}(j_2)} U_{i,j_1} U_{i,j_2} \in [L, U]$  with  $L := \sum_{i \in \mathcal{F}} (|U_{i,j_1} + U_{i,j_2}| - 1)$  and  $U := \sum_{i \in \mathcal{F}} (1 - |U_{i,j_1} - U_{i,j_2}|)$ .

Next, let us denote  $\mathbf{U}_{\cdot,j}$  the  $j$ -th column of  $\mathbf{U}$ . Since  $\mathbf{U}\mathbf{U}^\top \preceq \mathbf{I}$ , we have  $|\sum_{i \in \mathcal{I}(j_1) \cup \mathcal{I}(j_2)} U_{i,j_1} U_{i,j_2}| \leq \|\mathbf{U}_{\cdot,j_1}\|_2 \|\mathbf{U}_{\cdot,j_2}\|_2 \leq 1$  which implies that  $-\sum_{i \in \mathcal{I}(j_1) \cup \mathcal{I}(j_2)} U_{i,j_1} U_{i,j_2} \in [-1, 1]$ . Hence, it is sufficient to show that, under the assumption that  $|\mathcal{F}| = n - |\mathcal{I}(j_1) \cup \mathcal{I}(j_2)| \geq 4$ , we have  $L \leq -1$  and  $U \geq 1$ .

We have

$$\begin{aligned} L &= \sum_{i \in \mathcal{F}} (|U_{i,j_1} + U_{i,j_2}| - 1) = \sum_{i \in \mathcal{F}} |U_{i,j_1} + U_{i,j_2}| - |\mathcal{F}| \leq \sqrt{|\mathcal{F}|} \sqrt{\sum_{i \in \mathcal{F}} (U_{i,j_1} + U_{i,j_2})^2} - |\mathcal{F}| \\ &\leq \sqrt{2|\mathcal{F}|} - |\mathcal{F}|, \end{aligned}$$

where the first inequality holds due to Cauchy-Schwarz, and the second inequality holds because  $\|\mathbf{U}_{\cdot,j_1} \pm \mathbf{U}_{\cdot,j_2}\|_2^2 \leq 2$ . Hence, for  $|\mathcal{F}| \geq 4$ , we have  $L \leq -1$ . Similarly, for  $|\mathcal{F}| \geq 4$ , we have

$$\sum_{i \in \mathcal{F}} (1 - |U_{i,j_1} - U_{i,j_2}|) \geq |\mathcal{F}| - \sqrt{2|\mathcal{F}|} \geq 1.$$

Therefore,  $(\mathbf{Y}, \mathbf{U}, \mathbf{V})$  is feasible within one branch of our disjunction, and there is thus no hope of improving upon the root node relaxation with this disjunction.  $\square$

### EC.3. Branch-and-bound design decisions

In this section, we provide more detail on the design decisions we make in proposing our branch-and-bound algorithm, namely node selection (EC.3.1) and branching strategy (EC.3.2).

#### EC.3.1. Node selection: depth-first vs. breadth-first vs. best-first search

One of the most significant design decisions in a branch-and-bound scheme is the node selection strategy employed. The three node selection heuristics that we consider in this work are depth-first search (where nodes are selected in a last-in-first-out manner), breadth-first search (where nodes are selected in a first-in-first-out manner), and best-first search (where the node with the lowest remaining lower bound is selected at each iteration).

Breadth-first and best-first search ensure that the overall lower bound increases at most iterations, provided no ties exist. As argued by Lawler and Wood (1966), best-first search is potentially advantageous, because if the set of branching directions is fixed, then any nodes expanded under this strategy must also be expanded under any other strategy. However, both strategies incur a high memory cost from maintaining many unexplored nodes in the queue. On the other hand, depth-first search maintains a queue size which is a linear function of the problem size throughout the entire search process (Ibaraki 1976), although at the price of spending less time tightening the upper bound and, therefore, often needed to expand more nodes overall. We also observe empirically in Section 5 that best-first search usually performs better compared to the other strategies; therefore, we propose using best-first search unless the instance generates too many nodes to fit in available memory, at which point the depth-first strategy should be used.

Finally, while we assumed a breadth-first strategy to establish the convergence of Algorithm 1 in Theorem 1, the convergence result holds with any node selection strategy.

### **EC.3.2. Branching strategy**

When running our branch-and-bound algorithm on a rank- $k$  matrix, each node that is not fathomed generates  $q^k$ ,  $q \geq 2$ , child nodes corresponding to the  $q^k$  regions of the disjunction. For example, the disjunction (6) generates  $2^k$  child nodes, but, as explained for the rank-one case in Section 2.3, more fine-grained upper approximations of the  $\ell_2^2$  norm lead to disjunctions over  $3^k$  or  $4^k$  regions.

Accordingly, another algorithmic design decision is selecting the number of pieces  $q \geq 2$  which should be used in our disjunctive branching scheme. Indeed, increasing the number of child nodes generated at each iteration,  $q^k$ , improves the tightness of the bound at the expense of additional computational time for solving all  $q^k$  subproblems. We investigate this tradeoff numerically in Section 5.

## **EC.4. Numerical results**

### **EC.4.1. Root Node: Strengthened Relaxations**

Tables EC.1–EC.2 illustrate the trade-off between optimality gap at the root node (the upper bound is obtained via alternating minimization) and computational time for solving the strengthened relaxation in rank-one matrix completion. For each parameter setting, we either impose no Shor LMIs, all Shor LMIs on the minors in  $\mathcal{M}_4$  (4 filled entries), all Shor LMIs in  $\mathcal{M}_4$  and half of the Shor LMIs in  $\mathcal{M}_3$ , and all Shor LMIs in  $\mathcal{M}_4$  and  $\mathcal{M}_3$ . From Table EC.1, we see that larger instances (larger  $n$ ) with more observed entries (larger  $p$ ) are computationally more challenging, and that

$n$	$p$	$\gamma$	Time taken (s)			
			None	$\mathcal{M}_4$	$\mathcal{M}_4 \cup \mathcal{M}_3$ (half)	$\mathcal{M}_4 \cup \mathcal{M}_3$
10	2.0	20.0	0.011 68	0.028 91	0.1078	0.1553
		80.0	0.012 34	0.027 35	0.1227	0.2063
	3.0	20.0	0.012 06	0.047 75	0.3827	0.7328
		80.0	0.013 02	0.045 04	0.4023	0.7577
20	2.0	20.0	0.074 37	0.3082	1.505	2.854
		80.0	0.075	0.2884	1.678	4.091
	3.0	20.0	0.072 91	0.7584	6.02	21.57
		80.0	0.0772	0.6874	7.617	21.73
30	2.0	20.0	0.329	2.161	8.45	19
		80.0	0.3629	1.637	10.11	22.88
	3.0	20.0	0.3502	3.234	47.37	177.3
		80.0	0.3723	4.21	56.22	177.9
50	2.0	20.0	3.066	29.24	94.18	216.5
		80.0	3.357	25.38	119.4	271.4
	3.0	20.0	3.244	30.12	884	2787
		80.0	3.515	46.43	1090	2628
75	2.0	20.0	22.85	238.7	1002	2073
		80.0	24.62	253.7	1070	2473
	3.0	20.0	25.36	266.7	–	–
		80.0	27.07	395.5	–	–
100	2.0	20.0	111.2	1628	–	–
		80.0	121.7	1386	–	–
	3.0	20.0	112.1	1298	–	–
		80.0	125.4	1931	–	–

**Table EC.1** Time taken to solve the root node relaxation across rank-one matrix completion problems with  $pn \log_{10}(n)$  filled entries, with different Shor LMIs added (arithmetic mean taken over 20 instances per row). A “–” indicates parameter settings that did not complete within the 1-hour time limit.

imposing more Shor LMIs can significantly increase the computation time required. In return, Table EC.2 indicates that imposing more Shor LMIs can yield tighter semidefinite relaxations, successfully narrowing the root node relative gap in a majority of trials. This effect holds across all sizes, regularization parameters, and sparsity settings. Whilst the (arithmetic mean of the) root node gap appears to decrease marginally with the inclusion of all minors in  $\mathcal{M}_4$  and  $\mathcal{M}_3$  as compared to without, these averages are distorted by the large root node gaps of a few instances which see little benefit from the inclusion of  $\mathcal{M}_3$  minors. Including constraints on all minors typically increases the fraction of instances with a root node gap lower than  $10^{-4}$  by 10 to 50 percentage points (a 20–83% improvement), and this effect is stronger as the number of filled entries increases (larger  $p$ ) and regularization decreases (larger  $\gamma$ ). Including a subset of the minors in  $\mathcal{M}_4 \cup \mathcal{M}_3$  provides a finer control of this trade-off between computation time and relaxation strength.

		Root node gap								
		None		$\mathcal{M}_4$		$\mathcal{M}_4 \cup \mathcal{M}_3$ (half)		$\mathcal{M}_4 \cup \mathcal{M}_3$		
10	2.0	20.0	$3.70 \times 10^{-1}$	(5%)	$3.68 \times 10^{-1}$	(5%)	$3.64 \times 10^{-1}$	(25%)	$3.56 \times 10^{-1}$	(25%)
		80.0	$1.61 \times 10^0$	(0%)	$1.59 \times 10^0$	(0%)	$1.54 \times 10^0$	(5%)	$1.15 \times 10^0$	(10%)
	3.0	20.0	$1.73 \times 10^{-1}$	(10%)	$1.68 \times 10^{-1}$	(15%)	$1.51 \times 10^{-1}$	(40%)	$1.47 \times 10^{-1}$	(50%)
		80.0	$7.11 \times 10^{-1}$	(0%)	$6.51 \times 10^{-1}$	(0%)	$5.61 \times 10^{-1}$	(20%)	$5.40 \times 10^{-1}$	(45%)
20	2.0	20.0	$1.14 \times 10^{-1}$	(15%)	$1.36 \times 10^{-1}$	(25%)	$1.10 \times 10^{-1}$	(45%)	$1.31 \times 10^{-1}$	(55%)
		80.0	$1.65 \times 10^{-1}$	(0%)	$1.53 \times 10^{-1}$	(0%)	$1.23 \times 10^{-1}$	(0%)	$1.07 \times 10^{-1}$	(15%)
	3.0	20.0	$6.17 \times 10^{-2}$	(10%)	$5.81 \times 10^{-2}$	(25%)	$5.48 \times 10^{-2}$	(65%)	$5.46 \times 10^{-2}$	(70%)
		80.0	$2.71 \times 10^{-1}$	(0%)	$2.23 \times 10^{-1}$	(0%)	$1.70 \times 10^{-1}$	(60%)	$1.69 \times 10^{-1}$	(70%)
30	2.0	20.0	$3.93 \times 10^{-2}$	(15%)	$3.90 \times 10^{-2}$	(15%)	$3.63 \times 10^{-2}$	(45%)	$3.60 \times 10^{-2}$	(45%)
		80.0	$1.84 \times 10^{-1}$	(0%)	$1.79 \times 10^{-1}$	(0%)	$1.50 \times 10^{-1}$	(0%)	$1.40 \times 10^{-1}$	(30%)
	3.0	20.0	$2.07 \times 10^{-2}$	(25%)	$1.96 \times 10^{-2}$	(40%)	$1.89 \times 10^{-2}$	(70%)	$1.89 \times 10^{-2}$	(70%)
		80.0	$1.17 \times 10^{-1}$	(0%)	$9.68 \times 10^{-2}$	(0%)	$7.06 \times 10^{-2}$	(60%)	$6.99 \times 10^{-2}$	(70%)
50	2.0	20.0	$6.47 \times 10^{-3}$	(45%)	$6.33 \times 10^{-3}$	(45%)	$5.15 \times 10^{-3}$	(60%)	$4.85 \times 10^{-3}$	(70%)
		80.0	$4.85 \times 10^{-2}$	(0%)	$4.60 \times 10^{-2}$	(0%)	$3.02 \times 10^{-2}$	(20%)	$2.62 \times 10^{-2}$	(35%)
	3.0	20.0	$9.72 \times 10^{-3}$	(55%)	$9.40 \times 10^{-3}$	(55%)	$9.01 \times 10^{-3}$	(75%)	$8.78 \times 10^{-3}$	(80%)
		80.0	$5.29 \times 10^{-2}$	(0%)	$4.41 \times 10^{-2}$	(0%)	$3.32 \times 10^{-2}$	(65%)	$2.67 \times 10^{-2}$	(83%)
75	2.0	20.0	$4.26 \times 10^{-4}$	(65%)	$4.07 \times 10^{-4}$	(65%)	$3.94 \times 10^{-4}$	(75%)	$3.94 \times 10^{-4}$	(75%)
		80.0	$1.15 \times 10^{-2}$	(0%)	$1.01 \times 10^{-2}$	(0%)	$2.77 \times 10^{-3}$	(45%)	$1.89 \times 10^{-3}$	(50%)
	3.0	20.0	$2.19 \times 10^{-3}$	(65%)	$2.19 \times 10^{-3}$	(70%)	–	–	–	–
		80.0	$1.54 \times 10^{-2}$	(0%)	$1.15 \times 10^{-2}$	(0%)	–	–	–	–
100	2.0	20.0	$3.22 \times 10^{-3}$	(75%)	$3.21 \times 10^{-3}$	(75%)	–	–	–	–
		80.0	$1.75 \times 10^{-2}$	(0%)	$1.68 \times 10^{-2}$	(0%)	–	–	–	–
	3.0	20.0	$5.52 \times 10^{-3}$	(70%)	$5.52 \times 10^{-3}$	(70%)	–	–	–	–
		80.0	$2.21 \times 10^{-2}$	(10%)	$2.05 \times 10^{-2}$	(25%)	–	–	–	–

**Table EC.2** Relative gap at root node across rank-one matrix completion problems with  $pn \log_{10}(n)$  filled entries, with different Shor LMIs added (arithmetic mean taken over 20 instances per row). Values in parentheses denote proportion of trials which converged (root node gap less than  $10^{-4}$ ), out of those completed within the time limit. A “–” indicates parameter settings which did not complete within the 1-hour time limit.

#### EC.4.2. Branch-and-bound Design Decisions

In this section, we document the performance of our branch-and-bound scheme with various parameter settings: comparing our eigenvector disjunctions to a naive McCormick-based approach, changing the order of the nodes explored, and including alternating minimization at child nodes of the search tree. We evaluate all combinations of parameter settings, and record their relative optimality gap and time taken (capped at one hour) for rank-one matrix completion problems with regularization  $\gamma \in \{20.0, 80.0\}$  and  $pn \log_{10}(n)$  filled entries with  $p \in \{2.0, 3.0\}$ . The results are shown in Tables 1–2, EC.3–EC.4 (less entries  $p = 2.0$  and less regularization  $\gamma = 80.0$ ), EC.5–EC.6 (more entries  $p = 3.0$  and more regularization  $\gamma = 20.0$ ), and EC.7–EC.8 (more entries  $p = 3.0$  and less regularization  $\gamma = 80.0$ ) respectively.

$n$	Alternating minimization	With McCormick disjunctions			With eigenvector disjunctions		
		Best-first	Breadth-first	Depth-first	Best-first	Breadth-first	Depth-first
10	✗	$1.42 \times 10^0$	$1.50 \times 10^0$	$1.50 \times 10^0$	$2.21 \times 10^0$	$2.40 \times 10^0$	$1.91 \times 10^0$
10	✓	$7.97 \times 10^{-2}$	$1.14 \times 10^{-1}$	$6.26 \times 10^{-1}$	$5.39 \times 10^{-2}$	$8.89 \times 10^{-2}$	$3.18 \times 10^{-1}$
20	✗	$3.38 \times 10^{-1}$	$3.38 \times 10^{-1}$	$3.38 \times 10^{-1}$	$5.02 \times 10^{-1}$	$5.27 \times 10^{-1}$	$4.09 \times 10^{-1}$
20	✓	$8.61 \times 10^{-2}$	$8.77 \times 10^{-2}$	$8.77 \times 10^{-2}$	$5.11 \times 10^{-2}$	$6.50 \times 10^{-2}$	$2.25 \times 10^{-1}$
30	✗	$2.34 \times 10^{-1}$	$2.34 \times 10^{-1}$	$2.34 \times 10^{-1}$	$2.34 \times 10^{-1}$	$2.42 \times 10^{-1}$	$2.18 \times 10^{-1}$
30	✓	$1.08 \times 10^{-1}$	$1.13 \times 10^{-1}$	$1.13 \times 10^{-1}$	$4.71 \times 10^{-2}$	$5.42 \times 10^{-2}$	$1.04 \times 10^{-1}$
40	✗	$2.12 \times 10^{-1}$	$2.12 \times 10^{-1}$	$2.12 \times 10^{-1}$	$2.07 \times 10^{-1}$	$2.11 \times 10^{-1}$	$2.13 \times 10^{-1}$
40	✓	$1.05 \times 10^{-1}$	$1.05 \times 10^{-1}$	$1.05 \times 10^{-1}$	$3.08 \times 10^{-2}$	$3.38 \times 10^{-2}$	$8.51 \times 10^{-2}$
50	✗	$7.45 \times 10^{-2}$	$7.45 \times 10^{-2}$	$7.45 \times 10^{-2}$	$1.35 \times 10^{-1}$	$1.37 \times 10^{-1}$	$1.39 \times 10^{-1}$
50	✓	$4.53 \times 10^{-2}$	$4.53 \times 10^{-2}$	$4.53 \times 10^{-2}$	$2.15 \times 10^{-2}$	$2.32 \times 10^{-2}$	$6.64 \times 10^{-2}$

**Table EC.3** Final optimality gap across rank-one matrix completion problems with  $|\mathcal{I}| = pn \log_{10}(n)$  filled entries, averaged over 20 instances per row ( $p = 2.0$ ,  $\gamma = 80.0$ ).

$n$	Alternating minimization	With McCormick disjunctions			With eigenvector disjunctions		
		Best-first	Breadth-first	Depth-first	Best-first	Breadth-first	Depth-first
10	✗	3424.9	3425.3	3600.0	3430.1	3600.0	3600.0
10	✓	3162.1	3204.4	3600.0	3274.8	3600.0	3600.0
20	✗	3600.1	3600.1	3600.1	3600.1	3600.1	3600.1
20	✓	3600.1	3600.1	3600.1	3600.1	3600.1	3600.1
30	✗	3600.4	3600.3	3600.3	3600.3	3600.3	3600.4
30	✓	3600.4	3600.3	3600.2	3600.3	3600.3	3600.4
40	✗	3601.0	3601.1	3600.7	3600.8	3600.9	3600.9
40	✓	3601.2	3601.0	3600.8	3600.9	3601.1	3600.8
50	✗	3602.5	3602.1	3602.1	3602.0	3602.1	3602.7
50	✓	3602.4	3602.1	3602.1	3601.9	3602.5	3602.1

**Table EC.4** Computational time (s) across rank-one matrix completion problems with  $|\mathcal{I}| = pn \log_{10}(n)$  entries, using McCormick disjunctions (top), eigenvector disjunctions (bottom), averaged over 20 instances ( $p = 2.0$ ,  $\gamma = 80.0$ ).

The tables show that eigenvector disjunctions perform consistently better than McCormick disjunctions, and that best-first search on unexplored child nodes is usually a good node selection strategy. They also illustrate the power of performing alternating minimization at (some) child nodes, because better feasible solutions can be found with different initializations, which yield tight upper bounds and thereby accelerating the branch-and-bound procedure.

In Figure EC.1, we observe the effect of the number of pieces used  $q$  in our disjunctive scheme on the final relative gap after branch-and-bound for one hour on rank-one matrix completion problems with  $n \geq 50$ . As the problem size  $n$  increases, keeping the number of observed entries at  $pn \log_{10}(n)$

$n$	Alternating minimization	With McCormick disjunctions			With eigenvector disjunctions		
		Best-first	Breadth-first	Depth-first	Best-first	Breadth-first	Depth-first
10	✗	$3.35 \times 10^{-1}$	$3.55 \times 10^{-1}$	$3.57 \times 10^{-1}$	$1.73 \times 10^{-1}$	$1.73 \times 10^{-1}$	$3.13 \times 10^{-1}$
10	✓	$3.40 \times 10^{-2}$	$4.83 \times 10^{-2}$	$1.49 \times 10^{-1}$	$7.16 \times 10^{-3}$	$1.29 \times 10^{-2}$	$1.15 \times 10^{-1}$
20	✗	$4.63 \times 10^{-2}$	$4.63 \times 10^{-2}$	$4.63 \times 10^{-2}$	$1.08 \times 10^{-2}$	$1.16 \times 10^{-2}$	$1.20 \times 10^{-2}$
20	✓	$9.58 \times 10^{-3}$	$2.54 \times 10^{-2}$	$2.54 \times 10^{-2}$	$1.79 \times 10^{-4}$	$9.03 \times 10^{-4}$	$1.13 \times 10^{-2}$
30	✗	$6.18 \times 10^{-2}$	$6.18 \times 10^{-2}$	$6.18 \times 10^{-2}$	$9.70 \times 10^{-3}$	$9.57 \times 10^{-3}$	$4.65 \times 10^{-2}$
30	✓	$3.99 \times 10^{-3}$	$1.56 \times 10^{-2}$	$1.56 \times 10^{-2}$	$1.32 \times 10^{-4}$	$2.22 \times 10^{-4}$	$3.02 \times 10^{-3}$
40	✗	$1.27 \times 10^{-2}$	$1.27 \times 10^{-2}$	$1.27 \times 10^{-2}$	$4.43 \times 10^{-3}$	$4.55 \times 10^{-3}$	$8.10 \times 10^{-3}$
40	✓	$9.62 \times 10^{-3}$	$9.62 \times 10^{-3}$	$9.62 \times 10^{-3}$	$1.96 \times 10^{-4}$	$2.89 \times 10^{-4}$	$8.44 \times 10^{-3}$
50	✗	$9.11 \times 10^{-3}$	$9.11 \times 10^{-3}$	$9.11 \times 10^{-3}$	$1.01 \times 10^{-4}$	$1.07 \times 10^{-4}$	$2.31 \times 10^{-3}$
50	✓	$7.13 \times 10^{-3}$	$7.13 \times 10^{-3}$	$7.13 \times 10^{-3}$	$1.01 \times 10^{-4}$	$1.06 \times 10^{-4}$	$6.18 \times 10^{-3}$

**Table EC.5** Final optimality gap across rank-one matrix completion problems with  $|\mathcal{I}| = pn \log_{10}(n)$  filled entries, averaged over 20 instances per row ( $p = 3.0, \gamma = 20.0$ ).

$n$	Alternating minimization	With McCormick disjunctions			With eigenvector disjunctions		
		Best-first	Breadth-first	Depth-first	Best-first	Breadth-first	Depth-first
10	✗	2547.4	2553.2	3060.0	2152.4	2118.6	3060.1
10	✓	1871.5	1885.8	3060.0	1524.9	1990.3	2809.7
20	✗	3240.1	3240.1	3240.1	1746.0	2260.7	2880.1
20	✓	2704.7	3060.1	3060.1	1743.3	2262.4	2880.1
30	✗	3060.4	3060.3	3060.3	1340.8	1517.6	2880.4
30	✓	2565.5	2700.5	2700.4	854.99	1429.6	2340.5
40	✗	1981.3	1981.3	1981.3	976.3	1546.2	1981.2
40	✓	1981.4	1981.5	1981.2	597.66	1141.3	1981.4
50	✗	1983.7	1983.8	1983.3	262.55	406.79	1100.1
50	✓	1443.7	1443.9	1443.7	253.22	381.37	959.29

**Table EC.6** Computational time (s) across rank-one matrix completion problems with  $|\mathcal{I}| = pn \log_{10}(n)$  entries, using McCormick disjunctions (top), eigenvector disjunctions (bottom), averaged over 20 instances ( $p = 3.0, \gamma = 20.0$ ).

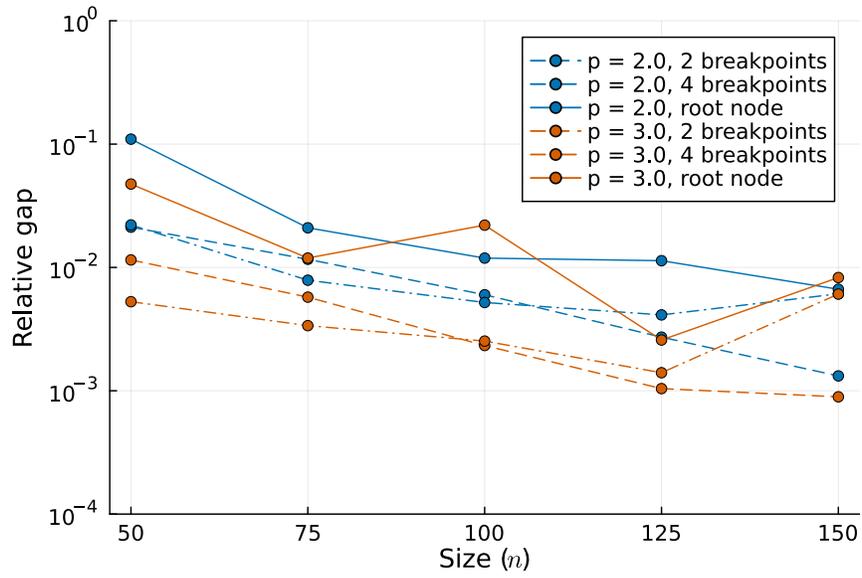
for  $p \in \{2.0, 3.0\}$ , the relative advantage of 4-piece disjunctions vanishes – this is likely due to the fact that, as  $n$  increases, the time taken for a single semidefinite relaxation increases and using 4-piece disjunctions introduces more child nodes, leading to an overall increase in computational time. Hence, we only recommend using 4-piece disjunctions for rank-one matrix completion problems of low to moderate size.

$n$	Alternating minimization	With McCormick disjunctions			With eigenvector disjunctions		
		Best-first	Breadth-first	Depth-first	Best-first	Breadth-first	Depth-first
10	✗	$5.26 \times 10^{-1}$	$5.93 \times 10^{-1}$	$5.96 \times 10^{-1}$	$4.73 \times 10^{-1}$	$5.40 \times 10^{-1}$	$5.76 \times 10^{-1}$
10	✓	$1.03 \times 10^{-1}$	$1.48 \times 10^{-1}$	$2.76 \times 10^{-1}$	$7.03 \times 10^{-2}$	$1.10 \times 10^{-1}$	$2.52 \times 10^{-1}$
20	✗	$4.63 \times 10^{-2}$	$4.63 \times 10^{-2}$	$4.63 \times 10^{-2}$	$1.08 \times 10^{-2}$	$1.16 \times 10^{-2}$	$1.20 \times 10^{-2}$
20	✓	$9.58 \times 10^{-3}$	$2.54 \times 10^{-2}$	$2.54 \times 10^{-2}$	$1.79 \times 10^{-4}$	$9.03 \times 10^{-4}$	$1.13 \times 10^{-2}$
30	✗	$6.18 \times 10^{-2}$	$6.18 \times 10^{-2}$	$6.18 \times 10^{-2}$	$9.70 \times 10^{-3}$	$9.57 \times 10^{-3}$	$4.65 \times 10^{-2}$
30	✓	$3.99 \times 10^{-3}$	$1.56 \times 10^{-2}$	$1.56 \times 10^{-2}$	$1.32 \times 10^{-4}$	$2.22 \times 10^{-4}$	$3.02 \times 10^{-3}$
40	✗	$1.27 \times 10^{-2}$	$1.27 \times 10^{-2}$	$1.27 \times 10^{-2}$	$4.43 \times 10^{-3}$	$4.55 \times 10^{-3}$	$8.10 \times 10^{-3}$
40	✓	$9.62 \times 10^{-3}$	$9.62 \times 10^{-3}$	$9.62 \times 10^{-3}$	$1.96 \times 10^{-4}$	$2.89 \times 10^{-4}$	$8.44 \times 10^{-3}$
50	✗	$9.11 \times 10^{-3}$	$9.11 \times 10^{-3}$	$9.11 \times 10^{-3}$	$1.01 \times 10^{-4}$	$1.07 \times 10^{-4}$	$2.31 \times 10^{-3}$
50	✓	$7.13 \times 10^{-3}$	$7.13 \times 10^{-3}$	$7.13 \times 10^{-3}$	$1.01 \times 10^{-4}$	$1.06 \times 10^{-4}$	$6.18 \times 10^{-3}$

**Table EC.7** Final optimality gap across rank-one matrix completion problems with  $|\mathcal{I}| = pn \log_{10}(n)$  filled entries, averaged over 20 instances per row ( $p = 3.0$ ,  $\gamma = 80.0$ ).

$n$	Alternating minimization	With McCormick disjunctions			With eigenvector disjunctions		
		Best-first	Breadth-first	Depth-first	Best-first	Breadth-first	Depth-first
10	✗	3329.9	3346.5	3600.0	3600.0	3600.0	3600.0
10	✓	3228.0	3308.0	3600.0	3600.0	3600.0	3600.0
20	✗	3600.1	3600.1	3600.1	3600.1	3600.1	3600.1
20	✓	3600.1	3600.1	3600.1	3600.1	3600.1	3600.1
30	✗	3600.3	3600.4	3600.3	3600.4	3600.3	3600.3
30	✓	3600.3	3600.4	3600.3	3600.4	3600.2	3600.4
40	✗	3601.2	3600.9	3600.8	3600.9	3600.8	3601.1
40	✓	3601.0	3601.0	3600.8	3601.1	3600.9	3601.0
50	✗	3602.4	3602.2	3602.2	3602.5	3602.0	3602.4
50	✓	3602.3	3602.9	3602.0	3602.3	3602.1	3602.9

**Table EC.8** Computational time (s) across rank-one matrix completion problems with  $|\mathcal{I}| = pn \log_{10}(n)$  entries, using McCormick disjunctions (top), eigenvector disjunctions (bottom), averaged over 20 instances ( $p = 3.0$ ,  $\gamma = 80.0$ ).

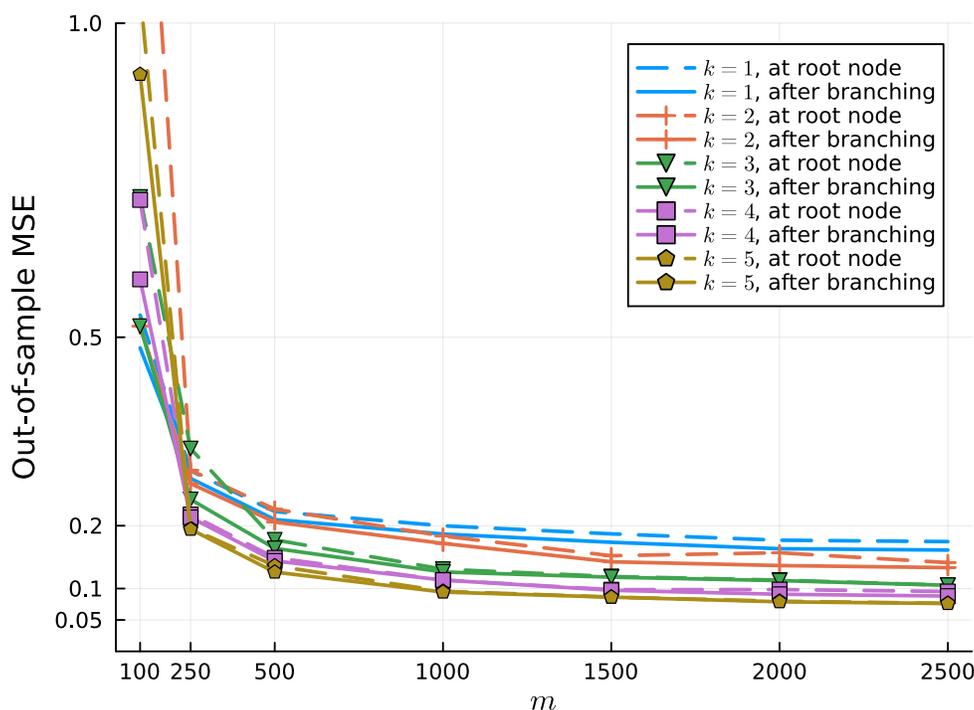


**Figure EC.1** Comparison of relative gap (at root node, and after branching with 2-piece and 4-piece disjunctions) for rank-one matrix completion problems with  $pn \log_{10}(n)$  filled entries, after one hour (averaged over 20 instances per point).

### EC.4.3. Scalability Experiments

Here we delve into a more detailed investigation of the scalability of our approach for matrix completion problems to large problem sizes and larger ranks.

Figure EC.2 shows the out-of-sample mean-squared error (MSE), at the root node and after branching for three hours, on the large rectangular  $50 \times m$  instances. These plots show that branching improves the out-of-sample MSE for all values of  $m$ , with the most significant improvement in out-of-sample MSE when  $m$  is small.



**Figure EC.2** Out-of-sample MSE at the root node and after branching, for rank- $k$  matrix completion problems of dimension  $50 \times m$ , with  $km \log_{10}(m)$  filled entries, varying  $m$  and  $k$ , with  $\gamma = 120.0$ , averaged over 10 random instances.

$k$	Relative improvement						
	$m = 100$	$m = 250$	$m = 500$	$m = 1000$	$m = 1500$	$m = 2000$	$m = 2500$
1	$8.22 \times 10^{-2}$	$4.15 \times 10^{-2}$	$5.31 \times 10^{-2}$	$5.66 \times 10^{-2}$	$6.16 \times 10^{-2}$	$6.33 \times 10^{-2}$	$6.65 \times 10^{-2}$
2	$5.84 \times 10^{-1}$	$6.68 \times 10^{-2}$	$8.21 \times 10^{-2}$	$6.40 \times 10^{-2}$	$4.91 \times 10^{-2}$	$1.02 \times 10^{-1}$	$4.39 \times 10^{-2}$
3	$1.33 \times 10^{-1}$	$8.83 \times 10^{-2}$	$6.30 \times 10^{-2}$	$3.11 \times 10^{-2}$	$9.48 \times 10^{-3}$	$7.07 \times 10^{-3}$	$-9.29 \times 10^{-6}$
4	$6.86 \times 10^{-2}$	$1.94 \times 10^{-2}$	$3.09 \times 10^{-2}$	$4.71 \times 10^{-4}$	$8.82 \times 10^{-3}$	$4.94 \times 10^{-2}$	$4.94 \times 10^{-2}$
5	$9.93 \times 10^{-2}$	$7.21 \times 10^{-3}$	$5.74 \times 10^{-2}$	$1.27 \times 10^{-2}$	$1.16 \times 10^{-3}$	$1.10 \times 10^{-2}$	$3.34 \times 10^{-3}$

**Table EC.9** Relative improvement in out-of-sample MSE after branching for 3 hours, for rank- $k$  matrix completion problems of dimension  $50 \times m$ , with  $km \log_{10}(m)$  entries, varying  $m$  and  $k$ , with  $\gamma = 120.0$ , averaged over 10 random instances.