Matrix Completion: Approximating the Minimum Diameter

Diptarka Chakraborty
National University of Singapore, Singapore

Sanjana Dey
National University of Singapore, Singapore

Abstract

In this paper, we focus on the matrix completion problem and aim to minimize the diameter over an arbitrary alphabet. Given a matrix M with missing entries, our objective is to complete the matrix by filling in the missing entries in a way that minimizes the maximum (Hamming) distance between any pair of rows in the completed matrix (also known as the *diameter* of the matrix). It is worth noting that this problem is already known to be NP-hard. Currently, the best-known upper bound is a 4-approximation algorithm derived by applying the triangle inequality together with a well-known 2-approximation algorithm for the radius minimization variant.

In this work, we make the following contributions:

- We present a novel 3-approximation algorithm for the diameter minimization variant of the matrix completion problem. To the best of our knowledge, this is the first approximation result that breaks below the straightforward 4-factor bound.
- Furthermore, we establish that the diameter minimization variant of the matrix completion problem is $(2-\varepsilon)$-inapproximable, for any $\varepsilon > 0$, even when considering a binary alphabet, under the assumption that $\mathsf{P} \neq \mathsf{NP}$. This is the first result that demonstrates a hardness of approximation for this problem.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases Incomplete Data, Matrix Completion, Hamming Distance, Diameter Minimization, Approximation Algorithms, Hardness of Approximation

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2023.17

Funding This work was supported by an MoE AcRF Tier 2 grant (MOE-T2EP20221-0009).

1 Introduction

With the advent of big data, the occurrence of missing values in data objects has become increasingly common. These missing entries can arise due to various errors occurring at different stages of data processing, including data collection, data transfer, and data cleaning, and they often even occur arbitrarily. Handling such missing data is widely recognized as a challenging task, and numerous methods, including heuristic, greedy, convex optimization, and statistical approaches, have been proposed in the context of practical applications [1]. One such popular technique is data imputation, which, albeit finds extensive use in data mining, machine learning, and computational biology, requires prior knowledge of the dataset or the adoption of certain statistical assumptions [33].

Addressing the issue of incomplete matrices by filling in missing values is a fundamental problem in data analysis, often approached as an optimization task [10, 20, 21, 29, 16, 17]. In the context of clustering, a popular objective function is to minimize the *cluster diameter* (e.g., [25, 13, 24, 29, 17]), which represents the maximum pairwise distance among data points within a cluster. When dealing with missing entries (or wildcards denoted by $*$), a fundamental question arises: Given an incomplete matrix $M \in (\Sigma \cup \{*\})^{n \times d}$ (over an arbitrary alphabet Σ), how can we fill the $*$-entries with symbols from Σ to obtain a

completion $\bar{M} \in \Sigma^{n \times d}$ that minimizes the maximum pairwise distance between rows? In this paper, we consider the Hamming distance as the underlying distance measure, which is arguably one of the most prevalent distance functions used in a wide range of applications. This problem is known as the Minimum Diameter Matrix Completion (DMC) problem.

The DMC problem is a combinatorial matrix completion problem with numerous applications in coding theory, computational biology, and data science. For instance, in computational biology, the DMC problem arises in assessing the degree of relatedness among genome sequences, where missing entries represent missing data points. The DMC problem is encountered in data science when completing entities with their attributes while satisfying pairwise dissimilarity constraints. The stringology literature extensively explores several consensus problems closely related to DMC [7, 4, 6, 10, 9, 11, 14, 26, 27, 37, 34, 32, 38].

The DMC problem, like most other variants of matrix completion problems, is known to be NP-hard. Koana, Froese, and Niedermeier [29] conducted a comprehensive complexity study on the DMC problem, considering diameter bounds and the maximum number of missing entries, and identified various polynomial-time solvable cases and NP-hard cases. The parameterized complexities of the DMC problem, specifically a more general k-clustering version, have also been investigated in terms of various parameters [16, 17]. Regarding approximation algorithms, only a 4-approximation algorithm is currently known for the DMC problem. This approximation factor is derived from the result of another closely related matrix completion problem called Minimum Radius Matrix Completion (RadMC) [28, 27]. In the RadMC problem, the objective is to find a completion and a "center" string such that the distance between each row of the completed matrix and the center string is minimized. A straightforward application of the triangle inequality shows that any c-approximation (for any $c \geq 1$) to the RadMC problem implies a $2c$-approximation to the DMC problem. Since a simple (folklore) 2-approximation algorithm[1] exists for the RadMC problem, it immediately provides a 4-factor approximate solution for the DMC problem. However, it has been proven that no $(2-\varepsilon)$-approximation algorithm for the RadMC problem exists unless $\mathsf{P} = \mathsf{NP}$ [12], and thus there is no hope of getting a better factor than 4 to the DMC problem by improving the approximation factor of the RadMC problem.

Currently, the possibility of improving the 4-factor approximation for the DMC problem remains an open question. Moreover, there is no known inapproximability result for the DMC problem, leaving room for the plausibility of achieving a polynomial-time approximation scheme (PTAS). In this paper, we refute this possibility by demonstrating that no polynomial-time $(2-\varepsilon)$-approximation algorithm for the DMC problem exists unless $\mathsf{P} = \mathsf{NP}$. Furthermore, we present a 3-approximation algorithm that surpasses the 4-factor bound obtained from a direct application of the triangle inequality, along with a 2-factor algorithm for the RadMC problem.

Our contributions and techniques. One of our primary contributions is a 3-approximation algorithm for the DMC problem, which to the best of our knowledge, is the first one to break below the straightforward 4-approximation bound.

▶ **Theorem 1.** *There is a polynomial-time algorithm that, given an incomplete matrix $M \in (\Sigma \cup \{*\})^{n \times d}$, computes a 3-approximate solution for the Minimum Diameter Matrix Completion (DMC) problem over an arbitrary alphabet Σ.*

[1] The 2-approximation is attained by first solving the LP relaxation of the standard ILP formulation of the RadMC problem and then applying a simple deterministic rounding. We use a similar argument to get a 2-approximation for a restricted version of the DMC problem, namely the DRMC problem (see Appendix A).

To show our result, we consider an intermediate restricted variant of the DMC problem, which we refer to as Minimum Diameter Restricted Matrix Completion (DRMC). In this variant, we add a *column restriction* – all the missing entries of a column must be filled in with the same symbol – for feasible completion of an incomplete matrix. The main advantage of putting this restriction is that now we can formulate this restricted variant as an ILP. Then we solve the corresponding LP relaxation, and finally, applying a simple deterministic rounding, we get a 2-approximation for the DRMC problem (Theorem 5).

Since the only distinction between the DMC and the DRMC problem is the column restriction imposed on feasible completions in the DRMC problem, any feasible solution to the DRMC problem is also a feasible solution to the DMC problem. Surprisingly, we show that for any input incomplete matrix M, the optimum objective value to the DRMC problem is at most 3/2 times that of the DMC problem (Lemma 3). By leveraging this finding along with the 2-approximation algorithm for the DRMC, we can effectively establish Theorem 1.

To build the relationship between the optimal solution of the DRMC and the DMC, we consider an (arbitrary) optimal completion $\bar{M}^*$ for the DMC problem, acknowledging that this completion may not satisfy the column restriction requirement for the DRMC problem. To overcome this, we modify the completion by taking any arbitrary row, say the first row $\bar{M}_1^*$, and then for each column ℓ depending on the symbol at the ℓ-th coordinate of the row $\bar{M}_1^*$ fill in all the missing entries of that column in the whole input (incomplete) matrix. This modification yields a completion $\tilde{M}$ that satisfies the column restriction and thus becomes a feasible solution to the DRMC problem. Let Δ and Δ_R represent the diameters of the completed matrices for DMC ($\bar{M}^*$) and DRMC ($\tilde{M}$), respectively. We aim to demonstrate that $\Delta_R \leq \frac{3}{2} \cdot \Delta$. To provide a high-level idea, let us consider any two rows i, j. It is not hard to see that by applying the triangle inequality, the distance between the rows $\tilde{M}_i$ and $\tilde{M}_j$ is at most twice Δ. However, that only shows $\Delta_R \leq 2\Delta$, which, when combined with a 2-approximation of the DRMC, only gives a 4-approximation to the DMC problem, which is no better than the already known bound. Overcoming this challenge requires developing an argument that surpasses this naive application of the triangle inequality. To tackle this challenge, we divide the rows $\tilde{M}_i$ and $\tilde{M}_j$ into three parts: The first one comprises coordinates where no missing entries are there in both the i-th and j-th row of the input matrix, the second one contains all the coordinates with missing entries in the i-th row but no missing entries in the j-th row, and the third one consists of all the coordinates with missing entries in the j-th row but no missing entries in the i-th row (we disregard the coordinates with missing entries in both the rows because these positions do not contribute to the Hamming distance due to the column restriction). We emphasize that this partitioning is solely for the sake of analysis. Next, we look into these three parts separately and analyze their contributions to the overall Hamming distance. Finally, by using the fact that in $\bar{M}^*$, all the pairwise distances between the rows $\bar{M}_1^*, \bar{M}_i^*$, and $\bar{M}_j^*$ are bounded by Δ, we establish that the distance between the rows $\tilde{M}_i$ and $\tilde{M}_j$ is at most $\frac{3}{2} \cdot \Delta$. The detailed argument is provided in Section 3.

Our next significant contribution is an inapproximability result for the DMC problem. We show that it is NP-hard to get a $(2-\varepsilon)$-approximation, the first inapproximability result for the DMC problem.

▶ **Theorem 2.** *Consider any $\varepsilon > 0$. There is no deterministic polynomial-time algorithm that, given an incomplete matrix $M \in \{0, 1, *\}^{n \times d}$, computes a $(2-\varepsilon)$-approximate solution for the Minimum Diameter Matrix Completion (DMC) problem, unless* P = NP.

We highlight that the aforementioned inapproximability result holds even for a binary alphabet. To establish the inapproximability bound, we employ a reduction from the well-known *Label Cover problem* to a gap version of the DMC problem (Definition 6). Informally

speaking, in the label cover problem, we are presented with a (left and right-regular) bipartite graph with each edge having a function (defined on a label set) as a constraint relation, and the objective is to come up with an assignment (of labels to each vertex) that satisfies "as many" edges as possible (see Definition 7). It is well-known that a gap version of the label cover problem – deciding whether an assignment (of labels) satisfies all the edge constraints or no assignment (of labels) can satisfy more than a small constant fraction of the edges – is NP-hard, even for constant-sized label set and constant left/right-degree bipartite graphs [2, 36]. Given such a label cover instance, we construct a "sparse" incomplete matrix (DMC instance), i.e., a matrix with only a small number of non-$*$ entries per row. For the construction, we utilize the concept of a dictatorship gadget [3] (see Section 4 for the reduction). The completeness of our reduction follows from the properties of the dictatorship gadget. However, for soundness, we need more intricate arguments. The crux of the argument lies in the fact that if the given label cover instance is a No instance (i.e., no assignment can satisfy more than a small constant fraction of the edges), then in our constructed incomplete matrix, for every possible completion we can find "a large" subset of rows where the sum of pairwise distances is large and as a consequence, by averaging a "distant" pair of rows exists.

We remark that a similar proof provides the same inapproximability bound to the restricted version of DMC, namely DRMC, that we consider as an intermediate problem to show our 3-approximation result, establishing that our 2-approximation algorithm for the DRMC is essentially optimal.

Other related works. Various optimization tasks with numerous applications have been investigated in the matrix completion problem [5, 39, 19, 18]. In addition to minimizing the diameter of a cluster, as in the case of the DMC problem, researchers have also studied the problem of radius minimization, known as the Minimum Radius Matrix Completion (RadMC) problem. Alternatively, the RadMC problem is also formulated as the closest string with wildcards problem (or the 1-center in Hamming distance with wildcards). The parameterized complexities of this problem have been explored in [27, 28]. A result of $(2-\varepsilon)$-inapproximability (for any $\varepsilon > 0$) was demonstrated in [12] under the assumption that $\mathsf{P} \neq \mathsf{NP}$, while a 2-approximation algorithm is commonly known. Notably, when there are no missing entries (i.e., without wildcards), the closest string problem admits a polynomial-time approximation scheme (PTAS) [31].

Both the DMC and the RadMC problems are specific variations of clustering problems. In [16, 17], the authors considered a more general version of the clustering problem, where the goal is to partition the rows of an incomplete matrix into clusters while minimizing the diameter or radius of each cluster. Besides radius and diameter, [20] investigated the minimization of rank and the number of distinct rows in the completed matrix. In [21], the authors explore the complexity of completing an incomplete matrix in a way that satisfies specific constraints and can be partitioned into low-rank subspaces. Within the clustering literature, numerous variants of non-combinatorial matrix completion, such as k-center and k-means clustering, have also been extensively studied from the perspective of designing approximation algorithms [22, 23, 30, 35, 15, 8].

2 Preliminaries

Notations. Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. For any $n \times d$ dimensional matrix M, we use M_i to denote the i-th row of M, and $M_i[j]$ (or sometimes for brevity M_{ij}) to denote the (i, j)-th entry of M. Further, for any subset of indices $J = \{j_1, j_2, \ldots, j_k\} \subseteq [d]$, we use $M_i[J]$

to denote the sequence $M_i[j_1]M_i[j_2]\cdots M_i[j_k]$. For two strings $x, y \in \Sigma^d$, we use $\mathcal{H}(x, y)$ to denote the Hamming distance between x and y, which counts the number of coordinates where the symbols of x and y do not match, i.e., $\mathcal{H}(x, y) := |\{i \in [d] \mid x[i] \neq y[i]\}|$.

Matrix Completion. For any incomplete matrix $M \in (\Sigma \cup \{*\})^{n\times d}$, we call $\bar{M} \in \Sigma^{n\times d}$ a valid (feasible) *completion* iff for all $i \in [n]$, $j \in [d]$ with $M_i[j] \neq *$, $\bar{M}_i[j] = M_i[j]$. Sometimes we refer to $\bar{M}$ as a *complete matrix* of M.

Given an incomplete matrix $M \in (\Sigma \cup \{*\})^{n\times d}$, for any feasible completion $\bar{M} \in \Sigma^{n\times d}$ of M, we refer to the quantity $\max_{i\neq j\in[n]} \mathcal{H}(\bar{M}_i, \bar{M}_j)$ as the objective value of $\bar{M}$, denoted by $\texttt{Obj}(\bar{M})$. For any completion $\bar{M}^*$ that minimizes $\texttt{Obj}(\bar{M})$, we denote $\texttt{Obj}(\bar{M}^*)$ by $\mathsf{OPT}_{\text{DMC}}(M)$ (or simply $\mathsf{OPT}(M)$ when the problem DMC is clear from the context). We call a feasible solution $\bar{M}$ a c-approximate solution (for $c \geq 1$) of M iff $\texttt{Obj}(\bar{M}) \leq c \cdot \mathsf{OPT}(M)$.

3 Approximation Algorithm for DMC

In this section, we describe a 3-approximation algorithm for the Minimum Diameter Matrix Completion (DMC) problem over an arbitrary alphabet Σ.

▶ **Theorem 1.** *There is a polynomial-time algorithm that, given an incomplete matrix $M \in (\Sigma \cup \{*\})^{n\times d}$, computes a 3-approximate solution for the Minimum Diameter Matrix Completion (DMC) problem over an arbitrary alphabet Σ.*

In proving the above theorem, we first consider a restricted version of the DMC problem, namely Minimum Diameter Restricted Matrix Completion (DRMC), which, given an incomplete matrix $M \in (\Sigma \cup \{*\})^{n\times d}$, asks to find a (valid) completion $\bar{M}$ of M with a restriction, referred to as *column restriction*, that

- For each column $\ell \in [d]$, for all $i, j \in [n]$, if $M_i[\ell] = M_j[\ell] = *$, then $\bar{M}_i[\ell] = \bar{M}_j[\ell]$,

while minimizing the objective value $\max_{i\neq j\in[n]} \mathcal{H}(\bar{M}_i, \bar{M}_j)$.

It is worth noting that the only difference between DMC and DRMC is that in a complete matrix for DMC, the missing entries of a single column can be completed with different symbols, whereas for DRMC, the missing entries of any particular column must be completed with the same symbol. Thus, it is easy to observe that any feasible solution to the DRMC problem is also a feasible solution to the DMC problem, although the converse may not be true. We provide an LP-based 2-approximation algorithm for the DRMC problem and then argue that it also gives us a 3-approximate solution to the DMC problem. The heart of the argument lies in the relationship between the optimum solution of the DMC problem and that of the DRMC problem.

Relationship between DMC and DRMC. For any incomplete matrix $M \in (\Sigma \cup \{*\})^{n\times d}$, let us use $\mathsf{OPT}_{\text{DMC}}(M)$ and $\mathsf{OPT}_{\text{DRMC}}(M)$ to denote the optimum objective value of the DMC and DRMC problem, respectively. Recall that no matter whether it is the DMC or DRMC problem, the objective value of a (feasible) complete matrix $\bar{M}$ is defined as $\texttt{Obj}(\bar{M}) = \max_{i\neq j\in[n]} \mathcal{H}(\bar{M}_i, \bar{M}_j)$.

▶ **Lemma 3.** *For any $M \in (\Sigma \cup \{*\})^{n\times d}$, $\mathsf{OPT}_{DMC}(M) \leq \mathsf{OPT}_{DRMC}(M) \leq \frac{3}{2} \cdot \mathsf{OPT}_{DMC}(M)$.*

Proof. First, observe that any feasible solution to the DRMC problem is also a feasible solution to the DMC problem. It immediately implies that

$$\mathsf{OPT}_{\text{DMC}}(M) \leq \mathsf{OPT}_{\text{DRMC}}(M).$$

We now focus on proving that $\mathsf{OPT}_{\mathrm{DRMC}}(M) \leq \frac{3}{2} \cdot \mathsf{OPT}_{\mathrm{DMC}}(M)$. For that purpose, let us first consider an (arbitrary) optimal completion $\bar{M}^*$ of M with respect to the DMC problem. Next, we use this solution to come up with a feasible solution (complete matrix) $\tilde{M}$ of M to the DRMC problem. We construct $\tilde{M}$ using $\bar{M}^*$ as follows:

- Consider any arbitrary row, say the first row, of $\bar{M}^*$, i.e., $\bar{M}^*_1$.
- Next, for each $i \in [n]$ and $j \in [d]$, if $M_i[j] = *$, then set $\tilde{M}_i[j] = \bar{M}^*_1[j]$; otherwise set $\tilde{M}_i[j] = M_i[j]$.

It is straightforward to see that $\tilde{M}$ is a feasible completion of M for the DRMC problem. Next, we claim the following.

▷ Claim 4. For all $i, j \in [n]$, $\mathcal{H}(\tilde{M}_i, \tilde{M}_j) \leq \frac{3}{2} \cdot \mathtt{Obj}(\bar{M}^*)$.

This claim is pivotal in proving our lemma. We will prove this claim later, and let us now conclude the proof of the lemma by assuming the above claim.

$$\begin{aligned}\mathtt{Obj}(\tilde{M}) = \max_{i \neq j \in [n]} \mathcal{H}(\tilde{M}_i, \tilde{M}_j) &\leq \frac{3}{2} \cdot \mathtt{Obj}(\bar{M}^*) && \text{(By Claim 4)} \\ &= \frac{3}{2} \cdot \mathsf{OPT}_{\mathrm{DMC}}(M) && \text{(Since } \bar{M}^* \text{ is an optimal solution to DMC).}\end{aligned}$$

Now, since $\tilde{M}$ is a feasible completion of M for the DRMC problem,

$$\mathsf{OPT}_{\mathrm{DRMC}}(M) \leq \mathtt{Obj}(\tilde{M}) \leq \frac{3}{2} \cdot \mathsf{OPT}_{\mathrm{DMC}}(M),$$

which concludes the proof of the lemma. ◀

It now remains to prove Claim 4. Before proceeding with the proof, let us recall that for any subset of indices $J = \{j_1, j_2, \ldots, j_k\} \subseteq [d]$, we use $M_i[J]$ to denote the sequence $M_i[j_1]M_i[j_2] \cdots M_i[j_k]$.

Proof of Claim 4. Consider any $i, j \in [n]$. Let us now consider the indices with $*$-entries in the i-th and the j-th row of the matrix $M \in (\Sigma \cup \{*\})^{n \times d}$. Formally,

$$I := \{\ell \in [d] \mid M_i[\ell] = *\}, \quad J := \{\ell \in [d] \mid M_j[\ell] = *\}, \text{ and } \quad K := [d] \setminus (I \cup J).$$

By the construction of $\tilde{M}$,

$$\tilde{M}_i[I] = \bar{M}^*_1[I], \quad \tilde{M}_j[J] = \bar{M}^*_1[J], \quad \tilde{M}_i[K] = \bar{M}^*_i[K], \quad \tilde{M}_j[K] = \bar{M}^*_j[K]. \tag{1}$$

Let us now focus on the Hamming distances between the subsequences induced by the index sets I, J, and K of the rows $\bar{M}^*_1, \bar{M}^*_i$, and $\bar{M}^*_j$. Let

$$\begin{aligned}
\alpha_1 &:= \mathcal{H}(\bar{M}^*_1[I], \bar{M}^*_i[I]), & \beta_1 &:= \mathcal{H}(\bar{M}^*_1[J], \bar{M}^*_i[J]), & \gamma_1 &:= \mathcal{H}(\bar{M}^*_1[K], \bar{M}^*_i[K]), \\
\alpha_2 &:= \mathcal{H}(\bar{M}^*_1[I], \bar{M}^*_j[I]), & \beta_2 &:= \mathcal{H}(\bar{M}^*_1[J], \bar{M}^*_j[J]), & \gamma_2 &:= \mathcal{H}(\bar{M}^*_1[K], \bar{M}^*_j[K]), \\
\alpha_3 &:= \mathcal{H}(\bar{M}^*_i[I], \bar{M}^*_j[I]), & \beta_3 &:= \mathcal{H}(\bar{M}^*_i[J], \bar{M}^*_j[J]), & \gamma_3 &:= \mathcal{H}(\bar{M}^*_i[K], \bar{M}^*_j[K]).
\end{aligned}$$

See Figure 1 for a pictorial representation of the distances. The Hamming distance between the rows $\tilde{M}_i$ and $\tilde{M}_j$ is

$$\begin{aligned}\mathcal{H}(\tilde{M}_i, \tilde{M}_j) &\leq \mathcal{H}(\tilde{M}_i[I], \tilde{M}_j[I]) + \mathcal{H}(\tilde{M}_i[J], \tilde{M}_j[J]) + \mathcal{H}(\tilde{M}_i[K], \tilde{M}_j[K]) \\ &= \mathcal{H}(\bar{M}^*_1[I], \bar{M}^*_j[I]) + \mathcal{H}(\bar{M}^*_i[J], \bar{M}^*_1[J]) + \mathcal{H}(\bar{M}^*_i[K], \bar{M}^*_j[K]) \quad \text{(By Equation 1)} \\ &= \alpha_2 + \beta_1 + \gamma_3.\end{aligned}$$

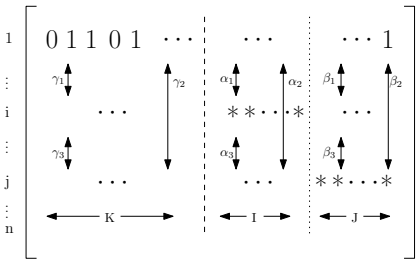

Figure 1 An example matrix M partitioned into coordinate sets I, J, K. For simplicity, in this figure, we assume I and J are disjoint (however, our proof works in full generality).

Thus, to prove our claim, it suffices to argue that

$$\alpha_2 + \beta_1 + \gamma_3 \leq \frac{3}{2} \cdot \texttt{Obj}(\bar{M}^*).$$

Observe that the Hamming distance between the rows $\bar{M}_1^*$ and $\bar{M}_i^*$ is

$$\begin{aligned} &\mathcal{H}(\bar{M}_1^*, \bar{M}_i^*) \leq \alpha_1 + \beta_1 + \gamma_1 \leq \texttt{Obj}(\bar{M}^*) \\ &\implies \alpha_1 + \beta_1 \leq \texttt{Obj}(\bar{M}^*) - \gamma_1. \end{aligned} \tag{2}$$

Similarly, the Hamming distance between the rows $\bar{M}_1^*$ and $\bar{M}_j^*$ is

$$\begin{aligned} &\mathcal{H}(\bar{M}_1^*, \bar{M}_j^*) \leq \alpha_2 + \beta_2 + \gamma_2 \leq \texttt{Obj}(\bar{M}^*) \\ &\implies \alpha_2 + \beta_2 \leq \texttt{Obj}(\bar{M}^*) - \gamma_2. \end{aligned} \tag{3}$$

Also, the Hamming distance between the rows $\bar{M}_i^*$ and $\bar{M}_j^*$ is

$$\begin{aligned} &\mathcal{H}(\bar{M}_i^*, \bar{M}_j^*) \leq \alpha_3 + \beta_3 + \gamma_3 \leq \texttt{Obj}(\bar{M}^*) \\ &\implies \alpha_3 + \beta_3 \leq \texttt{Obj}(\bar{M}^*) - \gamma_3. \end{aligned} \tag{4}$$

Next, the Hamming distance between $\bar{M}_i^*[K]$ and $\bar{M}_j^*[K]$ is

$$\begin{aligned} \gamma_3 &= \mathcal{H}(\bar{M}_i^*[K], \bar{M}_j^*[K]) \\ &\leq \mathcal{H}(\bar{M}_i^*[K], \bar{M}_1^*[K]) + \mathcal{H}(\bar{M}_1^*[K], \bar{M}_j^*[K]) \qquad \text{(By the triangle inequality)} \\ &= \gamma_1 + \gamma_2. \end{aligned} \tag{5}$$

Also, the Hamming distances between $\bar{M}_1^*[I]$ and $\bar{M}_j^*[I]$ is

$$\begin{aligned} \alpha_2 &= \mathcal{H}(\bar{M}_1^*[I], \bar{M}_j^*[I]) \\ &\leq \mathcal{H}(\bar{M}_1^*[I], \bar{M}_i^*[I]) + \mathcal{H}(\bar{M}_i^*[I], \bar{M}_j^*[I]) \qquad \text{(By the triangle inequality)} \\ &= \alpha_1 + \alpha_3 \leq \texttt{Obj}(\bar{M}^*) - \gamma_1 - \beta_1 + \alpha_3 \qquad \text{(By Equation 2)} \end{aligned}$$

which in turn implies that

$$\alpha_2 + \beta_1 \leq \texttt{Obj}(\bar{M}^*) - \gamma_1 + \alpha_3. \tag{6}$$

Similarly, from the Hamming distance between $\bar{M}_i^*[J]$ and $\bar{M}_j^*[J]$, we get the following

$$\beta_1 \leq \beta_2 + \beta_3 \leq \texttt{Obj}(\bar{M}^*) - \gamma_2 - \alpha_2 + \beta_3 \qquad \text{(By Equation 3)}$$

which in turn implies that

$$\alpha_2 + \beta_1 \leq \texttt{Obj}(\bar{M}^*) - \gamma_2 + \beta_3. \tag{7}$$

Adding Equation 6 and Equation 7, we get

$$\begin{aligned} 2(\alpha_2 + \beta_1) &\leq 2 \cdot \texttt{Obj}(\bar{M}^*) - (\gamma_1 + \gamma_2) + (\alpha_3 + \beta_3) \\ &\leq 2 \cdot \texttt{Obj}(\bar{M}^*) - \gamma_3 + (\texttt{Obj}(\bar{M}^*) - \gamma_3) \quad \text{(By Equation 5 and Equation 4)} \\ &\leq 3 \cdot \texttt{Obj}(\bar{M}^*) - 2\gamma_3 \end{aligned}$$

which implies that $\alpha_2 + \beta_1 + \gamma_3 \leq \frac{3}{2} \cdot \texttt{Obj}(\bar{M}^*)$. This completes the proof. ◁

2-approximation for DRMC. In this subsection, we design a 2-approximation algorithm for the DRMC problem, which, when combined with Lemma 3 provides a 3-approximation guarantee for the DMC problem.

▶ **Theorem 5.** *There is a polynomial-time algorithm that, given an incomplete matrix $M \in (\Sigma \cup \{*\})^{n \times d}$, computes a 2-approximate solution for the* Minimum Diameter Restricted Matrix Completion *(DRMC) problem over an arbitrary alphabet Σ.*

We first formulate the problem using an integer linear program (ILP), and then relax the integer constraints to get a linear program (LP), and finally apply a simple (deterministic) rounding scheme on an optimal solution to that LP. We defer the details to Appendix A.

Completing the proof of Theorem 1. Next, we combine Theorem 5 and Lemma 3 to get a 3-approximation algorithm for the DMC problem.

Proof of Theorem 1. Given an incomplete matrix $M \in (\Sigma \cup \{*\})^{n \times d}$, we run the algorithm mentioned in Theorem 5 to get a complete matrix $\bar{M} \in \Sigma^{n \times d}$. Since any feasible solution to the DRMC problem is also a feasible solution to the DMC problem, $\bar{M}$ is a feasible solution to the DMC problem for the input (incomplete) matrix M. Further,

$$\begin{aligned} \texttt{Obj}(\bar{M}) &\leq 2 \cdot \mathsf{OPT}_{\text{DRMC}}(M) && \text{(By Theorem 5)} \\ &\leq 2 \cdot \frac{3}{2} \cdot \mathsf{OPT}_{\text{DMC}}(M) && \text{(By Lemma 3)} \\ &= 3 \cdot \mathsf{OPT}_{\text{DMC}}(M). \end{aligned}$$

Thus $\bar{M}$ is a 3-approximate solution to the DMC problem, which completes the proof. ◀

4 Inapproximability of the DMC problem

In the previous section, we have seen a 3-approximation algorithm for the DMC problem. On the hardness side, so far, we only know that the DMC problem is NP-hard. No inapproximability result is known. In this section, we refute the possibility of getting better than a 2-factor approximation algorithm unless P = NP, even when the alphabet Σ is binary, i.e., $\Sigma = \{0, 1\}$. In particular, we prove Theorem 2.

▶ **Theorem 2.** *Consider any $\varepsilon > 0$. There is no deterministic polynomial-time algorithm that, given an incomplete matrix $M \in \{0, 1, *\}^{n \times d}$, computes a $(2 - \varepsilon)$-approximate solution for the* Minimum Diameter Matrix Completion *(DMC) problem, unless* P = NP.

To show the $(2-\varepsilon)$-inapproximability result, we consider the following gap-version of the DMC problem.

▶ **Definition 6.** *Consider an alphabet Σ and an $\varepsilon > 0$. Given an incomplete matrix $M \in (\Sigma \cup \{*\})^{n \times d}$ and a positive integer g, decide between the following two cases:*

YES: $\mathsf{OPT}(M) \leq g$,
NO: $\mathsf{OPT}(M) > (2-\varepsilon)g$.

Label cover problem and dictatorship gadget. To show the inapproximability result, we provide a reduction from the well-known *label cover* problem to the gap-version of the DMC problem. Let us start by defining the label cover problem.

▶ **Definition 7** (Label Cover Instance). *A* label cover *instance $\Psi = (U, V, E, \Pi)$ consists of*
- *A bipartite graph $G = (U, V, E)$ that is left and right regular. Let D_U and D_V be the degrees of each vertex in U and V respectively,*
- *Label sets $\mathcal{L}_U$ and $\mathcal{L}_V$ for U and V respectively,*
- *For each edge $e \in E$, a function $\pi_e : \mathcal{L}_V \to \mathcal{L}_U$. Let $\Pi = \{\pi_e : \mathcal{L}_V \to \mathcal{L}_U \mid e \in E\}$.*

A labelling *σ is a mapping that assigns each $u \in U$ a label $\sigma(u) \in \mathcal{L}_U$, and each $v \in V$ a label $\sigma(v) \in \mathcal{L}_V$. A labelling σ is said to satisfy an edge $e = (u, v) \in E$ iff $\pi_e(\sigma(v)) = \sigma(u)$. The* value of a labelling *σ, denoted by $\texttt{Val}(\Psi, \sigma)$, is defined as the fraction of edges of E satisfied by σ.*

It is known that a gap version of the label cover problem is NP-hard.

▶ **Theorem 8** ([2, 36]). *For every $\delta \in (0, 1)$, there exists $(1/\delta)^{O(1)}$-sized label sets $\mathcal{L}_U, \mathcal{L}_V$ such that, given a label cover instance $\Psi = (U, V, E, \Pi)$ with label sets $\mathcal{L}_U$ and $\mathcal{L}_V$ and the left degree and the right degree of the instance (bipartite) graph being at most $(1/\delta)^{O(1)}$, it is* NP*-hard to decide between the following two cases:*
- *There exists a labelling σ of Ψ such that $\texttt{Val}(\Psi, \sigma) = 1$,*
- *For every labelling σ of Ψ, $\texttt{Val}(\Psi, \sigma) \leq \delta$.*

One of the standard tools to provide a reduction from the label cover problem is the *dictatorship gadget*. Here, we use a construction of a dictatorship gadget presented in [3]. Before presenting a brief description of the dictatorship construction, let us first introduce a few notions. Let $\neg$ be a negation operator that works both on bits and strings, where the negation of a string is obtained by negating each of its bits individually. A function $f : \{0,1\}^m \to \{0,1\}$ is said to be *odd or folded* if for every x, $f(\neg x) = \neg f(x)$. The oddness of f allows us to store only the value of $f(x)$ for every pair $(x, \neg x)$. If $f(\neg x)$ is needed, we use $\neg f(x)$ instead.

Let us now describe the dictatorship gadget given in [3]. Consider a positive integer k. A *k-dictatorship gadget* is a CNF formula defined over 2^m (for some positive integer m) variables, where an assignment can be viewed as a function $f : \{0,1\}^m \to \{0,1\}$ and assumed to be folded. The set of constraints $\mathcal{C}$ on f is the set of all the clauses of the form $(f(x_1) \vee f(x_2) \vee \cdots \vee f(x_{2k+1}))$, where $x_1, \ldots, x_{2k+1}$ are such that for each $\ell \in [m]$,

$$\sum_{i=1}^{2k+1} x_{i,\ell} \geq k \tag{8}$$

where $x_{i,\ell}$ denotes the ℓ-th bit of the string x_i.

Before proceeding further, let us define a few basic notions. A function $f : \{0,1\}^m \to \{0,1\}$ is said to be a *dictatorship function* if there exists an $\ell \in [m]$ such that for every input $x_i \in \{0,1\}^m$, $f(x_i) = x_{i,\ell}$. For a function $f : \{0,1\}^m \to \{0,1\}$, we call a coordinate $\ell \in [m]$ *relevant* if there exists an input $x_i \in \{0,1\}^m$ such that $f(x_i) \neq f(x_i^{\oplus \ell})$, where $x_i^{\oplus \ell}$ denotes the input obtained by just flipping the ℓ-th bit of x_i. A function f is said to depend on r variables if there are r relevant coordinates. The following result about the dictatorship gadget plays a crucial role in our reduction.

▶ **Lemma 9** ([3]).

1. *If f is a dictatorship function, then it satisfies at least k literals of every clause in the constraint set $\mathcal{C}$.*
2. *Any assignment f that is odd and satisfies all the clauses in the constraint set $\mathcal{C}$ depends on at most $2k-1$ variables.*

It is worth remarking that Item 1 of the above lemma follows immediately from the construction of the dictatorship gadget, especially from Equation 8, whereas Item 2 of the above lemma (which is a weaker converse of Item 1) was shown in [3].

Reduction from the label cover problem. Consider a $\delta \in (0,1)$. Let us consider a label cover instance $\Psi = (U, V, E, \Pi)$, where $\Pi = \{\pi_e : \mathcal{L}_V \to \mathcal{L}_U \mid e \in E\}$. Let us assume that the sizes of both the label set $\mathcal{L}_V$ and $\mathcal{L}_U$ are upper bounded by some $L = (1/\delta)^{\Theta(1)}$. Also, the left degree and the right degree of the instance graph (U, V, E) are upper bounded by some $D = (1/\delta)^{\Theta(1)}$. We associate a function $f_u : \{0,1\}^{|\mathcal{L}_U|} \to \{0,1\}$ (intended to be a dictator of a label of u) to each vertex $u \in U$. Similarly, we associate $f_v : \{0,1\}^{|\mathcal{L}_V|} \to \{0,1\}$ to each $v \in V$. Let us partition the set $\{0,1\}^{|\mathcal{L}_U|}$ into two disjoint equal-sized sets T_U and F_U (arbitrarily) such that for each $x \in T_U$, $\neg x \in F_U$. Similarly, partition the set $\{0,1\}^{|\mathcal{L}_V|}$ into two disjoint equal-sized sets T_V and F_V. (The purpose of this partitioning is that we store the value of the functions only on T_U (and T_V) when the functions are folded.)

Let us consider a positive integer $k = (\min\{L, D, 1/\delta\})^{1/3}$ (which is at most $(1/\delta)^{\Theta(1)}$, and this choice of the value of k is used in the proof of Claim 13). We now construct an incomplete matrix M_Ψ (for brevity, we drop Ψ and simply refer to it as M). For each $u \in U$, consider the k-dictatorship gadget on f_u, and similarly, for each $v \in V$, consider the k-dictatorship gadget on f_v. For each $u \in U$ (resp., $v \in V$), there is a column corresponding to each $x \in T_U$ (resp., $x \in T_V$). (So each column is essentially indexed by either $f_u(x_i)$ for $u \in U$, $x_i \in T_U$, or $f_v(x_i)$ for $v \in V$, $x_i \in T_V$.) Thus, the number of columns is

$$d = |U| \cdot 2^{(|\mathcal{L}_u|-1)} + |V| \cdot 2^{(|\mathcal{L}_v|-1)}.$$

We create rows as follows:

- **Left Vertex Rows:** For each $u \in U$, consider the k-dictatorship gadget on f_u, and let $\mathcal{C}_u$ be the corresponding constraint set. Then add a row for each clause $C \in \mathcal{C}_u$ as follows: For each $x \in T_U$, if the literal represented by $f_u(x)$ is present in C, then set the corresponding entry in the row to be 1; if the literal represented by $f_u(\neg x)$ is present in C, then set the corresponding entry in the row to be 0; otherwise (none of $f_u(x)$ and $f_u(\neg x)$ is present in C), set the corresponding entry in the row to be $*$.
- **Right Vertex Rows:** For each $v \in V$, consider the k-dictatorship gadget on f_v, and let $\mathcal{C}_v$ be the corresponding constraint set. Then add a row for each clause $C \in \mathcal{C}_v$ in a way similar to the above.

- **Edge Rows:** For each edge $e = (u, v) \in E$, add rows as follows: For each possible k inputs $x_1, \ldots, x_k \in \{0,1\}^{|\mathcal{L}_U|}$ on the U side, and $k+1$ inputs $y_1, \ldots, y_{k+1} \in \{0,1\}^{|\mathcal{L}_V|}$ on the V side, we add a row if the following holds:

$$\text{For each label } \ell \in \mathcal{L}_V, \quad \sum_{j=1}^{k} x_{j,\pi_e(\ell)} + \sum_{j=1}^{k+1} y_{j,\ell} \geq k. \tag{9}$$

  In this added row, we set the entries as: If $x_i \in T_U$, then set the entry corresponding to the column $f_u(x_i)$ to be 1; otherwise ($x_i = \neg x'_i$ for some $x'_i \in T_U$), set the entry corresponding to the column $f_u(x'_i)$ to be 0. Similarly, if $y_i \in T_V$, then set the entry corresponding to the column $f_v(y_i)$ to be 1; otherwise ($y_i = \neg y'_i$ for some $y'_i \in T_V$), set the entry corresponding to the column $f_v(y'_i)$ to be 0. All the remaining entries of the row are set to $*$.

It is straightforward to observe that the number of rows n of the constructed matrix M is at most polynomial in the size of the label cover instance (due to our choice of k). Before arguing about the completeness and soundness of the above reduction, let us make a simple observation that immediately follows from the construction of M.

▶ **Observation 10.** *For any label cover instance Ψ, let M be the incomplete matrix constructed as mentioned above. Then each row of M contains exactly $2k+1$ non-$*$ entries.*

Proof. For any left vertex row or right vertex row, by the construction of the k-dictatorship gadget, it contains exactly $2k+1$ non-$*$ entries. For any edge row, by the construction of that row, it contains exactly $2k+1$ non-$*$ entries. ◀

Let us now state the completeness of the reduction, the proof of which is relatively direct from the construction and Lemma 9.

▶ **Lemma 11** (Completeness). *If there exists a labelling σ of Ψ such that $\texttt{Val}(\Psi, \sigma) = 1$, then $\mathsf{OPT}(M) \leq 2k+2$.*

Proof. Let σ be a labeling such that $\texttt{Val}(\Psi, \sigma) = 1$, i.e., for all the edges $e = (u, v) \in E$, $\pi_e(\sigma(v)) = \sigma(u)$. For each $u \in U$, let f_u be the dictatorship function of the label $\sigma(u)$, i.e., for every $x \in \{0,1\}^{|\mathcal{L}_U|}$, $f_u(x)$ is equal to the $\sigma(u)$-th bit of x. Similarly, for each $v \in V$, let f_v be the dictatorship function of the label $\sigma(v)$. Then create a string $s \in \{0,1\}^d$ as follows: For each $u \in U$ and $x \in T_U$ (resp., each $v \in V$ and $x \in T_V$), set the corresponding entry of s to be equal to $f_u(x)$ (resp., $f_v(x)$).

Let us now create a feasible completion $\bar{M}$ by setting each $*$-entry of any column r of M to be $s[r]$ (i.e., the r-th entry of the string s). Next, observe, for each left vertex row or right vertex row M_i, it immediately follows from Item 1 of Lemma 9 that $\mathcal{H}(\bar{M}_i, s) \leq k+1$ (since by Observation 10, there are only $2k+1$ non-$*$ entries in M_i). Also, for each edge row M_i, by the construction (Equation 9), it follows from Observation 10 that $\mathcal{H}(\bar{M}_i, s) \leq k+1$. Hence, for any two $i \neq j$, by the triangle inequality,

$$\mathcal{H}(\bar{M}_i, \bar{M}_j) \leq \mathcal{H}(\bar{M}_i, s) + \mathcal{H}(s, \bar{M}_j) \leq 2k+2.$$ ◀

Next, we consider the more intriguing case of soundness.

▶ **Lemma 12** (Soundness). *For any $\delta \in (0,1)$, there exists an $\varepsilon \in (0,1)$, such that if for every labelling σ of Ψ, $\texttt{Val}(\Psi, \sigma) \leq \delta$, then $\mathsf{OPT}(M) > (2-\varepsilon) \cdot 2k$.*

We devote the rest of this section to proving the soundness.

Proof of soundness. Let us fix a $\delta \in (0,1)$, and set $\varepsilon = (\min\{\delta, 1/k\})^2$ (which is $\delta^{\Theta(1)}$ and this choice of the value of ε is used in the proof of Claim 13). For each row M_i, let us denote the set of coordinates with non-$*$ entries by N_i, i.e.,

$$N_i := \{r \in [d] \mid M_i[r] \neq *\}.$$

We now show the soundness in two steps. First, we argue that if for every labelling σ of Ψ, $\texttt{Val}(\Psi, \sigma) \leq \delta$, then for every plausible completion (represented by a string $s \in \{0,1\}^d$) of any particular row, there exists a "large" subset of rows with every pair of rows having mutually disjoint sets of non-$*$ coordinates. Formally,

▷ Claim 13. If for every labelling σ of Ψ, $\texttt{Val}(\Psi, \sigma) \leq \delta$, then for every row M_p of $M \in (\Sigma \cup \{*\})^{n \times d}$ (where $\Sigma = \{0,1\}$), and for every (feasible) completion $s \in \{0,1\}^d$ of that row, there exists a subset $C_s \subseteq [n]$ of rows of M such that

- $|C_s| \geq 2/\varepsilon + 1$,
- For every $i \in C_s$ and every index $r \in N_i$, $M_i[r] \neq s[r]$,
- For every $i \in C_s$, $N_p \cap N_i = \emptyset$, and
- For every $i \neq j \in C_s$, $N_i \cap N_j = \emptyset$.

Proof. We prove the claim in two parts.

A "large" subset of non-$*$ disjoint rows exists. For a $\delta \in (0,1)$, fix a suitably small constant $\lambda \in (0,1)$ that depends on δ. First, we show that if for every labeling σ of Ψ, $\texttt{Val}(\Psi, \sigma) \leq \delta$, then for every row M_p of M, and for every (feasible) completion $s \in \{0,1\}^d$ of that row, there exists a subset $K_s \subseteq [n]$ of rows of M such that

- $|K_s| \geq (1-\lambda)n$, and
- For every $i \in K_s$ and every index $r \in N_i$, $M_i[r] \neq s[r]$.

The proof of this part resembles the argument used in [3]. We prove the contrapositive of the above statement. For that purpose, let us consider a row M_p, a feasible completion $s \in \{0,1\}^d$ of that row, and a subset $K_s \subseteq [n]$ of size at least $(1-\lambda)n$ such that

$$\text{For every } i \in K_s, \text{ there exists } r \in N_i, M_i[r] = s[r]. \tag{10}$$

Let U' denote the set of all $u \in U$ such that for all the left vertex rows $i \in [n]$ corresponding to u, there exists $r \in N_i$, such that $M_i[r] = s[r]$. Similarly, define $V' \subseteq V$. Also, let E' denote the set of all $e \in E$ such that for all the edge rows $i \in [n]$ corresponding to e, there exists $r \in N_i$, such that $M_i[r] = s[r]$. Recall that the label sets $\mathcal{L}_U$ and $\mathcal{L}_V$ are of size at most $L = (1/\delta)^{O(1)} \leq \mathsf{poly}(k)$ (for our choice of k), and thus the number of left vertex rows (resp., right vertex rows) for each $u \in U$ (resp., $v \in V$) is at most some constant that depends only on k. Also, the left degree and the right degree of the instance graph (U, V, E) are upper bounded by some $D = (1/\delta)^{O(1)} \leq \mathsf{poly}(k)$ (for our choice of k), and thus the number of edge rows is also at most $r(k) \cdot |U|$, where $r(k)$ is some constant that depends only on k. Thus for small enough λ, there exists a constant $\lambda' > 0$ such that

$$|U'| \geq (1-\lambda')|U|, \quad |V'| \geq (1-\lambda')|V|, \text{ and } |E'| \geq (1-\lambda')|E|.$$

Observe, by the construction, for each $u \in U'$ (resp., $v \in V'$), the substring of s corresponding to be positions of f_u (resp., f_v) (viewed as an assignment) satisfies the k-dictatorship gadget for u (resp., v). For simplicity, we refer to these substrings of s as the assignment f_u (resp., f_v). Thus by Item 2 of Lemma 9, f_u (resp., f_v) depends on at most $2k-1$ variables. For each $u \in U'$ (resp., $v \in V'$), let $S_u \subseteq \mathcal{L}_U$ (resp., $S_v \subseteq \mathcal{L}_V$) be the set of variables f_u (resp., f_v) depend on.

Next, we focus on a subset of E', which contains edges with both endpoints in U' and V'. Formally,

$$E'' := \{e = (u, v) \in E' \mid u \in U', v \in V'\}.$$

It is not hard to observe that $|E''| \geq (1 - 3\lambda')|E|$. Furthermore, we claim that for each $e \in E''$, $S_u \cap \pi_e(S_v) \neq \emptyset$. To see this, for the sake of contradiction, assume $S_u \cap \pi_e(S_v) = \emptyset$. Consider k inputs $x_1, \ldots, x_k \in \{0,1\}^{|\mathcal{L}_U|}$ on the U side such that $f_u(x_j) = 0$ and r-th bit of x_j is 1 for all $r \in \mathcal{L}_U \setminus S_u$, and $k+1$ inputs $y_1, \ldots, y_{k+1} \in \{0,1\}^{|\mathcal{L}_V|}$ on the V side such that $f_v(y_j) = 0$ and r-th bit of y_j is 1 for all $r \in \mathcal{L}_V \setminus S_v$. It is easy to verify that this set of inputs satisfies Equation 9. Thus, by the construction, all the bits of the corresponding row in M are different from that of the string s, which is a contradiction.

Now, if we assign labels to each $u \in U'$ and $v \in V'$ by picking labels uniformly at random from S_u and S_v respectively, then each edge $e = (u, v) \in E''$ is satisfied with probability $\frac{1}{|S_u| \cdot |S_v|} \geq \frac{1}{(2k-1)^2}$. This implies that there exists a labeling σ of Ψ such that $\texttt{Val}(\Psi, \sigma) \geq (1 - 3\lambda')/(2k-1)^2 \geq \delta$ (for our choice of k and λ).

A "large" subset of non-$*$ pairwise-disjoint rows exists. Next, let us consider any row M_p of M and any (feasible) completion $s \in \{0,1\}^d$ of that row. We have already argued that there exists a subset $K_s \subseteq [n]$ of rows of M such that

- $|K_s| \geq (1 - \lambda)n$, and
- For every $i \in K_s$ and every index $r \in N_i$, $M_i[r] \neq s[r]$.

We now claim that there exists a subset $C_s \subseteq K_s$ such that

1. $|C_s| \geq 2/\varepsilon + 1$,
2. For every $i \in C_s$ and every index $r \in N_i$, $M_i[r] \neq s[r]$,
3. For every $i \in C_s$, $N_p \cap N_i = \emptyset$, and
4. For every $i \neq j \in C_s$, $N_i \cap N_j = \emptyset$.

We construct a subset $C_s \subseteq K_s$ as follows: Consider the row M_p. If the row M_p is a left/right vertex row for a vertex u, then discard all other left/right vertex rows added for that vertex u, and also all the edge rows corresponding to any of the incident edges of u. If the row M_p is an edge row for an edge $e = (u, v)$, then discard all the other edge rows corresponding to that edge and all the edge rows corresponding to incident edges of u and v, and also all the left/right vertex rows corresponding to u and v. Then, pick a row arbitrarily from the remaining rows from K_s. Again, discard the rows as before and proceed unless we pick $2/\varepsilon + 1$ rows.

Note, $|K_s| \geq (1 - \lambda)n$. Further, recall the number of left vertex rows (resp., right vertex rows) for each $u \in U$ (resp., $v \in V$) is at most some constant that depends only on k, and also the number of edge rows for each edge is at most some constant that depends only on k. Thus, in the above construction of C_s, at each step, we discard at most some constant (that depends only on k) many rows. Hence, the above construction process does not terminate before picking $2/\varepsilon + 1$ rows.

Item 1, 3 and 4 are immediate from the construction. Since $C_s \subseteq K_s$, Item 2 also follows. This concludes the proof of the claim. ◁

Next, we argue that if for every string $s \in \{0,1\}^d$, such a subset C_s exists, then for every feasible completion $\bar{M}$ of M, $\texttt{Obj}(\bar{M}) \geq (2 - \varepsilon) \cdot 2k$.

▷ Claim 14. Let n and d denote the number of rows and columns of M, respectively. If for every row M_p, and any (feasible) completion $s \in \{0,1\}^d$ of that row, there exists a subset $C_s \subseteq [n]$ of rows of M such that

- $|C_s| \geq 2/\varepsilon + 1$,
- For every $i \in C_s$ and every index $r \in N_i$, $M_i[r] \neq s[r]$,
- For every $i \in C_s$, $N_p \cap N_i = \emptyset$, and
- For every $i \neq j \in C_s$, $N_i \cap N_j = \emptyset$.

then $\mathsf{OPT}(M) \geq (2-\varepsilon) \cdot 2k$.

Proof. Let us consider a feasible completion $\bar{M}$ of M, and then consider any arbitrary row, say the first row, of it. Let $s = \bar{M}_1$. Then, by the premise of the claim, there exists a subset $C_s \subseteq [n]$ such that

1. $|C_s| = c \geq 2/\varepsilon + 1$ (the value to be fixed),
2. For every $i \in C_s$ and every index $r \in N_i$, $M_i[r] \neq s[r]$,
3. For every $i \in C_s$, $N_1 \cap N_i = \emptyset$, and
4. For every $i \neq j \in C_s$, $N_i \cap N_j = \emptyset$.

Recall, for any subset of indices $J = \{j_1, j_2, \ldots, j_k\} \subseteq [d]$, we use $\bar{M}_i[J]$ to denote the sequence $\bar{M}_i[j_1]\bar{M}_i[j_2]\cdots\bar{M}_i[j_k]$. For any $i \neq j \in C_s$, let $\alpha_{ij} = \mathcal{H}(\bar{M}_i[N_i], \bar{M}_j[N_i])$ and $\alpha_{ji} = \mathcal{H}(\bar{M}_i[N_j], \bar{M}_j[N_j])$. Thus for any $i \neq j \in C_s$, we have that

$$\mathcal{H}(\bar{M}_i, \bar{M}_j) \geq \mathcal{H}(\bar{M}_i[N_i], \bar{M}_j[N_i]) + \mathcal{H}(\bar{M}_i[N_j], \bar{M}_j[N_j]) = \alpha_{ij} + \alpha_{ji}. \tag{11}$$

Further, consider any $i \neq j \in C_s$. Since $\bar{M}$ is a feasible completion of M, by Item 2, for every index $r \in N_j$, $\bar{M}_j[r] \neq \bar{M}_1[r]$. Now, since $\bar{M} \in \{0,1\}^{n \times d}$, for any index $r \in N_j$, $\bar{M}_i[r] = \bar{M}_1[r]$ if and only if $\bar{M}_i[r] \neq \bar{M}_j[r]$. Thus

$$\mathcal{H}(\bar{M}_1[N_j], \bar{M}_i[N_j]) = |N_j| - \alpha_{ij}. \tag{12}$$

Hence, for any $i \in C_s$, we get that

$$\begin{aligned}
\mathcal{H}(\bar{M}_i, \bar{M}_1) &\geq \sum_{j \in C_s} \mathcal{H}(\bar{M}_i[N_j], \bar{M}_1[N_j]) \\
&= |N_i| + \sum_{j \in C_s : j \neq i} (|N_j| - \alpha_{ij}) && \text{(By Item 2 and Equation 12)} \\
&= (2k+1)c - \sum_{j \in C_s : j \neq i} \alpha_{ij} && \text{(By Observation 10).}
\end{aligned} \tag{13}$$

Now, if for some $i \in C_s$, $\mathcal{H}(\bar{M}_i, \bar{M}_1) \geq 4k$, then clearly $\mathtt{Obj}(\bar{M}) \geq 4k$ and we are done with the proof. So let us assume that for all $i \in C_s$, $\mathcal{H}(\bar{M}_i, \bar{M}_1) \leq 4k$. Then by Equation 13, for every $i \in C_s$,

$$\sum_{j \in C_s : j \neq i} \alpha_{ij} \geq (2k+1)c - 4k = 2k(c-2) + c. \tag{14}$$

Then it follows from Equation 11,

$$\begin{aligned}
\sum_{i \in C_s} \sum_{j \in C_s : j \neq i} \mathcal{H}(\bar{M}_i, \bar{M}_j) &\geq \sum_{i \in C_s} \sum_{j \in C_s : j \neq i} (\alpha_{ij} + \alpha_{ji}) \\
&= 2 \sum_{i \in C_s} \sum_{j \in C_s : j \neq i} \alpha_{ij} \\
&\geq 2c\,(2k(c-2) + c) && \text{(By Equation 14).}
\end{aligned}$$

Then, by a simple averaging, there must exist $i \neq j \in C_s$ such that

$$\mathcal{H}(\bar{M}_i, \bar{M}_j) \geq \frac{2c\,(2k(c-2)+c)}{c(c-1)} > (2-\varepsilon) \cdot 2k$$

where the last inequality follows since $c \geq 2/\varepsilon + 1$. So we have argued that for any feasible completion $\bar{M}$ of M, $\mathtt{Obj}(\bar{M}) > (2-\varepsilon) \cdot 2k$, and hence $\mathsf{OPT}(M) > (2-\varepsilon) \cdot 2k$. ◁

Finally, by combining Claim 13 and Claim 14, we get our desired soundness Lemma 12.

▶ Remark 15. We want to remark that our reduction also establishes $(2-\varepsilon)$-inapproximability for the restricted variant of the DMC problem, namely the Minimum Diameter Restricted Matrix Completion (DRMC) problem, for which we provide a 2-approximation algorithm in Theorem 5. To understand why this is the case, first, observe that we indeed get a solution to the DRMC problem in our completeness proof. For soundness, using a similar (though much simpler) argument that is used in the proof of Lemma 12, we can show that if $\texttt{Val}(\Psi, \sigma) \leq \delta$, then for every string $s \in \{0,1\}^d$, we get at least two rows whose non-$*$ entries do not match with the corresponding entries of s and the set of non-$*$ coordinates are disjoint. Consequently, by an argument similar to that in Claim 14, their distance must be at least $(2-\varepsilon) \cdot 2k$.

5 Conclusion

In this paper, we focus on the task of completing an incomplete matrix while minimizing the diameter, which represents the maximum pairwise distance between any two rows. Currently, the only known approach is a 4-factor approximation algorithm derived from a straightforward utilization of the triangle inequality combined with a simple 2-approximation algorithm for the radius minimization variant. Although the problem is known to be NP-hard, no inapproximability result has been established until now. Our main contribution is the development of a novel 3-approximation algorithm. Notably, this result surpasses the existing 4-factor approximation, marking the first improvement in approximating this problem.

Additionally, we demonstrate that the problem is $(2-\varepsilon)$-inapproximable for any $\varepsilon > 0$, even when considering a binary alphabet. This represents the first inapproximability result for this problem. One of the intriguing open problems is to bridge the gap between the 3-factor approximation and the $(2-\varepsilon)$-inapproximability. Furthermore, it would be interesting to extend our approximation approach to a more general variant of k-clustering, with a focus on minimizing the diameter of each cluster.

References

1 Paul D Allison. *Missing data*. Sage publications, 2001.

2 S Arora, C Lund, R Motwani, M Sudan, and M Szegedy. Proof verification and intractability of approximation problems. In *Proceedings of the 33rd Annual IEEE Symposium on the Foundations of Computer Science, IEEE*, 1992.

3 Per Austrin, Venkatesan Guruswami, and Johan Håstad. $(2+\varepsilon)$-SAT is NP-hard. *SIAM Journal on Computing*, 46(5):1554–1573, 2017.

4 Vineet Bafna, Sorin Istrail, Giuseppe Lancia, and Romeo Rizzi. Polynomial and apx-hard cases of the individual haplotyping problem. *Theoretical Computer Science*, 335(1):109–125, 2005.

5 Laura Balzano, Arthur Szlam, Benjamin Recht, and Robert Nowak. K-subspaces with missing data. In *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pages 612–615. IEEE, 2012.

6 Manu Basavaraju, Fahad Panolan, Ashutosh Rai, MS Ramanujan, and Saket Saurabh. On the kernelization complexity of string problems. *Theoretical Computer Science*, 730:21–31, 2018.

7 Christina Boucher, Christine Lo, and Daniel Lokshantov. Consensus patterns (probably) has no eptas. In *Algorithms-ESA 2015: 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 239–250. Springer, 2015.

8 Vladimir Braverman, Shaofeng Jiang, Robert Krauthgamer, and Xuan Wu. Coresets for clustering with missing values. *Advances in Neural Information Processing Systems*, 34:17360–17372, 2021.

9 Laurent Bulteau, Vincent Froese, and Rolf Niedermeier. Tight hardness results for consensus problems on circular strings and time series. *SIAM Journal on Discrete Mathematics*, 34(3):1854–1883, 2020.

10 Laurent Bulteau, Falk Hüffner, Christian Komusiewicz, Rolf Niedermeier, et al. Multivariate algorithmics for NP-hard string problems. *Bulletin of EATCS*, 3(114), 2014.

11 Laurent Bulteau and Markus L Schmid. Consensus strings with small maximum distance and small distance sum. *Algorithmica*, 82(5):1378–1409, 2020.

12 Diptarka Chakraborty, Kshitij Gajjar, and Agastya Vibhuti Jha. Approximating the Center Ranking Under Ulam. In *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021)*, volume 213, pages 12:1–12:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.

13 Moses Charikar and Rina Panigrahy. Clustering to minimize the sum of cluster diameters. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 1–10, 2001.

14 Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. Lower bounds for approximation schemes for closest string. *arXiv preprint arXiv:1509.05809*, 2015.

15 Eduard Eiben, Fedor V Fomin, Petr A Golovach, William Lochet, Fahad Panolan, and Kirill Simonov. Eptas for k-means clustering of affine subspaces. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2649–2659. SIAM, 2021.

16 Eduard Eiben, Robert Ganian, Iyad Kanj, Sebastian Ordyniak, and Stefan Szeider. The parameterized complexity of clustering incomplete data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7296–7304, 2021.

17 Eduard Eiben, Robert Ganian, Iyad Kanj, Sebastian Ordyniak, and Stefan Szeider. Finding a cluster in incomplete data. In *30th Annual European Symposium on Algorithms (ESA 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

18 Ehsan Elhamifar. High-rank matrix completion and clustering under self-expressive models. *Advances in Neural Information Processing Systems*, 29, 2016.

19 Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013.

20 Robert Ganian, Iyad Kanj, Sebastian Ordyniak, and Stefan Szeider. Parameterized algorithms for the matrix completion problem. In *International Conference on Machine Learning*, pages 1656–1665. PMLR, 2018.

21 Robert Ganian, Iyad Kanj, Sebastian Ordyniak, and Stefan Szeider. On the parameterized complexity of clustering incomplete data into subspaces of small rank. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3906–3913, 2020.

22 Jie Gao, Michael Langberg, and Leonard J Schulman. Analysis of incomplete data and an intrinsic-dimension helly theorem. *Discrete & Computational Geometry*, 40:537–560, 2008.

23 Jie Gao, Michael Langberg, and Leonard J Schulman. Clustering lines in high-dimensional space: Classification of incomplete data. *ACM Transactions on Algorithms (TALG)*, 7(1):1–26, 2010.

24 Leszek Gasieniec, Jesper Jansson, and Andrzej Lingas. Approximation algorithms for hamming clustering problems. *Journal of Discrete Algorithms*, 2(2):289–301, 2004.

25 Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical computer science*, 38:293–306, 1985.

26 Jens Gramm, Rolf Niedermeier, Peter Rossmanith, et al. Fixed-parameter algorithms for closest string and related problems. *Algorithmica*, 37(1):25–42, 2003.

27 Danny Hermelin and Liat Rozenberg. Parameterized complexity analysis for the closest string with wildcards problem. *Theoretical Computer Science*, 600:11–18, 2015.

28 Tomohiro Koana, Vincent Froese, and Rolf Niedermeier. Parameterized algorithms for matrix completion with radius constraints. *arXiv preprint arXiv:2002.00645*, 2020.

29 Tomohiro Koana, Vincent Froese, and Rolf Niedermeier. Binary matrix completion under diameter constraints. In *38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

30 Euiwoong Lee and Leonard J Schulman. Clustering affine subspaces: hardness and algorithms. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 810–827. SIAM, 2013.

31 Ming Li, Bin Ma, and Lusheng Wang. On the closest string and substring problems. *Journal of the ACM (JACM)*, 49(2):157–171, 2002.

32 Ross Lippert, Russell Schwartz, Giuseppe Lancia, and Sorin Istrail. Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem. *Briefings in bioinformatics*, 3(1):23–31, 2002.

33 Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.

34 Christine Lo, Boyko Kakaradov, Daniel Lokshtanov, and Christina Boucher. Seesite: characterizing relationships between splice junctions and splicing enhancers. *IEEE/ACM transactions on computational biology and bioinformatics*, 11(4):648–656, 2014.

35 Yair Marom and Dan Feldman. k-means clustering of lines for big data. *Advances in Neural Information Processing Systems*, 32, 2019.

36 Ran Raz. A parallel repetition theorem. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 447–456, 1995.

37 Markus L Schmid. Finding consensus strings with small length difference between input and solution strings. *ACM Transactions on Computation Theory (TOCT)*, 9(3):1–18, 2017.

38 Lusheng Wang, Ming Li, and Bin Ma. *Closest String and Substring Problems*, pages 321–324. Springer New York, 2016.

39 Jinfeng Yi, Tianbao Yang, Rong Jin, Anil K Jain, and Mehrdad Mahdavi. Robust ensemble clustering by matrix completion. In *2012 IEEE 12th international conference on data mining*, pages 1176–1181. IEEE, 2012.

A Approximation Algorithm for the DRMC Problem

Let us start by restating the theorem.

▶ **Theorem 5.** *There is a polynomial-time algorithm that, given an incomplete matrix* $M \in (\Sigma \cup \{*\})^{n \times d}$*, computes a* 2*-approximate solution for the* Minimum Diameter Restricted Matrix Completion *(DRMC) problem over an arbitrary alphabet* Σ*.*

We now prove the above theorem. We first formulate the problem using an integer linear program (ILP), and then relax the integer constraints to get a linear program (LP), and finally apply a simple (deterministic) rounding scheme on an optimal solution to that LP. It is worth mentioning that the argument used here is very similar to the folklore 2-approximation algorithm for the RadMC problem.

LP relaxation and rounding. Let us first give an ILP formulation. For each column $\ell \in [d]$ and symbol $\sigma \in \Sigma$, we consider a $\{0,1\}$-variable $x_{\ell,\sigma}$. The variable $x_{\ell,\sigma}$ denotes whether all the $*$ entries of M in the ℓ-th column are set to the symbol σ. More specifically, if $x_{\ell,\sigma} = 1$, then all the $*$ entries of M in the ℓ-th column are set to σ.

Let us define δ_{ij} to be the Hamming distance between the non-$*$-entries of the i-th and j-th row of M. More formally, let

$$K_{ij} := \{\ell \in [d] \mid M_i[\ell] \neq * \text{ and } M_j[\ell] \neq *\}\,.$$

Then $\delta_{ij} := \mathcal{H}(M_i[K_{ij}], M_j[K_{ij}])$. For each row $i \in [n]$, let I_i denote the set of indices with non-$*$ entries in M. Formally,

$$I_i := \{\ell \in [d] \mid M_i[\ell] \neq *\}.$$

We use these notations to describe our ILP formulation.

Minimize z

$$s.t. \quad \sum_{\ell \in I_i \setminus K_{ij}} \left(\sum_{\sigma \neq M_i[\ell]} x_{\ell,\sigma} \right) + \sum_{\ell \in I_j \setminus K_{ij}} \left(\sum_{\sigma \neq M_j[\ell]} x_{\ell,\sigma} \right) + \delta_{ij} \leq z \quad \forall i, j \in [n] \tag{15}$$

$$\sum_{\sigma \in \Sigma} x_{\ell,\sigma} = 1 \quad \forall \ell \in [d] \tag{16}$$

$$x_{\ell,\sigma} \in \{0, 1\} \quad \forall \ell \in [d], \forall \sigma \in \Sigma \tag{17}$$

In the above ILP, the constraints 16 ensure that for each column, for all the $*$ entries, exactly one symbol is selected. It is easy to observe that the constraints 15 ask the Hamming distance between i-th and j-th row (for every pair of $i, j \in [n]$) of the output complete matrix to be at most z, which we minimize in the ILP. Hence, the above ILP provides an optimal solution to the DRMC problem on input M.

In order to convert it to LP, we relax the constraints 17 to

$$x_{\ell,\sigma} \in [0, 1], \quad \forall \ell \in [d], \forall \sigma \in \Sigma.$$

Let us consider an optimal solution $\left(x^*_{\ell,\sigma}\right)_{\ell \in [d], \sigma \in \Sigma}$ to the above LP. Next, we use the following simple (deterministic) rounding: For each $\ell \in [d]$, if there exists a symbol $\sigma \in \Sigma$ such that $x^*_{\ell,\sigma} \geq 1/2$ (break ties arbitrarily), then set $\bar{x}_{\ell,\sigma} = 1$, and set $\bar{x}_{\ell,\sigma'} = 0$ for all $\sigma' \neq \sigma$. For an $\ell \in [d]$, if for all $\sigma \in \Sigma$, $x^*_{\ell,\sigma} < 1/2$, then pick a symbol $\sigma \in \Sigma$ arbitrarily and set $\bar{x}_{\ell,\sigma} = 1$, and set $\bar{x}_{\ell,\sigma'} = 0$ for all $\sigma' \neq \sigma$.

It is straightforward to see that by the above rounding, the following holds:

$$\text{For each } \ell \in [d], \quad \sum_{\sigma \in \Sigma} \bar{x}_{\ell,\sigma} = 1.$$

Thus, it provides us with a feasible completion of M for the DRMC problem.

Approximation guarantee. Now we argue that the solution $(\bar{x}_{\ell,\sigma})_{\ell \in [d], \sigma \in \Sigma}$ obtained by the rounding provides a 2-approximate solution to the DRMC problem for the incomplete input matrix M.

Let z^* be the value of z of any optimal solution to our LP formulation. Let $\bar{z}$ be the minimum integer such that

$$\sum_{\ell \in I_i \setminus K_{ij}} \left(\sum_{\sigma \neq M_i[\ell]} \bar{x}_{\ell,\sigma} \right) + \sum_{\ell \in I_j \setminus K_{ij}} \left(\sum_{\sigma \neq M_j[\ell]} \bar{x}_{\ell,\sigma} \right) + \delta_{ij} \leq \bar{z} \qquad \forall i, j \in [n].$$

We want to claim that $\bar{z} \leq 2z^*$, and as a consequence, we get that the solution $\bar{x}_{\ell,\sigma}$ (for all $\ell \in [d], \sigma \in \Sigma$) obtained by the rounding, provides a 2-approximate solution to the DRMC problem.

To show that $\bar{z} \leq 2z^*$, we analyze each term separately in the constraints 15. By our rounding procedure, for any $j \in [n]$, $\ell \in [d]$, $\sum_{\sigma \neq M_j[\ell]} \bar{x}_{\ell,\sigma} = 1$ if and only if $x^*_{\ell,M_j[\ell]} \leq 1/2$ that means $\sum_{\sigma \neq M_j[\ell]} x^*_{\ell,\sigma} \geq 1/2$. Hence, $\sum_{\sigma \neq M_j[\ell]} \bar{x}_{\ell,\sigma} \leq 2 \cdot \sum_{\sigma \neq M_j[\ell]} x^*_{\ell,\sigma}$. Hence,

$$\sum_{\ell \in I_i \setminus K_{ij}} \left(\sum_{\sigma \neq M_i[\ell]} \bar{x}_{\ell,\sigma} \right) + \sum_{\ell \in I_j \setminus K_{ij}} \left(\sum_{\sigma \neq M_j[\ell]} \bar{x}_{\ell,\sigma} \right) + \delta_{ij} \leq 2z \qquad \forall i, j \in [n]$$

which implies $\bar{z} \leq 2z^*$. This concludes the proof of Theorem 5.