

What You Will Gain By Rounding: Theory and Algorithms for Rounding Rank

Stefan Neumann*
University of Vienna
Vienna, Austria
stefan.neumann@univie.ac.at

Rainer Gemulla
Data and Web Science Group
University of Mannheim, Germany
rgemulla@uni-mannheim.de

Pauli Miettinen
Max Planck Institute for Informatics
Saarland Informatics Campus, Germany
pmiettinen@mpi-inf.mpg.de

Abstract—When factorizing binary matrices, we often have to make a choice between using expensive combinatorial methods that retain the discrete nature of the data and using continuous methods that can be more efficient but destroy the discrete structure. Alternatively, we can first compute a continuous factorization and subsequently apply a rounding procedure to obtain a discrete representation. But what will we gain by rounding? Will this yield lower reconstruction errors? Is it easy to find a low-rank matrix that rounds to a given binary matrix? Does it matter which threshold we use for rounding? Does it matter if we allow for only non-negative factorizations? In this paper, we approach these and further questions by presenting and studying the concept of rounding rank. We show that rounding rank is related to linear classification, dimensionality reduction, and nested matrices. We also report on an extensive experimental study that compares different algorithms for finding good factorizations under the rounding rank model.

I. INTRODUCTION

When facing data that can be expressed as a binary matrix, the data analyst usually has two options: either she uses combinatorial methods—such as frequent itemset mining or various graph algorithms—that will retain the binary structure of the data, or she applies some sort of continuous-valued matrix factorization—such as SVD or NMF—that will represent the binary structure with continuous approximations. The different approaches come with different advantages and drawbacks. Retaining the combinatorial structure is helpful for interpreting the results and can preserve better other characteristics such as sparsity. Continuous methods, on the other hand, are often more efficient, yield better reconstruction errors, and may be interpreted probabilistically.

A third alternative, often applied to get “the best of both worlds,” is to perform a continuous factorization first and apply some function to the elements of the reconstructed matrix to make them binary afterwards. In probabilistic modelling, for example, the logistic function is commonly used to map real values into the unit range. We can obtain a binary reconstruction by rounding, i.e. by setting all values less than $1/2$ to 0 and the remaining values to 1. Alternatively, for $\{-1, 1\}$ matrices, we may take the sign of the values of a continuous factorization to obtain a discrete representation. Even though such methods are commonly used, relatively little is known

about the consequences of this thresholding process. There are few, if any, methods that aim at finding a matrix that rounds exactly to the given binary data, or finding a low-rank matrix that causes only little error when rounded (although there are methods that have such a behavior as a by-product). Almost nothing is known about the theoretical properties of such decompositions.

In this paper, we give a comprehensive treatise of these topics. We introduce the concept of *rounding rank*, which, informally, is defined to be the least rank of a real matrix that rounds to the given binary matrix. But does it matter how we do the rounding? How will the results change if we constrain ourselves to nonnegative factorizations? A solid theoretical understanding of the properties of rounding rank will help data miners and method developers to understand what happens when they apply rounding. Some of our results are novel, while others are based on results obtained from related topics such as *sign rank* and *dot product graphs*.

Studying rounding rank is not only of theoretical interest. The concept can provide new insight or points of view for existing problems, and lead to interesting new approaches. In essence, rounding rank provides another *intrinsic dimensionality* of the data (see, e.g. [33]). Rounding rank can be used, for example, to determine the minimum number of features linear classifiers need for multi-label classification or the minimum number of dimensions we need from a dimensionality reduction algorithm. There is also a close relationship to *nested matrices* [23], a particular type of binary matrices that occur, for example, in ecology. We show that nested matrices are equivalent to matrices with a non-negative rounding rank of 1 and use this property to develop a new algorithm for the problem of finding the closest nested matrix.

But just knowing about the properties of rounding rank will not help if we cannot find good decompositions. As data miners have encountered problems related to rounding rank earlier, there are already existing algorithms for closely related problems. In fact, any low-rank matrix factorization algorithm could be used for estimating (or, more precisely, bounding) the rounding rank, but not all of them would work equally well. To that end, we survey a number of algorithms for estimating the rounding rank and for finding the least-error fixed rounding rank decomposition. We also present some novel methods. One major contribution of this paper is an empirical evaluation of

*Part of the work was done when the author was with MPI Informatics, Saarland University and the Saarbrücken Graduate School of Computer Science.

these algorithms. Our experiments aim to help the practitioners in choosing the correct algorithm for the correct task: for example, if one wants to estimate the rounding rank of a binary matrix, simply rounding the truncated singular value decomposition may not be a good idea.

II. DEFINITIONS, BACKGROUND, AND THEORY

In this section we formally define the rounding rank of a binary matrix, discuss its properties, and compare it to other well-known matrix-ranks. Throughout this paper, we use \mathbf{B} to denote a binary $m \times n$ matrix.

A. Definitions

The *rounding function* w.r.t. *rounding threshold* $\tau \in \mathbb{R}$ is

$$\text{round}_\tau(x) = \begin{cases} 1, & \text{if } x \geq \tau, \\ 0, & \text{if } x < \tau. \end{cases}$$

We apply round_τ to matrices by rounding element-wise, i.e. if $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a real-valued matrix, then $\text{round}_\tau(\mathbf{A})$ denotes an $m \times n$ binary matrix with $[\text{round}_\tau(\mathbf{A})]_{ij} = \text{round}_\tau(\mathbf{A}_{ij})$.

Rounding rank. Given a rounding threshold $\tau \in \mathbb{R}$, the *rounding rank* of \mathbf{B} w.r.t. τ is given by

$$\text{rrank}_\tau(\mathbf{B}) = \min\{\text{rank}(\mathbf{A}) : \mathbf{A} \in \mathbb{R}^{m \times n}, \text{round}_\tau(\mathbf{A}) = \mathbf{B}\}. \quad (1)$$

The rounding rank of \mathbf{B} is thus the smallest rank of any real-valued matrix that rounds to \mathbf{B} . We often omit τ for brevity and write $\text{round}(\mathbf{A})$ and $\text{rrank}(\mathbf{B})$ for $\tau = 1/2$.

When \mathbf{B} has rounding rank k , there exists matrices $\mathbf{L} \in \mathbb{R}^{m \times k}$ and $\mathbf{R} \in \mathbb{R}^{n \times k}$ with $\mathbf{B} = \text{round}_\tau(\mathbf{L}\mathbf{R}^T)$. We refer to \mathbf{L} and \mathbf{R} as a *rounding rank decomposition* of \mathbf{B} .

Sign rank. The *sign matrix* of \mathbf{B} , $\mathbf{B}^\pm \in \{-1, +1\}^{m \times n}$, is obtained from \mathbf{B} by replacing every 0 by -1 . Given a sign matrix, its *sign rank* is given by

$$\text{srnk}(\mathbf{B}^\pm) = \min\{\text{rank}(\mathbf{A}) : \mathbf{A} \in \mathbb{R}_{\neq 0}^{m \times n}, \text{sign}(\mathbf{A}) = \mathbf{B}^\pm\}, \quad (2)$$

where $\mathbb{R}_{\neq 0} = \mathbb{R} \setminus \{0\}$. The sign rank is thus the smallest rank of any real-valued matrix \mathbf{A} without 0-entries and with $\mathbf{B}_{ij}^\pm = \text{sign}(\mathbf{A}_{ij})$ for all i, j . The sign rank is closely related to the rounding rank as $\text{rrank}_0(\mathbf{B}) \leq \text{srnk}(\mathbf{B}^\pm) \leq \text{rrank}_0(\mathbf{B}) + 1$. The first inequality holds because for any $\mathbf{A} \in \mathbb{R}_{\neq 0}^{m \times n}$ and with $\text{sign}(\mathbf{A}) = \mathbf{B}^\pm$, $\text{round}_0(\mathbf{A})^\pm = \text{sign}(\mathbf{A})$. The second inequality holds because if $\text{round}(\mathbf{A}) = \mathbf{B}$ and \mathbf{A} contains 0-entries, we can add a constant $0 < \varepsilon < \min_{a_{ij} < 0} |a_{ij}|$ to each entry of \mathbf{A} to obtain $\text{sign}(\mathbf{A} + \varepsilon) = \mathbf{B}^\pm$ and $\text{rank}(\mathbf{A} + \varepsilon) \leq \text{rank}(\mathbf{A}) + 1$. Even when $\tau \neq 0$, the differences remain small as suggested by Prop. 5.

Non-negative rounding rank. We define the *non-negative rounding rank* of \mathbf{B} w.r.t. τ , denoted $\text{rrank}_\tau^+(\mathbf{B})$, as the smallest k such that there exist non-negative matrices $\mathbf{L} \in \mathbb{R}_{\geq 0}^{m \times k}$ and $\mathbf{R} \in \mathbb{R}_{\geq 0}^{n \times k}$ with $\text{round}_\tau(\mathbf{L}\mathbf{R}^T) = \mathbf{B}$.

Minimum-error rounding rank problem. The rounding rank is concerned with exact reconstructions of \mathbf{B} . We relax this by introducing the *minimum-error rounding rank- k problem*: Find a binary matrix $\mathbf{C} \in \{0, 1\}^{m \times n}$ with $\text{rrank}(\mathbf{C}) \leq k$

which minimizes $\|\mathbf{B} - \mathbf{C}\|_F$, where $\|\cdot\|_F$ denotes the Frobenius norm. Note that $\|\mathbf{B} - \mathbf{C}\|_F^2$ corresponds to the number of entries in which \mathbf{B} and \mathbf{C} disagree. We denote the problem by *MINERRORRR- k* .

B. Related Work

A number of concepts closely related to rounding rank (albeit less general) have been studied in various communities.

There is a relationship between rounding rank and dot-product graphs [15], [24], [31], which arise in social network analysis [34]. Let G be a graph with n vertices and adjacency matrix \mathbf{M} . Then G is a *dot-product graph of rank k* if there exists a matrix $\mathbf{L} \in \mathbb{R}^{m \times k}$ such that $\mathbf{M} = \text{round}_1(\mathbf{L}\mathbf{L}^T)$. The rank of a dot-product graph corresponds to the *symmetric* rounding rank of its adjacency matrix. In this paper, we consider asymmetric factorizations and allow for rectangular matrices.

Sign rank was studied in the communication complexity community in order to characterize a certain communication model. Consider two players, Alice and Bob. Alice and Bob obtain private inputs $x, y \in \{0, 1\}^n$, respectively, and their task is to evaluate a function $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ on their inputs. The *communication matrix* \mathbf{M}_f of f is the $2^n \times 2^n$ binary matrix with $[\mathbf{M}_f]_{xy} = f(\text{bin}(x), \text{bin}(y))$, where $\text{bin} : 2^n \rightarrow \{0, 1\}^n$ denotes the n -bit binary encoding of its input number. The *probabilistic communication complexity* of f is the smallest number of bits Alice and Bob have to communicate in order to compute $f(x, y)$ correctly with probability larger than $\frac{1}{2}$. It is known that the probabilistic communication complexity of f and $\log(\text{srnk}(\mathbf{M}_f))$ differ by at most one [2], [16], [30]. Sign rank was also studied in learning theory to understand the limits of large margin classification [3], [4], [8], [17]; see Alon et al. [4] for a summary of applications of sign rank. These complexity results focus on achieving lower and upper bounds on sign rank as well as the separation of complexity classes. We review some of these results in subsequent sections and present them in terms of rounding rank, thereby making them accessible to the data mining community.

Ben-David et al. [8, Cor. 14] showed that only a very small fraction of the $n \times n$ sign matrices can be well-approximated (with “vanishing” error in at most $n^{-O(1)}$ entries) by matrices of sign-rank at most k unless $k = \omega(n^{1-O(1)})$ is very large. To the best of our knowledge, there are no known results for fixed relative error (e.g., 5% of the matrix entries) or for the *MINERRORRR- k* problem.

C. Characterization of Rounding Rank

Below we give a geometric interpretation of rounding rank that helps to relate it to problems in data mining. A similar theorem was presented in the context of communication complexity [30, Th. 5]. Our presentation is in terms of matrix ranks (instead of communication protocols) and gives a short proof that provides insights into the relationship between rounding rank and geometric embeddings.

Theorem 1. *Let $d \in \mathbb{N}$ and $\tau \in \mathbb{R}$. The following statements are equivalent:*

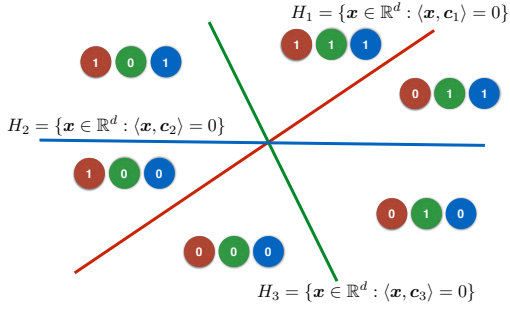


Figure 1. Three hyperplanes in \mathbb{R}^2 with the labels of the subspaces into which they dissect the space. Any $m \times 3$ binary matrix in which each row corresponds to one of the six label vectors has rounding rank at most 2.

- 1) $\text{rrank}_\tau(\mathbf{B}) \leq d$.
- 2) *There exist points $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{R}^d$ and affine hyperplanes H_1, \dots, H_n in \mathbb{R}^d with normal vectors $\mathbf{c}_1, \dots, \mathbf{c}_n \in \mathbb{R}^d$ given by $H_j = \{\mathbf{x} \in \mathbb{R}^d : \langle \mathbf{x}, \mathbf{c}_j \rangle = \tau\}$ such that $\text{round}_\tau(\langle \mathbf{x}_i, \mathbf{c}_j \rangle) = \mathbf{B}_{ij}$ for all i, j .*

Proof. 2 \Rightarrow 1: Consider points \mathbf{x}_i and hyperplanes H_j with the property asserted in the theorem. Define an $m \times d$ matrix \mathbf{L} with the \mathbf{x}_i in its rows, and an $n \times d$ matrix \mathbf{R} with the \mathbf{c}_j in its rows. Then $\text{round}_\tau(\mathbf{L}\mathbf{R}^T) = \mathbf{B}$, and hence $\text{rrank}_\tau(\mathbf{B}) \leq d$.

1 \Rightarrow 2: Let $\mathbf{B} = \text{round}_\tau(\mathbf{A})$ with $\text{rank}(\mathbf{A}) \leq d$. Pick any two real matrices \mathbf{L} and \mathbf{R} with d columns s.t. $\mathbf{L}\mathbf{R}^T = \mathbf{A}$. We can consider the rows \mathbf{L}_i of \mathbf{L} as points in \mathbb{R}^d ($\mathbf{x}_i = \mathbf{L}_i$) and the rows \mathbf{R}_j of \mathbf{R} as the normal vectors ($\mathbf{c}_j = \mathbf{R}_j$) of affine hyperplanes H_j with offset τ . Since $\mathbf{B} = \text{round}_\tau(\mathbf{A})$, we also get $\mathbf{B}_{ij} = \text{round}_\tau(\langle \mathbf{L}_i, \mathbf{R}_j \rangle)$ for all i, j . \square

Fig. 1 illustrates Th. 1 in \mathbb{R}^2 with $n = 3$ and $\tau = 0$. The three hyperplanes dissect the space into six convex, open regions. Each point $\mathbf{x} \in \mathbb{R}^2$ can be labeled with a binary vector according to whether it is “above” or “below” each of the hyperplanes H_j by using the rounding function $\text{round}_\tau(\langle \mathbf{x}, \mathbf{c}_j \rangle)$.

The second point of Th. 1 can be interpreted as follows: Pick a binary matrix \mathbf{B} and treat each of the n columns \mathbf{B}_j as the labels of a binary classification problem P_j on m points. We can find data points $\mathbf{x}_1, \dots, \mathbf{x}_m$ and affine hyperplanes H_1, \dots, H_n in \mathbb{R}^d which solve all linear classification problems P_j without error if and only if the rounding rank of \mathbf{B} is at most d . We then interpret the \mathbf{x}_i as data points and the \mathbf{c}_j as feature weights. Rounding rank decompositions thus describe the “best case” for multiple linear classification problems: if the rounding rank of \mathbf{B} is d , then we need at least d features to achieve perfect classification. In other words, we need to collect at least $\text{rrank}(\mathbf{B})$ features (or attributes) to have a chance to classify perfectly. Similarly, if we employ dimensionality reduction, linear classification cannot be perfect if we reduce to less than d dimensions.

Corollary 2 (informal). *Rounding rank provides a natural lower bound on how many features we need for linear classification. This provides us with lower bounds on data collection or dimensionality reduction.*

D. Comparison of the Ranks

We compare rounding rank with several well-known ranks. Many of the results in this subsection were obtained for sign rank in the communication complexity community; we present these results here in terms of rounding rank. To the best of our knowledge, we are the first to make the role of the rounding threshold explicit by introducing mixed matrices (see Prop. 5).

Boolean rank. For binary matrices $\mathbf{L} \in \{0, 1\}^{m \times k}$ and $\mathbf{R} \in \{0, 1\}^{n \times k}$, the *Boolean matrix product* $\mathbf{L} \circ \mathbf{R}^T$ is given by the $m \times n$ binary matrix with $[\mathbf{L} \circ \mathbf{R}^T]_{ij} = \bigvee_{\ell=1}^k (\mathbf{L}_{i\ell} \wedge \mathbf{R}_{\ell j})$ for all entries i, j . The *Boolean rank of a binary matrix* \mathbf{B} , denoted $\text{brank}(\mathbf{B})$, is the smallest $k \in \mathbb{N}$ s.t. there exist $\mathbf{L} \in \{0, 1\}^{m \times k}$ and $\mathbf{R} \in \{0, 1\}^{n \times k}$ with $\mathbf{B} = \mathbf{L} \circ \mathbf{R}^T$ [27]. The rounding rank is a lower bound on the Boolean rank.

Lemma 3. $\text{rrank}(\mathbf{B}) \leq \text{brank}(\mathbf{B})$.

Proof. Let $\text{brank}(\mathbf{B}) = k$. Then there exist matrices $\mathbf{L} \in \{0, 1\}^{m \times k}$ and $\mathbf{R} \in \{0, 1\}^{n \times k}$ s.t. $\mathbf{B} = \mathbf{L} \circ \mathbf{R}^T$. If we use the algebra of \mathbb{R} , we get $[\mathbf{L}\mathbf{R}^T]_{ij} \geq \frac{1}{2}$ iff $\mathbf{B}_{ij} = 1$. This implies $\text{round}(\mathbf{L}\mathbf{R}^T) = \mathbf{B}$ and $\text{rrank}(\mathbf{B}) \leq k = \text{brank}(\mathbf{B})$. \square

Real rank. Comparing rounding rank and real rank, we observe that $\mathbf{B} = \text{round}(\mathbf{B})$ for all binary matrices \mathbf{B} . Hence,

$$\text{rrank}(\mathbf{B}) \leq \text{rank}(\mathbf{B}).$$

This is in contrast to the relationship between Boolean rank and standard rank, which cannot be compared (i.e. neither serves as a lower bound to the other) [28].

Note that the rounding rank can be much lower than both real and Boolean rank. For example, the $n \times n$ “upper triangle matrix” with 1’s on the main diagonal and above has real and Boolean rank n , but rounding rank 1 (see Th. 11). As another example, we show in the extended version of this paper [29] that the $n \times n$ identity matrix has rounding rank 2 for all $n \geq 3$. In fact, while we know that a real-valued $n \times n$ matrix can have rank up to n , the situation is different for rounding rank: On the one hand, for large enough n , all $n \times n$ matrices \mathbf{B} have $\text{rrank}(\mathbf{B}) \leq (\frac{1}{2} + o(1))n$ [2, Cor. 1.2]. On the other hand, for each n , there exist $n \times n$ matrices with $\text{rrank}(\mathbf{B}) \geq \frac{n}{32}$, i.e., the rounding rank can indeed be linear in n [2, Cor. 1.2].

It is well-known that real-valued matrices with all entries picked uniformly at random from some bounded proper interval have full standard rank with probability 1. For rounding rank, an $n \times n$ binary matrix sampled uniformly at random has rounding rank $\Omega(n)$ with high probability (see the proof of Cor. 1.2 in [2]). Hence, the rounding ranks of random binary matrices are expected to be large. The real-world data matrices in our experiments often had small rounding ranks, though.

A lower bound on the rounding rank of a binary matrix \mathbf{B} can be derived from the singular values of the sign matrix \mathbf{B}^\pm .

Proposition 4. *Let $r = \text{rank}(\mathbf{B}^\pm)$ and let $\sigma_1(\mathbf{B}^\pm) \geq \dots \geq \sigma_r(\mathbf{B}^\pm) > 0$ be the non-zero singular values of \mathbf{B}^\pm . Then*

$$(\text{rrank}_0(\mathbf{B}) + 1) \sum_{i=1}^{\text{rank}_0(\mathbf{B})} \sigma_i^2(\mathbf{B}^\pm) \geq mn.$$

Prop. 4 is a slight modification of a result in [17, Th. 5] and we give the proof in the extended version [29].

Role of rounding threshold. We compare the rounding ranks of a fixed matrix for different rounding thresholds. We call a binary matrix *mixed*, if it contains no all-zero and no all-one columns (or rows).

Proposition 5. *For any \mathbf{B} and arbitrary $\tau \neq \tau' \in \mathbb{R}$, $\text{rrank}_\tau(\mathbf{B})$ and $\text{rrank}_{\tau'}(\mathbf{B})$ differ by at most 1. If additionally $\tau, \tau' \neq 0$, $\text{rrank}_\tau(\mathbf{B}) = \text{rrank}_{\tau'}(\mathbf{B})$ if $\text{sign}(\tau) = \text{sign}(\tau')$ or if \mathbf{B} is mixed.*

To prove Prop. 5 we need Lem. 6 below. The lemma is implied by the Hyperplane Separation Theorem [10, p. 46], and we prove it in [29].

Lemma 6. *Let A and B be two disjoint nonempty convex sets in \mathbb{R}^d , one of which is compact. Then for all nonzero $c \in \mathbb{R}$, there exists a nonzero vector $\mathbf{v} \in \mathbb{R}^d$, such that $\langle \mathbf{x}, \mathbf{v} \rangle > c$ and $\langle \mathbf{y}, \mathbf{v} \rangle < c$ for all $\mathbf{x} \in A$ and $\mathbf{y} \in B$.*

Proof of Prop. 5. First claim: Let $\tau, \tau' \in \mathbb{R}$ be arbitrary and pick $k \in \mathbb{N}$, $\mathbf{L} \in \mathbb{R}^{m \times k}$, $\mathbf{R} \in \mathbb{R}^{n \times k}$ such that $\text{round}_\tau(\mathbf{L}\mathbf{R}^T) = \mathbf{B}$. Set $c = \tau' - \tau$, then

$$\mathbf{B}_{ij} = \text{round}_\tau([\mathbf{L}\mathbf{R}^T]_{ij}) = \text{round}_{\tau'}([\mathbf{L}\mathbf{R}^T]_{ij} + c).$$

Set $\mathbf{L}' = (\mathbf{L} \ c\mathbf{1})$ and $\mathbf{R}' = (\mathbf{R} \ \mathbf{1})$, where $\mathbf{1}$ denotes the all-one vector. Then $\text{round}_{\tau'}(\mathbf{L}'\mathbf{R}'^T) = \mathbf{B}$ and thus $\text{rrank}_{\tau'}(\mathbf{B}) \leq k + 1$.

Second claim: Without loss of generality, assume that \mathbf{B} contains no all-zero and no all-one columns (otherwise tranpose the matrix). Let $\tau, \tau' \neq 0$ and let k and $\mathbf{L}\mathbf{R}^T$ be as before. If $\text{sign}(\tau) = \text{sign}(\tau')$, set $c = \tau'/\tau > 0$ and $\mathbf{R}' = c\mathbf{R}$. Then $\text{round}_\tau(\mathbf{L}\mathbf{R}) = \text{round}_{\tau'}(\mathbf{L}\mathbf{R}')$ by construction so that $\text{rrank}_{\tau'}(\mathbf{B}) \leq k$. By reversing the roles of τ and τ' in the argument, we establish $\text{rrank}_\tau(\mathbf{B}) = \text{rrank}_{\tau'}(\mathbf{B})$.

Suppose $\tau, \tau' \neq 0$ (not necessarily of same sign) and let \mathbf{B} be mixed. We now treat the rows of \mathbf{L} as points $\mathbf{L}_1, \dots, \mathbf{L}_m$ in \mathbb{R}^k and show that there exists an $n \times k$ matrix \mathbf{R}' consisting of normal vectors of affine hyperplanes in \mathbb{R}^k in its rows such that the hyperplanes separate the points with rounding threshold τ' , thereby establishing $\text{rrank}_{\tau'}(\mathbf{B}) \leq \text{rrank}_\tau(\mathbf{B})$. Again, by reversing the roles of τ and τ' , we obtain equality. To construct the j 'th row of \mathbf{R}' , let $C_j = \{\mathbf{L}_i : \mathbf{B}_{ij} = 1\}$ and $\bar{C}_j = \{\mathbf{L}_i : \mathbf{B}_{ij} = 0\}$. Notice that since \mathbf{B} is mixed, both C_j and \bar{C}_j are non-empty. We observe that the convex hulls of C_j and \bar{C}_j are separated by the affine hyperplane with the j 'th row of \mathbf{R} as its normal vector and offset from the origin τ . Thus, we apply Lem. 6 to obtain a vector \mathbf{r}'_j s.t. $\langle \mathbf{r}'_j, \mathbf{c} \rangle > \tau'$ for all $\mathbf{c} \in C_j$ and $\langle \mathbf{r}'_j, \bar{\mathbf{c}} \rangle < \tau'$ for all $\bar{\mathbf{c}} \in \bar{C}_j$. We set \mathbf{r}'_j to be the j 'th row of \mathbf{R}' . To obtain \mathbf{R}' , we repeating this procedure for each of its n rows. \square

The above proof can be adopted to show that if \mathbf{B} is mixed, even using a different (non-zero) rounding threshold for each row (or column) does not affect the rounding rank.

Non-negative rounding rank. While the gap between rank and non-negative rank can be arbitrarily large [6], for rounding rank and non-negative rounding rank this is not the case.

Proposition 7. $\text{rrank}_\tau^+(\mathbf{B}) \leq \text{rrank}_\tau(\mathbf{B}) + 2$.

This can be shown using ideas similar to the ones in [30] by a simple but lengthy computation. We give a proof in [29].

E. Computational Complexity

The following proposition asserts that rounding rank is NP-hard to compute regardless of the rounding threshold.

Proposition 8. *It is NP-hard to decide if $\text{rrank}_0(\mathbf{B}) \leq k$ for all $k > 2$. For $\tau \neq 0$, it is NP-hard to decide if $\text{rrank}_\tau(\mathbf{B}) \leq k$ for all $k > 1$.*

For sign rank (i.e. $\tau = 0$), this was proven in [9, Th. 1.2], [5, Sec. 3]. Moreover, Alon et al. [4] argue that computing the sign rank is equivalent to the existential theory of the reals. For $\tau \neq 0$, NP-hardness was proven in [24, Th. 10].

It is an open problem whether sign rank or rounding rank computation is in NP. Assume we store a matrix \mathbf{A} that achieves the rounding rank of \mathbf{B} by representing all entries with rational numbers. The following proposition asserts that in general, the space needed to store a \mathbf{A} can be exponential in the size of \mathbf{B} . Hence, the proposition rules out proving that computing rounding rank is in NP by nondeterministically guessing a matrix \mathbf{A} of small rank and rounding it.

Proposition 9. *For all sufficiently large n , there exist $n \times n$ binary matrices \mathbf{B} with $\text{rank}(\mathbf{B}) = 3$ s.t. for each matrix \mathbf{A} with $\text{rank}(\mathbf{A}) = 3$ and $\text{round}(\mathbf{A}) = \mathbf{B}$, it takes $\Theta(\exp(n))$ bits to store the entries of \mathbf{A} using rational numbers.*

Prop. 9 can be derived from the proof of [24, Th. 4].

Lemma 10. *The MINERRORRR- k problem is NP-hard to solve exactly. It is also NP-hard to approximate within any polynomial-time computable factor.*

Proof. Both claims follow from Prop. 8. If in polynomial time we could solve the MINERRORRR- k problem exactly or within any factor, then we could also decide if $\text{rrank}(\mathbf{B}) \leq k$ by checking if the result for MINERRORRR- k is zero. \square

III. COMPUTING THE ROUNDING RANK

In this section, we provide algorithms approximating $\text{rrank}(\mathbf{B})$ and for approximately solving the MINERRORRR- k problem. The algorithms are based on some of the most common paradigms for algorithm design in data mining. The ProjLP algorithm makes use of randomized projections, R-SVD uses truncated SVD, L-PCA uses logistic PCA, and ASSO is a Boolean matrix factorization algorithm. For each algorithm, we first discuss how to obtain an approximation to $\text{rrank}(\mathbf{B})$ (in the form of an upper bound) and then discuss extensions to solve MINERRORRR- k .

Projection-based algorithm (ProjLP). We first describe a Monte Carlo algorithm to decide whether $\text{rrank}(\mathbf{B}) \leq d$ for a given matrix \mathbf{B} and $d \in \mathbb{N}$. The algorithm can output YES

or UNKNOWN. If the algorithm outputs YES, it also produces a rounding rank decomposition. We use this algorithm for different values of d to approximate $\text{rrank}(\mathbf{B})$.

The decision algorithm is inspired by a simple observation: Considering an $m \times n$ binary matrix \mathbf{B} , we have $\mathbf{B} = \text{round}(\mathbf{B}\mathbf{I})$, where \mathbf{I} denotes the $n \times n$ identity matrix. We interpret each row \mathbf{B}_i of \mathbf{B} as a point in \mathbb{R}^n and each column \mathbf{I}_j of \mathbf{I} as the normal vector of a hyperplane in \mathbb{R}^n . The hyperplane given by \mathbf{I}_j separates the points \mathbf{B}_i into the classes $C_j = \{\mathbf{B}_i : \mathbf{B}_{ij} = 1\}$ and $\bar{C}_j = \{\mathbf{B}_i : \mathbf{B}_{ij} = 0\}$ by the j 'th column of \mathbf{B} , since $\mathbf{B}_{ij} = \text{round}(\langle \mathbf{B}_i, \mathbf{I}_j \rangle)$. The idea of `ProjLP` is to take the points \mathbf{B}_i (the rows of \mathbf{B}) and to project them into lower-dimensional space \mathbb{R}^d , $d \ll n$, to obtain vectors $\mathbf{L}_1, \dots, \mathbf{L}_m \in \mathbb{R}^d$. We use a randomized projection that approximately preserves the distances of the \mathbf{B}_i and—if \mathbf{B} has rounding rank at most d —try (or hope) to maintain the separability of the points by hyperplanes by doing so. Given the projected vectors in \mathbb{R}^d , we check separability by affine hyperplanes and find the corresponding normal vectors $\mathbf{R}_1, \dots, \mathbf{R}_n$ using a linear program. If the \mathbf{L}_i turn out to be separable, we have $\mathbf{B}_{ij} = \text{round}(\langle \mathbf{L}_i, \mathbf{R}_j \rangle)$ for all i, j and thus $\mathbf{B} = \text{round}(\mathbf{L}\mathbf{R}^T)$, where \mathbf{L} and \mathbf{R} have the \mathbf{L}_i 's and \mathbf{R}_j 's in their rows, respectively. We conclude $\text{rrank}(\mathbf{B}) \leq d$ and output YES. If the \mathbf{L}_i are not separable, no conclusions can be drawn and the algorithm outputs UNKNOWN.

The Johnson–Lindenstrauss Lemma [22] asserts that there exists a linear mapping \mathbf{A} that projects points from a high-dimensional space into a lower-dimensional space while approximately preserving the distances. We use the projections proposed by Achlioptas [1] to obtain \mathbf{A} . We set $\mathbf{L}_i = \mathbf{B}_i \mathbf{A}$. The linear program (LP) to compute the normal vector \mathbf{R}_j is

$$\begin{aligned} & \text{find } \mathbf{R}_j \in \mathbb{R}^d \\ & \text{subject to } \sum_{k=1}^d \mathbf{L}_{ik} \mathbf{R}_{jk} \geq \tau + \varepsilon \quad \text{if } \mathbf{B}_{ij} = 1, \\ & \quad \quad \quad \sum_{k=1}^d \mathbf{L}_{ik} \mathbf{R}_{jk} \leq \tau - \varepsilon \quad \text{if } \mathbf{B}_{ij} = 0. \end{aligned}$$

We enforce strict separability by introducing an offset $\varepsilon > 0$. In practice, we set ε to the smallest positive number representable by the floating-point hardware. Notice that the LP only aims at finding a feasible solution; it has m constraints and d variables.

To approximate the rounding rank, we repeatedly run the above algorithm with increasing values of d until it outputs YES; i.e., $d = 1, 2, \dots$. Alternatively, we could use some form of binary search to find a suitable value of d . In practice, however, solving the LP for large values of d slows down the binary search too much.

To solve `MINERRORRR- k` , we modify the LP of `ProjLP` to output an approximate solution. For this purpose, we introduce non-negative slack-variables \mathbf{c}_i as in soft-margin SVMs to allow for errors, and an objective function that minimizes the L_1 norm of the slack variables. We obtain the following LP:

$$\begin{aligned} & \min_{\substack{\mathbf{c} \in \mathbb{R}_{\geq 0}^m \\ \mathbf{R}_j \in \mathbb{R}^d}} \sum_{i=1}^m \mathbf{c}_i \\ & \text{subject to } \sum_{k=1}^d \mathbf{L}_{ik} \mathbf{R}_{jk} + \mathbf{c}_i \geq \tau + \varepsilon, \quad \text{if } \mathbf{B}_{ij} = 1, \\ & \quad \quad \quad \sum_{k=1}^d \mathbf{L}_{ik} \mathbf{R}_{jk} - \mathbf{c}_i \leq \tau - \varepsilon, \quad \text{if } \mathbf{B}_{ij} = 0. \end{aligned}$$

Rounded SVD algorithm (R-SVD). We use rounded SVD to approximate $\text{rrank}(\mathbf{B})$. The algorithm is greedy and similar to the one in [14]. Given a binary matrix \mathbf{B} , the algorithm sets $k = 1$. Then it computes the rank- k truncated SVD of \mathbf{B} and rounds it. If the rounded matrix and \mathbf{B} are equal, it returns k , otherwise, it sets $k = k + 1$ and repeats. The underlying reasoning is that the rank- k SVD is the real-valued rank k matrix minimizing the distance to \mathbf{B} w.r.t. the Frobenius norm. Hence, also its rounded version should be “close” to \mathbf{B} .

To approximately solve `MINERRORRR- k` , we compute the truncated rank- ℓ -SVD of \mathbf{B} for all $\ell = 1, \dots, k$ and return the rounded matrix with the smallest error.

Logistic Principal Component Analysis (L-PCA). The logistic function $f(x) = (1 + e^{-x})^{-1}$ is a differentiable surrogate of the rounding function and it can be used to obtain a smooth approximation of the rounding.

L-PCA [32] models each \mathbf{B}_{ij} as a Bernoulli random variable with success probability $f(\langle \mathbf{L}_i, \mathbf{R}_j \rangle)$, where $\mathbf{L} \in \mathbb{R}^{m \times k}$ and $\mathbf{R} \in \mathbb{R}^{n \times k}$ are unknown parameters. Given \mathbf{B} and $k \in \mathbb{N}$ as input, L-PCA obtains (approximate) maximum-likelihood estimates of \mathbf{L} and \mathbf{R} . If each $f(\langle \mathbf{L}_i, \mathbf{R}_j \rangle)$ is a good estimate of $\mathbf{B}_{ij} = 1$, then $\|\mathbf{B} - \text{round}(\mathbf{L}\mathbf{R}^T)\|_F$ should be small.

To approximate the $\text{rrank}(\mathbf{B})$, we run L-PCA on \mathbf{B} for $k = 1$ and check if $\text{round}(\mathbf{L}\mathbf{R}^T) = \mathbf{B}$. If this is the case, we return k , otherwise, we set $k = k + 1$ and repeat.

To use L-PCA to compute an approximation of `MINERRORRR- k` , we simply run L-PCA and apply rounding.

Permutation algorithm (Permutation). The only known algorithm to approximate the sign rank of a $n \times n$ matrix in polynomial time was given in [4]; it guarantees an upper bound within an approximation ratio of $O(n/\log n)$. By Prop. 5, we can use this method to approximate the rounding rank. The algorithm permutes the rows of the input matrix \mathbf{B} s.t. the maximum number of bit flips over all columns is approximately minimized. It then algebraically approximates $\text{rrank}(\mathbf{B})$ by evaluating a certain polynomial based on the occurring bit flips. The algorithm cannot solve the `MINERRORRR- k` problem.

Nuclear norm algorithm (Nuclear). The nuclear norm $\|\mathbf{X}\|_*$ of a matrix \mathbf{X} is the sum of the singular values of \mathbf{X} and is a convex and differentiable surrogate of the rank function of matrix. A common relaxation for minimum-rank matrix factorization is to minimize $\|\mathbf{X}\|_*$ instead of $\text{rank}(\mathbf{X})$.

In our setting, we obtain the following minimization problem:

$$\begin{aligned} \mathbf{X}^* = \arg \min_{\mathbf{X} \in \mathbb{R}^{m \times n}} \quad & \|\mathbf{X}\|_* \\ \text{subject to} \quad & \mathbf{X}_{ij} \geq \tau \quad \text{if } \mathbf{B}_{ij} = 1, \\ & \mathbf{X}_{ij} < \tau \quad \text{if } \mathbf{B}_{ij} = 0. \end{aligned}$$

This method has some caveats: While \mathbf{X}^* must have small singular values, it may still have *many*. Additionally, by Prop. 9, some entries of a matrix \mathbf{A} achieving the rounding rank might be extremely large. In such a case, some of the singular values of \mathbf{A} must also be large, and consequently the nuclear norm of the matrix is large. Thus, \mathbf{X}^* might have a too large rank. This algorithm cannot be extended to solve MINERRORRR- k .

IV. NESTED MATRICES

A binary matrix is nested if we can reorder its columns such that after the reordering, the one-entries in each row form a contiguous segment starting from the first column [25]. Intuitively, nested matrices model subset/superset relationships between the rows and columns of a matrix. Such structures are, for example, found in presence/absence data of locations and species [25].

We show that nested matrices are exactly the matrices with non-negative rounding rank 1. Formally, a binary matrix \mathbf{B} is *directly nested* if for each one-entry $\mathbf{B}_{ij} = 1$, we have $\mathbf{B}_{i'j'} = 1$ for all $i' \in \{1, \dots, i-1\}$ and $j' \in \{1, \dots, j-1\}$. A binary matrix \mathbf{B} is *nested* if there exist permutation matrices \mathbf{P}_1 and \mathbf{P}_2 , such that $\mathbf{P}_1 \mathbf{B} \mathbf{P}_2$ is directly nested.

Theorem 11. *Let $\mathbf{0} \neq \mathbf{B} \in \{0, 1\}^{m \times n}$. Then \mathbf{B} is nested if and only if $\text{rrank}^+(\mathbf{B}) = 1$.*

Proof. \Rightarrow : We reorder the rows and columns of \mathbf{B} by the number of 1s they contain in descending order. This gives us permutation matrices \mathbf{P}_1 and \mathbf{P}_2 s.t. $\mathbf{B}' = \mathbf{P}_1 \mathbf{B} \mathbf{P}_2$ is directly nested. Set $\mathbf{p} = \mathbf{B}' \mathbf{1}$, i.e., \mathbf{p} is the vector containing the row sums of \mathbf{B}' . Then for \mathbf{l}' and \mathbf{r}' with $l'_i = 2^{p_i-1}$ and $r'_j = 2^{-j}$, $\mathbf{B}' = \text{round}(\mathbf{l}' \cdot (\mathbf{r}')^T)$. Setting $\mathbf{l} = \mathbf{P}_1^T \mathbf{l}'$ and $\mathbf{r} = \mathbf{P}_2 \mathbf{r}'$, we get $\mathbf{B} = \text{round}(\mathbf{l} \cdot \mathbf{r}^T)$. Hence, we have $\text{rrank}(\mathbf{B}) = 1$.

\Leftarrow : Let $\mathbf{l} \geq 0$ and $\mathbf{r} \geq 0$ be s.t. $\mathbf{B} = \text{round}(\mathbf{l} \mathbf{r}^T)$. Then there exist permutation matrices \mathbf{P}_1 and \mathbf{P}_2 s.t. for $\mathbf{l}' = \mathbf{P}_1 \mathbf{l}$ we have $l'_1 \geq \dots \geq l'_m$ and for $\mathbf{r}' = \mathbf{P}_2^T \mathbf{r}$ we have $r'_1 \geq \dots \geq r'_n$. Set $\mathbf{B}' = \text{round}(\mathbf{l}' (\mathbf{r}')^T)$ and observe $l'_i r'_j \geq l'_{i+1} r'_j$ for all i, j . Therefore, for each entry of \mathbf{B}' , $\mathbf{B}'_{ij} = \text{round}(l'_i r'_j) \geq \text{round}(l'_{i+1} r'_j) = \mathbf{B}'_{(i+1)j}$. Similarly, $\mathbf{B}'_{ij} = \text{round}(l'_i r'_j) \geq \text{round}(l'_i r'_{j+1}) = \mathbf{B}'_{i(j+1)}$. Therefore, \mathbf{B}' is directly nested. We conclude that $\mathbf{B} = \mathbf{l} \mathbf{r}^T$ is nested since $\mathbf{B} = \text{round}(\mathbf{l} \mathbf{r}^T) = \mathbf{P}_1^T \text{round}(\mathbf{P}_1 (\mathbf{l} \mathbf{r}^T) \mathbf{P}_2) \mathbf{P}_2^T = \mathbf{P}_1^T \mathbf{B}' \mathbf{P}_2^T$. \square

Binary matrices with rounding rank 1 are also closely related to nested matrices.

Proposition 12. *Let $\mathbf{0} \neq \mathbf{B} \in \{0, 1\}^{m \times n}$. The following statements are equivalent:*

- 1) $\text{rrank}(\mathbf{B}) = 1$.

- 2) *there exist permutation matrices \mathbf{P}_1 and \mathbf{P}_2 and nested matrices \mathbf{B}_1 and \mathbf{B}_2 , such that*

$$\mathbf{B} = \mathbf{P}_1 \begin{pmatrix} \mathbf{B}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2 \end{pmatrix} \mathbf{P}_2.$$

The proof is in the extended version [29].

Algorithms. Mannila and Terzi [25] introduced the *Bidirectional Minimum Nestedness Augmentation* (BMNA) problem: Given a binary matrix \mathbf{B} , find the nested matrix \mathbf{A} which minimizes $\|\mathbf{B} - \mathbf{A}\|_F$. We will discuss three algorithms to approximately solve this problem.

[25] gave an algorithm, MT, which approximates a solution for the BMNA problem by iteratively eliminating parts of the matrix that violate the nestedness.

Next, we propose a alternating minimization algorithm, NNRR1, which exploits Th. 11. NNRR1 maintains two vectors $\mathbf{l} \in \mathbb{R}_{\geq 0}^m$ and $\mathbf{r} \in \mathbb{R}_{\geq 0}^n$ and iteratively minimizes the error $\|\mathbf{B} - \text{round}(\mathbf{l} \cdot \mathbf{r}^T)\|_F$. It starts by fixing \mathbf{r} and updates \mathbf{l} , such that the error is minimized. Then \mathbf{l} is fixed and \mathbf{r} is updated. This procedure is repeated until the error stops reducing or we have reached a certain number of iterations.

We describe an update of \mathbf{l} for fixed \mathbf{r} ; updating \mathbf{r} for given \mathbf{l} is symmetric. Observe that changing entry l_i only alters the i 'th row of $\mathbf{A} = \mathbf{l} \cdot \mathbf{r}^T$, and consequently \mathbf{A}_i is not affected by any l_k with $k \neq i$. Hence, we only describe the procedure for updating l_i . Define the set $V_i = \{r_j : \mathbf{B}_{ij} = 1\}$ of all values of \mathbf{r} where \mathbf{B}_i is non-zero. We make the following observations: If we set $l_i < \frac{1}{2 \max(\mathbf{r})}$, then \mathbf{A}_i only contains zeros after the update. If $\frac{1}{2 \max(\mathbf{r})} < l_i < \frac{1}{2 \max(V_i)}$, then after the update all non-zeros of \mathbf{A}_i will be in entries where \mathbf{B}_i has a zero. If $l_i > \frac{1}{2 \min(V_i)}$, we add too many 1s to \mathbf{A}_i . Thus, all values that we need to consider for updating l_i are $\frac{1}{2 \max(\mathbf{r})}$ and the values in $\{\frac{1}{2v} : v \in V_i\}$. The algorithm tries all possible values for l_i exhaustively and computes the error at each step.

We can also use the results of MT as initialization for NNRR1: We run MT and obtain a nested matrix \mathbf{B} . Now we use the construction from step 1 of the proof of Th. 11 to obtain \mathbf{l} and \mathbf{r} with $\mathbf{B} = \text{round}(\mathbf{l} \mathbf{r}^T)$, and try to improve using NNRR1.

Finally, we can use R-SVD to solve the BMNA problem approximately. By the Perron-Frobenius Theorem [21, Ch. 8.4], the principal left and right singular vectors of a non-negative matrix are also non-negative. Hence we may use the R-SVD algorithm to obtain the rank-1 truncated SVD and round. By Th. 11, the result must be nested.

V. EXPERIMENTS

We conducted an experimental study on synthetic and real-world datasets to evaluate the relative performance of each algorithm for estimating the rounding rank or for MINERRORRR- k .

A. Implementation Details

For L-PCA, we used the implementation by the authors of [32]. We implemented MT and Permutation in C and all other algorithms in Matlab. For Nuclear, we used the CVX package with the SeDuMi solver [20]. For solving the linear programs in ProjLP, we used Gurobi.

Due to numerical instabilities, `Nuclear` often returned a matrix with only positive singular values (i.e. of full rank). We countered this by zeroing the smallest singular values of the returned matrix that did not affect to the result of the rounding.

All experiments were conducted on a computer with eight Intel Xeon E5530 processors running at 2.4 GHz and 48 GB of main memory. All our algorithms and the synthetic data generators are available online.¹

B. Results With Synthetic Data

We start by studying the behavior of the algorithms under controlled synthetic datasets.

1) *Data generation*: We generated synthetic data by sampling two matrices $\mathbf{L} \in \mathbb{R}^{m \times k}$ and $\mathbf{R} \in \mathbb{R}^{n \times k}$ and then rounding their product to obtain $\mathbf{B} = \text{round}_\tau(\mathbf{L}\mathbf{R}^T)$ with rounding rank *at most* k . The actual rounding rank of \mathbf{B} can be lower, however, because there may be matrices $\mathbf{L}' \in \mathbb{R}^{m \times k'}$ and $\mathbf{R}' \in \mathbb{R}^{n \times k'}$ with $k' < k$ and $\text{round}_\tau(\mathbf{L}'\mathbf{R}'^T) = \mathbf{B}$. (In fact, we sometimes found such matrices.) In some experiments, we additionally applied noise by flipping elements selected uniformly at random. We report as *noise level* p the ratio of the number of flipped elements to the number of non-zeros in the original noise-free matrix.

We sampled every element of \mathbf{L} and \mathbf{R} i.i.d. using two families of distributions: uniform and normal distribution. For both distributions, we first pick a desired expected value $\mu = \mathbb{E}[(\mathbf{L}\mathbf{R}^T)_{ij}]$ of each entry in $\mathbf{L}\mathbf{R}^T$. We then parameterize the distributions such that the expected value for an element of \mathbf{L} or \mathbf{R} is $q = \sqrt{\mu/k}$. For the normal distribution, we set the variance to 1, and for the uniform distribution, we sampled from range $[q - 1/2, q + 1/2]$.

We generated two sets of matrices. In the first set, the matrices were very small, and it was used to understand the behavior of the slower algorithms. In the second set, the matrices were medium-sized, to give us more realistic-sized data, but we could use only some of the methods with these data. When generating the data, we varied four different parameters: number of rows m , the planted rank k , the expected value μ , and the level of noise p . In all experiments, we varied one of these parameters, while keeping the others fixed. We generated all datasets with rounding threshold $\tau = 1/2$. For the small data, we used $n = 100$ columns and the number of rows varied from 60 to 220 with steps of 40 with the default value being $n = 100$. The rank k in the small matrices varied from 5 to 30 with steps of 5, default being $k = 10$; the expected value μ varied from 0.1 to 0.7 with 0.1 steps (default was $\mu = 0.5$); the noise level p varied from 0.05 to 0.5 with steps of 0.05, and by default we did not apply any noise. We generated ten random matrices with each parameter setting to test the variation of the results.

For the medium-sized matrices, we used $n = 300$ columns and the number of rows varied from 400 to 600 with steps of 50 the default being $m = 500$; the planted rank k varied

from 40 to 100 with default value $k = 60$; the expected value and the noise were as with the small data. We generated five random matrices with each parameter setting.

2) *Rounding rank*: In our first set of experiments, we studied the performance of the different algorithms for estimating the rounding rank. The results for the small synthetic datasets are summarized in Fig. 2. The results are given for the uniformly distributed factor matrices; the results with normally distributed factors were largely similar and are postponed to [29].

We used `ProjLP`, `Nuclear`, `R-SVD`, and `L-PCA`. We also used `Permutation` in all experiments except when we varied the number of rows (`Permutation` only works with square matrices). We also computed a lower bound `Spectral LB` on rrank_0 using Prop. 4. Finally, in experiments with no noise, we also plot the planted rank (inner dimension of the factor matrices), which acts as an *upper bound* of the actual rounding rank.

As can be seen from Fig. 2, the estimated lower bound is almost always less than 3, even when the data contains significant amounts of noise. It seems reasonable to assume that the true rounding rank of the data is therefore closer to the upper bound of our planted rank than the estimated lower bound given by `Spectral LB`.

Of the algorithms tested here, `ProjLP`, and `Permutation` are the only ones that aim directly to find the rounding rank, with `Permutation` being the only one with approximation guarantees (albeit weak ones). Our experiments show that `Permutation` is not competitive to most other methods; good theoretical properties do not ensure a good practical behavior. `ProjLP` performs much better, being typically the second-best method. `R-SVD` is commonly employed in the literature, but our experiments show clearly that for computing the rounding rank, it is not recommended.

`L-PCA` consistently produced the smallest (i.e. best) rank estimate but it was also the second-slowest method. `ProjLP`, the second-best method for estimating the rank, was much faster. `R-SVD` often produced the worst estimates, but it is also the fastest method. The running times are broadly as expected: `Nuclear` has to solve a semidefinite programming problem, `L-PCA` solves iteratively dense least-squares problems, `ProjLP` only needs to solve linear equations, and `R-SVD` computes a series of orthogonal projections.

Varying the different parameters yielded mostly expected results with the most interesting result being how little the rank and noise had effect to the results. We assume that this is (at least partially) due to the robustness of the rounding rank: increasing the noise, say, might not have increased the rounding rank of the matrix. This is clearly observed when the rank is varied (Fig. 2(b)), where `L-PCA` actually obtains smaller rounding rank than the planted one.

3) *Minimum-error decomposition*: We now study the algorithms' capability to return low-error fixed-rank decompositions. We leave out `Permutation` and `Nuclear` as they only approximate rounding rank. Instead, we add a method to compare against: `T-SVD`. It computes the standard truncated SVD, that is, we do not apply any rounding. `T-SVD` is used

¹<http://dws.informatik.uni-mannheim.de/en/resources/software/rounding-rank/>

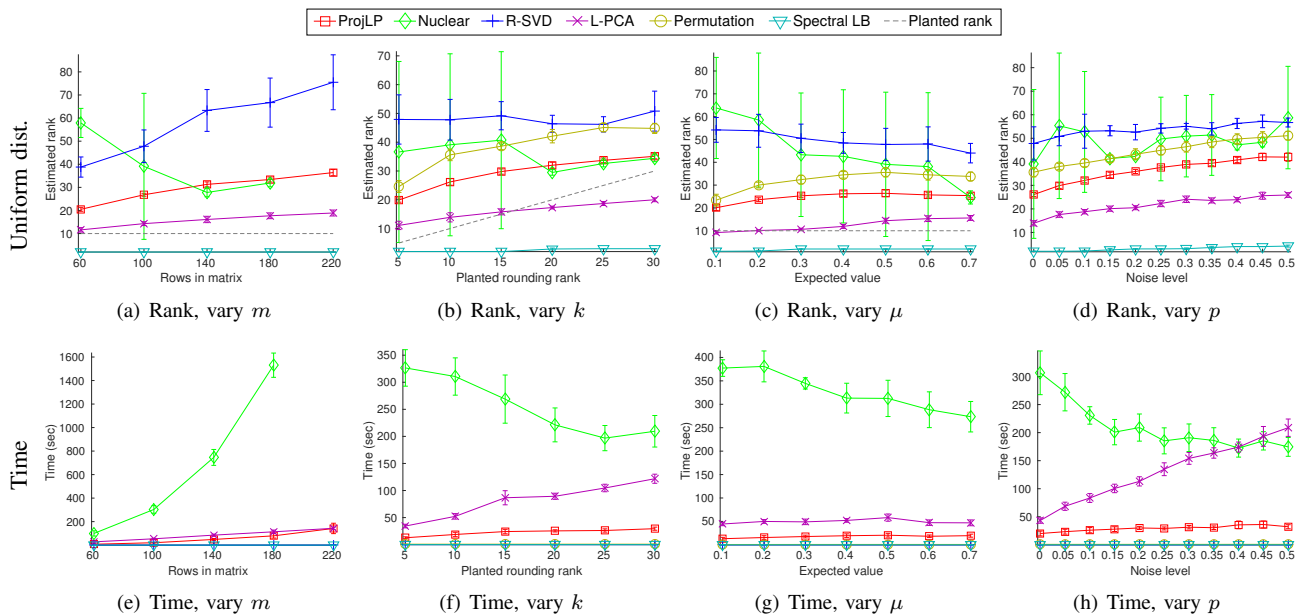


Figure 2. Estimated rounding ranks and running times on small synthetic data varying different parameters. The top row gives estimated rank for uniformly distributed factor matrices and the bottom row shows running times. *Permutation* can only run on square matrices and was excluded from the “vary m ” experiments. All data points are averages over 10 random matrices and the width of the error bars is twice the standard deviation.

for providing a baseline: in principle, the methods that apply rounding should give better results as they utilize the added information that the final matrix must be binary. At the same time, however, the rounding procedure may emphasize small errors (e.g., incorrectly representing a 1 with 0.49 contributes ≈ 0.26 to the sum of squares; after rounding, the contribution is 1). We also tested the *ASSO* [27] algorithm for Boolean matrix factorization (BMF). Like any BMF algorithm, *ASSO* returns a rounding rank decomposition restricted to binary factor matrices. The performance of *ASSO*’s approximations was so much worse than the performance of the other methods that we decided to omit it from the results.

To compare the algorithms, we use the relative reconstruction error, that is, the squared Frobenius norm of the distance between the data and its representation relative to the squared norm of the data. For all method except *T-SVD*, the relative reconstruction error agrees with the absolute number of errors divided by the number of non-zeros in the data.

The results for these experiments are presented in Fig. 3. We only report the reconstruction with uniformly distributed factors: the running times were as with the above experiments, and the results with normally distributed factors were generally similar to the reported ones. The other results are in [29]. As in the above experiments, *L-PCA* is the best method, and the slowest as well, taking sometimes an order of magnitude longer than *ProjLP*. The best all-rounder here, though, is the *R-SVD* method: it provided reasonable results and was by far the fastest method.

C. Results with Real-World Data

We now turn our attention to real-world datasets. For these experiments we used only *ProjLP*, *L-PCA*, and *R-SVD* to

estimate the rounding rank, and added *T-SVD* and *ASSO* for the minimum-error decompositions.

Datasets. The basic properties of the datasets are listed in Tab. I. The *ABSTRACTS* data set² is a collection of project abstracts that were submitted to the National Science Foundation of the USA in applications for funding. The data is documents-by-terms matrix giving the appearance of terms in documents. The *DBLP* data³ is an authors-by-conferences matrix containing information who published where. The *PALEO* data set⁴ contains information about the locations at which fossils of certain species were found. It was fetched by [18] and preprocessed according to [19]. The *DIALECT* data [11], [12] contains information about which linguistic features appear in the dialect spoken in various parts of Finland. The *APJ* dataset is a binary matrix containing access control rules from Hewlett-Packard [13].

Rounding rank. First we computed the upper bounds for the rounding ranks with the different methods. The results and running times are shown in Tab. I. As with the synthetic experiments, *L-PCA* is again giving the best results, followed by *ProjLP* and *R-SVD*, the latter of which returns often significantly worse results than the other two. In the running times the order is reversed, *L-PCA* taking orders of magnitude longer than *ProjLP*, which is still slower than *R-SVD*.

Note that the estimated rounding ranks in Tab. I are significantly less than the respective normal or Boolean ranks. For example, for the *APJ* data, the normal rank is 455, the Boolean rank is 453, but *L-PCA* shows that the rounding rank is at most 9. Similarly, the normal and Boolean ranks for *DBLP*

²<http://kdd.ics.uci.edu/databases/nsfabs/nsfawards.html>

³<http://dblp.uni-trier.de/db/>

⁴<http://www.helsinki.fi/science/now/>

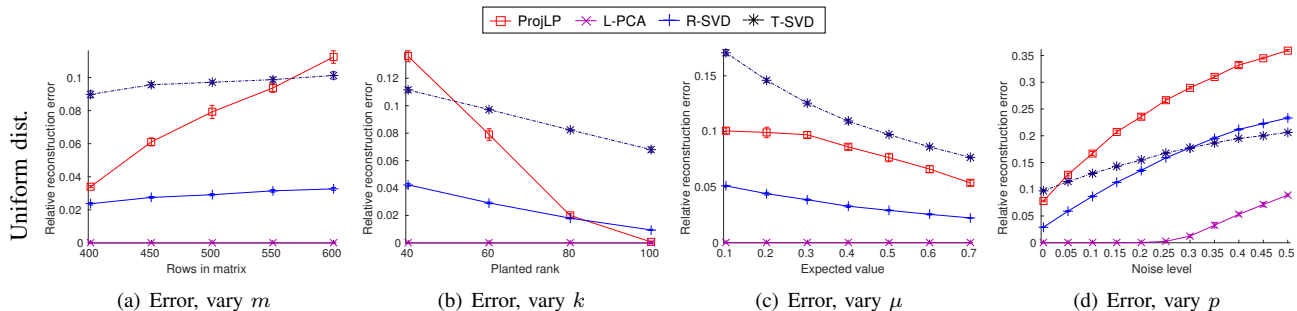


Figure 3. Relative reconstruction errors on medium-sized synthetic data with uniformly distributed factors. The results of `ASSO` are omitted as they were significantly worse than the other results. All data points are averages over 10 random matrices and the width of the error bars is twice the standard deviation.

Table I
UPPER BOUNDS FOR ROUNDING RANK WITH $\tau = 0.5$ FOR THE REAL-WORLD DATA. KNOWN BOOLEAN RANKS FROM [7]. L-PCA DID NOT FINISH ON THE ABSTRACTS DATA IN REASONABLE TIME.

Dataset	Dataset properties				Upper bounds on rrank		
	m	n	rank	brank	ProjLP	L-PCA	R-SVD
ABSTRACTS	12841	4894	4893	–	449 (437h)	–	4421 (9h)
APJ	2044	1164	455	453	29 (151s)	9 (109min)	443 (35s)
DBLP	19	6980	19	19	12 (46s)	11 (77min)	19 (2s)
DIALECT	1334	506	506	–	91 (527s)	78 (54h)	445 (17s)
PALEO	124	139	123	–	26 (10s)	13 (271s)	68 (1s)

are 19, while the rounding rank is no more than 11. In most cases, the rounding rank is about an order of magnitude smaller than the real rank. This shows that the expressive power of the methods significantly increases by applying the rounding.

Minimum-error decompositions. The relative reconstruction errors for the real-world datasets together with running times are presented in Tab. II. Again, L-PCA is often—but not always—the best method, especially with higher ranks. Again, the running time was high though. An exception to this is the ABSTRACTS data, where L-PCA is in fact faster than ProjLP (although it is still extremely slow). Again, ProjLP is often the second-best, and more consistently so with higher ranks.

D. Nestedness

Here we studied the possibility to use the non-negative rounding rank-1 decomposition to solve the BMNA problem. For these purposes, we generated nested matrices, perturbed them with noise, and tried to find the closest nested matrix using MT, NNRR1, their combination MT+NNR1, and R-SVD. All nested matrices were 200-by-300 and we varied the density of the data (from 0.1 to 0.7 with steps of 0.1) and the noise level (from 0.05 to 0.5 with steps of 0.05). A default density of $\mu = 0.5$ was used when the noise was varied, and noise level $p = 0.15$ was used when the density was varied.

Our results are shown in Fig. 4. MT and NNRR1 produced similar results, with MT being slightly better. The combined

MT+NNR1 is no better than MT, and R-SVD is significantly worse. In the running times, though, we see that MT takes much more time than the other approaches.

VI. CONCLUSIONS

Rounding rank is a natural way to characterize the commonly-applied rounding procedure. Rounding rank has some significant differences to real rank: for example, restricting the factor matrices to be non-negative has almost no consequences to rounding rank. Rounding rank provides a robust definition of an intrinsic dimension of a data, and as we saw in the experiments, real-world data sets can have surprisingly small rounding ranks. At the same time, rounding rank-related problems appear naturally in various different fields of data analysis; for example, the connection to nested matrices is somewhat surprising, and allowed us to develop new algorithms for the BMNA problem.

Unfortunately, computing the rounding rank, and the related minimum-error decomposition, is computationally very hard. We have studied a number of algorithms—based on common algorithm design paradigms in data mining—in order to understand how well they behave in our problems. None of the tested algorithms emerges as a clear winner, though.

The most obvious future research direction is to find better algorithms that aim directly for good rounding rank decompositions and scale to larger data sizes. Another question is if the factors obtained by a rounding rank decomposition reveal *interpretable* insights into the data. The connections of rounding rank to other problems also propose natural follow-up questions. For example, communities in graphs are often nested (sub-)matrices [26]. Could rounding rank decompositions be used to find non-clique-like communities?

REFERENCES

- [1] D. Achlioptas. Database-friendly random projections: Johnson–Lindenstrauss with binary coins. *J. Comput. System Sci.*, 66(4):671–687, 2003.
- [2] N. Alon, P. Frankl, and V. Rodl. Geometrical realization of set systems and probabilistic communication complexity. In *FOCS*, pages 277–280, 1985.
- [3] N. Alon, D. Haussler, and E. Welzl. Partitioning and geometric embedding of range spaces of finite Vapnik–Chervonenkis dimension. In *SoCG*, pages 331–340, 1987.
- [4] N. Alon, S. Moran, and A. Yehudayoff. Sign rank versus VC dimension. In *COLT*, pages 47–80, 2016.
- [5] R. Basri, P. Felzenszwalb, R. Girshick, D. Jacobs, and C. Klivans. Visibility constraints on features of 3D objects. In *CVPR*, pages 1231–1238, 2009.

Table II
RECONSTRUCTION ERRORS RELATIVE TO THE NUMBER OF NON-ZEROS AND RUNNING TIMES IN REAL-WORLD DATA.

$k =$	ABSTRACTS			APJ			DBLP			DIALECT			PALEO			
	10	50	100	5	10	15	5	10	15	10	30	50	5	10	20	
<i>Relative reconstruction error</i>																
ProjLP	1.152	1.091	0.842	0.626	0.302	0.099	0.408	0.060	0.003	0.378	0.130	0.036	0.701	0.360	0.037	
L-PCA	0.993	0.863	0.459	0.631	0.194	0.034	0.150	0.003	0.000	0.200	0.031	0.002	0.552	0.089	0.000	
R-SVD	0.995	0.937	0.843	0.641	0.611	0.573	0.488	0.225	0.058	0.258	0.137	0.094	0.697	0.516	0.260	
T-SVD	0.917	0.838	0.766	0.640	0.596	0.559	0.382	0.198	0.064	0.212	0.120	0.089	0.515	0.410	0.283	
Asso	0.988	0.971	0.960	0.663	0.637	0.603	0.531	0.347	0.187	0.442	0.358	0.333	0.793	0.706	0.602	
<i>Running time (seconds)</i>																
ProjLP	9627	27201	72956	54	64	63	12	13	12	26	83	169	2	2	2	
L-PCA	675	17849	39954	39	235	297	98	114	109	96	166	218	11	12	17	
R-SVD	3	6	13	1	1	1	1	1	1	1	1	1	1	1	1	
T-SVD	2	6	13	1	1	1	1	1	1	1	1	1	1	1	1	
Asso	2366	11701	23023	8	17	23	29	52	75	49	145	238	1	1	1	

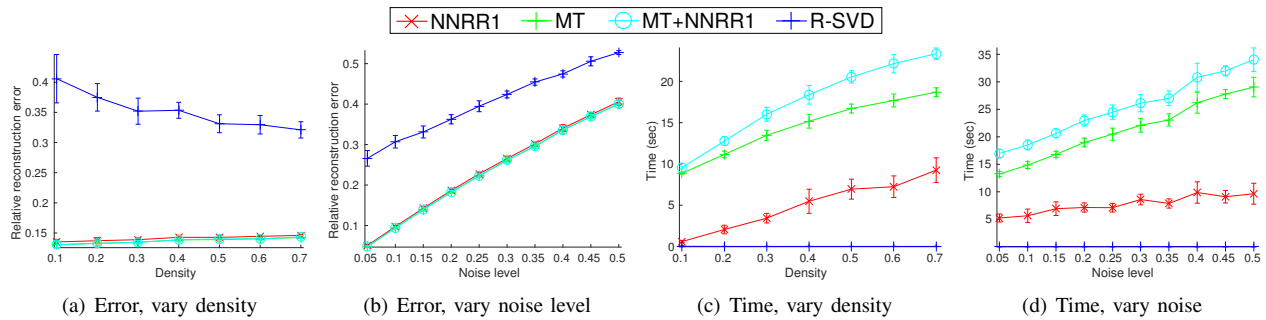


Figure 4. Relative reconstruction errors and running times on nested data. The running times for MT+NNRR1 exclude the running time of the MT algorithm. All data points are averages over 10 random matrices and the width of the error bars is twice the standard deviation.

- [6] L. B. Beasley and T. J. Laffey. Real rank versus nonnegative rank. *Linear Algebra Appl.*, 431(12):2330 – 2335, 2009.
- [7] R. Bělohávek and M. Trnečka. From-below approximations in Boolean matrix factorization: Geometry and new algorithm. *J. Comput. Syst. Sci.*, 81(8):1678–1697, Dec. 2015.
- [8] S. Ben-David, N. Eiron, and H. U. Simon. Limitations of learning via embeddings in Euclidean half spaces. *JMLR*, 3:441–461, 2003.
- [9] A. Bhangale and S. Kopparty. The complexity of computing the minimum rank of a sign pattern matrix. *CoRR*, abs/1503.04486, 2015.
- [10] S. Boyd and L. Vandenberghe. *Convex Optimization*. CUP, 2004.
- [11] S. Embleton and E. S. Wheeler. Finnish dialect atlas for quantitative studies. *J. Quant. Linguist.*, 4(1-3):99–102, 1997.
- [12] S. M. Embleton and E. S. Wheeler. Computerized dialect atlas of Finnish: Dealing with ambiguity. *J. Quant. Linguist.*, 7(3):227–231, 2000.
- [13] A. Ene, W. Horne, N. Milosavljevic, P. Rao, R. Schreiber, and R. E. Tarjan. Fast exact and heuristic methods for role minimization problems. In *SACMAT*, pages 1–10, 2008.
- [14] D. Erdős, R. Gemulla, and E. Terzi. Reconstructing graphs from neighborhood data. *Trans. Knowl. Discov. Data*, 8(4):23:1–23:22, 2014.
- [15] C. M. Fiduccia, E. R. Scheinerman, A. Trenk, and J. S. Zito. Dot product representations of graphs. *Discrete Math.*, 181(1–3):113–138, 1998.
- [16] J. Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *J. Comput. System Sci.*, 65(4):612–625, 2002.
- [17] J. Forster and H. U. Simon. On the smallest possible dimension and the largest possible margin of linear arrangements representing given concept classes. *Theor. Comput. Sci.*, 350(1):40 – 48, 2006.
- [18] M. Fortelius. New and old worlds database of fossil mammals (NOW). Online. <http://www.helsinki.fi/science/now/>, 2003. Accessed: 2015-09-23.
- [19] M. Fortelius, A. Gionis, J. Jernvall, and H. Mannila. Spectral ordering and biochronology of European fossil mammals. *Paleobiology*, 32(2):206–214, 2006.
- [20] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, 2014.
- [21] R. A. Horn and C. R. Johnson. *Matrix Analysis*. CUP, 2013.
- [22] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemp. Math.*, 26(189-206):189–206, 1984.
- [23] E. Junttila. *Patterns in permuted binary matrices*. PhD thesis, University of Helsinki, 2011.
- [24] R. J. Kang and T. Müller. Sphere and dot product representations of graphs. *Discrete Comput. Geom.*, 47(3):548–568, 2012.
- [25] H. Mannila and E. Terzi. Nestedness and segmented nestedness. In *KDD*, pages 480–489, 2007.
- [26] S. Metzler, S. Günnemann, and P. Miettinen. Hyperbolae are no hyperbole: Modelling communities that are not cliques. In *ICDM*, 2016.
- [27] P. Miettinen, T. Mielikäinen, A. Gionis, G. Das, and H. Mannila. The Discrete Basis Problem. *IEEE Trans. Knowl. Data Eng.*, 20(10):1348–1362, Oct. 2008.
- [28] S. D. Monson, N. J. Pullman, and R. Rees. A survey of clique and biclique coverings and factorizations of (0; 1)-matrices. *Bull. ICA*, 14:17–86, 1995.
- [29] S. Neumann, R. Gemulla, and P. Miettinen. What you will gain by rounding: Theory and algorithms for rounding rank. Technical Report <https://arxiv.org/abs/1609.05034>, 2016.
- [30] R. Paturi and J. Simon. Probabilistic communication complexity. *J. Comput. System Sci.*, 33(1):106–123, 1986.
- [31] J. Reiterman, V. Rödl, and E. Šiňajová. On embedding of graphs into euclidean spaces of small dimension. *J. Comb. Theory Ser. B*, 56(1):1–8, 1992.
- [32] A. I. Schein, L. K. Saul, and L. H. Ungar. A Generalized Linear Model for Principal Component Analysis of Binary Data. In *AISTATS*, 2003.
- [33] N. Tatti, T. Mielikäinen, A. Gionis, and H. Mannila. What is the Dimension of Your Binary Data? In *ICDM*, pages 603–612, 2006.
- [34] S. J. Young, E. R. Scheinerman, and F. R. K. Chung. Random dot product graph models for social networks. In *WAW*, pages 138–149, 2007.