

Using rank information to estimate missing values in data matrices

Tacildo de Souza Araújo

Department of Applied Mathematics, State University of Campinas
Cidade Universitária Zeferino Vaz - Barão Geraldo, Campinas - SP, 13083-970
tacildo.araujo@ifam.edu.br

Cristiano Torezzan

School of Applied Sciences, State University of Campinas
Rua Pedro Zaccaria, 1300, Limeira - SP, 13484-350
torezzan@unicamp.br

RESUMO

Um problema frequente em muitas aplicações que envolvem matrizes de dados é a falta, ou a imprecisão, de alguns de seus elementos. Nestes casos, é desejável dispor de alguma técnica de pré-processamento capaz de fornecer estimativas, ou boas aproximações, para os dados faltantes. Este problema pode ser modelado como um problema de otimização, sob a hipótese de que há algum tipo de redundância de informação na matriz de dados, de forma que o seu posto é muito menor do que a sua dimensão. Neste trabalho mostramos que o conhecimento prévio do posto da matriz resultante pode ser utilizado para calcular o parâmetro regularizador de métodos baseados em *Soft-Impute*, o que permite estimar, de forma precisa, dados faltantes em matrizes. Com base em propriedades de algoritmos proximais, uma análise de convergência é discutida e experimentos numéricos mostram que o método proposto supera os principais algoritmos do estado da arte. Para finalizar, como exemplo de aplicação, o método é utilizado para a recuperação de imagens ruidosas.

PALAVRAS CHAVE. Completamento de Matrizes, Algoritmos Proximais, Matriz de Posto Fixo.

Tópicos: Inteligência Computacional, Big Data e Analytics, Metaheurísticas

ABSTRACT

A frequent problem in many applications involving data matrices is the lack, or inaccuracy, of some of its elements. In these cases, it is desirable to have some pre-processing technique to estimate the missing data. This problem can be formulated as an optimization problem, under the assumption that there is some information redundancy in the data matrix, so that its rank is much smaller than its dimension. In this paper we show that the rank of the target matrix can be used to calculate the regularizing parameter of *Soft-Impute* methods, which allows to accurately estimate missing data in matrices. Based on properties of proximal algorithms, a convergence analysis is discussed and numerical experiments show that the proposed method outperforms the main state-of-the-art algorithms. Finally, as an application example, the method is used to recover noisy images.

KEYWORDS. Matrix Completion. Proximal algorithms. Fixed-rank matrix.

Paper topics: Computational Intelligence; Big Data and Analytics, Metaheuristics

1. Introduction

Suppose we are interested in a data matrix but only a subset of its entries is available. Is it possible to estimate the missing values? How accurate can these estimates be? Those questions are quite often in many applied situations and have attracted the attention of statisticians, mathematics, engineers and practitioners Nguyen et al. [2019]. For example, in Duarte et al. [2020] the problem of missing data is considered in the context of Data Envelopment Analysis, where the absence of some input or output data for a subset of DMUs may demand some preprocessing procedure. One very famous case involving matrix completion was the “Netflix Prize”, where the purpose was to predict user ratings for films based on previous ratings without any other information. The challenge encompassed a matrix with about 8.6 billion entries, which only 1.2% was known Bennett et al. [2007].

Obviously, it is not possible to retrieve *any matrix*¹ having only a subset of its entries, unless there is some additional structure to be explored, or precisely delimited. However, in many applications it is reasonable to suppose that there are some relationships between the elements of a data matrix. For instance, in the Netflix Prize, it is expected that a same user tends to rate films of a given category (or director) with similar values. In these cases, the relevant information can be, somehow, projected onto a vector subspace whose dimension is much lower than the size of the matrix. This is a fundamental assumption in areas such as big data and has been shown to be valid in many applications involving missing data. Such matrices are referred to be *low-rank* matrices.

Thus, under this assumption, the problem of recovering a matrix A can be formulated by the rank minimization problem

$$\begin{aligned} \min_X \text{rank}(X) \\ \text{subject to } X_{ij} = A_{ij}, \quad (i, j) \in \Omega, \end{aligned} \quad (1)$$

However, from a practical point of view, the formulation (1) is not very convenient, since this problem has been shown to be NP-hard by Chistov e Grigor’Ev [1984] and an efficient method to solve it is not known. To overcome this computational issue, relaxed versions of the problem (1) have been proposed and several heuristic methods have obtained good results. Two widely used approaches consider Frobenius norm minimization Tanner e Wei [2016]; Lee e Bresler [2010]; Wang et al. [2014] and nuclear norm minimization Fazel [2002]; Candès e Tao [2010]; Fornasier et al. [2011].

In Mazumder et al. [2010], the authors present an alternative formulation for (1), given by

$$\min_X \frac{1}{2} \|P_\Omega(A) - P_\Omega(X)\|_F^2 + \lambda \|X\|_*. \quad (2)$$

Here λ is a regularization parameter, Ω denotes the set of known entries of A , $P_\Omega(\cdot)$ is the projection operator, as defined in (3) and $\|X\|_*$ is the nuclear norm of X , defined in Section 2.

Based on this lagrangean relaxation, Mazumder et al. [2010] propose a heuristic algorithm called *Soft-Impute* that uses successive Singular Value Decomposition (SVD) to obtain a sequence of approximations to the solution of (2).

Although Soft-Impute has proven to be efficient in many cases, two important issues concerning its practical implementation are the main objects of study in this paper: (i) the algorithm is very sensitive to the choice of a descending sequence of λ s, one for each iteration, and there is no

¹Thing about a uniform random matrix: if each element of a matrix is uniformly distributed over a non-empty interval, it is impossible to predict its exact value.

specific way to calculate such parameters; (ii) the rank information of the target matrix is despised, but it may contain valuable information.

In this paper, we propose a strategy to overcome such issues and propose a modified method for matrix completion. In particular, we assume that the rank of the target matrix is r (which is supposed to be known) to perform a partial SVD by computing, for each iteration, only the first $r + 1$ singular values/vectors of the most recent approximation of the target matrix by using the PROPACK algorithm Larsen [1998], and we set $\lambda = \sigma_{r+1}$ as the regularization parameter. In addition, we propose an acceleration that considers a parameter tuning without compromising the special “sparse plus low-rank” structure required for computing the SVD efficiently.

Our approach is inspired in a few previous works that attempted to use rank information in the context of matrix completion. In Moreira et al. [2018] it is proposed to use rank information to improve the performance of Soft-Impute for Euclidean Distance Matrix. Here, we consider a more general case and propose an approach that do not compromises the “sparse plus low-rank” structure and also provide an acceleration equation, which is parameterized with a parameter that also depends on the rank and provide a significant improvement on the convergence rate of the algorithm. In addition, we also discuss a convergence analysis.

The rest of the paper is organized as follows. In Section 2, we set the notation and present some previous results and the background. Section 3 presents the proposed algorithm and a convergence analysis is discussed from proximal operator properties. Section 4 presents numerical experiments and comparisons with other algorithms. As an example of application, the method is applied to recover noisy images. Finally, we bring our conclusions and final remarks in Section 5.

2. Background and notations

In this section we set the notation and also present some definitions and previous results that will be used throughout the paper.

From now on, we use A^T to denote the transpose of a matrix A . The notation $\|A\|_F$ refers to the Frobenius norm of A , which is defined as $\|A\|_F = \sqrt{\left(\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2\right)}$. When necessary, the gradient of a function f will be denoted by ∇f .

The Singular Value Decomposition

Let $A \in \mathbb{R}^{m \times n}$ be a matrix of rank r , the *Singular Value Decomposition* (SVD) of A consists in the factorization of A into a product of three matrices $A = U\Sigma V^T$, where $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{r \times n}$ are matrices with orthonormal columns, $\Sigma \in \mathbb{R}^{m \times n}$ is a rectangular diagonal matrix, $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r, 0, \dots, 0)$ and the singular values σ_i are non-negative and satisfies $\sigma_i \geq \sigma_{i+1}$, $i = 1, 2, \dots, r - 1$.

The Nuclear Norm of a matrix A is defined by $\|A\|_* = \sum_{i=1}^r \sigma_i$.

The Projection Operator

Let $\Omega^\perp$ be the complement of the set Ω . Given a matrix A , the projection operator $P_\Omega(\cdot)$ is defined as

$$[P_\Omega(A)]_{ij} = \begin{cases} X_{ij} & \text{if } (i, j) \in \Omega \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

with $P_\Omega^\perp(\cdot)$ is defined by $P_\Omega^\perp(A) = A - P_\Omega(A)$

Proximal Algorithms

Consider the problem of minimizing a sum of two convex functions, $f(x) = g(x) + h(x)$. Assume that g is differentiable and ∇g is Lipschitz continuous with constant $L > 0$, but h can be not necessarily smooth. A well-known method to solve this problem is to consider a sequence generated by the *Proximal Algorithm*, defined as

$$x_{k+1} = \text{prox}_{tg}(v_k) = \arg \min_x \left\{ \frac{1}{2t} \|x - v_k\|_2^2 + h(x) \right\} = \text{prox}_{t_k h} \left(x^k - t_k \nabla f(x^k) \right), \quad (4)$$

where $\text{prox}_{t_k h}(\cdot)$ is the so called proximal operator. We recall that when $h(x) \equiv 0$ the Proximal Algorithm is reduced to the gradient method.

The next Theorem establish a sufficient condition on the parameter t , in terms of the Lipschitz parameter L , in order to guarantee that the sequence generated by the Proximal Algorithm converges to the solution, x^* , of the optimization problem discussed above.

Theorem 1 (Tibshirani [2010]) *A Proximal Algorithm with fixed step size $t \leq \frac{1}{L}$ satisfies*

$$f(x^k) - f(x^*) \leq \frac{\|x^0 - x^*\|_2^2}{2tk},$$

where $k \in \mathbb{N}$ denotes the number of iterations.

The above result states that the Proximal Algorithm converges to the optimal solution with a rate of $O(1/k)$. In other words, it tells us that the number of iterations necessary to obtain an ϵ -optimal solution $\bar{x}$, such that $f(\bar{x}) - f(x^*) \leq \epsilon$, is at most $\lceil C/\epsilon \rceil$, with $C = \frac{\|x^0 - x^*\|_2^2}{2t}$.

According to Tibshirani [2010], in the context of matrices the proximal operator can be defined as

$$\text{prox}_\tau(M) = \arg \min_X \left\{ \frac{1}{2\tau} \|M - X\|_F^2 + \lambda \|X\|_* \right\}.$$

The following Lemma connect the proximal operator of a matrix with the respective SVD and it is a fundamental result for all methods of matrix completion based on SVD.

Lemma 1 (Cai et al. [2010]) *Let $\lambda \geq 0$ and $U\Sigma V^T$ the SVD of the matrix $M \in \mathbb{R}^{m \times n}$. The solution to the optimization problem*

$$\min_X \frac{1}{2} \|M - X\|_F^2 + \lambda \|X\|_*,$$

is given by $S_\lambda(M) = U\Sigma_\lambda V^T$, where $\Sigma_\lambda = \text{diag}(\{(\sigma_i - \lambda)_+\}_{i=1}^r)$, with $t_+ = \max(t, 0)$.

In view of the previous results and the discussion provided in the Introduction, we present next the main contribution of this paper.

3. The proposed algorithm

In this section, we introduce the *Accelerated Rank-based Soft-Impute (ARSI)* and discuss its convergence analysis based on the properties of proximal algorithms presented in the previous section. The key idea behind ARSI is to perform, in each iteration, a partial SVD, truncated one level ahead of the rank of the target matrix (step 2 in Algorithm 1) and setting the regularization parameter λ as the last calculated singular value (steps 3 and 4 in Algorithm 1). In addition, we use

the accelerating equation equipped with a parameter β , which can be tuned to accelerate the method. These apparently small modifications have proven to be very valuable to improve the convergence of matrix completion. For situations where the rank of the desired matrix is not known, ARSI can still be used by performing a backtracking procedure on the rank.

3.1. The ARSI algorithm

Let A be a matrix with missing data and suppose we wish to complete the empty entries of A , such that the resulting matrix has rank r , keeping the known positions unchanged. In order to use the target rank to set the regularization parameter, we consider the following optimization problem, which is a suitable modification of (2).

$$\begin{aligned} \min_X \quad & \frac{1}{2} \|P_\Omega(A) - P_\Omega(X)\|_F^2 + \lambda \|X\|_* \\ \text{subject to} \quad & \text{rank}(X) = r \end{aligned} \quad (5)$$

Thus, we present the following algorithm, called *Accelerated Rank-based Soft-Impute* (ARSI), which uses the rank of the target matrix to set a path of solutions to (5).

Algorithm 1 Accelerated Rank-based Soft-Impute (ARSI)

Input: Matrix of missing data $A \in \mathbb{R}^{m \times n}$, rank r of the desired matrix $\tilde{A} \in \mathbb{R}^{m \times n}$, a tolerance $\epsilon > 0$, maximum number of iterations k , and $\beta \in \mathbb{N}$.

```

1: Initialize  $X^{old} = 0$  and  $B = 0$ ;
2: for  $i = 1$  to  $k$  do
3:   Calculate the truncated SVD of  $P_\Omega(A) + P_\Omega^\perp(X^{old})$  up to  $r + 1$ .
4:   Set  $\lambda = \sigma_{r+1}$ .
5:   Calculate  $X^{new} \leftarrow S_\lambda(P_\Omega(A) + P_\Omega^\perp(X^{old}))$ .
6:   if  $\frac{\|X^{old} - X^{new}\|_F}{\|X^{new}\|_F} < \epsilon$  then exit.
7:   end if
8:    $X^{old} \leftarrow X^{new} + \frac{k-1}{k+\beta}(X^{new} - B)$ .
9:    $B \leftarrow X^{new}$ .
10: end for
Output:  $X^{new}$ .

```

3.2. Convergence discussion

Algorithm 1 is a heuristic approach to solve the optimization problem defined in (5). One can note that, for each value of λ , the step 4 modifies the optimization problem given in (2). On the other hand, the algorithm preserves the characteristics of a proximal algorithm, with the difference that it changes the problem in each iteration, so it is possible to consider it as an inexact proximal algorithm.

There is a substantial amount of work that use inexact proximal algorithms for matrix completion. In Schmidt et al. [2011], it is provided a general convergence theory for inexact proximal algorithms and in Yao e Kwok [2018], it is proposed a inexact matrix completion method that uses different values of λ in each iteration, as in algorithm 1.

The above theory and considerations together with the empirical results obtained in the numerical experiments allows us to conjecture that the sequence generated by ARSI converges with

rate of convergence of $O(1/k^2)$, where k is the number of iterations. The parameter $\beta \in \mathbb{N}$ must be appropriately chosen depending on the three variables: the rank of the desired matrix r , the tolerance for relative error ϵ , and the missing data proportion p , that is, $\beta = \beta(r, \epsilon, p)$. For each triple (r, ϵ, p) the efficiency of the Algorithm 1 can be improved by more than 85% by simply adjusting the parameter β , as it can be seen in some numerical experimental described in the next section.

4. Numerical Experiments

In this section, we perform matrix completion experiments with synthetic data and on images recovery problem. The method is implemented in Matlab language and all the numerical results are performed on a PC with Intel Core i7-7500U CPU and 16 GB RAM. The experimental results are averaged over 5 repetitions.

4.1. Experiments with synthetic data

In our experiments, we generated $n \times n$ matrices of rank $r \ll n$ of the form $A = MN \in \mathbb{R}^{n \times n}$, where the entries of $M \in \mathbb{R}^{n \times r}$ and $N \in \mathbb{R}^{r \times n}$ are sampled i.i.d from the standard normal distribution. In order to simulate the missing data, we deleted, uniformly at random, a percentage p of the entries of matrix A . Thus, for performance evaluation, we use the relative error, defined by

$$er = \frac{\|A - \hat{A}\|_F}{\|A\|_F}, \quad (6)$$

where $\hat{A}$ is the recovered matrix.

Table 1 compares the performance of our algorithm, with respect to the number of iterations with Fixed-Rank Soft-Impute (FRSI) Moreira et al. [2018], Singular Value Thresholding (SVT) Cai et al. [2010], and Fixed Point Continuation (FPC) Ma et al. [2011]. The number of iterations is a suitable figure of merit for these tests because the methods have nearly the same cost by iteration. For this experiment, we set $n = 1000$, $p = 40\%$, $\epsilon = 10^{-4}$, and r varying in the range $\{10, 15, 20, 40, 80, 120, 150, 160\}$.

Tabela 1: Comparison between ARSI with FRSI, SVT, and FPC algorithms. Performance evaluation for $n = 1000$, $p = 40\%$, $\epsilon = 10^{-4}$, and r varying in the range $\{10, 15, 20, 40, 80, 120, 150, 160\}$.

rank(r)	Number of iterations			
	ARSI	FRSI	SVT	FPC
10	10	17	43	73
15	11	19	47	84
20	11	20	51	92
40	13	25	65	126
80	14	38	96	211
120	18	67	*	*
150	22	147	209	500
160	33	253	*	*

* No data available

As can be seen, our algorithm converges much faster than the other algorithms. Furthermore, the bigger is the rank r of the desired matrix the better is the performance of the ARSI when compared with FRSI (which is consistently faster than SVT and FPC), allowing us to reduce the number of iterations by up to 87%.

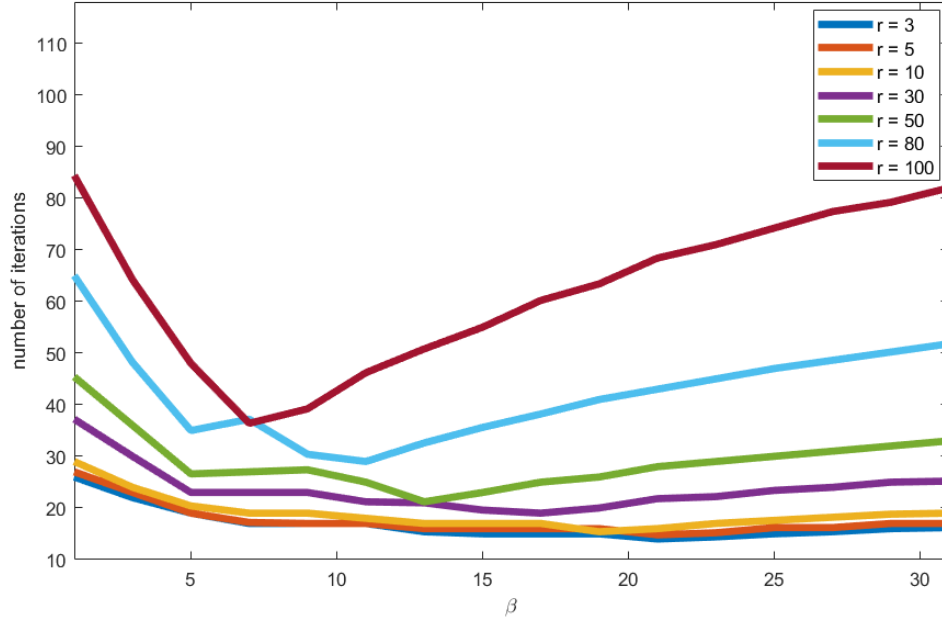


Figura 1: Optimal value for β with $n = 1000$, $r \in \{3, 5, 10, 30, 50, 80, 100\}$, $\epsilon = 10^{-5}$ and $p = 50\%$.

Empirical results show that the number of iterations of the ARSI can be highly reduced by a suitable choice of the parameter β , especially when the number of observed entries is very small. Figure 1 gives us an overview of how to set the value of β . For this experiment, we fixed $(n, \epsilon, p) = (1000, 10^{-5}, 50\%)$ and $r \in \{3, 5, 10, 30, 50, 80, 100\}$. As can be seen, the bigger is rank of the target matrix the smaller is the value of β . Extensive experiments show that the value of β decreases as the rank of the target matrix increases, regardless of the choice of the triple (n, ϵ, p) . Therefore, one can set a small value of β for a bigger rank and vice-versa.

4.2. Application on image recovery

Our algorithm is amenable to other contexts beyond the classic applications such as synthetic data and Euclidean distance matrices Moreira et al. [2018] or recommender systems Bennett et al. [2007]. In this section, we apply ARSI to a image recovery problem.

As an simple test, we use a image extracted from the website of SBPO SOBRAPÓ [2021]. For this experiment, we followed the steps: (a) the pixels of the original image were first scaled into the range between 0 and 1 (figure 2), (b) a truncated SVD is applied to reduce the rank of the original image (figure 3), (c) the truncated image has 65% of its entries randomly deleted (figure 4), and (d) we apply ARSI to the corrupted image (figure 4) to obtain the recovered image (figure 5).

As can be seen, our approach reaches a good performance in image recovery. This idea can be further explored for applications in areas such as image compressing or error correcting codes.



Figura 2: Original image

Figura 3: Truncated image



Figura 4: Corrupted image

Figura 5: Recovered image

5. Conclusion

Matrix completion aims to recover a low-rank matrix from a small number of its entries. In this paper, we set the path of solutions for the Soft-impute algorithm through truncated singular value decomposition (SVD) by assuming that we know the rank of the target matrix in advance. We propose an acceleration that improves the rate of convergence.

It is worth pointing out that the ARSI algorithm can recover matrices from a few percentage of its entries with very high accuracy in a few seconds. This result can be used for many applications. As an example, we show an application in a image recovery problem.

Remark: Part of the results contained in this paper will be presented at the 22nd *Conference of the International Federation of Operational Research Societies (IFORS 2021)* and also compose an extended version of an article to be submitted to a journal.

Referências

- Bennett, J., Lanning, S., et al. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, p. 35. Citeseer.
- Cai, J.-F., Candès, E. J., e Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982.
- Candès, E. J. e Tao, T. (2010). The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080.
- Chistov, A. L. e Grigor’Ev, D. Y. (1984). Complexity of quantifier elimination in the theory of algebraically closed fields. In *International Symposium on Mathematical Foundations of Computer Science*, p. 17–31. Springer.

- Duarte, L. T., Mussio, A. P., e Torezzan, C. (2020). Dealing with missing information in data envelopment analysis by means of low-rank matrix completion. *Annals of Operations Research*, 286(1):719–732.
- Fazel, M. (2002). *Matrix rank minimization with applications*. PhD thesis, PhD thesis, Stanford University.
- Fornasier, M., Rauhut, H., e Ward, R. (2011). Low-rank matrix recovery via iteratively reweighted least squares minimization. *SIAM Journal on Optimization*, 21(4):1614–1640.
- Larsen, R. M. (1998). Lanczos bidiagonalization with partial reorthogonalization. *DAIMI Report Series*, (537).
- Lee, K. e Bresler, Y. (2010). Admira: Atomic decomposition for minimum rank approximation. *IEEE Transactions on Information Theory*, 56(9):4402–4416.
- Ma, S., Goldfarb, D., e Chen, L. (2011). Fixed point and bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1):321–353.
- Mazumder, R., Hastie, T., e Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322.
- Moreira, N. J., Duarte, L. T., Lavor, C., e Torezzan, C. (2018). A novel low-rank matrix completion approach to estimate missing entries in euclidean distance matrix. *Computational and Applied Mathematics*, 37(4):4989–4999.
- Nguyen, L. T., Kim, J., e Shim, B. (2019). Low-rank matrix completion: A contemporary survey. *IEEE Access*, 7:94215–94237.
- Schmidt, M., Roux, N. L., e Bach, F. (2011). Convergence rates of inexact proximal-gradient methods for convex optimization. *arXiv preprint arXiv:1109.2415*.
- SOBRAPO (2021). SBPO 2021. LIII Simposio Brasileiro de Pesquisa Operacional. URL <https://sbpo2021.galoa.com.br>. Accessed in 2021-06-07.
- Tanner, J. e Wei, K. (2016). Low rank matrix completion by alternating steepest descent methods. *Applied and Computational Harmonic Analysis*, 40(2):417–429.
- Tibshirani, R. (2010). Proximal gradient descent and acceleration. *Lecture Notes*.
- Wang, Z., Lai, M.-J., Lu, Z., Fan, W., Davulcu, H., e Ye, J. (2014). Rank-one matrix pursuit for matrix completion. In *International Conference on Machine Learning*, p. 91–99. PMLR.
- Yao, Q. e Kwok, J. T. (2018). Accelerated and inexact soft-impute for large-scale matrix and tensor completion. *IEEE Transactions on Knowledge and Data Engineering*, 31(9):1665–1679.