A Heuristic for Statistical Seriation

Komal Dhull[1] Jingyan Wang[1] Nihar B. Shah[1] Yuanzhi Li[1] R. Ravi[2]

[1]School of Computer Science, Carnegie Mellon University
[2]Tepper School of Business, Carnegie Mellon University

Preliminary version

Abstract

We study the statistical seriation problem, where the goal is to estimate a matrix whose rows satisfy the same shape constraint after a permutation of the columns. This is a important classical problem, with close connections to statistical literature in permutation-based models and also has wide applications ranging from archaeology to biology. Specifically, we consider the case where the rows are monotonically increasing after an unknown permutation of the columns. Past work has shown that the least-squares estimator is optimal up to logarithmic factors, but efficient algorithms for computing the least-squares estimator remain unknown to date. We approach this important problem from a heuristic perspective. Specifically, we replace the combinatorial permutation constraint by a continuous regularization term, and then use projected gradient descent to obtain a local minimum of the non-convex objective. We show that the attained local minimum is the global minimum in certain special cases under the noiseless setting, and preserves desirable properties under the noisy setting. Simulation results reveal that our proposed algorithm outperforms prior algorithms when (1) the underlying model is more complex than simplistic parametric assumptions such as low-rankedness, or (2) the signal-to-noise ratio is high. Under partial observations, the proposed algorithm requires an initialization, and different initializations may lead to different local minima. We empirically observe that the proposed algorithm yields consistent improvement over the initialization, even though different initializations start with different levels of quality.

1 INTRODUCTION

Seriation refers to the problem of identifying a sequential ordering of the data such that “the position of each unit reflects its similarity to other units” [Marquardt, 1978]. For example, in archaeology seriation is used to identify the chronological ordering of historical artifacts (see [Marquardt, 1978] and references therein). Other applications include ecology (identifying ages of fossil sites [Mannila, 2008]), biology (discovering gene expression patterns [Caraux and Pinloche, 2004]), and operations research (understanding the interactions between organizations [McCormick et al., 1972]), just to name a few. From the statistical perspective, termed “statistical seriation”, seriation is formulated as a matrix estimation problem, where the rows of the matrix are assumed to satisfy the same shape constraint after an unknown permutation of the columns [Flammarion et al., 2019]. One common shape constraint is that the rows are monotonically increasing after the permutation of the columns, and in this paper we focus on this monotonic case. We refer the reader to the papers [Liiv, 2010, Flammarion et al., 2019] for surveys of (statistical) seriation in various applications.

Statistical seriation also forms a fundamental building block for many other problems, and ideas on solving statistical seriation may be applicable to estimation under closely-related “permutation-based” models, which involve matrices that are monotonic up to unknown permutations of rows and/or columns. Permutation-based models arise in a variety of applications including estimating pairwise comparison probabilities [Shah et al., 2017, Liu and Moitra, 2020, Mao et al., 2020], crowdsourced labeling [Shah et al., 2020], matrix completion [Shah et al., 2019], passive [Heckel et al., 2019] and active ranking [Shah and Wainwright, 2017]. A key challenge in these applications, as well as in the statistical seriation problem, is the presence of unknown permutations.

An additional application of statistical seriation is miscalibration in peer review [Wang and Shah, 2019]. This application involves a collection of reviewers and papers, where

Accepted for the 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021).

each reviewer provides ratings to their assigned subset of papers. In this context, the ratings of each reviewer is represented by a row in a matrix, and the papers represented by the columns inherit an ordering. The goal is to estimate an underlying ordering of the papers. A key challenge is that reviewers may be miscalibrated, that is, different reviewers may have different rating scales. One model for miscalibration is to assume that there exists an underlying true value for each paper, and each row of the matrix (respresenting a reviewer) is some monotonic transformation of these true values combined with noise. In such applications, one prominent benefit of the statistical seriation model is that the permutation-based assumption is general, and does not impose overly-simplistic assumptions such as the matrix being low rank or having a specific parameter-based form. Hence, the seriation model is robust in modeling a broad class of true matrices and has low bias in estimation compared to specialized models that make parameter-based assumptions.

1.1 PROBLEM FORMULATION

We now introduce the formulation of statistical seriation. Let n and d be positive integers, and let $Y \in \mathbb{R}^{n \times d}$ be a real-valued matrix. Let Π_d be the set of all permutations of size d. For any permutation $\pi \in \Pi_d$, let $\mathcal{M}_\pi \subseteq \mathbb{R}^{n \times d}$ be the set of all matrices whose columns satisfy the ordering given by π. That is, for every matrix $A \in \mathcal{M}_\pi$, we have $A_{i,\pi(1)} \leq A_{i,\pi(2)} \leq \ldots \leq A_{i,\pi(d)}$ for every $i \in [n]$. Let $\mathcal{M} := \cup_{\pi \in \Pi_d} \mathcal{M}_\pi$ denote the set of all $(n \times d)$ matrices whose columns can be permuted such that every row is non-decreasing from left-to-right after some permutation of the columns. Statistical seriation assumes that observations are made in the form of

$$Y = A^* + Z, \tag{1}$$

where we have an unknown true matrix $A^* \in \mathcal{M}$, and the unknown matrix Z is a zero-mean sub-Gaussian random matrix that represents the noise. The goal of statistical seriation is to estimate the matrix A^* (and/or the ordering $\pi^* \in \Pi_d$ associated with it). A natural estimator for this problem is the least-squares estimator [Flammarion et al., 2019]

$$\widehat{A}_{\text{LS}} \in \underset{A \in \mathcal{M}}{\operatorname{argmin}} \|A - Y\|_F^2. \tag{2}$$

The aforementioned description assumed that the matrix Y was fully observed, but this is rarely the case especially in applications such as peer grading or peer review, where each reviewer only evaluates a small subset of the items. Therefore, we also consider the setting of partial observations, where only a subset of entries $\Omega \subseteq [n] \times [d]$ in Y is observed. To this end, for any matrix $X \in \mathbb{R}^{n \times d}$, let $\|X\|_\Omega$ denote the Frobenius norm restricted to the set Ω, defined as $\|X\|_\Omega^2 = \sum_{(i,j) \in \Omega} X_{ij}^2$. Then the least-squares estimator under the case of partial observations finds the matrix within the domain $\mathcal{M}$ that best fits the observed entries:

$$\widehat{A}_{\text{LS}} \in \underset{A \in \mathcal{M}}{\operatorname{argmin}} \|A - Y\|_\Omega^2. \tag{3}$$

The least-squares estimators (2) and (3) have desirable statistical properties. When the noise is i.i.d. normal, then they correspond to the maximum likelihood estimator (MLE). Furthermore, Flammarion et al. [2019] shows that the least-squares estimator (2) is optimal up to logarithmic factors and adapts to matrices with a certain natural structure. However, despite the generality of the seriation model and the strong theoretical guarantees of the least-squares estimator, the unknown permutation π in (2) imposes computational challenges in solving (2) efficiently. If the permutation π were known, then A can be solved by isotonic regression taking $O(nd)$ time [Barlow et al., 1972]. However, in (2) the permutation π is unknown, and naively brute-forcing all possible choices of π takes exponential time in d. Computationally efficient algorithms for computing (2) are not known to date [Flammarion et al., 2019]. Moreover, no algorithms have been found that are both efficient and statistically optimal (whether using the least-squares formulation (2) or not), showing an unclosed statistical-computation gap for the statistical seriation problem.

1.2 OUR CONTRIBUTIONS

In this section, we outline the main contributions of this paper and summarize our results.

Approach: A Heuristic Approximation The goal of our work is to provide a practical algorithm that heuristically approximates the solution to (3). Specifically, we approach the problem by replacing the combinatorial permutation constraint in (3) by a continuous regularization term while still capturing the permutation constraint. Formally, we define the following objective function $L : \mathbb{R}^{n \times d} \to \mathbb{R}$, parameterized by a tuning parameter $\lambda \geq 0$:

$$L(A) = L_{Y,\Omega,\lambda}(A) := \|A - Y\|_\Omega^2 + \lambda R(A). \tag{4}$$

where $R : \mathbb{R}^{n \times d} \to \mathbb{R}^{\geq 0}$ is a carefully-designed regularizer term to be explained in Section 2. Then our solution is computed by minimizing the objective as

$$\underset{A \in [0,1]^{n \times d}}{\operatorname{argmin}} \; L(A). \tag{5}$$

Following Shah et al. [2017], we assume Bernoulli noise Z in (1), and therefore restrict the domain of optimization (5) to $[0, 1]^{n \times d}$. Now that the objective is continuous and the domain is a closed bounded set, we use projected gradient descent to obtain a local minimum of this non-convex objective. Our approach is quite different from past work – past work has primarily focused on designing efficient algorithms

Preliminary Version

that reduce the gap from the optimal estimator in terms of the statistical rates. On the other hand, we directly provide a heuristic for approximating the optimal estimator. We thus provide a new point of comparison in terms of the statistical and computational trade-off. Our approach thus provides new insights in terms of possible research directions to understand and address this statistical-computational gap.

Theoretical results We first theoretically analyze the stationary points of (5), and show that projected gradient descent converges to a stationary point (Section 4). Specifically, the attained stationary point recovers the exact input data in the noiseless case (Theorem 2) and has other desirable theoretical properties in the noisy case (Proposition 3 and Theorem 4). These theoretical results hold generally for any $\lambda \geq 0$. The theoretical results thus provide insights into our approach (5) to approximating statistical seriation, and provide justification for its validity.

Simulation results We then empirically evaluate our algorithm by simulation. Specifically, we examine the following aspects:

- **Accuracy-computational tradeoff of λ** We first observe that the tuning parameter λ induces an accuracy-computational tradeoff (Section 5.2.1). Specifically, when the value of λ increases, estimation achieves higher accuracy but gradient descent takes more iterations to converge.
- **Advantage under non-parametric models and high SNR** We then compare our estimator with various baselines under various models (Section 5.2.2). We observe that our estimator performs well when the true data violates simplistic parametric assumptions. This is because our estimator inherits the general formulation of statistical seriation, giving low bias in estimation. On the other hand, although the parametric baselines perform well when the true data is generated from such parametric models, they incur a large bias when the true data is not. In addition, our estimator especially performs well when the SNR is high. This is also expected, as noise is of low-rank in nature. Therefore, when the signal level relative to the noise is low, the noise overshadows the non-parametric structure of the true matrix.
- **Partial observations and initialization of gradient descent** Finally, we consider the case when the data is only partially observed (Section 5.3). In this case, the gradient descent algorithm requires an initialization on the unobserved entries of the matrix. Since the objective (4) is non-convex, gradient descent may converge to different local optima based on the initialization. We empirically observe that our algorithm consistently improves the estimation accuracy for different choices of initialization, although the amounts of error at the beginning of gradient descent are different for different initializations.

Putting the theoretical and empirical results together, our work demonstrates the effectiveness of our approach to approximating the solution of the least-squares estimator, and the generality of the approach inherited by the generality of the seriation model.

2 OUR PROPOSED ALGORITHM

We propose the following regularizer R for the objective (4):

$$R(A) = \sum_{i,i' \in [n] j,j' \in [d]} R_{i,i',j,j'}(A), \tag{6}$$

where $R_{ii'jj'}(A)$ is defined as

$$R_{i,i',j,j'}(A) := \begin{cases} 0 & \text{if } (A_{ij} - A_{ij'})(A_{i'j} - A_{i'j'}) \geq 0 \\ (A_{ij} - A_{ij'})^2 (A_{i'j} - A_{i'j'})^2 & \text{otherwise.} \end{cases} \tag{7}$$

The goal of the regularizer R is to capture the permutation constraint of the matrix. The main challenge with the constraint is that the permutation is unknown. In (7), we consider the four matrix entries in rows $\{i, i'\} \subseteq [n]$ and columns $\{j, j'\} \subseteq [d]$ of the matrix. We call these four entries as the "quadruple" (i, i', j, j'). We observe that $A \in \mathcal{M}$ if and only if the terms $(A_{ij} - A_{ij'})$ and $(A_{ij} - A_{ij'})$ have the same sign (or one or both of the terms equal 0) for all the quadruples in the matrix (including quadruples where some or all of the four entries are unobserved). Hence, the regularizer $R_{ii'jj'}$ is designed to penalize the difference in the sign between the pairs of terms $(A_{ij} - A_{ij'})$ and $(A_{i'j} - A_{i'j'})$. The quadratic form (7) of $R_{ii'jj'}$ can be viewed as a differentiable approximation to the step function $\mathbb{1}\{(A_{ij} - A_{ij'})(A_{i'j} - A_{i'j'}) < 0\}$. Finally, the regularizer R takes a summation over all the quadruples (i, i', j, j'). It can be verified that we have $A \in \mathcal{M}$ if and only if $R(A) = 0$.

Putting (4), (5) and (6) together, our estimator is defined as

$$\operatorname*{argmin}_{A \in [0,1]^{n \times d}} \|A - Y\|_\Omega^2 + \lambda \sum_{i,i' \in [n], j,j' \in [d]} R_{ii'jj'}(A), \tag{8}$$

where ties are broken arbitrarily. Equivalently, our estimator can be viewed as first reformulating the original problem (3) to an equivalent problem:

$$\operatorname*{argmin}_{\substack{A \in [0,1]^{n \times d} \\ R_{ii'jj'}(A) = 0 \quad \forall i,i' \in [n], j,j' \in [d]}} \|A - Y\|_\Omega^2. \tag{9}$$

Then optimization (8) can be considered as the Lagrangian of the optimization problem (9). Intuitively, a large value of λ corresponds to stricter enforcement of the permutation structure on the matrix A.

Preliminary Version

To solve (8) we use projected gradient descent. The projected gradient descent algorithm consists of two steps in each iteration. In the gradient step, the algorithm updates its current estimate by computing gradient of the objective and moving the current estimate in its objective-improving direction for a stepsize. In the projection step, the algorithm projects the current estimate back to the domain $[0,1]^{n \times d}$. Formally, we denote $\gamma_t \in \mathbb{R}$ as the stepsize in each iteration $t \geq 1$. We have

$$\text{Gradient step: } A_t = A_{t-1} - \gamma_t \nabla L_A(A). \tag{10a}$$

$$\text{Projection step: } \begin{array}{l} A_t \leftarrow \max\{0, A_t\}, \\ A_t \leftarrow \min\{1, A_t\}. \end{array} \tag{10b}$$

Note that we choose a quadratic form in (7) instead of a linear form such as the hinge loss, because the quadratic form is differentiable, and hence its gradient can be computed straightforwardly.

3 RELATED WORK

Seriation and estimation under monotonicity Flammarion et al. [2019] proposes the statistical model for seriation, and then shows that the least-squares estimator (2) is optimal up to logarithmic factors when the underlying constraint is either monotonic or unimodal. More generally, there is a rich line of literature on estimation under permutation constraints, where the data obeys certain underlying orderings, but the orderings are unknown. For example, Mao et al. [2020] consider the class of bivariate isotonic matrices, where the matrix follows an unknown row permutation and an unknown column permutation, and a subclass where one of the two permutations is known. Shah et al. [2017] analyze the class of stochastic transitivity (SST) matrices, which are bivariate isotonic matrices that are (shifted) skew-symmetric. A multivariate generalization is considered in Pananjady and Samworth [2020]. For such problems, the least-squares estimators are considered (e.g., [Shah et al., 2017, Flammarion et al., 2019, Shah et al., 2020]). However, efficient algorithms for computing such least-squares estimators are not known [Flammarion et al., 2019, Mao et al., 2020, Liu and Moitra, 2020]. Due to the computational inefficency of the least-squares estimator, other computational efficient estimators are proposed [Flammarion et al., 2019, Mao et al., 2020, Liu and Moitra, 2020]. Many of these efficient estimators are statistically suboptimal, with the exception of Liu and Moitra [2020] and Pananjady and Samworth [2020]. Specifically, Liu and Moitra [2020] considers bivariate isotonic matrix estimation where one of the two permutations is known, and proposes an estimator that runs in linear time achieving the optimal rate up to an $n^{o(1)}$ factor. Pananjady and Samworth [2020] proposes an estimator that is optimal when the dimension of the problem is $d \geq 3$ (but not $d = 2$). For statistical seriation, positive or negative results on efficient estimators achieving the optimal rate remains unknown [Flammarion et al., 2019].

Landscape design and properties of local optima Optimization-based approaches are widely used for many problems, where the solution is posed as the minimizer to an objective function and computed by standard techniques such as gradient descent. The objective often includes regularization terms. Designing proper regularization (also termed “landscape design”) that has desirable properties has been considered problems such as low-rank approximation [Ge et al., 2016] and neural networks [Ge et al., 2018]. In particular, Ge et al. [2016] considers low-rank approximation under a random design setting and proves that all local minima are global minima. Ma et al. [2018] considers a specific crowdsourced labeling setting with a rank-1 (Dawid-Skene) model, and shows that under arbitrary fixed design, all local minima are global minima for rank-1 matrix completion [Ma et al., 2018]. These theoretical results suggest that gradient descent converges to the global optimum for their problems. Ma et al. [2018] further proposes an exponentiated gradient descent algorithm to achieve polynomial-rate convergence. Since a rank-1 matrix is monotonic by definition (where the permutation is unknown), our theoretical results (Section 4) can be considered as a generalized setting of Ma et al. [2018]. Our idea of using projected gradient descent is also inspired by Ma et al. [2018].

On using regularization for permutation constraints, Tibshirani et al. [2011] proposes a regularizer to captures the permutation constraint in isotonic regression, where the permutation is known. On the other hand, we consider the case where the permutation is unknown.

Data imputation In the partial observation setting, our algorithm starts with an initialization. This initialization is related to data imputation, which is used in domains such as clustering. Methods such as naively taking the mean, nearest-neighbor (NN) [Beretta and Santaniello, 2016] and MICE [Azur et al., 2011] are proposed. In the simulation results, we consider initializing the missing data by the mean and the nearest-neighbor methods.

4 THEORETICAL PROPERTIES

In this section, we present theoretical properties of our algorithm. Specifically, we analyze the stationary points of the non-convex objective (8). We show desirable properties of any stationary point under the noiseless and the noisy settings. These results provide theoretical backing that the regularized objective proposed in (8) provides a natural approach to approximating the solution of (2).

The following result connects stationary points and gradient descent, stating that the gradient of the iterates obtained by projected gradient descent converges to 0.

Theorem 1. *Consider any matrix $Y \in [0,1]^{n \times d}$, any non-*

Preliminary version

empty observation set $\Omega \subseteq [n] \times [d]$, and any value of the parameter $\lambda \geq 0$. With any initialization, the gradient of the iterates given by projected gradient descent on objective (8) *converges to 0. Specifically, with a proper choice of a constant stepsize (dependent on n, d and λ), for any $\epsilon > 0$, the solution of projected gradient descent satisfies $\lim_{t\to\infty} \|\nabla L(\widehat{A}_t)\|_F^2 < \epsilon$.*

The proof of this theorem is provided in Appendix B. In what follows, we present properties of the stationary points of the objective (8). Note that the objective (8) is continuous and over a closed bounded set (that is, $[0,1]^{n\times d}$). Therefore, there always exists at least one global minimum [Rudin, 1976, Theorem 4.16], and hence at least one local minimum. In Lemma 6 of Appendix A.2, we show that all local minima on the boundary of the domain $[0,1]^{n\times d}$ are stationary points, so there exists at least one stationary point.

4.1 THE NOISELESS SETTING

We first consider the noiseless setting where we have $Y \in \mathcal{M}$. Our approach is inspired by the work of Ma et al. [2018]. Specifically, Ma et al. [2018] considers rank-1 matrix completion under any fixed-design, and shows that their proposed algorithm can perfectly recover the rank-1 matrix in the noiseless case. Without a second thought, one may be tempted to write off this result – there is a straightforward algorithm to perfectly recover noiseless rank-1 matrices, that is, picking any non-zero row of the matrix, and writing each remaining row as the product of a multiplicative factor and this row. However, the theoretical results in Ma et al. [2018] still provide non-trivial theoretical contributions and useful insights – the straightforward algorithm is heavily tailored to the noiseless case, and quickly becomes inapplicable when the data deviates from being rank-1. On the contrary, the theoretical guarantees by Ma et al. [2018] are shown on a much more general algorithm with any initialization, applicable to any arbitrary matrix Y.

In our problem, under the noiseless setting, the set of global minima to (8) is the set of monotonic matrices whose entries equal to Y on the observed set Ω. The following result shows that all stationary points are global minima. Since rank-1 matrices are monotonic by definition, our result supplements the result of Theorem 2 in Ma et al. [2018] by considering general monotonic matrices in small matrix sizes.

Theorem 2. *Consider any $Y \in \mathcal{M}$, any non-empty observation set $\Omega \subseteq [n] \times [d]$ and any value of the parameter $\lambda \geq 0$. Consider $n = 2$ or $d \leq 3$. Then any stationary point to the objective* (8) *is a global minimum.*

The proof of this theorem is provided in Appendix C. The proof relies on the first-order optimality condition, and uses combinatorial arguments to derive contradictions if any stationary point were not a global minimum.

Similar to the setting in Ma et al. [2018], under the noiseless setting, there also exists a straightforward algorithm to obtain all the global minima of (8) – by first finding the total ordering of the columns (or the set of all such total orderings) induced by the entries within each row, and filling each unobserved entry to be any value subject to this total ordering. On the contrary, our algorithm is applicable to any arbitrary matrix Y. With its generality, it is even unclear if the original noiseless matrix can be recovered under any arbitrary initialization without Theorem 2. Furthermore, the property of perfectly recovering noiseless data is not only natural but also important – given the generality of the seriation model, Theorem 2 contrasts our algorithm with prior approaches in matrix estimation and completion such as using parameter-based models or low-rank matrix decomposition, where a non-zero bias is incurred in this noiseless case.

4.2 THE NOISY SETTING

Now we move to consider the noisy setting where the matrix Y is not guaranteed to be monotonic. A quadruple (i, i', j, j') is called a "disagreement quadruple" if the signs of $(A_{ij} - A_{ij'})$ and $(A_{i'j} - A_{i'j'})$ are different. The following result shows that the set of disagreement quadruples at any stationary point to (8) is a subset of the disagreement quadruples in the original matrix Y.

Proposition 3. *Consider any matrix $Y \in [0,1]^{n\times d}$, any non-empty observation set $\Omega \subseteq [n]\times[d]$ and any value of the parameter $\lambda \geq 0$. Consider $n = 2$. Let $\widehat{A}$ be any stationary point of the objective* (8)*. For every $\{i, i'\} = \{1, 2\}$ and any $j, j' \in [d]$ such that*

$$\widehat{A}_{i,j} < \widehat{A}_{i,j'} \qquad \textit{and} \qquad \widehat{A}_{i',j} > \widehat{A}_{i',j'},$$

we have the same relation holds at the corresponding entries of the matrix Y:

$$\begin{aligned} & Y_{i,j} < Y_{i,j'}, && \textit{if } (i,j),(i,j') \in \Omega \\ \textit{and} \quad & Y_{i',j} > Y_{i',j'}, && \textit{if } (i',j),(i',j') \in \Omega. \end{aligned}$$

The proof of this result is provided in Appendix D. In words, this result shows that our estimator only reduces the disagreement quadruples in the observations Y and never introduces new ones that do not exist in Y, thus revealing another natural desirable property of our estimator (8).

Using Proposition 3 as a building block, the following result considers the case where there is a partition of the columns, and there is a total ordering describing the dominance relation of these columns in the matrix Y. Specifically, a set of columns $S \subseteq [d]$ is said to "dominate" another set of columns $S' \subseteq [d]$, if we have $Y_{ij} > Y_{ij'}$, for every $i \in [n], j \in S$ and $j' \in S'$ such that $(i,j),(i,j') \in \Omega$. The following theorem shows that any stationary point to (8) retains this dominance relation.

Preliminary version

Theorem 4. *Consider any matrix $Y \in [0,1]^{n\times d}$, any non-empty observation set $\Omega \subseteq [n] \times [d]$ and any value of the parameter $\lambda \geq 0$. Consider $n = 2$. Assume there exists a partition of columns $[d] = S_1 \cup \ldots \cup S_m$, such that S_{k+1} dominates S_k for each $k \in [m-1]$. Assume that for each $k \in [m-1]$, and each $j \in S_k, j' \in S_{k+1}$, we have*

$$\exists i \in \{1, 2\} \text{ such that } (i, j), (i, j') \in \Omega. \tag{11}$$

Then at any stationary point $\widehat{A}$ to the objective (8)*, we have $\widehat{A}_{ij} < \widehat{A}_{ij'}$ for any $i \in \{1, 2\}$ and any $j \in S_k, j' \in S_{k+1}$ with any $k \in [m-1]$.*

The proof of this result is provided in Appendix E. In words, the condition (11) in Theorem 4 requires that the ordering of two columns are directly comparable. Note that in the noiseless case, we can write the partition as $[d] = \{1\} \cup \ldots \cup \{d\}$. Hence, this result is a generalization of our result from the noiseless case (Theorem 2). Proposition 3 and Theorem 4 thus together show that in the noisy setting, any stationary point to the objective (8) has desirable properties under certain special cases. These theoretical properties are natural but at the same time non-trivial, providing theoretical insights and validation to our proposed estimator (8).

5 SIMULATIONS

In this section, we evaluate the performance of gradient descent on the objective (8) in different settings[1]. We first discuss the simulation set-up for a full-observation setting ($\Omega = [n] \times [d]$)) in Section 5.1. We provide the associated results in Section 5.2. In a nutshell, our algorithm performs better than the baselines when the underlying models do not satisfy specialized parametric assumptions, and also when the signal-to-noise (SNR) is high so that the noise does not overshadow the non-parametric structure of the data. We then simulate settings with only partial observations in Section 5.3. We consider several natural methods to initialize the matrix Y and we find that our algorithm consistently improves the performance as compared to the various common initialization methods. We also find that our algorithm is quite robust to the choice of the initialization method, although the choice of initialization could in theory lead to very different local minima.

5.1 SIMULATION SETUP

We now describe the design choices made for our estimator (8) and the simulation settings.

Reparameterizing the hyperparameter λ Instead of the objective (8) that weighs the two terms by 1 and λ, we reparametrize the hyperparameter λ and now weigh the two terms by $(1-\widetilde{\lambda})$ and $\widetilde{\lambda}$ with $\widetilde{\lambda} \in [0, 1)$. That is, we consider the objective

$$\underset{A\in[0,1]^{n\times d}}{\operatorname{argmin}} \quad (1-\widetilde{\lambda}) \cdot \|Y - A\|_\Omega^2 + \widetilde{\lambda} R(A). \tag{12}$$

Note that this objective (12) is equivalent to the previous objective (8), with a one-to-one correspondence between the values of λ and $\widetilde{\lambda}$. The reparameterized objective (12) reduces the variation on the magnitude of the objective through the range $\widetilde{\lambda} \in [0, 1)$, making it easier to choose a simple constant stepsize for gradient descent independent of the specific choice of $\widetilde{\lambda}$. For all the subsequent simulation results, we consider this reparameterized objective (12).

Gradient descent For simplicity, we choose a constant stepsize of 0.1 with a momentum of 0.9. We use the initialization $A_0 = Y$ under full observations. The choice of the initialization under partial observations is further discussed in Section 5.3. We terminate the algorithm when the normalized squared Frobenius norm of the gradient is smaller than 10^{-8}, that is, when $\frac{1}{nd}\|\nabla_A \widetilde{L}(A)\|_F^2 < 10^{-8}$, where $\widetilde{L}$ denotes the reparameterized objective (12). We implement our objective (12) and run gradient descent in PyTorch [Paszke et al., 2019].

Models We follow the observation models studied in Shah et al. [2017], but with an additional parameter that controls the relative levels of signal and noise. We consider square matrices with $n = d$. Let $A^* \in [0,1]^{n\times n}$ represent the true matrix whose value is specified later for different models. Bernoulli observations Y are generated[2] from A^*, that is, we have $\mathbb{P}(Y_{ij} = 1) = A^*_{ij}$ for each $i, j \in [n]$. We use the five SST models of A^* described in Shah et al. [2017, Section 4]; we also include the descriptions below for completeness.

(a) **Uniform:** The diagonal entries are 0.5. Then $\binom{n}{2}$ values are drawn independently and uniformly at random from $[\beta, 1]$, for a fixed choice of $\beta \in [0.5, 1]$, and sorted in the increasing order. The entries immediately above the diagonal are filled with the smallest $(n-1)$ values uniformly at random. Then the entries in the next diagonal above are filled uniformly at random with the smallest $(n-2)$ of the remaining values, and so on. The entries below the diagonal are filled in to make A^* skew symmetric.

(b) **Thurstone:** A vector $w^* \in \mathbb{R}^n$ is chosen uniformly at randomly from the set of w^* such that $\langle w^*, 1\rangle = 0$ and

[1] The code for the implementation of our estimator and for evaluation is provided at `https://github.com/jingyanw/heuristic-seriation`.

[2] Note that Shah et al. [2017] only generates i.i.d. Bernoulli observations in the upper diagonal with $i \leq j$, and set the entries in the lower diagonal as $Y_{ji} = 1 - Y_{ij}$. This is because Shah et al. [2017] requires the matrix to be skew-symmetric whereas with the seriation model we do not have this restriction.

Preliminary version

all entries of w^* are between $-0.5 - \beta$ and $0.5 + \beta$, for a fixed choice of β. Then the matrix A^* is filled in via $A^*_{ij} = F(w^*_i - w^*_j)$ for each $i, j \in [n]$, where F is the CDF of the standard normal distribution.

(c) **BTL:** Identical to the Thurstone model, except that F is given by the sigmoid function.

(d) **Noisy sorting:** The diagonal entries are 0.5. All entries above the diagonal are β, and all entries below the diagonal are $1 - \beta$, for a fixed choice of $\beta \in [\frac{1}{2}, 1]$. This is a classic model proposed by Braverman and Mossel [2008] and studied subsequently in the literature (e.g., Mao et al. [2018]).

(e) **Independent bands:** The diagonal entries are 0.5. The entries immediately above the diagonal are chosen i.i.d. uniformly at random from $[\beta, 1]$, for a fixed choice of $\beta \in [0.5, 1]$. The entries in the next diagonal is chosen uniformly randomly from the range lower bounded by the entries to its left and below. The entries below the diagonal are filled in a manner that makes A^* skew symmetric.

Metrics For any estimator $\widehat{A}$, we consider its risk in terms of the normalized squared Frobenius norm, $\frac{1}{nd}\|\widehat{A} - A^*\|_F^2$.

Baselines We compare our algorithm to the following baselines:

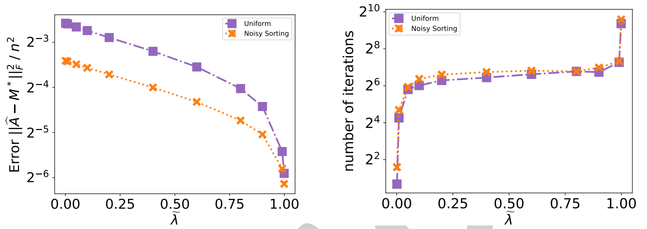
1. **Rank-1**: The estimate $\widehat{A}$ is computed as the rank-1 approximation of Y.
2. **Singular-value thresholding (SVT):** This estimator is studied in Shah et al. [2017, Section 3.2] (and also in various other works such as Chatterjee [2015]), with a parameter α denoting the (soft) threshold level applied on the singular values of Y. The value of α is required to be strictly greater than $2\sqrt{n}$, and Shah et al. [2017] uses $\alpha = 2.01\sqrt{n}$. For our settings, we consistently observe that a smaller value of α gives better performance, so we set $\alpha = 2.0000001\sqrt{n}$.

5.2 RESULTS FOR FULL OBSERVATIONS

We now present the results from our simulations pertaining to the full-observation setting.

5.2.1 Accuracy-computation tradeoff induced by $\widetilde{\lambda}$

We first inspect the performance of our algorithm for different choices of $\widetilde{\lambda} \in [0, 1)$, in terms of the accuracy (measured by the Frobenius error of estimation) and the computational time (measured by the number of iterations taken till convergence of gradient descent), shown in Figure 1. We use $n = 64$ and $\beta = 0.5$ (which matches the setting in Shah et al. [2017]). The error bars in Figure 1 and all subsequent results represent the standard error of the mean, computed over 10 trials. In Figure 1 and subsequent plots, the error bars are small and therefore not visible.



(a) Estimation error as a function of $\widetilde{\lambda}$

(b) Number of iterations as a function of $\widetilde{\lambda}$ for the uniform model

Figure 1: Tradeoff between accuracy (estimation error) and time (number of iterations) for different values of $\widetilde{\lambda} \in [0, 1)$.

We observe from Figure 1 that there is a tradeoff between accuracy and the computational time. As the value of $\widetilde{\lambda}$ increases, our algorithm attains a lower error (Figure 1(a)), but takes more time (Figure 1(b)). This tradeoff is expected, because the original least-square estimator intuitively corresponds to setting $\widetilde{\lambda} = 1$, which is known to be optimal in estimation and conjectured computationally inefficient. On the other hand, setting $\widetilde{\lambda} = 0$ is equivalent to outputting the observation matrix Y without any computation. For clarity, only a few models are shown in Figure 1, but we consistently observe these trends for numerous settings not shown.

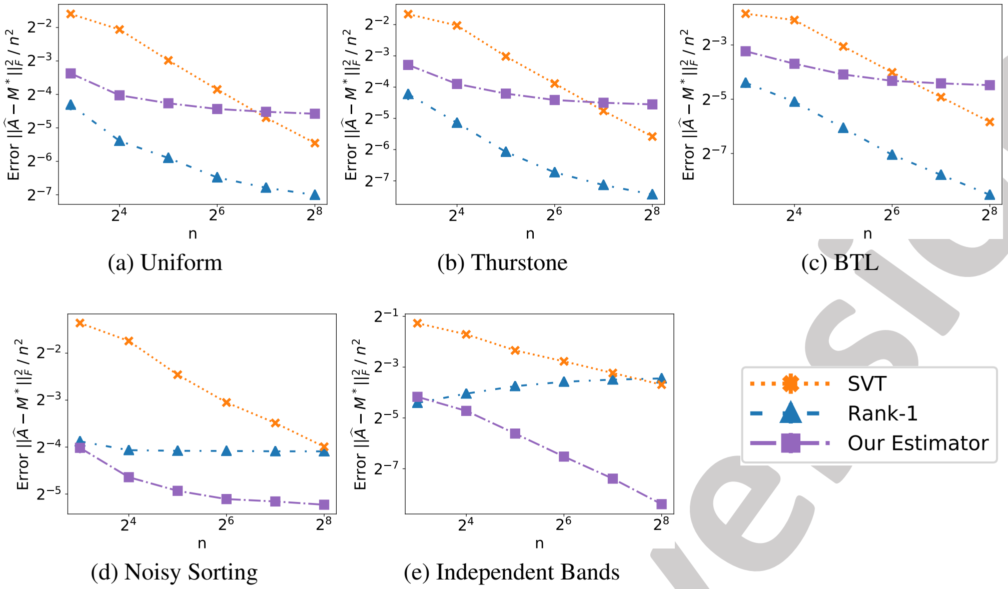
Consequently, for all subsequent simulations we set $\widetilde{\lambda} = 0.9$, which is a reasonably large value that attains low error without excessively slowing down the convergence. We now provide simulation results for the 5 models under the set-up described in Section 5.1.

5.2.2 Comparison to baselines

We run simulations comparing the performance of our algorithm with the baselines on the aforementioned models in two ways: varying the matrix size n (for fixed $\beta = 0.5$) and varying the signal relative to noise, β (for fixed $n = 64$). The results are shown in Figure 2 and Figure 3, respectively. The key findings from these simulations are as follows:

- The baselines work well when the underlying model is parametric or similar (Figure 2(b)(c)), but are inconsistent when such parametric assumptions do not hold (Figure 2(d)(e)). A similar observation about the Thurstone MLE is made in Shah et al. [2017]. The rank-1 estimator outperforms the (soft)-SVT estimator.
- Our estimator outperforms the baselines when the underlying model is more complex.
- When the noise level is high relative to the signal

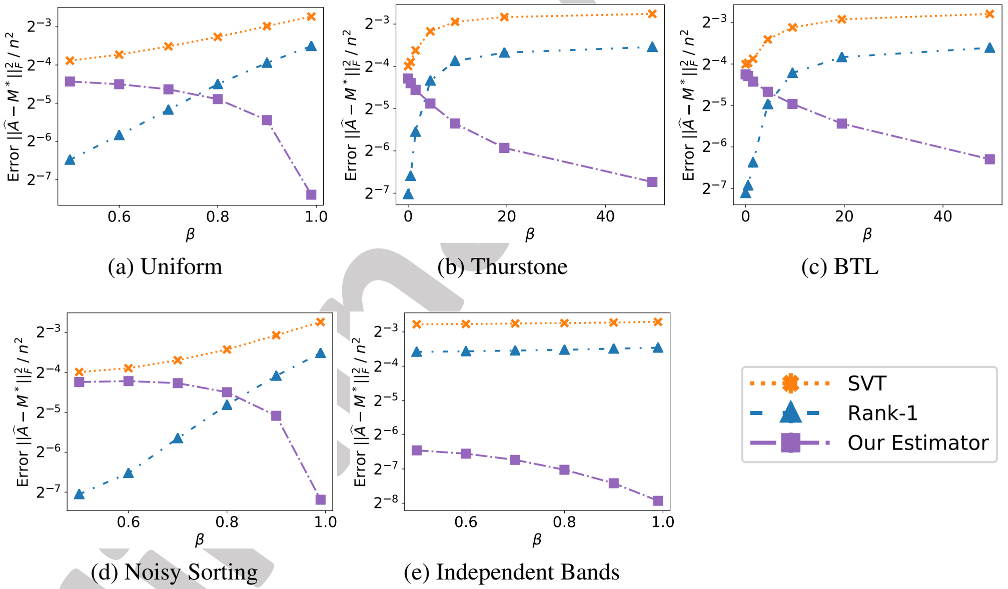
Preliminary versio



(a) Uniform (b) Thurstone (c) BTL

(d) Noisy Sorting (e) Independent Bands

Figure 2: Estimation error of different algorithms for different models of A^*.



(a) Uniform (b) Thurstone (c) BTL

(d) Noisy Sorting (e) Independent Bands

Figure 3: Estimation error of different algorithms under different levels of signal relative to noise.

(smaller values of β in Figure 3), the baselines perform well. This is because the estimation error dominates, and the baselines trim off a lot of noise.

- When the noise level is low relative to the signal (larger values of β in Figure 3), our estimator offers substantial improvements. In this regime, the approximation error is the dominating source of error, and the baselines incur a large approximation error since they also trim off a large part of the signal.

5.3 PARTIAL OBSERVATIONS

In what follows, we simulate settings where Y has missing entries, which is important in practice but has received much less attention in the literature. We consider our algorithm (8) and evaluate various initializations for gradient descent, as well as compare it to the baselines. The initialization potentially affects the performance of gradient descent, because gradient descent may converge to different local optima depending on the initialization.

Preliminary version

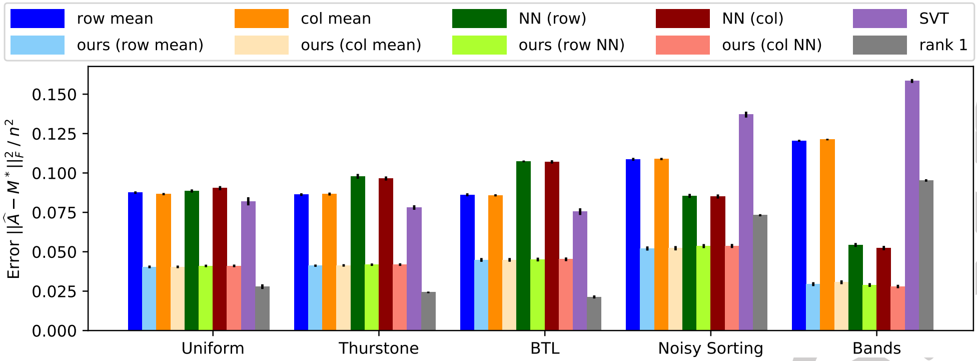


Figure 4: Performance with partial observations under different initialization methods.

5.3.1 Simulation setup

As before, we choose $n = 64$, and $\beta = 0.5$, matching the setting in Shah et al. [2017].

Random-design observations We consider a random design to construct Ω so that each matrix entry is observed with probability 0.3 independent of all else.

Initialization methods We consider the following initialization methods:

- **Row mean:** Each unobserved entry is initialized to the mean of the observed entries in its row.
- **Column mean:** Each unobserved entry is initialized to the mean of the observed entries in its column.
- **Row kNN:** Each unobserved entry is imputed as the mean of the 5 nearest rows among the rows. The distance between rows is measured in terms of the normalized Euclidean distance.
- **Column kNN:** Each unobserved entry is imputed as the mean of the 5 nearest columns among the columns. The distance between columns is measured in terms of the normalized Euclidean distance.

5.3.2 Results for partial observations

The simulation results for partial observations are shown in Figure 4, where the bars for the same initialization before and after running our algorithm are coded in a pair of similar colors. We also compare the performance of our algorithm with the baselines described earlier in Section 5.1. The figure shows the performance of each baseline with the initialization method for which it performs the best (which happens to be both row and column kNNs for both baselines). The salient findings from the simulations are as follows:

- The choice of the initialization method does not have strong influence on the performance of our algorithm.
- Our algorithm consistently improves upon different initialization methods.
- Similar to the full-observation setting, our method outperforms the baselines when the underlying model is more complex, whereas the baselines perform well when the underlying model is simpler.

6 CONCLUSIONS AND DISCUSSION

In this work, we contribute a heuristic-based perspective with respect to the spectrum of the statistical-computational gap in the statistical seriation problem. In terms of open problems, on the theory front, it is still certainly of interest to accurately characterize the statistical-computational gap. On the applied side, a wide range of applications have application-specific characteristics. For example, in peer review, reviewers' behaviors may not be entirely monotonic due to subjectivity, so that the true matrix may have only a partially monotonic structure. Our heuristic-based approach can provide a useful tool to tackle such challenges that are even more complex than the open problem of statistical seriation.

Acknowledgements

This work was supported in part by an NSF CAREER award CIF: 1942124, NSF grant CIF: 1763734, the U.S. Office of Naval Research N00014-21-1-2243, the Air Force Office of Scientific Research FA9550-20-1-0080, and a grant from the CMU Block Center for Technology and Society.

References

Melissa J. Azur, Elizabeth A. Stuart, Constantine Frangakis, and Philip J. Leaf. Multiple imputation by chained equations: What is it and how does it work? *International Journal of Methods in Psychiatric Research*, 20(1):40–49, March 2011.

Preliminary version

Preliminary version

R.E. Barlow, D.J. Bartholomew, J.M. Bremner, and H.D. Brunk. *Statistical Inference Under Order Restrictions: The Theory and Application of Isotonic Regression*. Wiley, 1972.

Lorenzo Beretta and Alessandro Santaniello. Nearest neighbor imputation algorithms: A critical evaluation. *BMC Medical Informatics and Decision Making*, 16, July 2016.

Mark Braverman and Elchanan Mossel. Noisy sorting without resampling. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 268–276. Society for Industrial and Applied Mathematics, 2008.

Gilles Caraux and Sylvie Pinloche. PermutMatrix: a graphical environment to arrange gene expression profiles in optimal linear order. *Bioinformatics*, 21(7):1280–1281, November 2004.

Sourav Chatterjee. Matrix estimation by universal singular value thresholding. *Annals of Statistics*, 43(1):177–214, 2015.

Nicolas Flammarion, Cheng Mao, and Philippe Rigollet. Optimal rates of statistical seriation. *Bernoulli*, 25(1): 623–653, February 2019.

Rong Ge, Jason D. Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, volume 29, pages 2973–2981. Curran Associates, Inc., 2016.

Rong Ge, Jason D. Lee, and Tengyu Ma. Learning one-hidden-layer neural networks with landscape design. In *6th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2018.

Reinhard Heckel, Nihar B Shah, Kannan Ramchandran, and Martin J Wainwright. Active ranking from pairwise comparisons and when parametric assumptions do not help. *The Annals of Statistics*, 47(6):3099–3126, 2019.

Innar Liiv. Seriation and matrix reordering methods: An historical overview. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 3(2):70–91, 2010.

Allen Liu and Ankur Moitra. Better algorithms for estimating non-parametric models in crowd-sourcing and rank aggregation. In *Conference on Learning Theory (COLT)*, volume 125 of *Proceedings of Machine Learning Research*, pages 2780–2829. PMLR, 2020.

Yao Ma, Alexander Olshevsky, Csaba Szepesvari, and Venkatesh Saligrama. Gradient descent for sparse rank-one matrix completion for crowd-sourced aggregation of sparsely interacting workers. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3335–3344. PMLR, 2018.

Heikki Mannila. Finding total and partial orders from data for seriation. In *Discovery Science*, pages 16–25, Berlin, Heidelberg, 2008. Springer.

Cheng Mao, Jonathan Weed, and Philippe Rigollet. Minimax rates and efficient algorithms for noisy sorting. In *Proceedings of Algorithmic Learning Theory*, volume 83 of *Proceedings of Machine Learning Research*, pages 821–847. PMLR, 2018.

Cheng Mao, Ashwin Pananjady, and Martin J. Wainwright. Towards optimal estimation of bivariate isotonic matrices with unknown permutations. *Ann. Statist.*, 48(6):3183–3205, December 2020.

William H. Marquardt. Advances in archaeological seriation. *Advances in Archaeological Method and Theory*, 1:257–314, December 1978.

William T. McCormick, Paul J. Schweitzer, and Thomas W. White. Problem decomposition and data reorganization by a clustering technique. *Operations Research*, 20(5): 993–1009, 1972.

Ashwin Pananjady and Richard J. Samworth. Isotonic regression with unknown permutations: Statistics, computation, and adaptation. *arXiv preprint arXiv:2009.02609*, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 1976.

Nihar B. Shah and Martin J. Wainwright. Simple, robust and optimal ranking from pairwise comparisons. *Journal of Machine Learning Research*, 18:199:1–199:38, 2017.

Nihar B. Shah, Sivaraman Balakrishnan, Adityanand Guntuboyina, and Martin J. Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. *IEEE Trans. Inf. Theor.*, 63(2): 934–959, February 2017.

Nihar B Shah, Sivaraman Balakrishnan, and Martin J Wainwright. Low permutation-rank matrices: Structural properties and noisy completion. *Journal of Machine Learning Research*, 2019.

Nihar B Shah, Sivaraman Balakrishnan, and Martin J Wainwright. A permutation-based model for crowd labeling: Optimal estimation and robustness. *IEEE Transactions on Information Theory*, 2020.

Ryan J. Tibshirani, Holger Hoefling, and Robert Tibshirani. Nearly-isotonic regression. *Technometrics*, 53(1):54–61, 2011.

Jingyan Wang and Nihar B. Shah. Your 2 is my 1, your 3 is my 9: Handling arbitrary miscalibrations in ratings. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, page 864–872. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

Preliminary version