Matrix Estimation for Individual Fairness

Cindy Y. Zhang [*1] **Sarah H. Cen** [*2] **Devavrat Shah** [2]

Abstract

In recent years, multiple notions of algorithmic fairness have arisen. One such notion is individual fairness (IF), which requires that individuals who are similar receive similar treatment. In parallel, matrix estimation (ME) has emerged as a natural paradigm for handling noisy data with missing values. In this work, we connect the two concepts. We show that pre-processing data using ME can improve an algorithm's IF without sacrificing performance. Specifically, we show that using a popular ME method known as singular value thresholding (SVT) to pre-process the data provides a strong IF guarantee under appropriate conditions. We then show that, under analogous conditions, SVT pre-processing also yields estimates that are consistent and approximately minimax optimal. As such, the ME pre-processing step does not, under the stated conditions, increase the prediction error of the base algorithm, i.e., does not impose a fairness-performance trade-off. We verify these results on synthetic and real data.

1. Introduction

As data-driven decision-making becomes more ubiquitous, there is increasing attention on the *fairness* of machine learning (ML) algorithms. Because what is deemed to be fair is context-dependent (e.g., reflects a given value system), there is no universally accepted notion of fairness.

One notion of algorithmic fairness is *individual fairness (IF)*, which is distinct from notions of group fairness (e.g., equalized odds). Stated informally, IF says that similar individuals should receive similar treatment. More precisely, an algorithm $f : \mathcal{X} \to \mathcal{Y}$ acting on a set of individuals $\mathcal{X}$ is individually fair if for any two individuals $a, b \in \mathcal{X}$,

$$D(f(a), f(b)) \leq L \cdot d(a, b), \tag{1}$$

[*]Equal contribution [1]Princeton University [2]Massachusetts Institute of Technology. Correspondence to: Cindy Y. Zhang <cindyz@princeton.edu>, Sarah H. Cen <shcen@mit.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

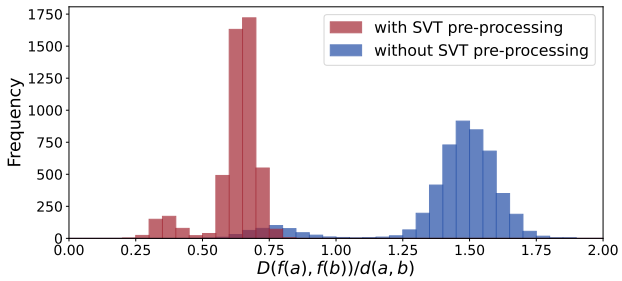

Figure 1. We run a deep neural network on synthetic data with and without SVT pre-processing (see Section 6, Experiment #1 for details). We randomly select pairs $a, b \in \mathcal{X}$ then compute the ratio $D(f(a), f(b))/d(a, b)$, where f denotes the neural network with (red) and without (blue) SVT pre-processing. As shown, applying SVT pre-processing results in lower ratios, which indicates that it improves individual fairness, as defined in (1). Indeed, we show in Section 4 that, under appropriate conditions, SVT pre-processing strengthens an algorithm's IF guarantee.

for the choice of distance metrics d and D. The *Lipschitz constant* L captures how strictly the IF condition is enforced. An algorithm f that satisfies IF ensures that the outcomes between two individuals who are close in feature space $\mathcal{X}$ also receive outcomes that are close in outcome space $\mathcal{Y}$, where the level of closeness is captured by L. A smaller Lipschitz constant therefore implies a stronger IF constraint.

In parallel, matrix estimation (ME) has arisen as a natural paradigm to handle data that is noisy and/or has missing values. In this work, we propose a two-step procedure in which the data (e.g., training data) is first pre-processed using a ME technique known as *singular value thresholding (SVT)* before being used by an inference algorithm h (e.g., a neural network). We show that, under appropriate conditions, this pre-processing step *strengthens* the IF guarantee of the inference algorithm, i.e., combining SVT with h results in a lower Lipschitz constant in (1) than h does alone.

Although SVT can improve an algorithm's IF, it is not clear whether such an improvement comes at a cost to the algorithm's performance. In this work, we show that the same thresholds that allow SVT to improve IF *also* imply that SVT has strong performance guarantees. In other words, under the appropriate conditions, SVT improves IF without imposing a performance cost in settings where ME can be

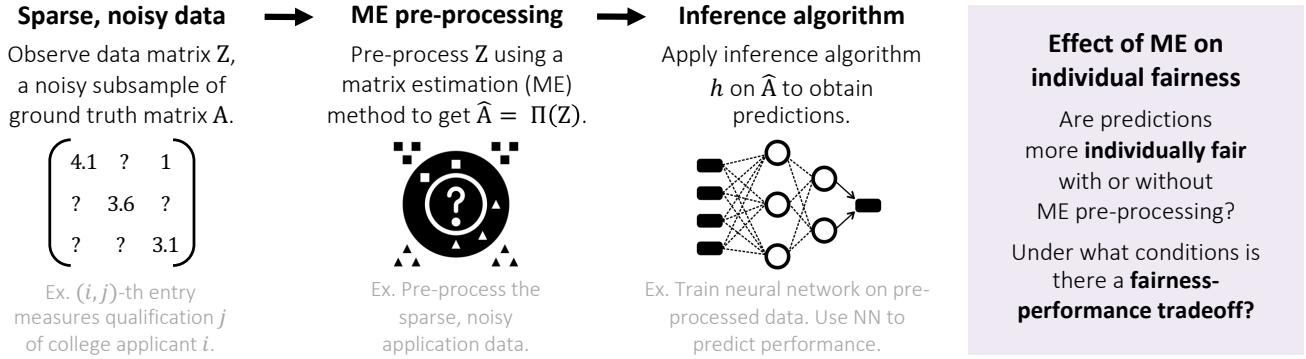

Figure 2. We study the effect of ME pre-processing on IF and performance in settings where we need to perform an inference task using sparse, noisy data. Our main results show that SVT, a popular ME method, provides strong IF guarantees and does not necessarily hurt performance when used as a pre-processing step.

applied. Our problem setup is visualized in Figure 2 and described in detail in Section 3.

Our main contributions can be summarized as follows:

- **We show SVT pre-processing has strong IF guarantees**. ME is used in high-dimensional inference to handle sparse, noisy data. One of the most popular ME methods is SVT. In Sections 4.2-4.3, we derive a set of conditions under which SVT pre-processing strengthens the IF guarantees of the inference algorithm with respect to the observed covariates and provides an approximate IF guarantee with respect to the (unknown) ground truth covariates. We then use this result to explore how SVT affects predictions in different data regimes.

- **We show that IF under SVT does not hurt asymptotic performance**. In Section 4.4, we show that achieving IF using SVT pre-processing does not necessarily hurt performance. Specifically, we show that the same conditions that are needed for SVT to guarantee IF mirror the conditions required under a popular method known as universal singular value thresholding (USVT). Because USVT has strong performance guarantees (it produces an estimator that is consistent and approximately minimax (Chatterjee, 2015)), this connection implies that SVT pre-processing can achieve IF without imposing a performance cost. Stated differently, enforcing IF via SVT pre-processing does not harm performance because it places no further restrictions on ME than the performance-based method USVT.

- **We empirically verify these results on real and synthetic datasets**. In Section 6, we demonstrate our findings on synthetic data and the MovieLens 1M dataset. We visualize the effect of SVT pre-processing on IF. Figure 1, for example, illustrates how the ratio $D(f(a), f(b))/d(a, b)$ decreases under SVT pre-processing. Smaller values indicate a stronger IF guarantee. We also demonstrate the effect of SVT pre-processing on performance.

To the best of our knowledge, this is the first work that establishes a theoretical link between IF and ME.

2. Background and Related Work

Matrix estimation (ME). ME studies the problem of estimating the entries of a matrix from noisy observations of a subset of the entries (Candès & Tao, 2010; Recht, 2011; Keshavan et al., 2010a; Negahban & Wainwright, 2012; Davenport et al., 2014; Chatterjee, 2015; Chen & Wainwright, 2015). ME is a class of methods that can be applied to data expressed in matrix form. Specifically, suppose there is a latent matrix, and one can only obtain noisy samples of a subset of its entries. The goal of ME is to estimate the values of every entry based on the noisy subsamples.

ME is used, for example, by recommender systems to estimate a user's interest in different types of content (Koren et al., 2009; Song et al., 2016; Borgs et al., 2017). In fact, the winning solution of the Netflix Prize was built on ME methods (Koren, 2009). ME has also been used to study social networks (Anandkumar et al., 2013; Abbe & Sandon, 2015; Hopkins & Steurer, 2017); to impute and forecast a time series (Agarwal et al., 2018; Amjad et al., 2018); to aggregate information in crowdsourcing (Shah & Lee, 2018); and more.

Singular value thresholding (SVT). There is an extensive literature on ME and the closely related areas of matrix completion and matrix factorization. While there are various approaches (Rennie & Srebro, 2005), spectral methods are among the most popular (Candès & Tao, 2010; Mazumder et al., 2010; Keshavan et al., 2010a;b)

One such method is SVT (Cai et al., 2010), which first factorizes the matrix of observations, then reconstructs it using only the singular values that exceed a predetermined threshold. It is well-known that SVT is a shrinkage operator that provides a solution to a nuclear norm minimization problem.

Universal singular value thresholding (USVT) builds on SVT by proposing an adaptive threshold that produces an estimator that is both consistent and approximately minimax (Chatterjee, 2015). We review SVT and USVT in Sections 4.1 and 4.4. As a pre-processing method, SVT has been used to improve the performance of an inference algorithm in time series analysis (Agarwal et al., 2018) and to improve adversarial robustness of deep neural networks (Yang et al., 2019).

Individual fairness (IF). IF is the notion that similar individuals should receive similar treatment (Dwork et al., 2012; Barocas et al., 2018), as formalized in (1). As an example, suppose individuals A and B apply for job interviews at the same time with similar (observed) qualifications a and b. Then, IF requires that A and B receive interview requests at similar rates. IF is distinct from notions of group fairness (e.g., statistical parity in the outcomes across demographic groups), but there are conditions under which IF implies group fairness (Dwork et al., 2012).

Under IF, similarity is captured by the choice of distance metrics D and d (cf, (1)), and IF is enforced as a Lipschitz constraint based on the chosen metrics. How to define "similarity" between individuals and their outcomes (i.e., how to choose the distance metrics) has been the subject of significant debate (Gajane & Pechenizkiy, 2017; Beutel et al., 2019; Ilvento, 2019; Beutel et al., 2019; Gillen et al., 2018; Bechavod et al., 2020). In this work, we allow for any D. One of our IF results is given for d as the ℓ^2 norm and the other for d as the ℓ^q norm.

Fairness and collaborative filtering. In recommendation, collaborative filtering algorithms leverage similarities between users to infer user preferences, and ME can be viewed as one such algorithm. There is some work on the fairness of collaborative filtering, and these typically study group fairness (Kamishima et al., 2012; Yao & Huang, 2017; Beutel et al., 2019; Foulds et al., 2020; Pitoura et al., 2021; Shao et al., 2022). A small number of works examine notions of fairness related to individuals (Serbos et al., 2017; Biega et al., 2018; Stratigi et al., 2020), but they are distinct from our notion of IF as formulated by Dwork et al. (Dwork et al., 2012). To our knowledge, we provide the first theoretical analysis connecting IF to ME and collaborative filtering, which can be found in Section 4.

Accuracy. One common thread of interest in algorithmic fairness is the fairness-accuracy trade-off (Farnadi et al., 2018; Zhu et al., 2018; Liu & Burke, 2018; Islam et al., 2020). By establishing a connection between IF and USVT, we show in Section 4.4 that IF can be achieved without significant performance costs in ME applications, including collaborative filtering.

3. Problem Statement

3.1. Setup

Consider a setting with m individuals. Suppose there is an unknown ground truth matrix $A \in \mathbb{R}^{m\times n}$, where each row in A corresponds to an individual such that the i-th row $\mathbf{A}_i \in \mathbb{R}^n$ is an unknown n-dimensional feature vector that describes individual $i \in [m]$. We assume that $A_{ij} \in [-1, 1]$ for all $i \in [m]$ and $j \in [n]$.[1]

Suppose that it is possible to observe a noisy subsample of A's entries. Formally, let $\Omega \subset [m] \times [n]$ denote the index set of observed entries and $\mathcal{Z} = [-1, 1] \cup \{\emptyset\}$. Let $Z \in \mathcal{Z}^{m\times n}$ denote the matrix of observations, where each entry of Z is a random variable; $\mathbb{E}Z_{ij} = A_{ij}$; $Z_{ij} \in [-1, 1]$ if $(i, j) \in \Omega$ and $Z_{ij} = \emptyset$, otherwise. As such, the i-th row $\mathbf{Z}_i \in \mathcal{Z}^n$ denotes the observed covariates for individual i.

Consider the following inference task. Make a prediction $\mathbf{y} \in \mathcal{Y}$ for individual $i \in [m]$ using the observations (i.e., training data) Z. Let $\mathcal{F} = \{f : [m] \times \mathcal{Z}^{m\times n} \to \mathcal{Y}\}$ denote the class of algorithms that perform this inference task. Note that the output of f could be a deterministic value or a distribution over possible values.

For the remainder of this work, let $\mathbf{B}_i$ denote the i-th row and $\mathbf{b}_i$ denote the i-th column of a matrix B.

3.2. Individual Fairness

Individual fairness (IF) is the notion that *similar individuals should receive similar treatments* (Dwork et al., 2012). IF is formulated as a (D, d)-Lipschitz constraint, as follows.

Definition 3.1 (IF with respect to observed covariates)**.** Consider an observation matrix $Z \in \mathcal{Z}^{m\times n}$. An algorithm $f \in \mathcal{F}$ is (D, d)-*individually fair on* Z if

$$D(f(i, Z), f(j, Z)) \leq L \cdot d(\mathbf{Z}_i, \mathbf{Z}_j),$$

for all $i, j \in [m]$, where $L \geq 0$ does not depend on i or j, $D : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ is a distance metric, and $d : \mathcal{Z}^n \times \mathcal{Z}^n \to \mathbb{R}_{\geq 0}$ is also a distance metric.

Definition 3.2 (IF with respect to latent covariates)**.** Consider an observation matrix $Z \in \mathcal{Z}^{m\times n}$ and ground truth matrix $A \in [-1, 1]^{m\times n}$. An algorithm $f \in \mathcal{F}$ is (D, d)-*individually fair on* A if

$$D(f(i, Z), f(j, Z)) \leq L \cdot d(\mathbf{A}_i, \mathbf{A}_j),$$

for all $i, j \in [m]$, where $L \geq 0$ does not depend on i and j, and $D : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ and $d : [-1, 1]^n \times [-1, 1]^n \to \mathbb{R}_{\geq 0}$ are distance metrics.

[1] For any A whose entries are bounded such that $|A_{ij}| < \infty$ for all $i \in [m]$ and $j \in [n]$, one can always translate and rescale A to be between -1 and 1, then adjust the final result accordingly.

Problem statement. We focus on a subclass of algorithms $\mathcal{F}(\mathcal{H}, \Pi) = \{f = h \circ \Pi : h \in \mathcal{H}\} \subset \mathcal{F}$, where $\mathcal{H} \subset \{h : [m] \times [-1, 1]^{m \times n} \to \mathcal{Y}\}$ and $\Pi : \mathcal{Z}^{m \times n} \to [-1, 1]^{m \times n}$. Intuitively, Π is a pre-processing method that takes in the (sparse and noisy) data Z and produces an estimate $\Pi(Z)$ of the unknown matrix A. The inference algorithm h is then applied on top of Π such that $f(i, Z) = h(i, \Pi(Z))$.

In this work, we examine the IF of f relative to h when Π is given by a ME method, i.e., how a ME pre-processing step affects the IF of an inference algorithm.

3.3. Examples

The setup in Section 3.1 can be applied to many problems in which the training data and algorithmic inputs are *noisy*, *sparse*, or both. Consider the following examples and the implications of IF.

Example 3.1 (Recommendation)**.** *Consider a platform that provides personalized movie recommendations to its m users based on sparse, noisy observations of their preferences. Suppose that the movie preferences of each user $i \in [m]$ can be described by an unknown n-dimensional vector $\mathbf{A}_i \in \mathbb{R}^n$. For instance, $a_{ij} \in [-1, 1]$ could denote the ground-truth preference of user i for movie $j \in [n]$.*

Although $A = [\mathbf{A}_1, \ldots, \mathbf{A}_m]^\top$ is unknown, the platform receives occasional feedback from users in the form of ratings and can also observe the users' viewing behaviors. Let these sparse, noisy observations be stored in Z, where $Z_{ij} = \emptyset$ implies that user i has not rated movie j. The goal of the platform is to estimate the users' movie preferences. Note that $f \in \mathcal{F}$ can leverage other information (e.g., ratings by other users, as done in collaborative filtering). In this example, IF on Z requires that users with similar viewing and rating behaviors receive similar recommendations. IF on A implies that users with similar latent (i.e., unknown) movie preferences receive similar recommendations.

Example 3.2 (Admissions)**.** *Consider an admissions setting in which there are m applicants. Suppose that, for the purposes of admissions, each applicant $i \in [m]$ is described by an unknown n-dimensional vector $\mathbf{A}_i \in \mathbb{R}^n$. Suppose each individual i submits an application $\mathbf{Z}_i$, which contains sparse, noisy measurements of $\mathbf{A}_i$. For example, one's standardized test score in math is a noisy measurement of one's math abilities. Data sparsity can occur when one applicant includes information that another does not (e.g., one may list "debate club" on their resume while another does not, but this sparsity does not necessarily imply that the latter is worse at public speaking). As an output, $f \in \mathcal{F}$ could produce an admissions score $\mathbf{y} \in [0, 1]$. In this example, IF on Z requires that applicants with similar applications receive similar admissions scores. IF on A implies that applicants whose true (but unknown) qualifications are similar receive similar admissions scores.*

Although IF on A is desirable, one generally requires IF on Z, i.e., that an algorithm ensures IF with respect to the information at its disposal. Consider Example 3.2. Suppose that two applicants i and j have similar ground-truth features but the first $n/2$ values of $\mathbf{Z}_i$ are $\emptyset$ while last $n/2$ values of $\mathbf{Z}_j$ are $\emptyset$. In other words, the types of qualifications that i reports contains no overlap with the types of qualifications j reports. Because i and j have similar ground-truth features, IF on A would require that a school treat i and j similarly even though the schools are given vastly different information about the two applicants.

4. Main Results

In this section, we show that pre-processing data with ME can improve IF with little to no performance cost under appropriate conditions. Before providing our main results, we begin in Section 4.1 by describing a ME method known as singular value thresholding (SVT). In Sections 4.2-4.3, we show that SVT pre-processing offers IF guarantees on both the observation matrix Z and the ground truth matrix A. In Section 4.4, we show that the class of SVT thresholds that guarantee IF align with the thresholds used by a well-known ME technique that has strong performance guarantees. This connection implies that SVT pre-processing can provide IF without imposing a high performance cost.

4.1. Singular Value Thresholding

Recall the inference task described in Section 3.1. In this section, we propose to use a popular ME method known as **singular value thresholding (SVT)** as the pre-processing step. That is, for algorithms in the class $\mathcal{F}(\mathcal{H}, \Pi) = \{f = h \circ \Pi : h \in \mathcal{H}\} \subset \mathcal{F}$, we propose that Π denote SVT.

More precisely, $\mathrm{SVT}(Z, \tau, \psi)$ takes in three values: the observation matrix $Z \in \mathcal{Z}^{m \times n}$, a threshold $\tau \geq 0$, and an increasing function $\psi : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. SVT then proceeds in four steps, as follows.

1. First, for any element in Z that is $\emptyset$, replace that value with 0, i.e., if $Z_{ij} = \emptyset$, re-assign it to $Z_{ij} = 0$. [2]
2. Perform the singular value decomposition (SVD):

$$Z = \sum_{\ell=1}^{\min(m,n)} \sigma_\ell \mathbf{u}_\ell \mathbf{v}_\ell^\top,$$

   where $\sigma_\ell \geq 0$ is the ℓ-th singular value, $\mathbf{u}_\ell \in \mathbb{R}^{m \times 1}$ is the ℓ-th left singular vector, and $\mathbf{v}_\ell \in \mathbb{R}^{n \times 1}$ is the ℓ-th right singular vector.
3. For any index ℓ such that $\sigma_\ell > \tau$, add ℓ to the set $S(\tau)$ such that $S(\tau) = \{\ell : \sigma_\ell > \tau\}$.

[2] Several methods exist for handling missing values in ME literature; common approaches include replacing missing values with 0 or the expected value of all entries.

4. Finally, construct estimate of A:

$$\hat{A} = \min\left(1, \max\left(-1, \sum_{\ell \in S(\tau)} \psi(\sigma_\ell)\mathbf{u}_\ell \mathbf{v}_\ell^\top\right)\right).$$

Intuitively, SVT detects and removes components of the observation matrix Z that correspond to noise while preserving the remaining components $S(\tau)$. The threshold τ determines the boundary between signal and noise, where a higher value for τ means that fewer components are kept.

4.2. IF With Respect to Observed Covariates

In the previous section, we proposed to pre-process Z using SVT before applying an inference algorithm h on top of it. In this section, we show that using SVT for pre-processing guarantees IF on Z. For the remainder of this section, we fix the Z of interest.

Consider a specific threshold τ and function ψ. Recall that σ_ℓ, $\mathbf{u}_\ell$, and $\mathbf{v}_\ell$ are the ℓ-th singular value, left singular vector, and right singular vector of Z, respectively. Recall further that $S(\tau) = \{\ell : \sigma_\ell > \tau\}$. Let

$$K_2 = \left\| \sum_{\ell \in S(\tau)} \frac{\psi(\sigma_\ell)\mathbf{u}_\ell \mathbf{v}_\ell^\top}{\sigma_\ell^2} \right\|_\infty \sqrt{n} \max_k \|\mathbf{z}_k\|_1 .$$

Theorem 4.1. *Suppose that h is $(\mathcal{D}, \ell^2)$-individually fair with constant K_1, i.e.,*

$$D(h(i,B), h(j,B)) \leq K_1 ||\mathbf{B}_i - \mathbf{B}_j||_2,$$

for all $i, j \in [m]$ and $B \in [-1,1]^{m \times n}$. Then, for $f = h \circ \mathrm{SVT}(Z, \tau, \psi)$,

$$D(f(i,Z), f(j,Z)) \leq K_1 K_2 \|\mathbf{Z}_i - \mathbf{Z}_j\|_1 ,$$

for all $i, j \in [m]$, i.e., f is (D, ℓ^1)-individually fair on Z with constant $K_1 K_2$.

Theorem 4.1 states that when h is IF with Lipschitz constant K_1, applying SVT pre-processing preserves IF with constant $K_1 K_2$ with respect to the observed covariates. In order for h with SVT pre-processing to have stronger IF than h alone, we need $K_2 \ll 1$, as we examine next.

Corollary 4.2. *Suppose $\psi(x) = \beta x$ and Z satisfies the strong incoherence condition*[3] *with parameter μ_1, i.e.,*

$$\left\| \sum_{\ell \in S(\tau)} \mathbf{u}_\ell \mathbf{v}_\ell^\top \right\|_\infty \leq \sqrt{\frac{\mu_1 r}{mn}},$$

where $r = |S(\tau)|$ denotes the rank of Z. Then for any threshold τ, $K_2 \leq \beta\sqrt{\mu_1 r m}/\tau$.

Corollary 4.2 characterizes common conditions under which K_2 scales as $O(\sqrt{rm}/\tau)$.

Specifically, suppose that $\tau \geq \sqrt{2n\beta}$. Then, $K_2 = O(\sqrt{rm/n})$. This indicates that combining h with SVT pre-processing would *improve* the IF of h as long as $n = \omega(rm)$. In other words, as long as there is enough data n per individual relative to the number of individuals m and the rank r, then $K_2 \to 0$ as $n \to \infty$.[4] We discuss the implications of this result further in Section 5.

4.3. IF With Respect to Latent Covariates

In the previous section, we showed that SVT pre-processing can improve IF on Z. In this section, we show that SVT pre-processing can also ensure IF on A as long as its estimates $\hat{A}$ are close to the ground-truth values.

Theorem 4.3. *Let d denote the ℓ^q norm. Suppose that h is $(\mathcal{D}, d)$-individually fair with constant K_1, i.e.,*

$$D(h(i,B), h(j,B)) \leq K_1 ||\mathbf{B}_i - \mathbf{B}_j||_q,$$

for all $i, j \in [m]$ and $B \in [-1,1]^{m \times n}$. Then, for $f = h \circ \mathrm{SVT}(Z, \tau, \psi)$,

$$\begin{aligned} &D(f(i,Z), f(j,Z)) \\ &\qquad \leq K_1 \|\mathbf{A}_i - \mathbf{A}_j\|_q + 2K_1 ||\hat{A} - A||_{q,\infty} \end{aligned}$$

for all $i, j \in [m]$.

Theorem 4.3 states when h is IF with Lipschitz constant K_1, then f is approximately IF on A and approaches exact IF as $\hat{A} \to A$. Note that Theorem 4.3 holds for any method Π.

This result implies that SVT pre-processing preserves the IF guarantee of h on A as the estimation error of SVT approaches 0. We show in the next section (Proposition 4.5) that, under an appropriate choice of threshold, the estimation error of SVT indeed goes to 0 (specifically, that $||\hat{A} - A||_{2,\infty} \to 0$) as $m, n \to \infty$. Together, these two results imply that adding SVT pre-processing to h ensures IF on A under the same conditions that guarantee that SVT (or, more generally, ME) is accurate.[5]

Remark 4.4. Theorem 4.3 shows that it is possible to achieve approximate IF on A, and the tightness of this guarantee depends on the accuracy of Π. Even though IF on A may be

[3] Strong incoherence is a standard assumption in the ME literature (Keshavan et al., 2010a; Negahban & Wainwright, 2012; Chen, 2015). It requires that the singular vectors of a matrix are not sparse, which can make it difficult to estimate the underlying latent matrix when given limited samples.

[4] The rank r indicates the "complexity" of the ground-truth matrix A. Although it is computed using Z, it reflects the amount of "signal" in Z, which generally depends on A.

[5] Note that the condition in both theorems that $D(h(i,B), h(j,B)) \leq K_1 ||\mathbf{B}_i - \mathbf{B}_j||_q$ for all $i, j \in [m]$ and $B \in [-1,1]^{m \times n}$ is not strong. In fact, if it is not met, then there is no method Π such that f is IF.

desirable, IF on Z is important because both individuals and algorithm designers generally cannot make claims based on the unknown ground-truth matrix A; they must often point to evidence, i.e., observations Z.

4.4. Performance Under Individual Fairness

Recall from Theorem 4.1 that, as long as the threshold τ is sufficiently large, SVT pre-processing guarantees IF on Z. However, it is unclear if the threshold chosen for IF is good for prediction performance. We now show that an adaptive threshold that is known to provide high accuracy coincides with thresholds that guarantee IF on Z. Because this adaptive threshold guarantees that $\hat{A} \to A$, it also guarantees IF on A, as per Theorem 4.3. As a result, SVT pre-processing under the appropriate threshold ensures that IF on both Z and A at little to no performance cost.

Consider a well-known ME method known as **universal singular value thresholding (USVT)**. USVT refines SVT by proposing a universal formula for the threshold τ, thereby removing the need to tune τ by hand. Under mild assumptions on A and Ω, USVT has strong performance guarantees. In order to study performance, let the mean-squared error (MSE) of ME be defined as

$$\text{MSE}(\hat{A}) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbb{E}\left[(\hat{A}_{ij} - A_{ij})^2\right]. \tag{2}$$

Let $\|M\|_*$ denote the nuclear norm of matrix M. We begin with a well-known performance guarantee on USVT.

Theorem 4.5 (Modified from Theorem 1.1. in Chatterjee (2015))**.** *Suppose the elements of Z are independent random variables, each independently observed with probability $p \in [0,1]$. Let $\hat{p}$ be the proportion of observed values, $\psi(x) = x/\hat{p}$, $\epsilon \in (0,1]$, and $w = (2+\eta)^2$ for $\eta \in (0,1)$. Let $\rho_1 = \max(m,n)$ and $\rho_2 = \min(m,n)$. Then, if $p \geq \rho_1^{\epsilon-1}$ for some $\epsilon > 0$ and $\tau = \sqrt{w\rho_1\hat{p}}$,*

$$\textit{MSE}\left(\text{SVT}(Z,\tau,\psi)\right) \leq C(\eta) \min\left(\frac{\|A\|_*}{\rho_2\sqrt{\rho_1 p}}, \frac{\|A\|_*^2}{\rho_1\rho_2}, 1\right) + C(\epsilon,\eta)\exp(-c(\eta)\rho_1 p),$$

where $C(\eta), c(\eta) > 0$ depend only on η and $C(\epsilon,\eta)$ depends only on η and ϵ.[6]

[6] This upper bound can be improved when the additional condition that $\text{Var}(Z_{ij}) \leq \sigma^2$ for all i,j and $\sigma \leq 1$ holds. Then, if $\tau \geq \sqrt{wn\hat{q}}$, where $\hat{q} = \hat{p}\sigma^2 + \hat{p}(1-\hat{p})(1-\sigma^2)$, $q \geq n^{\epsilon-1}$, and $q = p\sigma^2 + p(1-p)(1-\sigma^2)$ (Chatterjee, 2015),

$$\text{MSE}(\hat{A}) \leq C(\eta) \min\left(\frac{\|A\|_* \sqrt{q}}{mp\sqrt{n}}, \frac{\|A\|_*^2}{mn}, 1\right) + C(\epsilon,\eta)\exp(-c(\eta)nq).$$

Theorem 4.5 states that when $\tau = \sqrt{w\rho_1\hat{p}}$ and p is large enough, the MSE of SVT decays at a rate of $o((mn)^{-1})$. As an immediate extension, Theorem 4.5 tells us that if the loss of h when given perfect information A is small, then the loss of $f = h \circ \text{SVT}(Z, \sqrt{\omega\rho_1\hat{p}}, \psi)$ is also small as $n, m \to \infty$ because the estimate $\hat{A}$ produced by USVT is close to A.

Remark 4.6. Chatterjee (2015) also show that the MSE of USVT is within a constant multiplicative factor and an exponentially small, additive term of the MSE of the minimax estimator, which implies that one cannot do much better than USVT (cf. Theorem 1.2 in Chatterjee (2015)).

As such, SVT is consistent and approximately minimax under the threshold $\tau = \sqrt{w\rho_1\hat{p}}$. Next, we connect this finding to our earlier results on IF.

Performance under IF on Z. Suppose that $n > m$. Then, $\rho_1 = n$ and Theorem 4.5 indicates that SVT pre-processing with the threshold $\tau = \sqrt{w\hat{p}n}$ has good performance. Under Corollary 4.2, such a threshold also ensures that f *with* SVT pre-processing is *more* individually fair on Z than f *without* SVT pre-processing for large enough n such that $n = \omega(rm)$. Therefore, there is no trade-off between performance and IF under SVT pre-processing when n grows at the rate $\omega(rm)$.

Performance under IF on A. Recall from Theorem 4.3 that ME pre-processing is approximately individually fair on A, and it is fully individually fair on A when $||\Pi(Z) - A||_{q,\infty} = 0$. Therefore, the relationship between IF on A and performance under ME is straightforward: the lower the estimation error $\|\Pi(Z) - A\|_{q,\infty}$, the more individually fair f is on A.

5. Discussion

In this section, we interpret the results and discuss the conditions under which SVT pre-processing guarantees IF and good performance simultaneously.

Combining the results. Under Theorem 4.5, SVT yields good performance guarantees as $n \to \infty$ when $\tau = \sqrt{w\hat{p}n}$ and $n \geq m$. Under Corollary 4.2, this same τ guarantees IF on Z with Lipschitz constant K_1K_2, where K_1 is the Lipschitz constant for h without SVT pre-processing and $K_2 = O(\sqrt{rm\hat{p}/n})$. SVT pre-processing can therefore improve IF on Z *without* sacrificing performance when $K_2 \ll 1$, So, when is $K_2 \ll 1$, and why is K_2 sometimes greater than 1?

To answer this question, we examine two data regimes: (i) when $n = o(rm\hat{p})$ and (ii) when $n = \omega(rm\hat{p})$.

First data regime. In the first data regime, Corollary 4.2 tells us that $K_2 > 1$, which implies that SVT pre-processing does *not* necessarily improve IF. This phenomenon occurs

because, when there is not much information by which to distinguish between individuals (i.e., n, the number of features per individual, is small), SVT pre-processing produces a $\hat{A}$ that is smoothed across rows. That is, it causes f to treat individuals similarly *on the whole*.

This can, at times, work against IF, which requires that *similar* individuals be treated similarly, but not that *the population* be treated similarly. To see why the latter can work against IF, consider $g_1(x) = x$ and $g_2(x) = \text{round}(x)$ for $x \in [0, 1]$. Under g_2, individuals can only receive outcomes 0 or 1, so the algorithm treats individuals similarly *on the whole*. By this, we mean that individuals fall into one of two buckets, so the treatment is relatively homogeneous.

On the other hand, under g_1, individuals receive one of infinitely many outcomes in the range $[0, 1]$. Which of the two is individually fair? Although g_2 treats individuals similarly on the whole, g_1 is IF since $d(g_1(x), g_1(x')) = d(x, x')$ while g_2 is *not* because $g_2(0.5 - \delta) = 0$ while $g_2(0.5 + \delta) = 1$ for arbitrarily small $\delta > 0$. A similar logic can be used to show that SVT pre-processing does not always improve IF in this first data regime.[7]

Second data regime. In the second data regime, Corollary 4.2 tells us that $K_2 < 1$, which implies that SVT pre-processing *improves* IF. Intuitively, when $n = \omega(rm\hat{p})$, the number of features per individual grows faster than the number of individuals, rank, and observation probability. In this case, SVT smooths the data in a different way. It produces a $\hat{A}$ that is smoothed across columns. It therefore removes noise from individual (row) vectors $\mathbf{Z}_i$ but leaves enough signal in $\mathbf{Z}_i$ to differentiate individual i from other individuals, thereby avoiding the phenomenon that can occur in the first data regime (that individuals are treated similarly on the whole). The fact that the observational data is smoothed but individuals remain differentiable allows SVT to improve IF in this data regime.

Putting it together. SVT pre-processing smooths the data before sending it to h, and this smoothing operation affects IF differently in different data regimes. We show, however, that under an appropriately chosen threshold, IF on Z, IF on A, and good performance are simultaneously guaranteed as $n \to \infty$. More precisely, *when* $\tau = \sqrt{w\hat{p}n}$, $n = \omega(rm\hat{p})$, *and* n *is sufficiently large, SVT pre-processing not only strengthens IF on* Z*, but it also guarantees IF on* A *and good prediction performance.*

[7] Although SVT pre-processing does not necessarily improve IF in the first data regime, some might argue that the "smoothing" that SVT does can prevent f from unnecessarily differentiating between individuals. For example, suppose that f determines how much COVID-19 relief each household gets. Suppose that, due to the short turnaround time, $n = o(r^2 m)$, e.g., the government has little information on how each household has been affected by COVID-19. One might argue that, in such situations, the government cannot reliably distinguish between households and should send the same amount of monetary relief to all households rather than tailor the amounts based on limited data. The reasoning goes: in this data regime, it is easy to overfit and use spurious information to distinguish between individuals. In this way, one may debate the importance of IF in the first data regime.

6. Experiments

We ran several experiments in order to test the effect of SVT pre-processing on IF and performance. The inference task is to estimate the unknown n-dimensional feature vector $\mathbf{A}_i$ for each individual $i \in [m]$ using the observations Z.

The results show that SVT pre-processing improves IF, both in simulation and in the MovieLens1M dataset. We also examine the performance of an inference algorithm with and without SVT pre-processing. As expected, we find that adding SVT pre-processing increases the MSE but only by a small amount; by Theorem 4.5, we would expect this amount to decay to 0 as the amount of data grows.

Below, we divide our discussion into three parts. In the first two parts, we describe our experimental setups for the *synthetic data* and on the *MovieLens 1M dataset*. In the third part, we discuss the results. Additional results and implementation details can be found in the Appendix.

6.1. Setup for Experiment #1: Synthetic Data

In Experiment #1, we test h with and without SVT pre-processing on synthetic data, as follows.

Generating the ground truth matrix A. Consider $m = 200$ individuals. We sample m feature vectors of length $n = 800$, each corresponding to an individual, to form the ground truth matrix $A \in [-1, 1]^{m \times n}$. The feature vectors (i.e., the rows of A) are sampled from $c = 10$ clusters, where each cluster is given by a multivariate normal distribution. The mean of each cluster is a vector of length n drawn uniformly at random from $(-1, 1)$, and the covariance of each cluster is an $n \times n$ diagonal matrix with whose diagonal values are sampled uniformly at random from $(0, 0.1)$. The feature vectors are then clipped so that all values fall within $[-1, 1]$.

Generating the observation matrix Z. Recall that Ω denotes the set of observed entries. We generate Z as follows:

$$Z_{ij} = \begin{cases} \text{clip}(A_{ij} + \eta_{ij}, [-1, 1]), & \text{if } (i, j) \in \Omega, \\ \emptyset, & \text{otherwise,} \end{cases}$$

where $\eta_{ij} \sim \mathcal{N}(0, 0.1)$. In this section, $(i, j) \in \Omega \subset [m] \times [n]$ with probability p. This is aligned with the conditions in Theorem 4.5. In the Appendix, we provide results under a different choice of Ω (specifically, when the probability of observing an individual i's j-th feature depends on the cluster to which i belongs).

Inference algorithm. Recall that the inference task is to predict the feature vector $\mathbf{A}_i$ for individual i given data B (where B may or may not have undergone SVT pre-processing). In the synthetic data setting, we let the algorithm $h : [m] \times [-1, 1]^{m\times n} \to [-1, 1]^n$ be given as follows.

Let $h' : [m] \times [n] \to [0, 1]$ denote a deep neural net (DNN) trained on data B. Let the DNN be composed of three fully connected layers of size 300, 100, and 1 with ReLU activation after the hidden layers and sigmoid after the output layer. Lastly, let

$$h(i, B) = 2[h'(i, 1), h'(i, 2), \ldots, h'(i, n)]^\top - 1.$$

Pre-processing. We compare the IF and performance of h with and without SVT pre-processing. When there is no pre-processing step, the data B on which h' is trained on is Z (missing entries are replaced with zeros).

When SVT pre-processing is used, the data B on which h' is trained is $\mathrm{SVT}(Z, \tau, \psi)$, where $\hat{p} = |\Omega|/(mn)$, $\psi(x) = x/\hat{p}$, $\hat{q} = 0.01^2\hat{p} + \hat{p}(1 - \hat{p})(1 - 0.01^2)$, and $\tau = \sqrt{2.01n\hat{q}}$. This form of SVT is consistent with USVT (see Theorem 4.5 and Footnote 6).

6.2. Setup for Experiment #2: MovieLens 1M Dataset

In Experiment #2, we test h with and without SVT pre-processing on a popular, real-world dataset known as the MovieLens 1M Dataset.

Dataset. The MovieLens 1M dataset (Harper & Konstan, 2015) contains movie ratings data for 6040 users and 3952 movies. In the context of this work, this ratings data can be placed in the $m \times n$ matrix Z, where $m = 6040$ and $n = 3952$. Each entry Z_{ij} contains user i's rating of movie j if (i, j) is observed, and $Z_{ij} = \emptyset$ if user i has not rated movie j. The ratings are normalized to be between 0 and 1.

As a real-world dataset, there is no ground-truth matrix A. As such, we cannot evaluate performance relative to A—our MovieLens discussion instead focuses on IF.

Inference algorithm. Recall that the inference task is to predict the feature vector $\mathbf{A}_i$ for individual i given data B (where B may or may not have undergone SVT pre-processing). In the MovieLens setting, we let the inference algorithm $h : [m] \times [-1, 1]^{m\times n} \to [-1, 1]^n$ be the K-nearest neighbors (K-NN) algorithm (Sarwar et al., 2001).[8]

K-NN produces an estimate $\mathbf{Y}_i$ by taking the weighted average of the K users most similar to user i. In this work, we let $K = 10$ and the similarity between users i and j be measured using adjusted cosine similarity:

$$\mathrm{sim}(i, j) = \frac{\sum_{k\in[n]}(B_{ik} - \bar{B}_k)(B_{jk} - \overline{B}_k)}{\sqrt{\sum_{k\in[n]}(B_{ik} - \bar{B}_k)^2 \sum_{k'\in[n]}(B_{jk'} - \bar{B}_{k'})^2}},$$

where $\bar{B}_k$ represents the average of the k-th item's ratings.

Table 1. Results on IF and performance in Experiment #1.

	$\hat{p} = 0.05$	$\hat{p} = 0.1$	$\hat{p} = 0.2$
$\mathrm{MSE}(h)$	0.33 ± 0.003	0.21 ± 0.002	0.10 ± 0.003
$\mathrm{MSE}(f)$	0.35 ± 0.004	0.21 ± 0.001	0.11 ± 0.001
$\mathrm{IF}_1^h(Z)$	0.23 ± 0.005	0.18 ± 0.003	0.13 ± 0.001
$\mathrm{IF}_1^f(Z)$	0.02 ± 0.001	0.02 ± 0.001	0.03 ± 0.001
K_2	0.01 ± 0.001	0.01 ± 0.002	0.02 ± 0.001
$\mathrm{IF}_2^h(A)$	0.45 ± 0.011	0.63 ± 0.012	0.81 ± 0.005
$\mathrm{IF}_2^f(A)$	0.49 ± 0.015	0.65 ± 0.006	0.82 ± 0.003

Pre-processing. We compare the IF of h with and without SVT pre-processing. When there is no pre-processing step, the data B used by K-NN is Z (missing entries are replaced with zeros). When SVT pre-processing is used, the data B used by K-NN is $\mathrm{SVT}(Z, \tau, \psi)$, where $\hat{p} = |\Omega|/(mn)$, $\psi(x) = x/\hat{p}$, $\tau = \sqrt{2.01n\hat{p}}$, as consistent with USVT (see Theorem 4.5).

6.3. Results

Metrics. For a function $g : [m] \times \mathcal{Z}^{m\times n} \to \mathbb{R}$, let

$$\mathrm{MSE}(g) = \frac{1}{mn - |\Omega|} \sum_{(i,j)\notin\Omega} (g_j(i, Z) - \mathbf{A}_{ij})^2,$$

where $g_j(i, Z)$ is the j-th element of the vector $g(i, Z)$. For a matrix $X \in \mathbb{R}^{m\times n}$, let

$$\mathrm{IF}_q^g(X) = \frac{1}{m^2} \sum_{i,j\in[m]} \frac{\|g(i, Z) - g(j, Z)\|_2}{\|\mathbf{X}_i - \mathbf{X}_j\|_q}.$$

$\mathrm{IF}_q^f(Z)$ and $\mathrm{IF}_q^h(Z)$ measure IF on Z with and without SVT pre-processing, respectively. $\mathrm{IF}_q^f(A)$ and $\mathrm{IF}_q^h(A)$ measure IF on A with and without SVT pre-processing, respectively.[9] A smaller ratio indicates a stronger IF guarantee.

Results. Table 1 summarizes the results for Experiment #1. The values are averaged over 10 simulations, and the error bars give $+/-$ two standard deviations. Figures 1-3 visualize the effect of SVT pre-processing on IF on Z for Experiments #1 and #2. We discuss our findings below.

[8] We use K-NN in order to investigate the effect of SVT pre-processing on another common class of algorithms. In particular, K-NN smooths data in a way that already encourages IF, which makes it particularly meaningful if SVT pre-processing is able to *further* improve IF.

[9] We use the ℓ^1 norm for Z as per our result in Theorem 4.1 and the ℓ^2 norm for A due to the connection between Theorem 4.3 and Theorem 4.5.

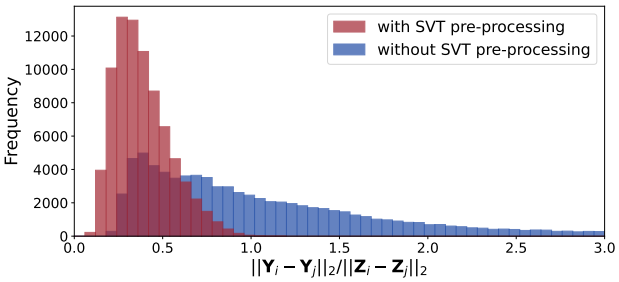

Figure 3. Frequencies of $||\mathbf{Y}_i - \mathbf{Y}_j||_2 / \|\mathbf{Z}_i - \mathbf{Z}_j\|_2$ across randomly selected pairs (i, j) in Experiment #2. Y denotes the estimate produced by K-NN on the MovieLens 1M dataset with (red) and without (blue) SVT pre-processing.

Effect of SVT pre-processing on IF on Z. Table 1 verifies that SVT pre-processing improves IF on Z in Experiment #1. In particular, $\text{IF}_1^f(Z)$ is much smaller than $\text{IF}_1^h(Z)$. Figures 1 and 3 visualize this effect for Experiments #1 and #2, showing that SVT pre-processing causes the difference in two individuals' outcomes relative to the difference in their features to be smaller than without SVT pre-processing.

Effect of SVT pre-processing on IF on A. IF on A is comparable though slightly weaker with SVT pre-processing than without it. In particular, $\text{IF}_2^f(A)$ is slightly larger than $\text{IF}_2^h(A)$ in Table 1. This is in line with Theorem 4.3, which tells us that adding a pre-processing step Π may weaken IF on A if the estimation error of Π is non-zero. Since SVT cannot estimate A perfectly, it yields some estimation error and, as a result, slightly weakens f's IF on A.

As illustrated in Table 1, this effect is small. Moreover, the gap between $\text{IF}_2^f(A)$ and $\text{IF}_2^h(A)$ gets smaller as $\hat{p}$ increases. This is consistent with our results because the estimation error of SVT decreases as $\hat{p}$ increases (see Theorem 4.5), which means that the IF on A guarantee improves as $\hat{p}$ increases (see Theorem 4.3).

Effect of SVT pre-processing on performance. The rows in Table 1 corresponding to $\text{MSE}(h)$ and $\text{MSE}(f)$ measure the error of the DNN without and with SVT pre-processing, respectively, in Experiment #1. As expected from our discussion in Section 4.4, they show that SVT pre-processing has a minimal effect on prediction performance, i.e., that there is little to no fairness-performance trade-off.

7. Conclusion

In this work, we propose using a well-known matrix estimation (ME) method known as singular value thresholding (SVT) to pre-process sparse, noisy data before applying an inference algorithm (e.g., a neural network). We show that pre-processing data using SVT before applying an inference algorithm comes with strong individual fairness (IF) guarantees. Specifically, we derive conditions under which SVT pre-processing *improves* IF. We then show that, under these same conditions, SVT pre-processing has strong performance guarantees. Together, these results imply that, under the appropriate conditions, SVT pre-processing provides a way to improve IF without imposing a performance cost. We verify our results on synthetic data and the MovieLens 1M dataset.

Acknowledgements

We thank our reviewers for their time and helpful comments. We also thank Michael Zhang for providing feedback on earlier versions of this work. This work was supported in parts by the MIT-IBM project on "Representation Learning as a Tool for Causal Discovery" and the NSF TRIPODS Phase II grant towards Foundations of Data Science Institute.

References

Abbe, E. and Sandon, C. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 670–688. IEEE, 2015.

Agarwal, A., Amjad, M. J., Shah, D., and Shen, D. Model agnostic time series analysis via matrix estimation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(3):1–39, 2018.

Amjad, M., Shah, D., and Shen, D. Robust synthetic control. *The Journal of Machine Learning Research*, 19(1):802–852, 2018.

Anandkumar, A., Ge, R., Hsu, D., and Kakade, S. A tensor spectral approach to learning mixed membership community models. In *Conference on Learning Theory*, pp. 867–881. PMLR, 2013.

Barocas, S., Hardt, M., and Narayanan, A. Fairness and machine learning: Limitations and opportunities, 2018.

Bechavod, Y., Jung, C., and Wu, Z. S. Metric-free individual fairness in online learning. *arXiv preprint arXiv:2002.05474*, 2020.

Beutel, A., Chen, J., Doshi, T., Qian, H., Wei, L., Wu, Y., Heldt, L., Zhao, Z., Hong, L., Chi, E. H., et al. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2212–2220, 2019.

Biega, A. J., Gummadi, K. P., and Weikum, G. Equity of attention: Amortizing individual fairness in rankings. In *The 41st International ACM SIGIR Conference on*

Research & Development in Information Retrieval, pp. 405–414, 2018.

Borgs, C., Chayes, J., Lee, C. E., and Shah, D. Thy friend is my friend: Iterative collaborative filtering for sparse matrix estimation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4718–4729, 2017.

Cai, J.-F., Candès, E. J., and Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982, 2010.

Candès, E. J. and Tao, T. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

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

Chen, Y. Incoherence-optimal matrix completion. *IEEE Transactions on Information Theory*, 61(5):2909–2923, 2015.

Chen, Y. and Wainwright, M. J. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *arXiv preprint arXiv:1509.03025*, 2015.

Davenport, M. A., Plan, Y., Van Den Berg, E., and Wootters, M. 1-bit matrix completion. *Information and Inference: A Journal of the IMA*, 3(3):189–223, 2014.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pp. 214–226, 2012.

Farnadi, G., Kouki, P., Thompson, S. K., Srinivasan, S., and Getoor, L. A fairness-aware hybrid recommender system. *arXiv preprint arXiv:1809.09030*, 2018.

Foulds, J. R., Islam, R., Keya, K. N., and Pan, S. An intersectional definition of fairness. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pp. 1918–1921. IEEE, 2020.

Gajane, P. and Pechenizkiy, M. On formalizing fairness in prediction with machine learning. *arXiv preprint arXiv:1710.03184*, 2017.

Gillen, S., Jung, C., Kearns, M., and Roth, A. Online learning with an unknown fairness metric. *arXiv preprint arXiv:1802.06936*, 2018.

Harper, F. M. and Konstan, J. A. The MovieLens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

Hopkins, S. B. and Steurer, D. Efficient Bayesian estimation from few samples: Community detection and related problems. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 379–390. IEEE, 2017.

Ilvento, C. Metric learning for individual fairness. *arXiv preprint arXiv:1906.00250*, 2019.

Islam, R., Keya, K. N., Zeng, Z., Pan, S., and Foulds, J. Neural fair collaborative filtering. *arXiv preprint arXiv:2009.08955*, 2020.

Kamishima, T., Akaho, S., Asoh, H., and Sakuma, J. Enhancement of the neutrality in recommendation. In *Decisions@ RecSys*, pp. 8–14. Citeseer, 2012.

Keshavan, R. H., Montanari, A., and Oh, S. Matrix completion from a few entries. *IEEE transactions on information theory*, 56(6):2980–2998, 2010a.

Keshavan, R. H., Montanari, A., and Oh, S. Matrix completion from noisy entries. *The Journal of Machine Learning Research*, 11:2057–2078, 2010b.

Koren, Y. The Bellkor solution to the Netflix grand prize. *Netflix prize documentation*, 81(2009):1–10, 2009.

Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37, 2009.

Liu, W. and Burke, R. Personalizing fairness-aware reranking. *arXiv preprint arXiv:1809.02921*, 2018.

Mazumder, R., Hastie, T., and Tibshirani, R. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11: 2287–2322, 2010.

Negahban, S. and Wainwright, M. J. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *The Journal of Machine Learning Research*, 13:1665–1697, 2012.

Pitoura, E., Stefanidis, K., and Koutrika, G. Fairness in rankings and recommendations: An overview. *arXiv preprint arXiv:2104.05994*, 2021.

Recht, B. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(12), 2011.

Rennie, J. D. and Srebro, N. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pp. 713–719, 2005.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, 2001.

Serbos, D., Qi, S., Mamoulis, N., Pitoura, E., and Tsaparas, P. Fairness in package-to-group recommendations. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 371–379, 2017.

Shah, D. and Lee, C. Reducing crowdsourcing to graphon estimation, statistically. In *International Conference on Artificial Intelligence and Statistics*, pp. 1741–1750. PMLR, 2018.

Shao, P., Wu, L., Chen, L., Zhang, K., and Wang, M. FairCF: Fairness-aware collaborative filtering. *Science China Information Sciences*, 65(12):1–15, 2022.

Song, D., Lee, C. E., Li, Y., and Shah, D. Blind regression: Nonparametric regression for latent variable models via collaborative filtering. *Advances in Neural Information Processing Systems*, 29:2155–2163, 2016.

Stratigi, M., Nummenmaa, J., Pitoura, E., and Stefanidis, K. Fair sequential group recommendations. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pp. 1443–1452, 2020.

Yang, Y., Zhang, G., Katabi, D., and Xu, Z. ME-Net: Towards effective adversarial robustness with matrix estimation. *arXiv preprint arXiv:1905.11971*, 2019.

Yao, S. and Huang, B. Beyond parity: Fairness objectives for collaborative filtering. *arXiv preprint arXiv:1705.08804*, 2017.

Zhu, Z., Hu, X., and Caverlee, J. Fairness-aware tensor-based recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 1153–1162, 2018.

Appendix

A. Proof of Theorem 4.1

Before proving Theorem 4.1, we require two lemmas, as follows.

Lemma A.1. *Suppose $T \in \mathbb{R}^{m \times n}$ and $\boldsymbol{x} \in \mathbb{R}^{n \times 1}$. Then*

$$\|T\boldsymbol{x}\|_2 \leq \|T\|_\infty \|\boldsymbol{x}\|_1 \sqrt{m}.$$

Proof.

$$\begin{aligned}
\|T\mathbf{x}\|_2 &= \left(\sum_{i \in [m]} \left(\sum_{j \in [n]} T_{ij} x_j \right)^2 \right)^{1/2} \\
&\leq \left(\sum_{i \in [m]} \left(\sum_{j \in [n]} |T_{ij}||x_j| \right)^2 \right)^{1/2} \\
&\leq \left(\sum_{i \in [m]} \left(\sum_{j \in [n]} \|T\|_\infty |x_j| \right)^2 \right)^{1/2} \\
&= \|T\|_\infty \left(\sum_{i \in [m]} \left(\sum_{j \in [n]} |x_j| \right)^2 \right)^{1/2} \\
&= \|T\|_\infty \sqrt{m} \left(\sum_{j \in [n]} |x_j| \right) \\
&= \|T\|_\infty \|\mathbf{x}\|_1 \sqrt{m}.
\end{aligned}$$

□

Lemma A.2. *Suppose $T \in \mathbb{R}^{m \times n}$ and $\boldsymbol{x} \in \mathbb{R}^{n \times 1}$. Then*

$$\|T\boldsymbol{x}\|_1 \leq \|\boldsymbol{x}\|_1 \max_j \|\mathbf{t}_j\|_1 .$$

Proof. Recall $\mathbf{T}_i$ denotes the i-th row of T and $\mathbf{t}_i$ denotes the i-th column of T.

$$\begin{aligned}
\|T\mathbf{x}\|_1 &= \sum_{i \in [m]} |\mathbf{T}_i^\top \mathbf{x}| \\
&= \sum_{i \in [m]} \left| \sum_{j \in [n]} T_{ij} x_j \right| \\
&\leq \sum_{i \in [m]} \sum_{j \in [n]} |T_{ij} x_j| \\
&\leq \sum_{i \in [m]} \sum_{j \in [n]} |T_{ij}||x_j| \\
&= \sum_{j \in [n]} |x_j| \sum_{i \in [m]} |T_{ij}|
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{j\in[n]} \left(|x_j| \max_{j'} \left(\sum_{i\in[m]} |T_{ij'}| \right) \right) \\
&= \sum_{j\in[n]} |x_j| \left(\max_{j'} \|\mathbf{t}_{j'}\|_1 \right) \\
&= \|\mathbf{x}\|_1 \max_j \|\mathbf{t}_j\|_1 .
\end{aligned}
$$

□

Theorem 4.1. *Suppose that h is $(\mathcal{D}, \ell^2)$-individually fair with constant K_1, i.e.,*

$$D(h(i,B), h(j,B)) \leq K_1 ||\mathbf{B}_i - \mathbf{B}_j||_2,$$

for all $i, j \in [m]$ and $B \in [-1,1]^{m\times n}$. Then, for $f = h \circ \mathrm{SVT}(Z, \tau, \psi)$,

$$D(f(i,Z), f(j,Z)) \leq K_1 K_2 \|\mathbf{Z}_i - \mathbf{Z}_j\|_1 , \tag{3}$$

for all $i, j \in [m]$, i.e., f is (D, ℓ^1)-individually fair on Z with constant K_1K_2.

Proof. Let the singular value decomposition (SVD) of Z be given by $Z = \sum_{\ell=1}^{\min(m,n)} \sigma_\ell \mathbf{u}_\ell \mathbf{v}_\ell^\top$, where $\sigma_i, \mathbf{u}_i, \mathbf{v}_i$ denote the i-th singular value, left singular vector, and right singular vector of Z respectively. Given $f = h \circ \mathrm{SVT}(Z, \tau, \psi)$, the input $\hat{A}$ to h is the output of running SVT on Z, i.e.,

$$\hat{A} = \sum_{\ell\in S(\tau)} \psi(\sigma_\ell) \mathbf{u}_\ell \mathbf{v}_\ell^\top .$$

We can expand $||\hat{\mathbf{A}}_i - \hat{\mathbf{A}}_j||_2$ to get

$$
\begin{aligned}
||\hat{\mathbf{A}}_i - \hat{\mathbf{A}}_j||_2 &= \left\| \sum_{\ell\in S(\tau)} \psi(\sigma_\ell) u_{\ell i} \mathbf{v}_\ell^\top - \sum_{\ell\in S(\tau)} \psi(\sigma_\ell) u_{\ell j} \mathbf{v}_\ell^\top \right\|_2 && (4) \\
&= \left\| \sum_{\ell\in S(\tau)} \psi(\sigma_\ell)(u_{\ell i} - u_{\ell j}) \mathbf{v}_\ell^\top \right\|_2 . && (5)
\end{aligned}
$$

Next we rewrite $u_{\ell i} - u_{\ell j}$ in terms of $\mathbf{Z}_i$ and $\mathbf{Z}_j$. Since $\mathbf{u}_\ell$ is the ℓ-th left singular vector of Z, it is the ℓ-th eigenvector of $ZZ^\top$. Let λ_ℓ be the ℓ-th eigenvalue of $ZZ^\top$. Note that $\lambda_\ell = \sigma_\ell^2$. Then,

$$\lambda_\ell \mathbf{u}_\ell = ZZ^\top \mathbf{u}_\ell .$$

Looking at only the i-th row,

$$
\begin{aligned}
\lambda_\ell u_{\ell i} &= \mathbf{Z}_i^\top Z^\top \mathbf{u}_\ell \\
\implies u_{\ell i} &= \frac{\mathbf{Z}_i^\top Z^\top \mathbf{u}_\ell}{\lambda_\ell} \\
\implies u_{\ell i} - u_{\ell j} &= \frac{(\mathbf{Z}_i - \mathbf{Z}_j)^\top Z^\top \mathbf{u}_\ell}{\sigma_\ell^2} .
\end{aligned}
$$

Plugging this back into equation (5), we get

$$||\hat{\mathbf{A}}_i - \hat{\mathbf{A}}_j||_2 = \left\| \sum_{\ell\in S(\tau)} \frac{\psi(\sigma_\ell)(\mathbf{Z}_i - \mathbf{Z}_j)^\top Z^\top \mathbf{u}_\ell \mathbf{v}_\ell^\top}{\sigma_\ell^2} \right\|_2 \tag{6}$$

$$= \left\| \left(\sum_{\ell \in S(\tau)} \frac{\psi(\sigma_\ell)}{\sigma_\ell^2} \left(\mathbf{u}_\ell \mathbf{v}_\ell^\top \right)^\top \right) Z(\mathbf{Z}_i - \mathbf{Z}_j) \right\|_2 . \tag{7}$$

Next we apply Lemma A.1 to (7) to get

$$||\hat{\mathbf{A}}_i - \hat{\mathbf{A}}_j||_2 \leq \left\| \sum_{\ell \in S(\tau)} \frac{\psi(\sigma_\ell) \mathbf{u}_\ell \mathbf{v}_\ell^\top}{\sigma_\ell^2} \right\|_\infty \sqrt{n} \left\| Z(\mathbf{Z}_i - \mathbf{Z}_j) \right\|_1 . \tag{8}$$

Applying Lemma A.2 to $\left\| Z(\mathbf{Z}_i - \mathbf{Z}_j) \right\|_1$ in (8) gives us

$$||\hat{\mathbf{A}}_i - \hat{\mathbf{A}}_j||_2 \leq \left\| \sum_{\ell \in S(\tau)} \frac{\psi(\sigma_\ell) \mathbf{u}_\ell \mathbf{v}_\ell^\top}{\sigma_\ell^2} \right\|_\infty \sqrt{n} \max_k \left\| \mathbf{z}_k \right\|_1 \left\| \mathbf{Z}_i - \mathbf{Z}_j \right\|_1 . \tag{9}$$

Since $D(h(i, B), h(j, B)) \leq K_1 ||\mathbf{B}_i - \mathbf{B}_j||_2$,

$$D(f(i, Z), f(j, Z)) = D(h(i, \hat{A}), h(j, \hat{A})) \tag{10}$$

$$\leq K_1 \left(\left\| \sum_{\ell \in S(\tau)} \frac{\psi(\sigma_\ell) \mathbf{u}_\ell \mathbf{v}_\ell^\top}{\sigma_\ell^2} \right\|_\infty \sqrt{n} \max_k \left\| \mathbf{z}_k \right\|_1 \right) \left\| \mathbf{Z}_i - \mathbf{Z}_j \right\|_1 \tag{11}$$

$$= K_1 K_2 \left\| \mathbf{Z}_i - \mathbf{Z}_j \right\|_1 . \tag{12}$$

□

B. Proof of Corollary 4.2

Corollary 4.2. *Suppose $\psi(x) = \beta x$ and Z satisfies the strong incoherence condition with parameter μ_1 (Chen, 2015), i.e.,*

$$\left\| \sum_{\ell \in S(\tau)} \mathbf{u}_\ell \mathbf{v}_\ell^\top \right\|_\infty \leq \sqrt{\frac{\mu_1 r}{mn}},$$

where $r = |S(\tau)|$ denotes the rank of Z. Then for any threshold τ, $K_2 \leq \beta \sqrt{\mu_1 r m} / \tau$.

Proof. Recall that

$$K_2 = \left\| \sum_{\ell \in S(\tau)} \frac{\psi(\sigma_\ell) \mathbf{u}_\ell \mathbf{v}_\ell^\top}{\sigma_\ell^2} \right\|_\infty \sqrt{n} \max_k \left\| \mathbf{z}_k \right\|_1 .$$

Given $\psi(x) = \beta x$, we have

$$K_2 = \beta \left\| \sum_{\ell \in S(\tau)} \frac{\mathbf{u}_\ell \mathbf{v}_\ell^\top}{\sigma_\ell} \right\|_\infty \sqrt{n} \max_k \left\| \mathbf{z}_k \right\|_1 .$$

Recall $S(\tau) = \{\ell : \sigma_\ell > \tau\}$ is the set of components whose singular values exceed τ, so the value of any σ_ℓ in the denominator must be at least τ, giving us

$$K_2 \leq \frac{\beta}{\tau} \left\| \sum_{\ell \in S(\tau)} \mathbf{u}_\ell \mathbf{v}_\ell^\top \right\|_\infty \sqrt{n} \max_k \left\| \mathbf{z}_k \right\|_1 .$$

Given Z satisfies the strong incoherence condition,

$$K_2 \leq \frac{\beta}{\tau} \cdot \sqrt{\frac{\mu_1 r}{mn}} \cdot \sqrt{n} \max_k \left\| \mathbf{z}_k \right\|_1 .$$

Since each entry $Z_{ij} \in [-1, 1]$ and there are m entries in each column of Z, $\|\mathbf{z}_k\|_1 \leq m$. Hence

$$K_2 \leq \frac{\beta}{\tau} \cdot \sqrt{\frac{\mu_1 r}{mn}} \cdot \sqrt{n} \cdot m \leq \frac{\beta\sqrt{\mu_1 rm}}{\tau},$$

concluding our proof. □

C. Proof of Theorem 4.3

Theorem 4.3. *Let d denote the ℓ^q norm. Suppose that h is $(\mathcal{D}, d)$-individually fair with constant K_1, i.e.,*

$$D(h(i, B), h(j, B)) \leq K_1 ||\mathbf{B}_i - \mathbf{B}_j||_q,$$

for all $i, j \in [m]$ and $B \in [-1, 1]^{m\times n}$. Then, for $f = h \circ \mathrm{SVT}(Z, \tau, \psi)$,

$$D(f(i, Z), f(j, Z)) \leq K_1 \|\mathbf{A}_i - \mathbf{A}_j\|_q + 2K_1 ||\hat{A} - A||_{q,\infty} \tag{13}$$

for all $i, j \in [m]$.

Proof. Recall that $\|M\|_{q,\infty} = \max_i \|\mathbf{m}_i\|_q$. This result follows from the application of the triangle inequality.

$$\begin{aligned}
D(f(i, Z), f(j, Z)) &= D(h(i, \hat{A}), h(j, \hat{A})) \\
&\leq K_1 \left\|\hat{\mathbf{A}}_i - \hat{\mathbf{A}}_j\right\|_q \\
&\leq K_1(||\hat{\mathbf{A}}_i - \mathbf{A}_i||_q + ||\hat{\mathbf{A}}_j - \mathbf{A}_j||_q + ||\mathbf{A}_i - \mathbf{A}_j||_q) \\
&= K_1||\mathbf{A}_i - \mathbf{A}_j||_q + 2K_1||\hat{A} - A||_{q,\infty}.
\end{aligned}$$

The first equality follows from the fact that $f = h \circ \mathrm{SVT}(Z, \tau, \psi)$, the first inequality follows from the assumption that h is IF with constant K_1, the second inequality follows from applying the triangle inequality twice with intermediate points $\mathbf{A}_i$ and $\mathbf{A}_j$, and the final equality combines like terms, which gives the result as stated. □

D. Modification of Theorem 1.1 in Chatterjee (2015)

Theorem 1.1 from Chatterjee (2015). *Suppose that we have a $m \times n$ matrix M, where $m \leq n$ and the entries of M are bounded by 1 in absolute value. Let X be a matrix whose elements are independent random variables, and $\mathbb{E}(x_{ij}) = m_{ij}$ for all i and j. Assume that the entries of X are also bounded by 1 in absolute value, with probability one. Let p be a real number belonging to the interval $[0, 1]$. Suppose that each entry of X is observed with probability p, and unobserved with probability $1 - p$, independently of the other entries.*

We construct an estimator $\hat{M}$ of M based on the observed entries of X using the Universal Singular Value Thresholding (USVT) algorithm with threshold $(2 + \eta)\sqrt{n\hat{p}}$. Suppose that $p \geq n^{-1+\varepsilon}$ for some $\varepsilon > 0$. Then

$$\mathit{MSE}(\hat{M}) \leq C \min\left(\frac{\|M\|_*}{m\sqrt{np}}, \frac{\|M\|_*^2}{mn}, 1\right) + C(\varepsilon)e^{-cnp},$$

where C and c are positive constants that depend only on the choice of η and $C(\varepsilon)$ depends only on ε and η.

In our work, we have modified Theorem 1.1 from Chatterjee (2015) for our specific setup. The modifications only involve the renaming of variables to keep our notation consistent and to clarify the dependencies between variables. The changes are summarized in the following table.

Table 2. Modifications to notation in Theorem 1.1 of Chatterjee (2015).

NOTATION IN CHATTERJEE (2015)	OUR NOTATION
M	A
X	Z
$\hat{M}$	$\text{SVT}(M, \tau, \psi)$
n	ρ_1
m	ρ_2
C	$C(\eta)$
c	$c(\eta)$
$C(\varepsilon)$	$C(\varepsilon, \eta)$

Our modified theorem is as follows.

Theorem 4.5. (Modified from Theorem 1.1. in Chatterjee (2015))**.** *Suppose the elements of Z are independent random variables, each independently observed with probability $p \in [0, 1]$. Let $\hat{p}$ be the proportion of observed values, $\psi(x) = x/\hat{p}$, $\epsilon \in (0, 1]$, and $w = (2 + \eta)^2$ for $\eta \in (0, 1)$. Let $\rho_1 = \max(m, n)$ and $\rho_2 = \min(m, n)$. Then, if $p \geq \rho_1^{\epsilon-1}$ for some $\epsilon > 0$ and $\tau = \sqrt{w\rho_1\hat{p}}$,*

$$\textit{MSE}\left(\text{SVT}(Z, \tau, \psi)\right) \leq C(\eta) \min\left(\frac{\|A\|_*}{\rho_2\sqrt{\rho_1 p}}, \frac{\|A\|_*^2}{\rho_1\rho_2}, 1\right) + C(\epsilon, \eta)\exp(-c(\eta)\rho_1 p),$$

where $C(\eta), c(\eta) > 0$ depend only on η and $C(\epsilon, \eta)$ depends only on η and ϵ.

E. Experimental Setup

Below are additional details about our experimental setup:

Training the DNN. Given input matrix B, the training set of the deep neural net (DNN) described in Section 6.1 consists of (input, target) tuples of the form $(\begin{bmatrix}\mathbf{B}_i & \mathbf{b}_j\end{bmatrix}, B_{ij})$. Missing entries in $\mathbf{B}_i$ and $\mathbf{b}_j$ are replaced with zeros. Out of the observed entries $(i, j) \in \Omega$, 80 percent are used for training and the remaining 20 percent are used for validation; the unobserved entries form our test set. We use a batch size of 128 and 2000 steps of training.

F. Additional Experiments

F.1. Experiment #3: Observing entries non-uniformly at random

Setup. Recall the setup for Experiment #1 in Section 6.1. We sample $m = 200$ feature vectors of length $n = 800$, each corresponding to an individual, to form the ground truth matrix $A \in [-1, 1]^{m\times n}$. The feature vectors are sampled from $c = 10$ clusters, where each cluster is a multivariate normal distribution.

Next we generate the observation matrix Z. Recall that Ω denotes the set of observed entries. Instead of selecting each entry independently with probability p, we instead observe entries with different probabilities depending on the cluster it belongs to. For each cluster k, there is an associated random vector $\mathbf{p}_k \in \mathbb{R}^n$ with entries summing to $p \cdot n$. For each individual i in cluster k, the entry (i, j) is observed with probability $p_i[j]$. The expected number of observed entries is $p \cdot n \cdot m$, so the proportion observed is as desired, but the entries are no longer drawn uniformly at random as the probability an entry is drawn is dependent on the cluster it is in. The remaining setup is identical to that for Experiment #1.

Results. Table 3 summarizes the results for Experiment #3. The values are averaged over 10 simulations, and the error bars give $+/-$ two standard deviations. We observe that $\text{IF}_1^f(Z)$ is much smaller than $\text{IF}_1^h(Z)$, which again verifies SVT pre-processing improves IF on Z.

Note that the entries $(i, j) \in \Omega$ not being selected uniformly at random violates one of the conditions of Theorem 4.5, which states that each entry of A is independently observed with probability p. Despite violating this condition, we observe in Table 3 that there is minimal decrease in performance when applying SVT pre-processing. This indicates the performance guarantees of SVT are robust to relaxations of the independence condition stated in Theorem 4.5.

Table 3. Results on IF and performance in Experiment #3.

	$\hat{p} = 0.05$	$\hat{p} = 0.1$	$\hat{p} = 0.2$	$\hat{p} = 0.4$
MSE(h)	0.31 ± 0.002	0.23 ± 0.002	0.16 ± 0.002	0.14 ± 0.001
MSE(f)	0.33 ± 0.003	0.23 ± 0.001	0.17 ± 0.001	0.14 ± 0.001
$\text{IF}_1^h(Z)$	0.24 ± 0.005	0.17 ± 0.003	0.11 ± 0.002	0.06 ± 0.001
$\text{IF}_1^f(Z)$	0.02 ± 0.001	0.02 ± 0.001	0.03 ± 0.001	0.03 ± 0.001
K_2	0.05 ± 0.003	0.11 ± 0.007	0.25 ± 0.009	0.49 ± 0.013
$\text{IF}_2^h(A)$	0.46 ± 0.010	0.63 ± 0.011	0.75 ± 0.015	0.75 ± 0.007
$\text{IF}_2^f(A)$	0.49 ± 0.014	0.64 ± 0.006	0.75 ± 0.008	0.73 ± 0.014

F.2. Experiment #4: Varying length of feature vectors

Setup. Consider $m = 500$ individuals. We sample m feature vectors of length n from $c = 20$ clusters, where each cluster is a multivariate normal distribution. The mean of each cluster is a vector of length n drawn uniformly at random from $(-1, 1)$, and the covariance of each cluster is an $n \times n$ diagonal matrix with whose diagonal values are sampled uniformly at random from $(0, 0.1)$. The feature vectors are then clipped so that all values fall within $[-1, 1]$. When generating the observation matrix Z, we observe a proportion $p = 0.2$ of entries uniformly at random. Instead of varying the value of p, we instead create datasets for varying values of n, the length of the feature vector. The remaining setup is identical to that for Experiment 1, described in Section 6.1.

Table 4. Results on IF and performance over different values of n in Experiment #4.

	$n = 25$	$n = 100$	$n = 400$	$n = 800$
MSE(h)	0.36 ± 0.004	0.28 ± 0.003	0.13 ± 0.001	0.10 ± 0.001
MSE(f)	0.36 ± 0.004	0.28 ± 0.003	0.12 ± 0.001	0.10 ± 0.001
$\text{IF}_1^h(Z)$	0.43 ± 0.004	0.26 ± 0.002	0.17 ± 0.001	0.13 ± 0.001
$\text{IF}_1^f(Z)$	0.15 ± 0.003	0.07 ± 0.001	0.05 ± 0.001	0.03 ± 0.001
K_2	0.21 ± 0.020	0.06 ± 0.002	0.03 ± 0.001	0.02 ± 0.001
$\text{IF}_2^h(A)$	0.51 ± 0.004	0.63 ± 0.005	0.80 ± 0.007	0.83 ± 0.007
$\text{IF}_2^f(A)$	0.62 ± 0.011	0.67 ± 0.007	0.82 ± 0.011	0.83 ± 0.010

Results. Table 4 summarizes the results for Experiment #4. Recall from Section 5 that our theoretical guarantees for SVT pre-processing simultaneously strengthening IF and having good prediction performance hold when $n = \omega(rm\hat{p})$. In the above setting, both $n = 25$ and $n = 100$ fall within $o(rm\hat{p})$. However, we observe that $\text{IF}_1^f(Z)$ is much smaller than $\text{IF}_1^h(Z)$ for all the values of n, and there are very minimal differences between MSE(h) and MSE(f). This means that even when $n = o(rm\hat{p})$, we still see large improvements in IF with respect to Z with little to no effect on prediction performance when applying SVT pre-processing. This demonstrates that the empirical IF and performance benefits of SVT pre-processing are not restricted to when $n = \omega(rm\hat{p})$.