Matrix Factorization with Interval-Valued Data

Mao-Lin Li, Francesco Di Mauro, K Selçuk Candan, and Maria Luisa Sapino

Abstract—With many applications relying on multi-dimensional datasets for decision making, matrix factorization (or decomposition) is becoming the basis for many knowledge discoveries and machine learning tasks, from clustering, trend detection, anomaly detection, to correlation analysis. Unfortunately, a major shortcoming of matrix analysis operations is that, despite their effectiveness when the data is scalar, these operations become difficult to apply in the presence of non-scalar data, as they are not designed for data that include non-scalar observations, such as intervals. Yet, in many applications, the available data are inherently non-scalar for various reasons, including imprecision in data collection, conflicts in aggregated data, data summarization, or privacy issues, where one is provided with a reduced, clustered, or intentionally noisy and obfuscated version of the data to hide information. In this paper, we propose matrix decomposition techniques that consider the existence of interval-valued data. We show that naive ways to deal with such imperfect data may introduce errors in analysis and present factorization techniques that are especially effective when the amount of imprecise information is large.

Index Terms—Matrix factorization, Interval valued data.

1 Introduction

With many machine-learning applications requiring latent semantics underlying the data sets, matrix factorization has emerged as a successful tool for discovering latent patterns in data [1], [2]: matrices are used to encode relationships among pairs of entities and data are analyzed for their latent semantics through matrix decomposition operations, such as singular value decomposition, SVD [3] or principal component analysis, PCA [4].

1.1 Challenge: Interval-Valued Data

In many applications, data need to be represented as ranges or *interval*s of possible values, as opposed to scalar data:

- *Summarized data.* Analyzing reduced or summarized data sets can be more efficient, especially for implementing interactive applications [5], [6]. When several observations are grouped and collapsed into a single observation, data may need to be represented as value ranges. While it may sometimes be possible to associate statistical meanings to the intervals and (assuming that appropriate generative models, probability distributions, and conditioning strategies are found) it might be possible to leverage probabilistic matrix factorization techniques, such as [7], this approach may be infeasible or ineffective due to the lack of appropriate statistical representations and/or the cost.
- *Data with conflicts.* When a data set reflects knowledge integrated from different data sources, it might not be possible to assign a single scalar weight to each observation and an interval of possible values might be a more appropriate representations [6]. Moreover, when analyzing such integrated data, the resulting intervals may not have a statistical interpretation, beyond presenting the spread (i.e., minimum and maximum values) of the data.
- *Anonymized data.* Various privacy-preserving data publishing algorithms, such as recoding techniques [8], replace precise scalar values with less precise value ranges or intervals (such as those obtained through value generalization [8]). The resulting intervals (intentionally) do not represent any specific data distribution; consequently, associating a statistical interpretation to the interval is not necessarily appropriate. This means that probabilistic techniques for data analysis are not appropriate for anonymized data sets. In Section 6, we see that the proposed approach is highly effective for interval-valued data generated through generalization.
- *Imprecise data.* Data imprecision may be caused by various reasons, including limitations in measurement. For example, minute variations in multiple facial images from a single individual may be represented using interval-valued data (see [9] and Section 6.1.2). As we further discuss in Section 6.1.3, ambiguities in users' ratings in a collaborating filtering application may also be captured using interval-valued data [10], [11].

The key challenge in performing decompositions over interval-valued matrices is that definitions of basic algebraic operations, such as multiplication and inversion (needed to implement factorization operations), are not as straightforward for intervals as they are for scalars (see Section 2.1). Also, unlike scalars which are totally ordered, intervals are often partially ordered. Furthermore, a naive approach (which would exhaustively enumerate all possible decompositions) would significantly increase the computational complexity of the problem (which is already high for the case with scalar weights). Therefore, many basic operations need to be redefined to accommodate such non-scalar data and these need to be implemented in ways that avoid increases in computational costs.

- *Mao-Lin Li and K. Selçuk Candan are with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, 85281, USA.*
  E-mail: {mao-lin.li,candan}@asu.edu
- *Francesco Di Mauro and Maria Luisa Sapino are with the Department of Computer Science, University of Turin, Italy.*
  E-mail: {dimauro,mlsapino}@di.unito.it

Authorized licensed use limited to: Texas A M University. Downloaded on August 18,2020 at 05:59:53 UTC from IEEE Xplore. Restrictions apply.

1.2 Contributions of this Paper

In this paper, we study the problem of obtaining decompositions of interval-valued matrices:

- We present the *decomposition* problem for interval valued data sets. We introduce interval-valued algebra and discuss the core challenges presented by the decomposition of interval-valued data.
- We propose an interval-valued latent semantics alignment scheme and, relying on this, we develop algorithms for obtaining eigenvalue-based (such as SVD [3]) or probabilistic (such as PMF [7]) decomposition for interval-valued data matrices.
- We finally study the effectiveness of the proposed schemes in several applications, including face image analysis and collaborative filtering.

1.3 Organization of the Paper

The paper is organized as follows: We introduce the background and review the related work in Section 2. Section 3 introduces the problem and presents the key observations and mathematical formulations regarding interval-valued latent spaces. Section 4 presents the proposed *interval singular value decomposition (ISVD)* algorithm to obtain SVD decompositions in the presence of interval-valued data. Section 5 shows that the proposed semantic alignment technique can also be used in probabilistic matrix factorization scenarios. Section 6 reports our experimental results under diverse scenarios. We conclude the paper in Section 7.

2 Background and Related Works

2.1 Interval Algebra

We first formalize the definitions for interval-valued data and its algebraic operations:

Definition 1 (Interval representation)**.** *An interval $a_\dagger$ is a pair*

$$a_\dagger = [a_*, a^*], a_* \leq a^*,$$

where a_* is the minimum value and a^* is the maximum value of the interval $a_\dagger$. If $a_* = a^*$, then $a_\dagger$ is scalar.

Definition 2 (Interval span)**.** *Given an interval $a_\dagger$, the corresponding span is computed by:*

$$span(a_\dagger) = span([a_*, a^*]) = (a^* - a_*) \in \mathbb{R}$$

Definition 3 (Interval algebraic operations)**.** *Given two intervals, $[a_*, a^*]$ and $[b_*, b^*]$, we adopt the following interval algebraic operations on them [12]:*

- *addition:* $[a_*, a^*] + [b_*, b^*] = [a_* + b_*, a^* + b^*]$,
- *subtraction:* $[a_*, a^*] - [b_*, b^*] = [a_* - b^*, a^* - b_*]$,
- *multiplication:* $[a_*, a^*] \times [b_*, b^*] = [min(a_* \times b_*, a_* \times b^*, a^* \times b_*, a^* \times b^*), max(a_* \times b_*, a_* \times b^*, a^* \times b_*, a^* \times b^*)]$.

When one of the values, say a, is scalar, the multiplication $[a, a] \times [b_*, b^*]$ can be written as $[min(a \times b_*, a \times b^*), max(a \times b_*, a \times b^*)]$ and the corresponding value of the span is $span(a \times [b_*, b^*]) = a \times span([b_*, b^*])$.

Note that given the above definition of interval algebraic operations, more complex interval-valued operations, such as interval-valued matrix algebra, can be defined by replacing scalar addition, subtraction, and multiplication operations, with their interval-valued counterparts.

2.2 Matrix Factorization

Feature selection and dimensionality reduction techniques [13] usually involve some (often linear) transformation of the vector space containing the data to help focus on a few features (or combinations of features) that best discriminate the data in a given corpus. Among these transformations, Karhunen-Loeve Transform, KLT (also known as the principal component analysis, PCA [14]), and singular value decomposition, SVD [3] have the key property that the vectors selected as the dimensions of the space are *mutually orthogonal* and, hence, linearly independent (i.e., there is no redundancy among the dimensions). The resulting basis vectors are referred to as the *latent variables* [15] or the *latent semantics* of the data [3]. While KLT and SVD may result in negative values, in non-negative matrix factorization (NMF) [16], [17], factor matrices are non-negative and enable probabilistic interpretation of the results and discovery of generative models. Below, we outline three common matrix factorization schemes: singular value decomposition (SVD [3]), non-negative matrix factorization (NMF [16], [17]), and probabilistic matrix factorization (PMF [7]).

2.2.1 Singular Value Decomposition (SVD)

Let $M \in \mathbb{R}^{n \times m}$ represent the input matrix. Let the rank, r, be a positive integer $r \leq min(n, m)$. In this paper, we denote the value of i^{th} row and j^{th} column of M as $M[i, j]$. The j^{th} column vector of M is similarly denoted as $M[j]$. M can be decomposed into $M = U\Sigma V^T$ through singular value decomposition (SVD), where

- $U \in \mathbb{R}^{n \times r}$, and $UU^T = I_n$;
- $\Sigma \in diag(\mathbb{R}^r_+)$;
- $V^T = transpose(V)$, $V \in \mathbb{R}^{m \times r}$, and $VV^T = I_m$.

The columns of U, also called the left singular vectors of matrix M, are the eigenvectors of the $n \times n$ matrix, MM^T. The columns of V, or the right singular vectors of M, are the eigenvectors of the $m \times m$ matrix, $M^T M$. Note that both columns of U and columns of V are mutually orthogonal.

2.2.2 Non-negative Matrix Factorization (NMF)

Given a non-negative matrix $M \in \mathbb{R}^{n \times m}_+$, NMF factorizes M into two non-negative matrices $U \in \mathbb{R}^{n \times r}_+$ and $V \in \mathbb{R}^{m \times r}_+$ with target rank r, which minimize the L_2 loss function

$$\mathcal{L}_{NMF} = \|M - UV^T\|_F^2,$$

where $U \geq 0$, $V \geq 0$ and $\|.\|_F^2$ denotes the Frobenius norm. The approximated solutions for U and V are commonly found by *iterative update rules*, such as [17]

$$U[i, j] \leftarrow U[i, j] \frac{(MV)[i, j]}{(UV^T V)[i, j]}$$
$$V^T[i, j] \leftarrow V^T[i, j] \frac{(U^T M)[i, j]}{(U^T U V^T)[i, j]}.$$

Authorized licensed use limited to: Texas A M University. Downloaded on August 18,2020 at 05:59:53 UTC from IEEE Xplore. Restrictions apply.

[9] extended these to interval-valued matrices as follows:

$$\mathcal{L}_{I-NMF} = \|M_* - UV_*^T\|_F^2 + \|M^* - UV^{*T}\|_F^2$$
$$U[i,j] \leftarrow U[i,j]\frac{(MV)[i,j]}{(UV^TV)[i,j]}$$
$$V_*^T[i,j] \leftarrow V_*^T[i,j]\frac{(U^TM_*)[i,j]}{(U^TUV_*^T)[i,j]}$$
$$V^{*T}[i,j] \leftarrow V^{*T}[i,j]\frac{(U^TM^*)[i,j]}{(U^TUV^{*T})[i,j]}.$$

Note that this scheme, called I-NMF, factorizes the matrix into a scalar-valued U and an interval-valued $V_\dagger = [V_*, V^*]$.

2.2.3 Probabilistic Matrix Factorization (PMF)

Probabilistic matrix factorization (PMF) [7] assumes matrix entries are drawn from Gaussian distribution. In particular, given a matrix $M \in \mathbb{R}^{n\times m}$, the conditional distribution over the observed values is defined as

$$p(M[i,j]|U,V,\sigma^2) = \sum_{i=1}^{n}\sum_{j=1}^{m}[\mathcal{N}(M[i,j]|U_{[i,:]}{V_{[j,:]}}^T,\sigma^2)]^{\mathcal{I}_{ij}},$$

where $\mathcal{N}$ is the probability density function of Gaussian distribution with mean μ and variance σ^2, $\mathcal{I}_{ij}$ is the indicator function that is equal to 1 if $M[i,j]$ is not null and 0 otherwise. Above, $U_{[i,:]}$ and $V_{[j,:]}$ are row vectors[1] in U and V, such that $M[i,j] \simeq U_{[i,:]}{V_{[j,:]}}^T$, and they place zero-mean spherical Gaussian priors on latent semantics[2]. The factors, U and V, are computed via the loss function

$$\mathcal{L}_{PMF} = \|M - UV^T\|_F^2 + \lambda_U\|U\|_F^2 + \lambda_V\|V^T\|_F^2,$$

where λ_U=σ^2/σ_U^2, λ_V=σ^2/σ_V^2, and $\|.\|_F^2$ denotes the Frobenius norm. A local minimum of loss function $\mathcal{L}_{PMF}$ can be found via gradient descent in $U_{[i,:]}$ and ${V_{[j,:]}}^T$

$$\frac{\partial\mathcal{L}_{PMF}}{\partial U_{[i,:]}} = \sum_{j=1}^{m}(U_{[i,:]}{V_{[j,:]}}^T - M[i,j])V_{[j,:]} + \lambda U_{[i,:]}$$

$$\frac{\partial\mathcal{L}_{PMF}}{\partial {V_{[j,:]}}^T} = \sum_{i=1}^{n}(U_{[i,:]}{V_{[j,:]}}^T - M[i,j])U_{[i,:]}{}^T + \lambda {V_{[j,:]}}^T.$$

2.3 Analysis of Symbolic and Interval-valued Data

In the real world, data rarely comes in simple scalar form. Often variables of interest may take complex, often symbolic, forms, including sets, histograms, vectors, intervals, or probability distributions [18], [19], [20], [21]. This is especially true when data is aggregated [22] or anonymized [8]. Consequently, several data analysis tools, including regression [23], [24], canonical analysis [25], and multi-dimensional scaling [26], have been developed for symbolic and interval-valued data. Given the popularity of PCA in data analysis, several interval-valued PCA algorithms have also been proposed [27], [28], [29], [30], most of which leverage the specific statistical and geometric meanings of principal components of a system of variables. As discussed above, interval NMF and PMF [9] also have been studied to resolve alignment approximation in face analysis and rating approximation in collaborative filtering. In contrast, we develop a more general interval-valued latent semantic alignment algorithm which can be integrated in common matrix factorization approaches that directly leverages interval-valued properties.

1. In this paper, we use $X_{[i,:]}$ to denote i^{th} row vector and $X_{[:,j]}$ to denote j^{th} column vector of matrix X.

2. Note that here, and in the rest of the paper, we use $\simeq$ to denote "*approximately equal*"

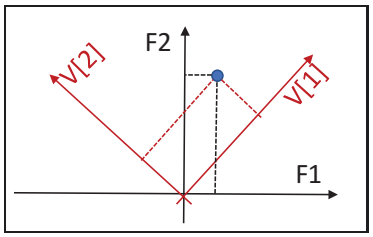
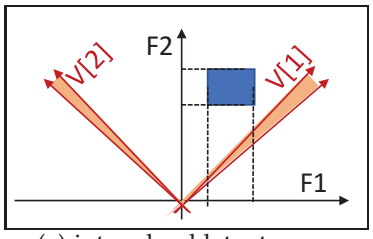
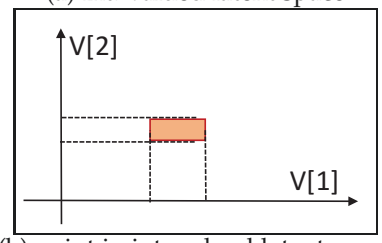


Fig. 1: Scalar latent semantic spaces



(a) int.-valued latent space



(b) point in int.-valued latent space

Fig. 2: Interval-valued latent semantic spaces

3 Interval-Valued Latent Spaces

In Section 2.2.1, we have seen that a scalar matrix can be decomposed into factor matrices (U and V) and core matrix (Σ). The columns in the factor matrices are referred to as latent semantics (LS) and, preferably, they are mutually orthogonal to serve as basis of the transformed space. Figure 1 shows a scalar-valued latent semantic space superimposed on the original space; the figure also shows a scalar-valued data point projected onto both original and latent spaces. Unfortunately, scalar-valued latent spaces are not sufficient to present interval-valued data.

3.1 Interval-Valued Decomposition

Here, we first extend the definition of singular valued decomposition taking into account the presence of interval-valued data.

Definition 4 (Interval-valued Decomposition)**.** *Given an interval-valued matrix, $M_\dagger \in \mathbb{R}^{n\times m}$, and a target rank $r \leq min(n,m)$, interval-valued SVD would decompose $M_\dagger$ into $M_\dagger \simeq U_\dagger\Sigma_\dagger V_\dagger^T$, such that*

- *$U_\dagger \in \mathbb{R}^{n\times r}$ and $V_\dagger \in \mathbb{R}^{m\times r}$ (potentially interval-valued) matrices, such that the columns of $U_\dagger$ and $V_\dagger$ are* quasi-orthonormal*; i.e., given column indexes h and l,*

Authorized licensed use limited to: Texas A M University. Downloaded on August 18,2020 at 05:59:53 UTC from IEEE Xplore. Restrictions apply.

– $U_{\dagger[:,h]} \cdot U_{\dagger[:,h]} \simeq 1; V_{\dagger[:,h]} \cdot V_{\dagger[:,h]} \simeq 1$ *and*
– $\forall_{h \neq l} \quad U_{\dagger[:,h]} \cdot U_{\dagger[:,l]} \simeq 0; V_{\dagger[:,h]} \cdot V_{\dagger[:,l]} \simeq 0.$

In other words, $U_\dagger U_\dagger^T \simeq I_n$ *and* $V_\dagger V_\dagger^T \simeq I_m$.

- $\Sigma_\dagger \in diag(\mathbb{R}_+^r)$ *and is potentially interval-valued.*

As we see in Figure 2(a), an interval-valued data entry has minimum and maximum values for each of the original basis vectors and can be considered as a (high-dimensional) box in the original space. As we also see in the figure, the transformation obtained through the decomposition process would represent each basis vector of the latent semantic space with an interval-valued column vector, $V_\dagger[:,h]$, bounded with a minimum and a maximum. Given this interval-valued latent space defined by the matrix $V_\dagger$, the projection of the data entry, defined by the row $M_\dagger[i,:]$, onto the basis vector $V_\dagger[:,h]$ would be represented as an interval defined by $U_\dagger[i,h] \times \Sigma_\dagger[h,h]$, where $\times$ is an interval algebraic multiplication as defined in Section 2.1 (Figure 2(b)).

3.2 Imprecision of Interval-Valued Latent Spaces

Unlike in the basic SVD operation, we cannot seek for an exact decomposition of the input matrix $M_\dagger$, nor we can treat the constraint $U_\dagger U_\dagger^T = V_\dagger V_\dagger^T = I$ as a hard constraint: The reason for both of these relaxations is implicit in the definition of interval-valued multiplication given in Section 2.1.

Theorem 1 (Scalar Theorem for $\times$)**.** *Let* $a = [a_*, a^*]$ *and* $b = [b_*, b^*]$ *be two non-zero intervals. Let* $c = [c_*, c^*] = a \times b$*, where* $\times$ *is interval-valued matrix multiplication. Then,*

$$(c_* = c^*) \rightarrow (a_* = a^*) \wedge (b_* = b^*).$$

Proof 1. *As we have seen in Section 2.1,* $[a_*, a^*] \times [b_*, b^*] = [c_*, c^*]$*, where*

$$\begin{aligned} c_* = \quad & min(a_* \times b_*, a_* \times b^*, a^* \times b_*, a^* \times b^*), \\ c^* = \quad & max(a_* \times b_*, a_* \times b^*, a^* \times b_*, a^* \times b^*)]. \end{aligned}$$

If, however, we are further given that $c = a \times b$ *is scalar, this implies that*

$$\begin{aligned} min(a_* \times b_*, a_* \times b^*, a^* \times b_*, a^* \times b^*) = \\ max(a_* \times b_*, a_* \times b^*, a^* \times b_*, a^* \times b^*)], \end{aligned}$$

which can only happen if either a *or* b *is a zero interval or we have* $(a_* = a^*)$ *and* $(b_* = b^*)$. ◊

In other words, the only way the multiplication of two interval values would return a scalar value is when both $[a_*, a^*]$ and $[b_*, b^*]$ themselves are scalars; i.e., $a_* = a^*$ and $b_* = b^*$. This further implies that the constraint $U_\dagger U_\dagger^T = V_\dagger V_\dagger^T = I$ cannot be exactly satisfied.

Theorem 2 (Scalar Theorem for Dot Product)**.** *Let* x *be a* k *dimensional interval-valued vector. Then the dot product of* x *with itself (i.e.,* $x.x$*) is scalar-valued only if all the entries in* x *are scalar-valued.*

Proof 2. *From the definition of dot product, we have*

$$x.x = \sum_{i=1\ldots k} x[i] \times x[i], where\ x \in \mathbb{R}^k$$

which is, by definition of interval-valued matrix multiplication,

$$x.x = \sum_{i=1\ldots k} [min(x_*[i]^2, x^*[i]^2), max(x_*[i]^2, x^*[i]^2)].$$

Since both $x_*[i]^2$ *and* $x^*[i]^2$ *are non-negative, the only way the equality*

$$\sum_{i=1\ldots k} min(x_*[i]^2, x^*[i]^2) = \sum_{i=1\ldots k} max(x_*[i]^2, x^*[i]^2)$$

holds is if for all i*, we have* $x_*[i] = x^*[i]$. ◊

Note that, as a corollary of above theorem, we cannot have XX^T scalar-valued unless X is scalar-valued.

Corollary 1. *Let* X *be an interval-valued matrix and let* S *be a scalar-valued matrix. Then, the following is true:*

$$(XX^T = S) \rightarrow X_* = X^*.$$

◊

This is because for S to be scalar-valued, the diagonal of S should be scalar-valued. Since for all h, we also have $S[h] = X[h].X[h]$, then by Theorem 2, if S is scalar-valued than each and every row, $X[h]$ (and consequently the entire matrix X) must also be scalar-valued.

Corollary 2. *Corollary 1 further implies that* $U_\dagger U_\dagger^T$ *and* $V_\dagger V_\dagger^T$ *cannot be equal to scalar-valued matrix,* I*, which means that an exact decomposition of interval-valued matrices is not possible.* ◊

Given the inherent imprecision in interval-valued latent spaces, we define decomposition accuracy as follows:

Definition 5 (Decomposition Accuracy)**.** *Given the above, let* $M_\dagger$ *be an interval-valued matrix, decomposed into* $U_\dagger$*,* $\Sigma_\dagger$*, and* $V_\dagger^T$ *and* $\tilde{M}_\dagger$ *be the interval-valued matrix obtained by*

$$\tilde{M}_\dagger = U_\dagger \Sigma_\dagger V_\dagger^T,$$

recombining $U_\dagger$*,* $\Sigma_\dagger$*, and* $V_\dagger^T$ *using the dot product operation, defined in terms of the interval algebraic operations introduced in Section 2.1. Let* $\tilde{M}_*$ *and* $\tilde{M}^*$*be the scalar-valued matrices obtained by considering the minimum and maximum values of the intervals in* $\tilde{M}_\dagger$*, respectively. We measure the* reconstruction accuracy *in terms of the Frobenius norms of the difference matrices:*

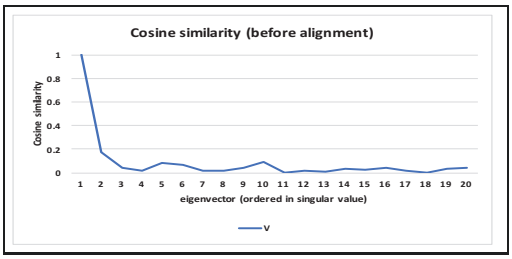
$$\Delta(M_*, \tilde{M}_*) = \left(\frac{\|M_* - \tilde{M}_*\|}{\|M_*\|}\right); \Delta(M^*, \tilde{M}^*) = \left(\frac{\|M^* - \tilde{M}^*\|}{\|M^*\|}\right).$$

To obtain a single measure of accuracy, we convert these into accuracies, $\Theta(M_*, \tilde{M}_*) = max(0, 1 - \Delta(M_*, \tilde{M}_*))$ *and* $\Theta(M^*, \tilde{M}^*) = max(0, 1 - \Delta(M^*, \tilde{M}^*))$*, and combine them using Harmonic mean:* $\Theta_{HM}(M, \tilde{M}) = HarmonicMean(\Theta(M_*, \tilde{M}_*), \Theta(M^*, \tilde{M}^*))$.

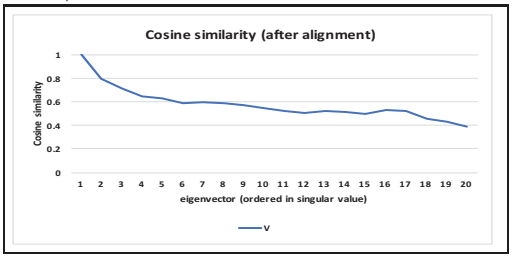
3.3 Interval-Valued Latent Semantic Alignment (ILSA)

Let $V_\dagger$ be an interval-valued factor matrix representing the latent semantics of the interval-valued data, $M_\dagger$. As we have discussed in Section 3.1 (and visualized in Figure 2(a)) we would expect that the columns of the $V_\dagger$ will be quasi-orthogonal (i.e., for all $i \neq j$, $V_{\dagger[:,i]} \cdot V_{\dagger[:,j]} \simeq 0$ and, for all i, $V_{\dagger[:,i]} \cdot V_{\dagger[:,i]} \simeq 1$) even though, as we have seen in Section 3.2, we cannot expect that $V_{\dagger[:,i]} \cdot V_{\dagger[:,j]} = 0$ or $V_{\dagger[:,i]} \cdot V_{\dagger[:,i]} = 1$. In fact, since per Theorem 2, the only way the dot product of a vector, x, with itself is scalar-valued is when this vector itself is scalar-valued, the only way we can even approximate the quasi-orthonormality constraint is when for all i, we

Authorized licensed use limited to: Texas A M University. Downloaded on August 18,2020 at 05:59:53 UTC from IEEE Xplore. Restrictions apply.



(a) $cos(V_{*[:,i]}, V^*_{[:,i]})$ before the alignment (the higher the better)



(b) $cos(V_{*[:,i]}, V^*_{[:,i]})$ after the alignment (the higher the better)

Fig. 3: Cosine similarities among min and max vectors: the higher the cosine similarity, the more similar the minimum and maximum factor matrices are, indicating higher precision in the decomposition (averaged for 100 matrices as described in Table 1, Section 6, default configuration)

have $V_{*[:,i]} \simeq V^*_{[:,i]}$, where $V_{*[:,i]}$ is the interval-valued vector consisting of the minimum entries of the intervals in $V_{\dagger[:,i]}$, whereas $V^*_{[:,i]}$ consists of the maximums.

Given this observation, we can now present an interval latent semantic alignment problem, which would optimally combine minimum and maximum vectors to form an interval-valued latent space. In particular, since we have seen above that the interval-valued vectors which define the latent semantics of the data must be such that $v_* \simeq v^*$, we can formulate this as a stable marriage (or stable matching) problem between two equally sized sets of vectors:

Problem 1 (Stable Min-Max Vector Alignment)**.** *Let us be given two sets of unit-length vectors, $V_1 = \{\vec{v}_{*1}, \ldots, \vec{v}_{*r}\}$ and $V_2 = \{\vec{v}^*_1, \ldots, \vec{v}^*_r\}$. Let us also be given the preference function between two vectors, x and y, as $abs(cos(x, y))$. Our goal is to find pairing/mapping $\mu = \{\mu_1, \ldots, \mu_r\}$ (where $\mu_i = \langle \mu_{l,i}, \mu_{h,i} \rangle$) among the vectors from V_1 and V_2, such that there are no two minimum and maximum vectors, $v_a \in V_1$ and $v_b \in V_2$, which would both align better with each other than their current assignments.*

While it is known that this problem has an $O(r^2)$ solution [31], this does not guarantee that the solution is also optimal in terms of maximizing overall alignments among the paired minimum and maximum vectors. Therefore, we consider an alternative formulation of the problem:

Problem 2 (Optimal Min-Max Vector Alignment)**.** *Let us be given two sets of unit-length vectors, $V_1 = \{\vec{v}_{*1}, \ldots, \vec{v}_{*r}\}$ and $V_2 = \{\vec{v}^*_1, \ldots, \vec{v}^*_r\}$. Let us also be given the preference function between two vectors, x and y, as $abs(cos(x, y))$. Our goal is to find pairing/mapping $\mu = \{\mu_1, \ldots, \mu_r\}$ (where $\mu_i = \langle \mu_{l,i}, \mu_{h,i} \rangle$) among the vectors from V_1 and V_2, such that $\sum_{1 \leq i \leq r} abs(cos(\vec{v}_{*\mu_{l,i}}, \vec{v}^*_{\mu_{h,i}}))$ is maximum, where abs means absolute value and cos means cosine similarity.*

This is an instance of the *linear assignment* problem [32] and can be solved using one of the many algorithms, such as the Hungarian algorithm with $O(r^3)$ time complexity.

Once a mapping, μ, is identified, then for any $1 \leq i \leq r$ such that $cos(\vec{v}_{*\mu_{l,i}}, \vec{v}^*_{\mu_{h,i}})$ is negative, we can realign their directions by multiplying $\vec{v}^*_i$ with -1 so that both $\vec{v}_{*i}$ and $\vec{v}^*_i$ are pointing in a similar direction. As we see in Figure 3, the algorithm helps improve the alignment between the minimum an maximum basis vectors, especially for low rank entries which are often critical in many analysis and recommendation applications.

3.4 Alternative Decomposition Strategies

We note that, given an interval-valued matrix, $M_\dagger$, different applications may seek different types of factor matrices and core matrix: some applications may need all of the resulting matrices be interval-valued, some others may need to capture the underlying imprecision in the form of interval-valued factor and/or core matrices, while others may require all matrices to be scalar. In particular, we consider three distinct application semantics:

3.4.1 Decomp. Target (a): Interval-Valued $U_\dagger$, $\Sigma_\dagger$, and $V_\dagger$

In the most general decomposition strategy, which we refer to as **decomposition target-a (option-a)**, the application captures the imprecision in the data and represents it in terms of imprecise factor and core matrices. We construct interval-valued $U_\dagger$, $\Sigma_\dagger$, and $V_\dagger$ matrices by combining the corresponding entries from U_*, Σ_*, V_* and from U^*, Σ^*, V^*, respectively. More specifically, for all $1 \leq i \leq m$ and $1 \leq j \leq r$, we have

$$U_\dagger[i,j] = \begin{cases} \langle \vec{u}_{*\mu_{l,j}}[i], \vec{u}^*_{\mu_{h,j}}[i] \rangle, \text{if } \vec{u}_{*\mu_{l,j}}[i] \leq \vec{u}^*_{\mu_{h,j}}[i] \\ avg(\vec{u}_{*\mu_{l,j}}[i], \vec{u}^*_{\mu_{h,j}}[i]), \text{otherwise} \end{cases}$$

In particular, when the minimum and maximum entries in U_* and U^* do not lead to a valid interval, we replace the entry with the mean of the values to represent the best approximation we have for that entry. For all $1 \leq j \leq r$, $\Sigma_\dagger[j,j]$ and, for all $1 \leq i \leq m$ and $1 \leq j \leq r$, $V_\dagger[i,j]$ are computed similarly.

3.4.2 Decomp. Target (b): Scalar U and V; Interval $\Sigma_\dagger$

One difficulty with the above construction is that the columns of the $U_\dagger$ and $V_\dagger$ matrices are interval-valued and, thus, it is difficult to interpret them as the alternative basis of a vector space. In fact, while for all $1 \leq j \leq r$, $\vec{u}_{*j}$ and $\vec{u}^*_j$ are themselves unit length, the same cannot be said for the interval-valued vector $\vec{u}_j$ since the norm operation is not defined for interval-valued vectors. Therefore, in **decomposition target b (option-b)**, the factors matrices are maintained scalar-valued (i.e., precise), whereas the core which represents the relationships among the factors are interval-valued to capture the underlying imprecision. We first obtain scalar U by (a) first constructing an X matrix, such that, for all

Authorized licensed use limited to: Texas A M University. Downloaded on August 18,2020 at 05:59:53 UTC from IEEE Xplore. Restrictions apply.

$1 \leq i \leq n$ and $1 \leq j \leq r$, $X[i,j] = avg(\vec{u}_{*\mu_{l,j}}[i], \vec{u}^*_{\mu_{h,j}}[i])$ and then (b) renormalizing the columns of the X matrix:

$$U[i,j] = \frac{X[i,j]}{\sqrt[2]{\sum_{1 \leq h \leq n} X[h,j]^2}}.$$

Similarly, the scalar V matrix is obtained by first constructing a Y matrix and renormalizing the columns of Y. Intuitively, U and V matrices are scalar-valued matrices, both consisting of unit length columns. Given these, we then obtain the core matrix $\Sigma_\dagger[j,j]$ for $1 \leq j \leq r$ as

$$\begin{cases} \rho_j \times \langle \Sigma_*[\mu_{l,j}, \mu_{l,j}], \Sigma^*[\mu_{h,j}, \mu_{h,j}] \rangle, \\ \quad \text{if } \Sigma_*[\mu_{l,j}, \mu_{l,j}] \leq \Sigma^*[\mu_{h,j}, \mu_{h,j}] \\ \rho_j \times avg(\Sigma_*[\mu_{l,j}, \mu_{l,j}], \Sigma^*[\mu_{h,j}, \mu_{h,j}]), \\ \quad \text{otherwise} \end{cases}$$

where $\rho_j = \left(\sum_{1 \leq h \leq n} X[i,j]\right) \times \left(\sum_{1 \leq h \leq m} Y[i,j]\right)$. Intuitively, we rescale the non-zero values of the interval-valued core matrix $\Sigma_\dagger$ to account for the renormalization factors of the corresponding columns of U and V.

Option-b leads to scalar-valued factor matrices, U and V, and an interval-valued core matrix, $\Sigma_\dagger$. Intuitively, this provides alternative basis vectors for the rows and columns of the input matrix M and an interval-valued *strength* for each basis vector (captured by the corresponding entries of the diagonal matrix, $\Sigma_\dagger$).

3.4.3 Decomp. Target (c): Scalar U, Σ, and V

As a third option, we can obtain scalar-valued approximation for Σ by (a) first renormalizing U and V in L2-Norm as in Section 3.4.2 and, then, (b) replacing each diagonal entry of Σ with the mean value of the corresponding interval. This approach, which we refer to as **decomposition target-c (option-c)**, leads to a scalar-valued core matrix, Σ, along with scalar-valued factor matrices, U and V. In this case, despite the imprecision in the input matrix, the application requires scalar-valued factors and core (e.g. to be compatible with algorithms and tools that assume scalar-valued factor and core matrices to support recommendations).

4 Interval SVD (ISVD)

In this section, we will introduce alternative decomposition strategies to obtain SVD of interval-valued matrices, as defined in Section 3.1. We start with a naive approach.

4.1 Naive Approach (ISVD$_0$): Average and Decompose

As we discussed in Section 3.2, we are not seeking an exact decomposition of interval-valued matrices, but only an approximation. Therefore, the very first approach to consider for solving the decomposition problem is to simplify the input matrix into a simple scalar matrix by representing each interval by its average (see the column labeled **ISVD$_0$** in Figure 4): Let $M[i,j]$ be an interval-valued entry $[a_*, a^*]$ in M and let the corresponding entry in the average matrix be $M_{avg}[i,j] = \frac{a_* + a^*}{2}$. We approximate the singular value decomposition of M with rank r as

$$ISVD_0(M,r) \simeq SVD_{avg}(M,r) = SVD(M_{avg}, r).$$

This approach results in U and V matrices that are scalar and orthonormal and a Σ core matrix that is also scalar-valued (i.e., it is compatible only with the **decomposition target-c**, discussed in Section 3.4).

4.2 ISVD$_1$: Decompose and Align

In this section, we note that, in order to support a quasi-orthogonal latent-semantics space visualized in Figure 2(a), the vectors in each of V_1 (and similarly in V_2) must be as mutually orthogonal as possible. Therefore, we can obtain V_1 and V_2 by decomposing the minimum and maximum matrices, M_* and M^*, of the given interval-valued matrix, $M_\dagger$, separately and aligning the resulting factor/core matrices: Let us split $M_\dagger$ into two scalar-valued matrices M_* and M^* such that if $M_\dagger[i,j]$ is an interval $[a_*, a^*]$, then $M_*[i,j] = a_*$ and $M^*[i,j] = a^*$. Given these, we can obtain two separate matrix decompositions of rank r, $SVD(M_*, r) = U_* \Sigma_* V_*^T$ and $SVD(M^*, r) = U^* \Sigma^* V^{*T}$.

4.2.1 Latent Semantic Alignment

Once we obtain $U_\dagger = [U_*, U^*]$,$\Sigma_\dagger = [\Sigma_*, \Sigma^*]$,$V_\dagger = [V_*, V^*]$ after decomposition, we then apply interval latent semantic alignment (ILSA, Section 3.3) to align the eigenvectors in matrix $V_\dagger$ and adjust the rank-order of the rows of $\Sigma_\dagger$ and the rank-order and direction of the columns in $U_\dagger$. This is visualized under the column labeled **ISVD$_1$** in Figure 4.

Note that in the discussion in Section 3.3, $V_\dagger$ is taken as an interval matrix, where entries in V_* are smaller than the corresponding entries in V^* (similarly for $\Sigma_\dagger$, and $U_\dagger$). This, however, does not necessarily hold in practice since $SVD(M_*, r) = U_* \Sigma_* V_*^T$ and $SVD(M^*, r) = U^* \Sigma^* V^{*T}$ can lead to situations where $V_*[i,j] > V^*[i,j]$. This, however, does not impact the latent semantic alignment process as the algorithm does not rely on $V_*[i,j]$ being $\leq V^*[i,j]$ and only seeks an alignment where $V_*[i,j] \simeq V^*[i,j]$.

4.2.2 Construction of $U_\dagger$, $\Sigma_\dagger$, and $V_\dagger$ Matrices

The final step is to construct the $U_\dagger$, $\Sigma_\dagger$, and $V_\dagger$ matrices corresponding to the singular value decomposition of the interval-valued input matrix, $M_\dagger$, according to the decomposition target described in Section 3.4: this leads to three schemes, **ISVD$_1$-a**, with interval-valued $U_\dagger$, $\Sigma_\dagger$, and $V_\dagger$ matrices, **ISVD$_1$-b**, with scalar-valued U and V and interval-valued $\Sigma_\dagger$, and **ISVD$_1$-c**, with scalar-valued U, Σ, and V.

4.3 ISVD$_2$: Decompose, Solve, Align

The ISVD$_1$ algorithm described in the previous section first splits the interval-valued input matrix, $M_\dagger$, into its minimum and maximum valued components, M_* and M^*, decomposes them independently, and then aligns the resulting latent semantics before recomposing the resulting $U_\dagger$, $\Sigma_\dagger$, and $V_\dagger$ matrices. A potential problem with this strategy is that M_* and M^* are decomposed independently and this may introduce errors in the overall decomposition. An alternative strategy would be to first finding left and/or right singular vectors of the interval-valued matrix $M_\dagger$ and using these to seek for the final $U_\dagger$, $\Sigma_\dagger$, and $V_\dagger$ matrices.

Let us remember from Section 2.2.1 that when M is a scalar-valued matrix and $M = U \Sigma V^T$ is its singular value decomposition, then we have the following: (a) the columns of U, also called the left singular vectors of matrix M, are the eigenvectors of the $m \times m$ matrix, MM^T; (b) the columns of V, or the right singular vectors of M, are the eigenvectors of the $n \times n$ matrix, $M^T M$; and (c) the diagonal entries of Σ, also known as the singular values of M, are the square

Authorized licensed use limited to: Texas A M University. Downloaded on August 18,2020 at 05:59:53 UTC from IEEE Xplore. Restrictions apply.

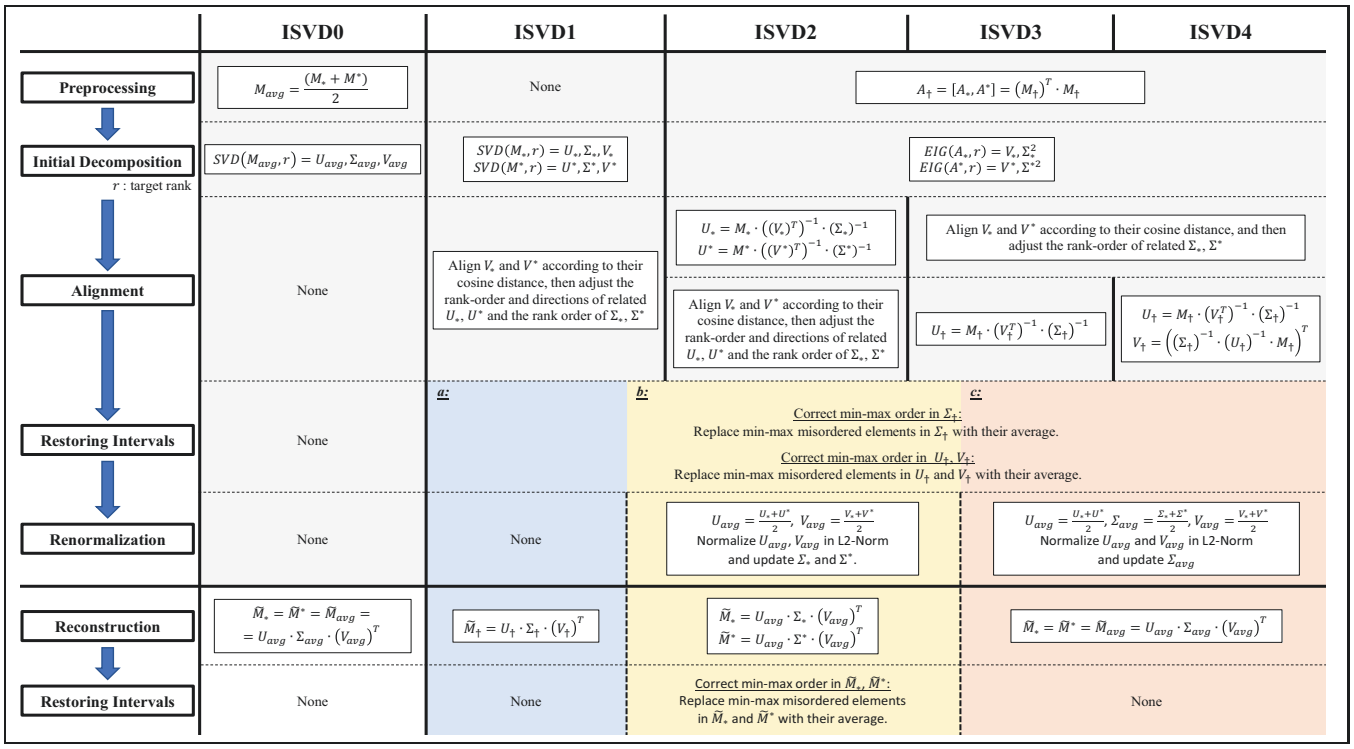


Fig. 4: ISVD decomposition strategies for interval-valued input data

roots of the eigenvalues of MM^T and M^TM. Therefore, a more principled way to seek the SVD of an interval-valued matrix would be to first compute an $n \times n$ square matrix, $A = MM^T$, through interval-valued matrix multiplication (discussed in Section 2.1) and, then, seek the eigenvectors and eigenvalues of this matrix.

4.3.1 Eigen-Decomposition of a Square Matrix

[33], [34] define the eigen-decomposition of an interval valued matrix $A_\dagger$ as a solution to the $A_\dagger \vec{v_\dagger} = \lambda \vec{v_\dagger}$ problem, where $\vec{v_\dagger}$ is an interval-valued eigen-vector and λ is a scalar. [33], [35] provide bounds and linear-programming based solutions to solve this problem. In contrast, [36] reformulates the problem as seeking interval eigen-values of scalar-valued eigen-vectors. In this paper, we consider a more general solution, where we seek *interval-valued eigen-values, corresponding to interval eigen-vectors*: Let $A_\dagger = M_\dagger^T M_\dagger$ be an interval-valued square matrix. For simplicity, let us assume that $A_\dagger$ is a 3×3 interval matrix:

$$A_\dagger = \begin{bmatrix} [a_*, a^*] & [b_*, b^*] & [c_*, c^*] \\ [d_*, d^*] & [e_*, e^*] & [f_*, f^*] \\ [g_*, g^*] & [h_*, h^*] & [i_*, i^*] \end{bmatrix}.$$

Let $\Lambda_\dagger = [\lambda_*, \lambda^*]$ be an interval-valued eigenvalue of A and $\vec{v} = [x, y, z]^T$ be the corresponding interval-valued eigenvector. Since (by definition) $A_\dagger \vec{v_\dagger} = \lambda_\dagger \vec{v_\dagger}$, we have

$$\begin{bmatrix} [a_*, a^*] & [b_*, b^*] & [c_*, c^*] \\ [d_*, d^*] & [e_*, e^*] & [f_*, f^*] \\ [g_*, g^*] & [h_*, h^*] & [i_*, i^*] \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = [\lambda_*, \lambda^*] \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

By applying interval-valued algebra and simple matrix algebra, we can rewrite the above equality as

$$\begin{bmatrix} [a_*x + b_*y + c_*z, a^*x + b^*y + c^*z] \\ [d_*x + e_*y + f_*z, d^*x + e^*y + f^*z] \\ [g_*x + h_*y + i_*z, g^*x + h^*y + i^*z] \end{bmatrix} = \begin{bmatrix} [\lambda_*x, \lambda^*x] \\ [\lambda_*y, \lambda^*y] \\ [\lambda_*z, \lambda^*z] \end{bmatrix}.$$

It is easy to see that the above equality can be split into a minimum and maximum component:

$$\begin{bmatrix} a_*x + b_*y + c_*z \\ d_*x + e_*y + f_*z \\ g_*x + h_*y + i_*z \end{bmatrix} = \begin{bmatrix} \lambda_*x \\ \lambda_*y \\ \lambda_*z \end{bmatrix}$$
$$\begin{bmatrix} a^*x + b^*y + c^*z \\ d^*x + e^*y + f^*z \\ g^*x + h^*y + i^*z \end{bmatrix} = \begin{bmatrix} \lambda^*x \\ \lambda^*y \\ \lambda^*z \end{bmatrix}.$$

In other words, to obtain the eigen-decomposition of the original interval-valued matrix, $A_\dagger$, we can instead seek the eigen-decompositions of the matrices A_* and A^*, where

$$A_* = \begin{bmatrix} a_* & b_* & c_* \\ d_* & e_* & f_* \\ g_* & h_* & i_* \end{bmatrix} \text{ and } A^* = \begin{bmatrix} a^* & b^* & c^* \\ d^* & e^* & f^* \\ g^* & h^* & i^* \end{bmatrix}$$

to obtain $IEigDec(A_*, r) = (V_*, \Lambda_*)$ and $IEigDec(A^*, r) = (V^*, \Lambda^*)$, where the matrices V_* and V^* capture the eigenvectors and the vectors Λ_* and Λ^* are diagonal matrices encoding the square roots of the eigenvalues of matrices, A_* and A^*.

4.3.2 Recovery of the U_* and U^* Matrices

Given the eigen-decomposition of $A_\dagger = M_\dagger^T M_\dagger$, which provides us V_* and V^* and the Λ_* and Λ^*, the next step of the process involves computation of the U_* and U^* matrices using V_*, V^*, $\Lambda_*(= \Sigma_*^2)$, and $\Lambda^*(= \Sigma^{*2})$ matrices. This can be achieved by directly relying on the definition of SVD: $U_* = M_*((V_*)^T)^{-1}(\Sigma_*)^{-1}$ and $U^* = M^*((V^*)^T)^{-1}(\Sigma^*)^{-1}$.

4.3.3 Latent Semantic Alignment and Construction of $U_\dagger$, $\Sigma_\dagger$, and $V_\dagger$ Matrices

Once the U_*, U^*, V_* and V^* are obtained, we can construct the $U_\dagger$, $\Sigma_\dagger$, and $V_\dagger$ matrices by following the same

Authorized licensed use limited to: Texas A M University. Downloaded on August 18,2020 at 05:59:53 UTC from IEEE Xplore. Restrictions apply.

steps in Section 4.2.1. These steps are visualized under the column labeled **ISVD$_2$** in Figure 4. Note that, as also discussed in Section 4.2.1, the decompositions of A_* and A^* can lead to situations where $V_*[i,j] > V^*[i,j]$ and/or $\Sigma_*[i,j] > \Sigma^*[i,j]$. Similarly, U_* and U^* computed using these matrices can contain cases where $U_*[i,j] > U^*[i,j]$. As before, these entries do not negatively impact the latent semantic alignment process and, thus, are corrected as part of the final step of the decomposition process (Section 3.4) which returns (**ISVD$_2$-a**) interval-valued $U_\dagger$, $\Sigma_\dagger$, and $V_\dagger$ matrices, (**ISVD$_2$-b**) scalar-valued U and V and interval-valued $\Sigma_\dagger$, or (**ISVD$_2$-c**) scalar-valued U, Σ, and V.

4.4 ISVD$_3$: Decompose, Align, Solve

In this section, we further consider an alternative approach where, instead of obtaining U_* and U^* matrices independently, we (a) first solve for the $V_\dagger$ and $\Sigma_\dagger$ matrices, (b) align the V_* and V^* matrices, and then (c) use these to compute the $U_\dagger$ matrix through interval-valued matrix inversion (Figure 4). Let, as before, $A_\dagger$ be an $n \times n$ square matrix, $A_\dagger = M_\dagger M_\dagger^T$, obtained through interval-valued matrix multiplication (Section 2.1). Let matrices V_* and V^* capture the eigenvectors and Σ_* and Σ^* be diagonal matrices encoding the square roots of the eigenvalues of matrices A_* and A^*, respectively, as described in Section 4.3.1.

4.4.1 Latent Semantic Alignment

Given the V_* and V^* matrices along with the corresponding Σ_* and Σ^*, we first seek the latent semantic alignment through ILSA, as described in Section 3.3. Given the latent alignment, we can next compute the interval-valued $V_\dagger$ matrix as in Section 3.4.1. The interval-valued core matrix $\Sigma_\dagger$ is also computed similarly, using the Λ_* and Λ^* matrices.

4.4.2 Recovery of the matrix $U_\dagger$

Given the interval valued matrices $V_\dagger$ and $\Sigma_\dagger$, we can then compute the interval-valued $U_\dagger$ matrix relying on the definition of SVD: $U_\dagger = M_\dagger((V_\dagger)^T)^{-1}(\Sigma_\dagger)^{-1}$. This, however, requires us to invert interval-valued matrices. We discuss how to achieve this next.

4.4.2.1 *Inverse of a Non-Negative Interval-Valued, $\Sigma_\dagger$:* Let $S = \Sigma_\dagger$ be a $r \times r$ diagonal matrix, where the entries in the diagonal may be interval values and the rest of the entries are 0. The entries in the diagonal are non-negative since they are obtained through square-roots of eigenvalues. We seek an $r \times r$ diagonal matrix, S^{-1}, such that $S\ S^{-1} = \tilde{I}$, where $\tilde{I}$ is a $k \times k$ interval-valued matrix, *approximately* equal to the identity matrix. More specifically, for all $1 \leq i \leq r$, we have $\tilde{I}(i,i) = [1-\epsilon_i, 1+\epsilon_i]$, where $0 \leq \epsilon_i \leq 1$.

Given the above, let $S(i,i) = [s_{i*}, s_i^*]$ and $S^{-1}(i,i) = [\sigma_{i*}, \sigma_i^*]$. We seek σ_{i*} and σ_i^* values that minimize the value of ϵ_i subject to the constraints (1) $s_{i*} \times \sigma_{i*} = 1-\epsilon_i$, (2) $s_i^* \times \sigma_i^* = 1+\epsilon_i$, (3) $\sigma_{i*} \leq \sigma_i^*$, and (4) $0 \leq \epsilon_i \leq 1$. Note that the first two constraints can be combined to obtain $\sigma_{i*} = \frac{1-\epsilon_i}{s_{i*}}$ and $\sigma_i^* = \frac{1+\epsilon_i}{s_i^*}$, which, together with the third constraint, lead to the inequality $0 \leq \frac{s_i^* - s_{i*}}{s_i^* + s_{i*}} \leq \epsilon_i \leq 1$.

This constraint shows that ϵ_i is minimum when it is equal to $\frac{s_i^* - s_{i*}}{s_{i*} + s_{i*}}$, and this case works when $\sigma_{i*} = \sigma_i^*$. Therefore, elements in interval inverse matrix S^{-1} have only scalar values, where $\sigma_{i*} = \sigma_i^*$. Additionally, when $\sigma_{i*} = \sigma_i^* = \sigma_i$, the equations, $s_{i*} \times \sigma_i = 1-\epsilon_i$ and $s_i^* \times \sigma_i = 1+\epsilon_i$, together can be used to get $\sigma_i = \frac{2}{s_{i*} + s_i^*}$.

4.4.2.2 *Inverse of an Interval-Valued Matrix, $V_\dagger$:* Note that the above inversion process is applicable to only square, diagonal interval-valued matrices. Unfortunately, the factor matrix, $V_\dagger$, is not a diagonal matrix. Therefore, its inverse cannot be obtained using this method. We therefore, need to approximate $V_\dagger^{-1}$, by (a) first computing V_{avg} by replacing each interval in $V_\dagger$ with its mean, (b) inverting V_{avg}, and (c) returning V_{avg}^{-1} as $V_\dagger^{-1}$. Note that, one difficulty with this approach is that V_{avg} may not always be invertible. We, therefore, first check whether V_{avg} is *well-conditioned* and, if that is not the case, use Moore-Penrose pseudo-inverse for V_{avg}^{-1} (or alternatively, revert back to ISVD$_2$). This can be considered as a generalization of the inverse matrix – we adapt SVD to compute pseudo-inverse and replace singular values smaller than 0.1 with zero.

4.4.3 Construction of $U_\dagger$, $\Sigma_\dagger$, and $V_\dagger$ Matrices

Given $M_\dagger$ and the inverses of the factor matrix $V_\dagger$ and the core matrix $\Sigma_\dagger$, we compute the factor matrix $U_\dagger$ using the equation $U_\dagger = M_\dagger((V_\dagger)^T)^{-1}(\Sigma_\dagger)^{-1}$. This is visualized under the column labeled **ISVD$_3$** in Figure 4.

As discussed in Sections 4.2.1 and 4.3.3, the decompositions of A_* and A^* can lead to situations where $V_*[i,j] > V^*[i,j]$ and/or $\Sigma_*[i,j] > \Sigma^*[i,j]$. These entries do not negatively impact the latent semantic alignment process. They, moreover, also do not impact the computation of $U_\dagger = M_\dagger((V_\dagger)^T)^{-1}(\Sigma_\dagger)^{-1}$, since (as described above) both $\Sigma_\dagger)^{-1}$ and $((V_\dagger)^T)^{-1}$ involve averaging of the interval-valued entries. Therefore, these misordered elements are corrected as part of the final step of the decomposition process described in Section 3.4[3], which returns, depending on the decomposition target, (**ISVD$_3$-a**) interval-valued $U_\dagger$, $\Sigma_\dagger$, and $V_\dagger$ matrices, (**ISVD$_3$-b**) scalar-valued U and V and interval-valued $\Sigma_\dagger$, or (**ISVD$_3$-c**) scalar-valued U, Σ, and V.

4.5 ISVD$_4$: Decompose, Align, Solve, Recompute

In this section, we consider a variant of ISVD$_3$, which inherits its benefits, but further reduces the degree of imprecision in the resulting interval-valued factor matrices.

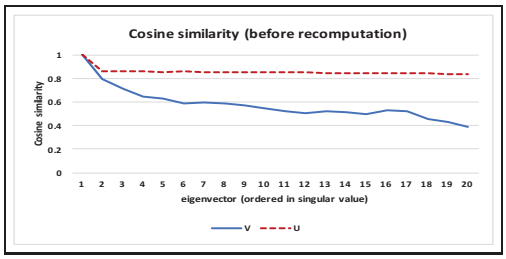
4.5.1 Recomputation of the Left Singular Vectors

As we discussed in the previous sections, the decomposition of interval-valued matrices potentially results in interval-valued factor matrices – as discussed in Section 3.3, the smaller are the sizes of the intervals in these factor matrices, the more precise are the resulting factors. Therefore, as discussed before, it is important to keep these intervals smaller (or equivalently, keep the minimum and maximum versions of the eigenvectors similar) as long as this does not introduce inaccuracies in the overall decomposition.

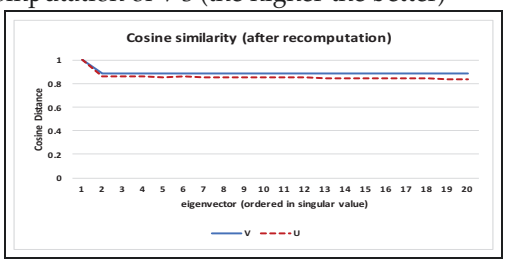
The ISVD$_4$ approach follows the same steps of ISVD$_3$, up to the computation of the left singular vectors matrix, $U_\dagger$, using the (aligned) right singular-values, $V_\dagger$, obtained through eigen-decomposition of minimum and maximum matrices. At that point, instead of using the original $V_\dagger$ factor matrix, ISVD$_4$ introduces an extra step to recompute the right singular vector matrix relying on the definition of SVD: $V_\dagger = M(V_\dagger^T)^{-1}\Sigma_\dagger^{-1}$. Intuitively, this recomputation step identifies minimum and maximum factor matrices, V_* and V^*, that are compatible with the $U_\dagger$ factor, which itself benefits from the alignment step for the left-singular vectors.

3. This also applies to **ISVD$_4$** described next.

Authorized licensed use limited to: Texas A M University. Downloaded on August 18,2020 at 05:59:53 UTC from IEEE Xplore. Restrictions apply.



(a) $cos(V_*[i], V^*[i])$ and $cos(U_*[i], U^*[i])$ before the recomputation of Vs (the higher the better)



(b) $cos(V_*[i], V^*[i])$ and $cos(U_*[i], U^*[i])$ after the recomputation of Vs (the higher the better)

Fig. 5: Cosine similarities among min and max vectors: the higher the cosine similarity, the more similar the minimum and maximum factor matrices are, indicating higher precision in the decomposition (averaged for 100 matrices as described in Table 1, Section 6, default configuration)

Figure 5 visualizes how matrix re-computation affects eigenvectors' alignment. In the figure, we plot cosine similarities between the corresponding eigenvectors in V_* and V^*, considered in increasing order of singular values – note that, the more similar the minimum and maximum versions of the eigenvectors is (i.e., the more precise the factors are), the higher is the cosine similarity. The figure reports the cosine similarity values *before recomputation* and *after recomputation*. As we see in Figure 5, for all phases, the first few eigenvectors (with the largest singular values) have the highest cosine similarities (indicating that they are more precise). On the other hand, Figure 5(a) shows that the alignment between the corresponding vectors in U_* and U^* (obtained in ISVD$_3$ through the equation $U_\dagger = M_\dagger((V_\dagger)^T)^{-1}(\Sigma_\dagger)^{-1}$) is very high, despite using less precise $V_\dagger$ vectors in the construction: intuitively, the equation used for the U_* and U^* matrices weigh the contribution of the more precise entries in the V_* and V^* matrices through multiplication with the inverses of the singular values.

ISVD$_4$ leverages this "corrective" behavior to re-obtain the V_* and V^* matrices using an extra step $V_\dagger = M(V_\dagger^T)^{-1}\Sigma_\dagger^{-1}$. The result is visualized in Figure 5(b): as we see here, after the recomputation step, the V_* and V^* matrices become much more similar, indicating more precise factor matrices, an improvement which (as we detail in Section 6) contributes to more accurate decompositions.

4.5.2 Construction of $U_\dagger$, $\Sigma_\dagger$, and $V_\dagger$ Matrices

After re-computing interval-valued factor matrices V_* and V^*, we can follow the same process described in the previous sections to reconstruct $U_\dagger$, $\Sigma_\dagger$, and $V_\dagger$. The steps are visualized in the column labeled **ISVD$_4$** in Figure 4. As before, depending on the decomposition target, this process can lead to interval-valued matrices in **ISVD$_4$-a**, scalar-valued U, V and interval-valued $\Sigma_\dagger$ in **ISVD$_4$-b**, or scalar-valued U, Σ and V in **ISVD$_4$-c**.

5 Interval-Valued PMF (IPMF)

In the previous section, we discussed several techniques to leverage latent semantic alignment to implement SVD on interval-valued data. We now show that latent alignment of minimum and maximum basis vectors can also be effective in alternative factorization schemes, such as probabilistic matrix factorization (PMF) introduced in Section 2.2.3.

We first consider interval-valued PMF (I-PMF) formulation presented in [9], which takes as input an interval-valued matrix $M_\dagger \in \mathbb{R}^{n\times m}$ (of probabilities) and introduces extra constraints, representing the minimum and maximum terms, in the loss function presented in Section 2.2.3:

$$\mathcal{L}_{I-PMF} = \|M_* - UV_*^T\|_F^2 + \|M^* - UV^{*T}\|_F^2 + \lambda_U\|U\|_F^2 + \lambda_V(\|V_*\|_F^2 + \|V^*\|_F^2),$$

where V_* and V^* represent the minimum and maximum in interval matrix $V_\dagger$. Given this loss formulation, a local minimum can be sought by applying a gradient decent in $U_{[i,:]}$, $V_{*[j,:]}$ and $V^*_{[j,:]}$ using the following partial derivatives:

$$\frac{\partial\mathcal{L}_{I-PMF}}{\partial U_{[i,:]}} = \sum_{j=1}^{m}[(U_{[i,:]}{V_*}_{[j,:]}^T - M_*[i,j])V_{*[j,:]} + (U_{[i,:]}V^{*T}_{[j,:]} - M^*[i,j])V^*_{[j,:]}] + \lambda_U U_{[i,:]}$$

$$\frac{\partial\mathcal{L}_{I-PMF}}{\partial {V_*}_{[j,:]}^T} = \sum_{i=1}^{n}(U_{[i,:]}{V_*}_{[j,:]}^T - M_*[i,j])U_{[i,:]}^T + \lambda_V {V_*}_{[j,:]}^T$$

$$\frac{\partial\mathcal{L}_{I-PMF}}{\partial {V^*_{[j,:]}}^T} = \sum_{i=1}^{n}(U_{[i,:]}V^{*T}_{[j,:]} - M^*[i,j])U_{[i,:]}^T + \lambda_V V^{*T}_{[j,:]}.$$

In this paper, we note that, while the above formulation adopts extra constraints in loss function to capture interval property of the data, it fails to consider the need for alignment in interval-valued latent semantic spaces, as discussed in Section 3.3. Hence, we propose semantically aligned interval-valued PMF formulation (AI-PMF), which adjusts the rank-order of eigen-vectors in V_* and V^* in each gradient descent iteration, after obtaining U, V_* and V^* from the above equations:

$$V_{\dagger aligned} = [V_{*aligned}, V^*_{aligned}] = ILSA(V_*, V^*).$$

As we see in the next section, this latent semantic alignment provides significant improvements in decomposition accuracy of interval-valued (probabilistic) data.

6 Experiments

In this section, we evaluate the proposed techniques in the presence of different distributions of interval-valued data. We also demonstrate that the proposed approaches can be applied to various real-world applications, including face image analysis and collaborative filtering.

6.1 Datasets

We use both synthetic and real datasets to evaluate the proposed matrix decomposition techniques in the context of several data reconstruction and classification scenarios.

Authorized licensed use limited to: Texas A M University. Downloaded on August 18,2020 at 05:59:53 UTC from IEEE Xplore. Restrictions apply.

TABLE 1: Experiment parameters for synthetic data (**bold values are defaults)**

Parameter	Values
Matrix dimension	$\mathbf{40 \times 250}$, 250×40, 25×400, 400×250, 250×400
Matrix density	percentage of 0-values: $\mathbf{0}\%$, 50%, 90%
Interval density	5%, 25%, 50%, 75%, 90%, 95%, 99%, $\mathbf{100}\%$ (on non-zeros)
Interval intensity	10%, 25%, 50%, 75%, $\mathbf{100}\%$
Target rank	uniform data: 5, 10, **20**, 40, 100, full anonymized and social media data: $5\%; 50\%; 100\%$

6.1.1 Synthetic Datasets

Table 1 provides an overview of the synthetic data scenarios we have considered in this paper. As the table shows, we have considered matrices of different sizes, dimensionalities, *densities*, and *intensities*: interval-values are generated by picking a cell in the matrix (based on the value of the *interval density parameter*) and replacing the corresponding scalar value by an interval whose size is determined by an *interval intensity parameter* – given the intensity value X, the scope of the interval is uniformly selected between 0% and $X\%$ of the minimum value of the cell.

We considered two types of interval-valued data:

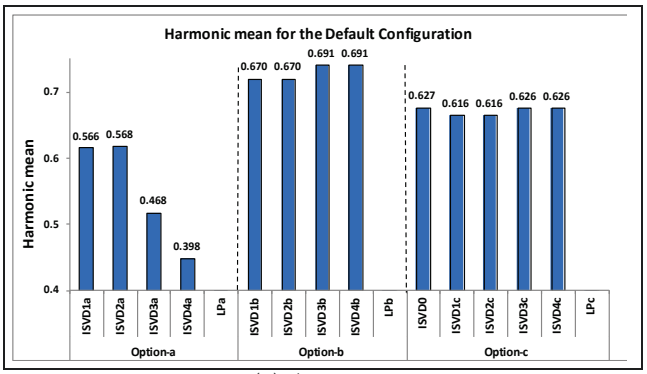
- **uniform** matrices do not have inherent structures – the cells and interval-values are selected uniformly randomly subject to constraints described in Table 1;
- **anonymized** matrices are obtained through generalization of random matrices; therefore, the resulting interval-valued matrices represent the required degree of anonymization. In particular, we consider 4 generalization levels: (a) $L1$ divides the domain into 100, (b) $L2$ into 50, (c) $L3$ into 20, and (d) $L4$ into 5 generalization intervals – note that *the higher the generalization level, the larger the corresponding intervals and the more anonymized the data*. Given these, we generalize the input data to generate three matrices with different mixtures of anonymization levels: *high anon.* (L1:10%, L2:20%, L3:30%, L4:40%), *medium anon.* (L1:25%, L2:25%, L3:25%, L4:25%), and *low anon.* (L1:40%, L2:30%, L3:20%, L4:10%).

For each scenario, we created 100 random matrices and the results presented are averages of the corresponding runs.

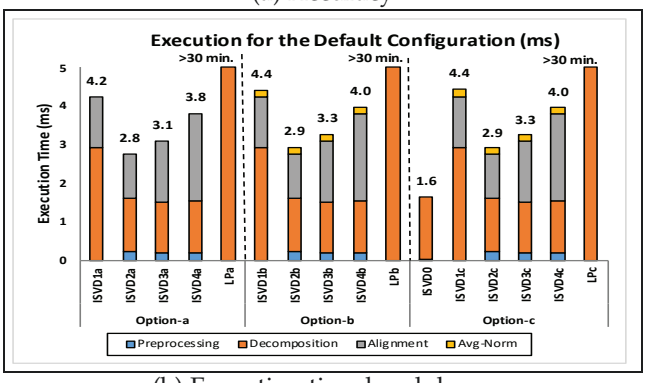
6.1.2 ORL Face Dataset

ORL face dataset [9] contains 32x32 pixels facial images of the 40 individuals. The data set contains 10 images for each individual. These 400 images are organized in the form of a matrix $M \in \mathbb{R}^{n \times m}$, where $n = 400$, $m = 1024$; i.e., each row represents a person's face and each column represents the pixel value at specific coordinates. We then follow the strategy proposed in [9] to generate an interval-valued matrix for the same data set and use these matrices for the following applications:

- Reconstruction: In this task, we attempt to reconstruct the original data matrix from the low-rank approximations and use root-mean square error to assess the reconstruction accuracy.



(a) Accuracy



(b) Execution time breakdown

Fig. 6: Comparison of the alternative approaches for SVD decomposition with interval-valued data (default configuration, the higher the Harmonic mean, the better the result)

- NN-Classification: In this task, the goal is to use the (scalar-valued) U factor for NMF based schemes and $U \times S$ (more specifically $[U_* \times S_*, U^* \times S^*]$, since we have interval valued matrices) factor for SVD based schemes to identify an individual among the 40 that are in the data set. For this purpose, we randomly select 50% rows per individual as training data and remaining rows are used as test data. The classification is done using 1-NN with Euclidean distance. When necessary, for interval-valued data, we use the following interval-valued Euclidean distance function:

$$dist(a_{int}, b_{int}) = \sqrt{(a_* - b_*)^2 + (a^* - b^*)^2}.$$

  We evaluate accuracy using F1-score.
- Clustering-based Classification: With the ORL face data set, we also consider classification through K-means (where $K = 40$) clustering, using Euclidean distances as described above. *Normalized mutual information* (NMI) [37] is used to assess cluster quality relative to given class labels.

We repeated each task 100 times and are reporting averages.

6.1.3 Social Media Data Sets

6.1.3.1 *Ciao and Epinions Datasets:* We have performed experiments with two ratings data sets: *Epinions*, *Ciao* [11]. For both *Epinions* (22K users, 27 categories, matrix density=0.26, interval density= 0.49, interval intensity=2.44/4, full rank=27) and *Ciao* (7K users, 28 categories, matrix density=0.28, interval density= 0.44, interval intensity=2.20/4, full rank=28) data sets, we have considered

Authorized licensed use limited to: Texas A M University. Downloaded on August 18,2020 at 05:59:53 UTC from IEEE Xplore. Restrictions apply.

TABLE 2: Comparison of the alternative approaches (for Option-b) with varying parameters (the higher the Harmonic mean, the better the result)

Int. density	$ISVD_0$	$ISVD_1$-b	$ISVD_2$-b	$ISVD_3$-b	$ISVD_4$-b
10%	**0.702**	0.689	0.689	**0.702**	**0.702**
25%	0.678	0.659	0.659	0.682	**0.683**
75%	0.641	0.657	0.657	**0.681**	**0.681**
100%	0.627	0.670	0.670	**0.691**	**0.691**

(a) Varying interval densities

Int. intensity	$ISVD_0$	$ISVD_1$-b	$ISVD_2$-b	$ISVD_3$-b	$ISVD_4$-b
10%	0.708	**0.709**	**0.709**	**0.709**	**0.696**
25%	0.702	0.704	0.704	**0.708**	**0.709**
75%	0.658	0.681	0.681	**0.698**	**0.698**
100%	0.627	0.670	0.670	**0.691**	**0.691**

(b) Varying interval intensities

Mat. density	$ISVD_0$	$ISVD_1$-b	$ISVD_2$-b	$ISVD_3$-b	$ISVD_4$-b
0%	0.627	0.670	0.670	**0.691**	**0.691**
50%	0.493	0.511	0.511	0.530	**0.530**
90%	0.425	0.431	0.431	0.451	**0.451**

(c) Varying matrix densities

Matrix conf.	$ISVD_0$	$ISVD_1$-b	$ISVD_2$-b	$ISVD_3$-b	$ISVD_4$-b
25-by-400	0.697	0.758	0.758	**0.789**	**0.789**
40-by-250	0.627	0.670	0.670	**0.691**	**0.691**
250-by-40	0.627	0.670	0.670	**0.691**	**0.691**
400-by-250	0.503	0.535	0.535	0.541	**0.542**
250-by-400	0.503	0.535	0.535	0.541	**0.542**

(d) Varying matrix configurations

Rank	$ISVD_0$	$ISVD_1$-b	$ISVD_2$-b	$ISVD_3$-b	$ISVD_4$-b
5	0.501	0.537	0.537	**0.539**	**0.539**
10	0.546	0.585	0.585	0.591	**0.592**
20	0.627	0.670	0.670	**0.691**	**0.691**
40	0.758	0.814	0.814	**0.895**	**0.896**

(e) Varying target ranks

For these data scenarios, the LP class of competitors return $\simeq$ 0.0 H-mean, indicating that they are not effective approaches for interval-valued matrix decomposition.

`user-category` rating matrices, where each non-zero matrix cell corresponds to a range of ratings a user provided for items of a given category.

6.1.3.2 *MovieLens Dataset:* We use MovieLens-100K dataset [10] for collaborative filtering evaluation. The data set includes 19 genres, 100K ratings, 943 users (objects) and 1682 movies (features). For this data set, we generate a `user-genre` interval matrix, where each interval corresponds to the range of ratings a user-provided for movies of a given genre; consequently, the MovieLens data set has a full rank 19. More specifically, we can consider these ratings as a scalar matrix $R \in \mathbb{R}^{n \times m}$, where $n = 943$ and $m = 1682$, and generate an interval matrix $R_\dagger \in \mathbb{R}^{n \times m}$ from R:

$$\delta_{ij} := \alpha \times std(\{R[i^{'}, j^{'}] | (i^{'} = i \vee j^{'} = j) \wedge (i^{'}, j^{'}) \in (i, j)\})$$
$$R[i,j]_\dagger = [R[i,j]_*, R[i,j]^*] = [R[i,j] - \delta_{ij}, R[i,j] + \delta_{ij}]$$

Above, $\alpha \in \mathbb{R}^+$ is a multiplicative scale coefficient. Given this interval matrix $R_\dagger$, unknown ratings can be computed by reconstructing the original matrix after low-rank approximation. See [9] for more details. We use RMSE to evaluate the accuracy of the reconstruction-based rating predictions.

6.2 Competitors

As we discussed in Section 4.3.1, [33], [35] proposed a linear-programming based approach to solve this task. We have, therefore, implemented and experimented also with their proposed solution. We denote their linear-programming based approaches as "LP_x", where the subscript x denote one of the three application semantics (or options), a, b or c.

As described in Section 6.1.2, we also compare proposed ISVD approaches with NMF and I-NMF [9] for the face analysis tasks: data reconstruction and classification. For collaborative filtering with social media data, discussed in Section 6.1.3, we used PMF and I-PMF [9] as competitors.

6.3 Results based on Synthetic Data Sets

6.3.1 Uniform Data

Figure 6 provides an overview of the accuracy and execution time results for the default configuration:

- we obtain the highest accuracies using $ISVD_\#$-b class of techniques (returning both scalar-valued factors and interval-valued core) – highest overall accuracy is provided by $ISVD_4$-b, which leverages both semantic alignment and latent space recomputation techniques;
- the $ISVD_\#$-c class of techniques (returning scalar valued factor and core matrices) approximate the accuracy of the $ISVD_0$ technique – however, these include redundant work;
- linear-programing based competitors [33], [35] have poor accuracies and massive execution times; the reason for this poor performance is that, as also acknowledged by the authors, these approaches are effective only when the interval ranges are very small, while our proposed approaches are able to handle intervals of varying sizes effectively.

Tables 2(a) through (e) show the accuracy results for $ISVD_\#$-b class of techniques (which as we have seen above provide the best overall accuracies) under various configurations reported in Table 1. We also include $ISVD_0$ as a fast alternative. The results show that $ISVD_4$-b provides the best accuracy under all data scenarios considered and the faster alternative, $ISVD_0$, is not able to provide competitive accuracy.

6.3.2 Anonymized Data

In Figure 7 (a)$\sim$(c), we see decomposition results for interval data generated through three privacy levels. The first thing to note is that, once again, decomposition target $ISVD_\#$-b (which has interval-valued core and scalar factor matrices) provides the best overall accuracy, except for the very low-rank (5%) decomposition under the low-privacy scenario (with smaller number of large intervals). In that case, decomposition target $ISVD_\#$-a, which maintains interval information not only for the core, but also for the factor matrices, leads to slightly better results. Once again, these results confirm that $ISVD_4$-b provides the overall best accuracy.

6.4 Results based on ORL Face Dataset

6.4.1 Reconstruction

Figure 8(a) shows reconstruction performance for target ranks $(10, 100, 200)$. As we see here, $ISVD_0$, $ISVD_4$-b, and $ISVD_4$-c provide best reconstruction accuracies. The competitor techniques, NMF and I-NMF, provide significantly

Authorized licensed use limited to: Texas A M University. Downloaded on August 18,2020 at 05:59:53 UTC from IEEE Xplore. Restrictions apply.

Anonymization High Privacy [10%, 20%, 30%, 40%]		100% rank H-mean	100% rank Order	50% rank H-mean	50% rank Order	5% rank H-mean	5% rank Order
Option a	ISVD1	0.749	13	0.652	12	0.512	5
	ISVD2	0.751	12	0.653	11	0.512	5
	ISVD3	0.770	10	0.654	10	0.501	12
	ISVD4	0.769	11	0.634	13	0.470	13
Option b	ISVD1	0.869	6	0.691	3	0.512	3
	ISVD2	0.869	6	0.691	3	0.512	3
	ISVD3	0.935	**1**	0.703	2	0.513	**1**
	ISVD4	0.935	**1**	0.703	**1**	0.513	**1**
Option c	ISVD0	0.878	3	0.685	5	0.501	7
	ISVD1	0.832	8	0.673	8	0.501	10
	ISVD2	0.832	8	0.673	8	0.501	10
	ISVD3	0.876	4	0.685	7	0.501	9
	ISVD4	0.876	4	0.685	6	0.501	8

(a) Anonymized data – high privacy

Anonymization Medium Privacy [25%, 25%, 25%, 25%]		100% rank H-mean	100% rank Order	50% rank H-mean	50% rank Order	5% rank H-mean	5% rank Order
Option a	ISVD1	0.740	13	0.650	13	0.513	5
	ISVD2	0.742	12	0.652	12	0.513	5
	ISVD3	0.773	11	0.659	11	0.507	12
	ISVD4	0.814	10	0.661	10	0.490	13
Option b	ISVD1	0.867	6	0.691	6	0.514	3
	ISVD2	0.867	6	0.691	6	0.514	3
	ISVD3	0.934	2	0.703	2	0.514	**1**
	ISVD4	0.934	**1**	0.703	**1**	0.514	**1**
Option c	ISVD0	0.900	3	0.693	3	0.508	7
	ISVD1	0.846	8	0.681	8	0.508	10
	ISVD2	0.846	8	0.681	8	0.508	10
	ISVD3	0.898	5	0.693	5	0.508	8
	ISVD4	0.898	4	0.693	4	0.508	8

(b) Anonymized data – medium privacy

Anonymization Low Privacy [40%, 30%, 20%, 10%]		100% rank H-mean	100% rank Order	50% rank H-mean	50% rank Order	5% rank H-mean	5% rank Order
Option a	ISVD1	0.583	11	0.479	10	0.247	**1**
	ISVD2	0.602	10	0.489	9	0.247	**1**
	ISVD3	0.546	12	0.446	13	0.238	7
	ISVD4	0.546	12	0.456	12	0.218	13
Option b	ISVD1	0.765	3	0.479	10	0.245	3
	ISVD2	0.765	3	0.583	3	0.245	3
	ISVD3	0.805	**1**	0.599	**1**	0.245	5
	ISVD4	0.805	**1**	0.599	**1**	0.245	5
Option c	ISVD0	0.763	5	0.578	4	0.236	11
	ISVD1	0.734	8	0.565	7	0.236	8
	ISVD2	0.734	8	0.565	7	0.236	8
	ISVD3	0.763	5	0.578	6	0.236	10
	ISVD4	0.763	5	0.578	5	0.236	11

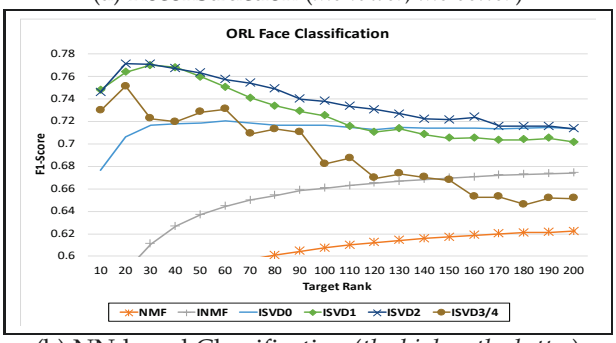
(c) Anonymized data – low privacy

The LP class of competitors have ≤ 0.01 H-mean, indicating that they are not effective approaches for interval-valued matrix decomposition.

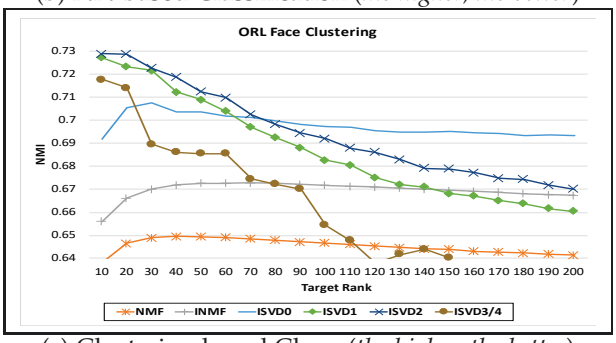
Fig. 7: Accuracy comparison for anonymized datasets for different target ranks (the greener the cell, the better the result – the tables are best viewed in color)

ORL Face Reconstruction		rank=200 RE	rank=200 Order	rank=100 RE	rank=100 Order	rank=10 RE	rank=10 Order
Option a	ISVD1	26.8	14	24.3	14	26.1	12
	ISVD2	23.3	13	24.1	13	25.7	7
	ISVD3	14.9	6	16.0	6	53.0	13
	ISVD4	21.6	12	19.7	8	113.5	14
Option b	ISVD1	19.3	8	20.3	9	26.0	8
	ISVD2	19.5	10	20.5	11	26.0	10
	ISVD3	8.8	4	11.7	4	23.1	4
	ISVD4	5.0	**2**	9.2	**2**	21.6	**2**
Option c	ISVD0	4.5	**1**	8.7	**1**	21.2	**1**
	ISVD1	19.3	8	20.3	9	26.0	8
	ISVD2	19.5	10	20.5	11	26.0	10
	ISVD3	8.8	4	11.7	4	23.1	4
	ISVD4	5.0	**2**	9.2	**2**	21.6	**2**
	NMF	16.7	7	18.3	7	23.8	6
	I-NMF	19.0	8	20.3	9	25.1	7

(a) Reconstruction (*the lower, the better*)



(b) NN-based Classification (*the higher, the better*)



(c) Clustering-based Class. (*the higher, the better*)

Fig. 8: (a) Reconstruction and (b,c) classification results for the ORL face data sets

lower reconstruction accuracies. We note that, while the relatively simpler $ISVD_0$ appears to be competitive against $ISVD_4$-b in the reconstruction task, in the classification tasks discussed next, performs less effectively.

6.4.2 NN-based Classification

Figure 8(b) shows classification performance for various algorithms. As we see here, the best classification results are obtained using $ISVD_1$ and $ISVD_2$ techniques, both of which leverage latent semantic alignment, under low (20-30) rank decomposition. Since classification task only relies on U factors and the core matrix, the $ISVD_3$ and $ISVD_4$ techniques (which attempt to further improve the latent space defined by the V factor) do not provide additional benefits (and in fact hurt the classification accuracies). It is important to note that NMF and I-NMF competitors, both, provide significantly lower accuracies than ISVD-based approaches.

6.4.3 Clustering-based Classification

The clustering results, shown in Figure 8(c), re-confirm the NN-based classification results discussed above. Once again, the best results are obtained using $ISVD_1$ and $ISVD_2$ techniques, both of which leverage latent semantic alignment, under low ($\simeq$10) rank decomposition.

Finally, in Table 3 we consider the original face data (scalar vectors) and processed interval-valued vectors and compare them to ISVD2 based clustering for both accuracy and execution time. As we see in this table, adding interval information (scalar vectors vs. interval vectors) significantly improves accuracy, but at a significant execution time cost. The proposed interval-valued decomposition strategy cuts down the clustering time significantly, while matching the accuracy (at rank 20) provided by the interval vectors.

Authorized licensed use limited to: Texas A M University. Downloaded on August 18,2020 at 05:59:53 UTC from IEEE Xplore. Restrictions apply.

TABLE 3: Accuracy and execution time for clustering-based classification using original data vector, interval-valued data vectors, and low-rank interval-value based data

Accuracy in terms of NMI; the higher,the better			
res.	*scalar vectors*	*interval vectors*	*ISVD2-b* ($r = 20$)
32×32	0.69	0.72	0.72
64×64	0.68	0.71	0.71

Exec.Time (decomp.+k-means) in seconds			
res.	*scalar vectors*	*interval vectors*	*ISVD2-b* ($r = 20$)
32×32	0.04	1.14	0.11 (0.08+0.03)
64×64	0.14	5.06	1.48 (1.44+0.04)

6.5 Results based on Social Media Data

6.5.1 Reconstruction Results

Figure 9 presents reconstruction accuracies for different target ranks and decomposition semantics on three interval-valued data sets, Ciao, Epinions, and MovieLens.

These results re-confirm that, in general, decomposition target **option-b** (with interval-valued core and scalar factors) leads to best results, except for very low-rank decompositions, where **option-a** (with interval-valued core and factors) becomes a better option.

For Ciao and Epinion data sets, we see that, as in the data sets considered earlier, $ISVD_3$ and $ISVD_4$, which perform latent alignment early, lead to better results. $ISVD_1$ and $ISVD_2$, which perform late-alignment of the factors, are more effective than $ISVD_3$ and $ISVD_4$ only for very low rank decompositions. The MovieLens data set also has a similar behavior: for relatively large decomposition ranks, $ISVD_3$ and $ISVD_4$, which perform latent alignment early, lead to better results. For low rank decompositions ($r \leq 10$) the benefit of early alignment is lost and $ISVD_1$ and $ISVD_2$ are (slightly) more effective.

6.5.2 Collaborative Filtering/Prediction Results

Figure 10 shows collaborative filtering performance in RMSE with MovieLens-100K dataset for PMF, I-PMF and AI-PMF. As the chart shows, the prediction accuracy of all algorithms improves as we consider higher decomposition ranks and the proposed latent semantic alignment based approach, AI-PMF, leads to better prediction performance than both PMF and I-PMF, for decomposition ranks > 60. Most importantly, however, AI-PMF always performs better than I-PMF indicating that, as expected, latent semantic alignment helps achieve better handling of interval-valued factors, also when considering factors with probabilistic interpretations, rather than eigen-vector based interpretations as considered in Section 3.3.

Ciao		100% rank (=28)		50% rank (=14)		5% rank (=1)	
		H-mean	Order	H-mean	Order	H-mean	Order
Option a	ISVD1	0.645	13	0.475	12	0.248	**1**
	ISVD2	0.661	11	0.484	10	0.248	**1**
	ISVD3	0.645	12	0.474	13	0.242	7
	ISVD4	0.671	10	0.484	11	0.228	13
Option b	ISVD1	0.805	6	0.552	3	0.246	3
	ISVD2	0.805	6	0.552	3	0.246	3
	ISVD3	0.838	**1**	0.564	2	0.246	5
	ISVD4	0.838	**1**	0.564	**1**	0.246	6
Option c	ISVD0	0.807	3	0.552	3	0.240	12
	ISVD1	0.781	8	0.541	8	0.241	8
	ISVD2	0.781	8	0.541	8	0.241	8
	ISVD3	0.807	3	0.552	7	0.240	10
	ISVD4	0.807	3	0.552	6	0.240	11

(a) Ciao data set

Epinion		100% rank (=27)		50% rank (=14)		5% rank (=1)	
		H-mean	Order	H-mean	Order	H-mean	Order
Option a	ISVD1	0.583	11	0.479	11	0.247	**1**
	ISVD2	0.602	10	0.489	10	0.247	**1**
	ISVD3	0.546	13	0.446	13	0.238	7
	ISVD4	0.571	12	0.456	12	0.218	13
Option b	ISVD1	0.765	3	0.583	3	0.245	3
	ISVD2	0.765	3	0.583	3	0.245	3
	ISVD3	0.805	**1**	0.599	2	0.245	5
	ISVD4	0.805	**1**	0.599	**1**	0.245	6
Option c	ISVD0	0.763	5	0.578	5	0.236	11
	ISVD1	0.734	8	0.565	8	0.236	8
	ISVD2	0.734	8	0.565	8	0.236	8
	ISVD3	0.763	5	0.578	7	0.236	10
	ISVD4	0.763	5	0.578	6	0.236	11

(b) Epinion data set

MovieLens-100k		100% rank (=19)		50% rank (=10)		5% rank (=1)	
		H-mean	Order	H-mean	Order	H-mean	Order
Option a	ISVD1	0.633	6	0.616	6	0.547	**1**
	ISVD2	0.737	5	0.720	5	0.547	**1**
	ISVD3	0.527	12	0.537	12	0.445	7
	ISVD4	0.215	13	0.267	13	0.242	13
Option b	ISVD1	0.765	3	0.730	**1**	0.536	3
	ISVD2	0.765	3	0.730	**1**	0.536	3
	ISVD3	0.768	**1**	0.728	4	0.533	5
	ISVD4	0.768	**1**	0.729	3	0.532	6
Option c	ISVD0	0.581	9	0.553	10	0.400	11
	ISVD1	0.589	7	0.564	7	0.412	8
	ISVD2	0.589	7	0.564	7	0.412	8
	ISVD3	0.581	9	0.554	9	0.405	10
	ISVD4	0.581	9	0.553	11	0.399	12

(c) Movielens data set

The LP class of competitors have ≤ 0.01 H-mean, indicating that they are not effective approaches for interval-valued matrix decomposition.

Fig. 9: Accuracy comparison for social media datasets for different target ranks (the greener the cell, the better the result – the tables are best viewed in color)

7 Conclusions

In this paper, we noted that many applications generate interval-valued data, yet, existing data analysis tools (such as matrix-decomposition operations) assume that all observations are scalar-valued. Building on this observation, we presented eigen-vector based and probabilistic matrix-decomposition techniques (*interval singular value decomposition (ISVD)* and *aligned interval probability matrix factorization (AI-PMF)*, respectively) that are designed for interval-valued data sets. Experiments, under diverse data scenarios and application semantics, showed that state-of-the-art linear-programming based eigen-decomposition techniques are ineffective for interval-valued SVD. Results also showed that by carefully combining interval- and matrix-algebra operations, we can improve matrix decomposition accuracies and further improve the performance in image classification, images clustering and collaborative filtering in the presence of interval-valued data.

Acknowledgments

Supported by NSF#1633381, NSF#1610282, and EU-H2020 grant ("FourCmodeling") No 690817.

References

[1] D. Li, C. Chen, Q. Lv, J. Yan, L. Shang, and S. Chu, "Low-rank matrix approximation with stability," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 295–303.

Authorized licensed use limited to: Texas A M University. Downloaded on August 18,2020 at 05:59:53 UTC from IEEE Xplore. Restrictions apply.

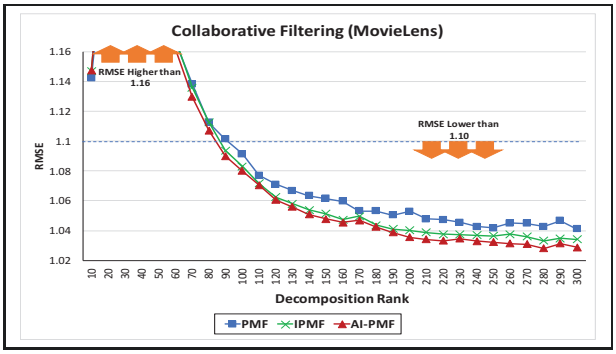


Fig. 10: Collaborative filtering (RMSE,*the lower, the better*)

[2] T. Rukat, C. C. Holmes, M. K. Titsias, and C. Yau, "Bayesian boolean matrix factorisation," in *ICML*, vol. 70, 2017, pp. 2969–2978.

[3] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[4] I. T. Jolliffe, *Principal component analysis*, ser. Springer series in statistics. New York: Springer-Verlag, 1986.

[5] K. S. Candan, H. Cao, Y. Qi, and M. L. Sapino, "Alphasum: size-constrained table summarization using value lattices," in *EDBT*, 2009, pp. 96–107.

[6] H. Cao, K. S. Candan, and M. L. Sapino, "Skynets: searching for minimum trees in graphs with incomparable edge weights," in *CIKM*, 2011, pp. 1775–1784.

[7] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *NIPS*.

[8] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 571–588, Oct. 2002.

[9] Z. Shen, L. Du, X. Shen, and Y. Shen, "Interval-valued matrix factorization with applications," in *2010 IEEE International Conference on Data Mining*, Dec 2010, pp. 1037–1042.

[10] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 19:1–19:19, Dec. 2015.

[11] J. Tang, H. Gao, and H. Liu, "mTrust: Discerning multi-faceted trust in a connected world," in *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012, pp. 93–102.

[12] T. Sunaga, "Theory of an interval algebra and its application to numerical analysis," *Japan Journal of Industrial and Applied Mathematics*, vol. 26, no. 2, pp. 125–143, October 2009.

[13] Z. Z. A. Zhao, *Spectral feature selection for data mining*, ser. Chapman & Hall/CRC data mining and knowledge discovery series. Boca Raton, FL: CRC Press, 2012.

[14] M. Berry and D. Martin., "Component analysis for information retrieval. in handbook of parallel computing and statistics." 2004.

[15] J. Sun, S. Papadimitriou, and C. Faloutsos, "Online latent variable detection in sensor networks," in *21st International Conference on Data Engineering (ICDE'05)*, April 2005, pp. 1126–1127.

[16] S. S. Bucak and B. Gunsel, "Video content representation by incremental non-negative matrix factorization," in *2007 IEEE International Conference on Image Processing*, vol. 2, Sept 2007, pp. II – 113–II – 116.

[17] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[18] L. Billard. and E. Diday, Eds., *Symbolic Data Analysis: Conceptual Statistics and Data Mining*. Wiley, 2007.

[19] M. Noirhomme-Fraiture and P. Brito, "Far beyond the classical data models: symbolic data analysis," *Statistical Analysis and Data Mining*, vol. 4, no. 2, pp. 157–170, 2011.

[20] E. Diday, "Principal component analysis for bar charts and metabins tables," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 6, no. 5, pp. 403–430, 2013.

[21] S. Makosso-Kallyth and E. Diday, "Adaptation of interval pca to symbolic histogram variables," *Advances in Data Analysis and Classification*, vol. 6, no. 2, pp. 147–159, Jul 2012.

[22] E. Diday, "Thinking by classes in data science: the symbolic data analysis paradigm," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 8, no. 5, pp. 172–205, 2016.

[23] H. Wang, R. Guan, and J. Wu, "Linear regression of interval-valued data based on complete information in hypercubes," *Journal of Systems Science and Systems Engineering*, vol. 21, no. 4, pp. 422–442, Dec 2012.

[24] P. Giordani, "Lasso-constrained regression analysis for interval-valued data," *Advances in Data Analysis and Classification*, vol. 9, no. 1, pp. 5–19, Mar 2015.

[25] N. C. Lauro, R. Verde, and A. Irpino, *Generalized Canonical Analysis*. John Wiley and Sons, Ltd, 2008, pp. 313–330.

[26] P. Groenen, S. Winsberg, O. Rodrguez, and E. Diday, "I-scal: Multidimensional scaling of interval dissimilarities," *Computational Statistics and Data Analysis*, vol. 51, no. 1, pp. 360 – 378, 2006.

[27] L. Billard and J. Le-Rademacher, "Principal component analysis for interval data," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, no. 6, pp. 535–540, 2012.

[28] H. Wang, R. Guan, and J. Wu, "Cipca: Complete-information-based principal component analysis for interval-valued data," *Neurocomputing*, vol. 86, pp. 158 – 169, 2012.

[29] F. Gioia and C. N. Lauro, "Principal component analysis on interval data," *Computational Statistics*, vol. 21, no. 2, pp. 343–363, Jun 2006.

[30] A. Douzal-Chouakria, L. Billard, and E. Diday, "Principal component analysis for interval-valued observations," *Statistical Analysis and Data Mining*, vol. 4, no. 2, pp. 229–246, 2011.

[31] K. Iwama and S. Miyazaki, "A survey of the stable marriage problem and its variants," 02 2008, pp. 131–136.

[32] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. Society for Industrial and Applied Mathematics, 2012.

[33] A. Deif, "The interval eigenvalue problem," *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift fr Angewandte Mathematik und Mechanik*, vol. 71, no. 1, pp. 61–64, 1991.

[34] J. Rohn, "Interval matrices: Singularity and real eigenvalues," *SIAM Journal on Matrix Analysis and Applications*, vol. 14, no. 1, pp. 82–91, 1993.

[35] N. P. Seif, S. Hashem, and A. S. Deif, "Bounding the eigenvectors for symmetric interval matrices," *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift fr Angewandte Mathematik und Mechanik*, vol. 72, no. 3, pp. 233–236, 1992.

[36] A. Deif, "Singular values of an interval matrix," *Linear Algebra and its Applications*, vol. 151, pp. 125 – 133, 1991.

[37] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. New York, NY, USA: Wiley-Interscience, 2006.

Mao-Lin Li is a computer science PhD student at the Arizona State University. His research interests include management and analysis of imprecise, graph structured data, with applications including collaborative filtering and predictive analysis.

Francesco Di Mauro is a PhD student at the Computer Science Department of the University of Torino. His research interests include decomposition of imprecise (such as interval-valued) matrices and tensors to support efficient and effective data analysis with real-world data sets.

K. Selçuk Candan is a professor of computer science and engineering at Arizona State University. His research is in the area of management and analysis of non-traditional and imprecise (such as multimedia, web, and scientific) data. He is an ACM Distinguished Scientist.

Maria Luisa Sapino is a Full Professor at the University of Torino. Her research interests are in the area of heterogeneous and multimedia data management, with emphasis on the development of efficient techniques for tensor based big data analysis and on indexing, classification, and querying of (possibly multivariate) time series.

Authorized licensed use limited to: Texas A M University. Downloaded on August 18,2020 at 05:59:53 UTC from IEEE Xplore. Restrictions apply.