Unit-Consistent Tensor Completion with Applications to Recommender Systems

Tung Nguyen and Jeffrey Uhlmann
University of Missouri - Columbia

Abstract

In this paper we introduce a new consistency-based approach for defining and solving nonnegative/positive matrix and tensor completion problems. The novelty of the framework is that instead of artificially making the problem well-posed in the form of an application-arbitrary optimization problem, e.g., minimizing a bulk structural measure such as rank or norm, we show that a single property/constraint — preserving unit-scale consistency — guarantees both existence of a solution and, under relatively weak support assumptions, uniqueness. The framework and solution algorithms also generalize directly to tensors of arbitrary dimension while maintaining computational complexity that is linear in problem size for fixed dimension $d$. In the context of recommender system (RS) applications, we prove that two reasonable properties that should be expected to hold for any solution to the RS problem are sufficient to permit uniqueness guarantees to be established within our framework. This is remarkable because it obviates the need for heuristic-based statistical or AI methods despite what appear to be distinctly human/subjective variables at the heart of the problem. Key theoretical contributions include a general unit-consistent tensor-completion framework with proofs of its properties, including algorithms with optimal runtime complexity, e.g., $O(1)$ term-completion with preprocessing complexity that is linear in the number of known terms of the matrix/tensor.

Keywords: matrix completion, tensor completion, unit consistency, recommender systems, algorithm fairness.
1 Introduction

Matrix/tensor completion is inherently ill-posed and thus demands additional
constraints to ensure the existence of a nonempty and nontrivial family of so-
lutions. Historically the focus has been on the matrix as a mathematical ob-
ject rather than on properties that should be preserved based on the geometry
or other structural constraints implicit to the application. Consequently, the
conventional means for defining matrix completion is to formulate it as an op-
timization problem such as minimizing a specified norm of the matrix. How-
ever, at a minimum the choice should be consistent with properties required
by the application. For example, if the geometric structure of the application’s
solution space is understood to be rotation-invariant, then clearly a unitary-
invariant norm is not inappropriate. On the other hand, if the application
involves variables with incommensurate units of measure (e.g., length and pres-
sure variables defined in arbitrary metric or imperial units), then minimizing a
unitary-invariant norm is meaningless because solutions will be dependent on
the arbitrary choice of units made for the application’s state variables.

We take the position that a well-posed formulation of the matrix/tensor
completion problem should derive from properties of the solution space that
must be enforced or conserved. In particular, we assume that the vast majority
of nontrivial real-world problems involve some number of known and unknown
variables with incommensurate units. Therefore the solution for a given prob-
lem must be consistent with respect to those units in the sense that arbitrary
relative linear scalings of them, e.g., changing from meters to centimeters for
lengths, should yield the same unique solution but in the new units. We refer to
this as a unit-consistent (UC) solution. To intuitively appreciate the significance
of this perspective, consider the alternative in which a change from millimeters
to centimeters yields a fundamentally different solution. Which solution is “bet-
ter”? If some meta-criterion selects one over the other, then does that mean
another choice of units might yield an even better solution?

In many areas of applied mathematics and engineering, the default reflex
when faced with an ill-posed problem (e.g., an underdetermined set of equa-
tions) is to apply a toolbox LMSE method to define a unique solution. This
urge is so strong as to go almost unnoticed, as if there is no need to reflect on
whether the resulting solution is meaningful in terms of the given application.
It could be argued that a slightly more sophisticated urge would be to try to
identify a formulation of the ill-posed problem so as to minimize the computa-
tional complexity of the algorithm needed to solve that formulation. Such an
approach almost presupposes that there is no particular “best” solution from the
perspective of the application, so computational efficiency becomes the principal
consideration.

For some practical applications it could be the case that computational con-
siderations must supersede all else because some solution is better than no
solution. This may motivate a rank-minimization formulation of the matrix
completion problem as a means for producing a decomposable (factorable) re-
sult that can reduce problem complexity for subsequent operations [6, 5]. But
will the resulting solutions be unitary-invariant or unit-consistent? The application may demand one of these or some other property or properties be enforced, and identifying which is appropriate should precede consideration of possible problem formulations if only to characterize limitations of the ultimately chosen solution method.

The first contribution of this paper is a fully general, computationally efficient method for transforming a given tensor to a scale-invariant canonical form. This form ensures that whatever operation is applied to it will be scale invariant. For example, a matrix function $f(A)$, can be made scale-invariant as $f(S(A))$, where the function $S$ ensures $A = D \cdot S(DAE) \cdot E$ and $S(DAE) = S(A)$ is uniquely determined and holds for all strictly positive diagonal matrices $D$ and $E$. Alternatively, the operation can be made unit-consistent as $D \cdot f(S(A)) \cdot E$. In summary, the unique canonical scaling $S(A)$ is invariant with respect to non-singular diagonal scalings of its matrix argument, and this canonical scaling generalizes to arbitrary $d$-dimensional tensors.

The second contribution of this paper is a tensor completion algorithm based on this unit-scale invariant canonical form. We argue that human/subjective variables that are presumed to be unknowable but critical to effective recommender system (RS) solutions can be effectively understood as a lack of knowledge of units that are implicitly applied when humans rate products, e.g., that humans implicitly rank using subjective units that may depend on the product (or its class or attributes). Evidence for this conclusion comes from the fact that a unit-consistency constraint on its own is sufficient to perform comparably to state-of-the-art specialized RS systems according to crude/arbitrary metrics such as RMSE. Potentially more important from an analytic perspective, the UC constraint alone also determines a unique solution (assuming full support is provided by the given entries), and from it we are able to prove a consensus ordering theorem that any admissible RS solution should be expected to satisfy: if all users agree on a rank-ordering of a set of products, then recommendations (entry completions) will also satisfy that ordering.

The complexity to transform to scale-invariant form is $\Omega(n)$, where $n$ is the number of known entries. However, our most general formulation of the RS problem for tensors can involve constructions for which each user is represented as a vector of attributes, and products are similarly generalized, and each of these attributes can in principle be recursively expanded to capture whatever relationships are deemed to be of predictive interest. In practice one can expect $n$ to be relatively manageable because the collection of information is practically bounded even if the implicit index space becomes exorbitant, but the $O(1)$ per-entry completion complexity, which assumes fixed $d$, can be compromised because we permit predictions to be defined across arbitrary $k$-dimensional subtensors, i.e., our generalization can allow for formulations with $O(\binom{d}{k})$ entry-completion complexities. Fortunately, RS applications of interest will typically involve formulations for which $k = d - 1$, thus preserving the optimal complexity.

\footnote{Examples include scale-invariant or unit-consistent PCA, MDS, neural networks and machine learning methods, etc., to replace conventional least-squares or other norm-specific criteria. See \cite{21, 22} for use of this in the context of robot-control applications.}
ities achieved in the matrix case to arbitrary $d$-dimensional tensors.

The format of the paper is as follows. We begin by introducing the recommender system problem and arguing that unit consistency is a necessary property. We then formally define a general canonical scaling algorithm (CSA) for tensors, followed by the tensor completion algorithm based on that canonical scaling, which provide a direct completion method for recommender-system applications. We then prove general UC properties of the completion algorithm, e.g., relating to uniqueness, followed by proofs of properties that are of specific relevance to recommender systems. Notably, we prove the consensus-ordering theorem, which essentially says that if all users agree on an ordering-by-preference for a set of products, the estimated ratings (recommendations) obtained from the completion algorithm will respect that ordering. We also generalize the interpretation of the consensus-ordering theorem to apply with respect to user attributes and/or product attributes and/or the tensor extension of any other state variable. We conclude with a discussion of results and their implications for generalized recommender systems.

2 Recommender Systems

A motivating application for our unit-consistent (UC) framework is recommender systems in which a table (matrix) of user ratings of products is used to estimate values for unfilled entries, i.e., predict ratings for particular products not rated by particular users. For a matrix $A$ with user $i$ and product $j$, we consider the score of user $i$ on product $j$ as $A_{ij} = A(i,j)$ for $(i,j) \in \sigma(A)$. Define the list of entries in $\sigma(A)$ as $A_r$ and the list of entries in $\sigma(A)$ as $A_{nr}$ (similar to absent entries). A recommendation process $RS$ bases on the entries of $A_r$ to approximate the entries of $A_{nr}$. Our goal is to provide a formulation of $RS$ such that the resulting entries filled in $A_{nr}$ satisfies unit consistency and consensus-ordering constraints.

Our approach involves taking the matrix matrix of user-product ratings and transforming it to a scale-invariant canonical by applying a left and right diagonal scaling so that the product of nonzero (and unfilled) entries in each row and column is unity. This scaled form is provably unique. Each unfilled entry is then replaced with 1, thus preserving the canonical form, and the inverse of the original scaling is applied to obtain the completed form of the original matrix. This algorithm preserves unit consistency because canonical form is invariant with respect to positive scalings of the rows and columns of the given matrix.

To appreciate the need for RS unit consistency, consider a user Alice who rates products in terms of a personal “unit of quality” that derives in unknowable ways from various aspects of her personality and experience [23]. Suppose the RS suggests a rating of $x$ for a particular film she has not yet rated. Now consider an alternative scenario in which Alice’s personal “unit of quality” is arbitrarily scaled by a factor of 1.05, i.e., all of her ratings become 5% larger. In this scenario we should expect the RS to suggest a rating of $1.05 \times x$, i.e., the same value but now consistent with her ratings using the new unit of measure.
If it does not, then the RS is not unit consistent.

To further impress the need for unit consistency, suppose in the example above that the original RS rating of $x$ for the new film is the same value Alice gave as a rating for a previous film she had seen. In other words, the RS is implicitly suggesting that she will like the new film almost identically to that previous film. If, however, the RS is not unit consistent, then the simple scaling of all of Alice’s rating by 5% – as described in the alternative scenario above – would then cause the RS-suggested rating for the new film to no longer be same as the previous film. In other words, the RS-predicted relative ranking of the new film among the films that Alice has previously rated could change. Of course, unit consistency would, by definition, ensure that the predicted ranking of the new film among Alice’s rated films is unchanged because the new rating would be scaled consistently, e.g., the original RS rating of $x$ for the film would become $1.05 \times x$ in the alternative scenario in which all of Alice’s ratings are scaled by 1.05.

A different perspective on the need for unit consistency is to consider the influence of Alice’s ratings on the recommendations made by the RS to other users. Unit consistency implies that a fixed scaling of all of Alice’s ratings, e.g., by a factor of 1.05, will have no effect on RS-suggested ratings for other users. This is because ratings are invariant with respect to scale factors applied separately to any row or column of the ratings matrix. This invariance also ensures that no individual user is advantaged or penalized in terms of their influence on system ratings due to the magnitude of their personal “unit of quality” they use (implicitly) to produce their ratings.

By contrast, an approach that optimizes RS ratings to minimize a measure of squared error – hence cannot provide unit consistency – will implicitly apply more weight to larger-magnitude user ratings than to smaller ones. This means that users who tend to be more reserved in their ratings will have less influence on the system’s ratings than users who tend to give higher ratings on average by some factor. Returning to the earlier example, the scaling of Alice’s ratings by 5% in a system that minimizes squared error would have the effect of giving her ranked ordering of films more influence on the system’s ratings than it had before. This potential source for manipulation of the RS is avoided under the unit-consistency constraint.

There are, of course, opportunities for users to selectively influence any RS system because its suggested ratings are necessarily derived from user-provided ratings. In the case a unit-consistent RS, for example, if a user feels that film A is 5% better than film B, and film B is 3% better than film C, then the influence of this degree of relative preferences of the user on system ratings cannot be increased or decreased simply by scaling the ratings, e.g., scaling ratings so film A has the highest rating. The user can only affect the system’s ratings for other users by changing his relative ratings for the three films, e.g., by giving them all the highest possible rating. Doing so will not change the magnitude of the user’s influence on system ratings, it will only cause the system to believe he equally likes all three films. In other words, he cannot artificially bias a unit-consistent RS toward ratings that are more aligned with his personal relative preferences.
of the three films, he can only affect what the system assumes to be his relative preferences. In this sense, unit consistency provides a natural and intuitive form of both robustness and fairness.

Previous works ([24, 25, 26]) have exploited unitary invariance via the SVD to achieve state-of-the-art probabilistic bounds on the minimum number of entries sufficient for retrieval. Other methods exploit the SVD indirectly via the Moore-Penrose pseudoinverse to achieve unitary and global-scale invariance. These include collaborative filtering ([27]), the parameter-decrease method for matrix factorization ([28]), SIFT for recommendation ([29, 30]), and social choice theory for recommendation ([31]). Unfortunately, unitary (e.g., rotational) invariance is inherently and unavoidably sensitive to the choice of units on key variables. Consequently, we argue them to be inadmissible for nonspatial applications (i.e., where preservation of Euclidean distances between objects has no meaningful relevance), of which recommender systems represent a prime example.

In the following sections, we formally describe our framework and its properties relating to recommender systems. In Section 6 we provide empirical results suggesting that unit consistency alone is sufficient to provide performance competitive with state-of-the-art methods on standard benchmark datasets.

3 Canonical Scaling

In this section we present the our general canonical scaling algorithm (CSA) and formally establish both its correctness and the uniqueness of its result for the canonical scaling problem (CSP) of a $d$-dimensional tensor $A$. First we define our notation:

1. $A \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is a $d$-dimensional tensor with fixed dimensional extents $n_1, \ldots, n_d$. (The restriction to strictly positive, as opposed to nonnegative, entries is unnecessary for canonical scaling but will prove convenient when considering recommender-system applications of the tensor completion algorithm.)

2. $\bar{\alpha} = \{\alpha_1, \ldots, \alpha_d\} \in \mathbb{Z}^d$ is a $d$-dimensional vector that specifies an entry of $A$ as $A(\bar{\alpha})$.

3. $A_i$ is the $i^{th}$ element of the ordered set of all $k$-dimensional subtensors of $A$, where each subtensor is equal to $A$ but with a distinct subset of the $d - k$ extents restricted to 1.

4. $S_k$ is a strictly positive-valued vector of length equal to the number of $k$-dimensional subtensors of $A$. Then $S_{k,i}$ is the $i^{th}$ element of $S_k$.

5. $\sigma(A) = \{\alpha | A(\alpha)\}$ is the set of known/defined entries of tensor $A$; and its complement, $\overline{\sigma}(A)$, is the set of absent/missing entries of $A$.

6. Integer $N \doteq |\sigma(A)| + |\overline{\sigma}(A)| = |A|$ is the cardinality of the total index space of $A$. 
7. Integer \( n = |\sigma(A)| \) is the cardinality of known entries of \( A \).

8. Integer \( k < d \) denotes the dimensionality of a given subtensor.

9. \([m] = \{1, 2, ..., m\}\) is an index set of the first \( m \) natural numbers.

The following are two key definitions.

**Definition 3.1 - Sets of subtensors:** Let \( V_i = \mathbb{R}^{n_i} \) and assume a \( d \)-dimensional tensor \( A \in V_1 \times V_2 \times \cdots \times V_d \) and a positive integer \( 1 \leq k < d \). From the set of dimensional indices \( [d] = \{1, \ldots, d\} \) we can obtain all \( \binom{d}{k} \) possible tuples of \( k \) dimensional indices. Specifically, for any tuple \( \{i_1, \cdots, i_k\} \subseteq [d] \), we enumerate all possible \( k \)-dimensional subtensors of \( A \) in dimensions \( V_{i_1} \times \cdots \times V_{i_k} \) to obtain the set of subtensors \( (i) \). The union of the set of subtensors is called the ordered set of all \( (k) \)-dimensional subtensors of \( A \). We denote this set as \( A \) and the finite cardinality of the set \( A \) is \( |A| \). Also, its subtensor element is assumed to be labelled as \( A_{i} \). By formulaic notation:

\[
A = \{A_i, 1 \leq i \leq |A|\} = \bigcup_{i=1}^{(d)} (i) \quad (1)
\]

**Definition 3.2 - Scaling \( k \)-dimensional subtensors of a tensor:** For \( d \)-dimensional tensor \( A \), the product \( A' = S_{k} * A \) is defined as a scaling of each \( k \)-dimensional subtensor \( A_{i} \) of \( A \) to \( A'_{i} \) of \( A' \) as \( A'_{i} = S_{k,i} \cdot A_{i} \) or, equivalently, for each \( \vec{\alpha} \in \sigma(A) \)

\[
A' (\vec{\alpha}) \equiv A(\vec{\alpha}) \cdot \prod_{i: \vec{\alpha} \in A_{i}} S_{k,i}. \quad (2)
\]

The structured scaling of Definition 3.2 is a fundamental component of the solution described in the next section for the canonical scaling problem, which requires the transformation of a given nonnegative tensor to a unique scale-invariant canonical form.

### 3.1 Canonical scaling problem

We now summarize the formulation of the CSA for arbitrary scaling of \( k \)-dimensional subtensors of a given nonnegative tensor \( A \), which may be obtained by replacing the elements of a given tensor \( A' \) with their magnitudes\(^2\). Using definition 3.2, we have

**Canonical Scaling Problem (CSP):** Find the \( d \)-dimensional tensor \( A' \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d} \)

---

\(^2\)The decomposition of a given tensor into the Hadamard product of a nonnegative scaling tensor and a tensor with unit-magnitude negative or complex (or quaternion, octonion, or other basis form) entries can be thought of as a scale/unit-consistent analog of a polar decomposition. In direct analogy to the positive-definite component of the matrix polar decomposition, the nonnegative scaling tensor developed in this section is unique.
and a positive vector $S_k$ such that $A' = A * k_S k$ and the product of the known entries of each $i^{th}$-index $k$-dimensional subtensor $A_i'$ is 1.

The CSP formulation can be transformed to the following equivalent problem by taking logarithms of known entries and replacing the unit-product constraint with a zero-sum constraint.

**Log Canonical Scaling Problem (LCSP):** Find the $d$-dimensional tensor $a'$ and a vector $s_k$ such that $A' \equiv a'(\vec{\alpha}) = a(\vec{\alpha}) + \sum_{i:\vec{\alpha} \in A_i} s_{k,i}$, $\forall \vec{\alpha} \in \sigma(A)$, and each $i^{th}$-index $k$-dimensional subtensor $a_i'$ with the sum of its known entries equal to zero.

In previous work [1] we established based on [7] and [12] that LCSP is equivalent to the following Convex Optimization Problem (COP):

\[
\min_{x \in \mathbb{R}^{n_1 \times \cdots \times n_d}} 2^{-1} \sum_{\vec{\alpha} \in \sigma(A)} (x(\vec{\alpha}) - a(\vec{\alpha}))^2 \quad \text{subject to} \quad \sum_{\vec{\alpha} \in \sigma(A_i)} x(\vec{\alpha}) = 0, \quad \forall \text{subtensor } A_i.
\]  

(3)

### 3.1.1 Uniqueness

In the proofs of [1], we show that the equivalency between COP and CSP guarantees a feasible solution, hence the existence of a tensor $A'$ that satisfies CSP and has $\sigma(A) = \sigma(A')$. Specifically, we show that the COP problem is equivalent to the following optimization problem: Briefly, let $u \in \mathbb{R}^p, b \in \mathbb{R}^q$, and $C \in \mathbb{R}^{p \times q}$ be the original convex optimization problem of finding a vector $u' \in \mathbb{R}^p$ and $\omega \in \mathbb{R}^q$ such that

\[
\begin{align*}
    u'^T &= u^T + \omega^T C \\
    Cu' &= b
\end{align*}
\]  

(4)  

and

(5)

It is proven in [12] that this program is equivalent to the following optimization problem, for which the properties of uniqueness and existence can be established:

\[
\min_{x \in \mathbb{R}^n} 2^{-1} \sum_{j=1}^p (x_j - u_j)^2 \quad \text{subject to} \quad Cx = b.
\]  

(6)

Thus, the convergence theorem of the CSP follows from the aforementioned optimization problem, with the specific proofs given in our previous work [1]. Moreover, we know that if there exists a solution $x$ such that $Ax = b$, and vector $\omega$ such that $A\omega = 0$, then $x + \omega$ is another solution. This leads to the following theorem (formally proven in [1]) regarding solution uniqueness:
Theorem 3.1 (Uniqueness of $A'$) There exists at most one tensor $A'$ for which there exists a strictly positive vector $S_k$ such that the solution $(A', S_k)$ satisfies CSP. Furthermore, if a positive vector $T_k$ satisfies

$$\prod_{i: \bar{a} \in A_i} T_{k,i} = 1 \quad \forall \bar{a} \in \sigma(A),$$

(7)

then $(A', S_k \circ T_k)$ is also a solution, where $\circ$ is the Hadamard product.

3.1.2 CSA for tensor:

We now present Algorithm 1 for obtaining a unique scaled tensor $A' = \text{CSA}(A, k)$.

**Algorithm 1: CSA for tensor**

| Input: | d-dimensional tensor $A$. |
| Output: | $A'$ and scaling vector $S_k$. |
| Function CSA($A, k$): |
| - Step 1: Iterative step over constraints: Initialize $count \leftarrow 0$, variance variable $v \leftarrow 0$, and let $p$ be a zero vector of conformant length. Let $a$ be the logarithm conversion of $A$, i.e., all known entries are replaced with their logs. |
| for each subtensor $A_i$ with index $i$ do |
| $\rho_i = -[|\sigma(A_i)|]^{-1} \sum_{\bar{a} \in \sigma(A_i)} a^{\text{step}}(\bar{a})$ |
| $a(\bar{a}) \leftarrow a(\bar{a}) + \rho_i, v \leftarrow v + \rho_i^2$, for $\bar{a} \in \sigma(A_i)$ |
| $s_{k,i} \leftarrow s_{k,i} - \rho_i$ |
| - Step 2: Convergence: If $v$ is less than a selected threshold $\epsilon$, then exit loop. Otherwise, set $count \leftarrow count + 1$ and return to step 1. |
| return $A' = \exp(a)$ and $S_k = \exp(s_k)$. |

The time complexity of this algorithm is $O(|\sigma(A)|)$ and is unaffected by the final step of converting back from the log-space solution to the desired solution as $A(\bar{a}) = \exp(a(\bar{a}))$, which is also $O(|\sigma(A)|)$. A detailed examination of the contributions of each of the set of subtensors $(i)$ to this time complexity can be found in [1].

In the case of $d=2$ when $A$ is a matrix, i.e., CSA($A, k=1$), the set of $k=1$ subtensors is simply the set of rows and columns. This is a specialized instance of a matrix problem studied in [12] with rows and columns explicitly distinguished for the problem of scaling line products of a matrix to chosen positive values (for which a solution is not guaranteed to exist except in the case we use of all scaling values equal to 1) but from our generalized tensor formulation it can be seen that such a distinction is unnecessary.
4 Tensor Completion Algorithm

The CSA process provides the basis for the following tensor completion algorithm.

**Algorithm 2:** Tensor completion algorithm (TCA)

**Input:** \(d\)-dimensional tensor \(A\) and \(k\).

**Output:** \(A'\)

**Function TCA** \((A,k)\):

- **Step 1:** CSA process.
  \[ S_k \leftarrow \text{CSA}(A,k) \]

- **Step 2:** Tensor completion process:
  \[
  A' = A \\
  \text{for } \vec{\alpha} \in \sigma(A) \text{ do} \\
  \quad A'(\vec{\alpha}) \leftarrow \prod_{i: \vec{\alpha} \in A_i} S_{k,i}^{-1}.
  \]

return \(A'\).

4.1 Uniqueness and Full Support

Although the canonical scaled tensor is unique, the scaling vectors may not be unless there are sufficient known entries to provide full support, which is now defined.

**Definition 4.1** For two vectors \(\vec{\alpha}, \vec{\alpha}' \in \mathbb{Z}_{>0}^d\) such that \(\alpha_j \neq \alpha'_j\) for all \(j \leq d\), define a hypercube ordered set of \(2^d - 1\) vectors with respect to \(\vec{\alpha}\) as \(H(\vec{\alpha}, \vec{\alpha}') = \{ \vec{\beta}_i \neq \vec{\alpha}, 1 \leq i \leq 2^d - 1 \}\), where each \(\vec{\beta}_i\) has the \(j\)-th component \(\beta_{ij}\) equals either \(\alpha_j\) or \(\alpha'_j\), for all \(j \leq d\). Then given a positive tensor \(A\), define a tensor \(A\) as fully-supported if for every entry \(\vec{\alpha} \in \sigma(A)\), there exists another vector \(\vec{\alpha}'\) such that \(\alpha_j \neq \alpha'_j\) for all \(j \leq d\) and \(\forall \vec{\beta}_i \in H(\vec{\alpha}, \vec{\alpha}'), \vec{\beta}_i \in \sigma(A)\).

Definition [4.1] essentially says that every unknown entry forms a vertex of a \(d\)-dimensional hypercube with known entries at the remaining vertices. In the \(d = 2\) matrix case, for example, an unknown entry \((i, j)\) must have known entries at \((i+p, j), (i, j+q),\) and \((i+p, j+q)\) for some \(p \neq 0\) and \(q \neq 0\). In this case, \(\vec{\alpha} = (i,j)\) and \(\vec{\alpha}' = (i+p, j+q)\). Thus, full support can be expected to hold even with very high sparsity. From this point forward we will assume full support unless otherwise stated.

4.1.1 Completion Uniqueness

To facilitate the subsequent theorem and proof, we introduce the following definition.

**Definition 4.2** Given \(\vec{\alpha}\) as a coordinate vector of tensor \(A\). A vector \(\vec{v} \in \mathbb{Z}_{>0}^k\) for \(k \leq d\) is considered a \(k\)-dimensional subvector of \(\vec{\alpha}\) if \(\vec{v} = (\alpha_{\gamma_1}, \ldots, \alpha_{\gamma_k})\) for \(\{\gamma_1, \gamma_2, \ldots, \gamma_k\} \subseteq [d]\). We define the set of those \(k\)-dimensional subvectors of a tensor \(A\) as \(V_k(\vec{\alpha})\).
Observation 4.1 For any $\vec{\alpha}$, each choice of $k$-dimensional subvector $\vec{v}$ has a 1-to-1 relation with a unique $k$-dimensional subtensor of $A$.

Using this definition, we obtain the following theorem regarding uniqueness of the recommendation/entry-completion result.

Theorem 4.1 (Uniqueness) The result from TCA($A, k$) is uniquely determined even if there are distinct sets of scaling vectors $\{S_k\}$ that yield the same, unique, CSA($A, k$).

Proof 4.1.1 For $A' = \text{TCA}(A, k)$, since entry $\vec{\alpha} \in \sigma(A)$ has $A'(|\vec{\alpha}|) = A(|\vec{\alpha}|)$ by the uniqueness theorem 3.1, we need only prove uniqueness of any completion $\vec{\alpha} \in \mathcal{P}(A)$. From the uniqueness result of Theorem 3.1, and assuming that TCA($A, k$) admits two distinct scaling vectors $S_k$ and $S'_k$, we can infer that an entry $S_k$ from $S_k$ represents a unique $k$-dimensional subtensor of $A$ for each $i$. From definition 4.1, there exists a vector $\vec{\alpha}'$ such that $\alpha_i \neq \alpha'_i$ for all $j \leq d$ and a hypercube set $H(\vec{\alpha}, \vec{\alpha}')$ of vectors $\vec{\beta}_i$ such that $\vec{\beta}_i \in \sigma(A)$. From theorem 3.1, the scaling vector $S'_k$ equals $S_k \circ T_k$ such that such that for $\vec{v} \in V_k(\vec{\beta}_i)$, the coefficient notation associated with $\vec{v}$ as $T_\vec{v}$ is equivalent to notation (4) of section 2: $T_\vec{v} \equiv T_k, j$. Here the index $j$ represents an $j$-th $k$-dimensional subtensor $A_j$ of $A$ such that $\vec{\beta}_i \in A_j$, and thus

$$\prod_{\vec{v} \in V_k(\vec{\beta}_i)} T_\vec{v} = 1. \quad (9)$$

We now show that $A'(|\vec{\alpha}|) = \prod_{S_k, i, \vec{\alpha} \in A_i} S_k^{-1}$ is unique, which is equivalent to $\prod_{S_k, i, \vec{\alpha} \in A_i} S_k, i = \prod_{\vec{v} \in V_k(\vec{\alpha})} S_{\vec{v}},$

$$\prod_{\vec{v} \in V_k(\vec{\alpha})} S_{\vec{v}} = \prod_{\vec{v} \in V_k(\vec{\alpha})} S_{\vec{v}} \quad (10)$$

or equivalently from theorem 3.1

$$\prod_{\vec{v} \in V_k(\vec{\alpha})} T_\vec{v} = 1. \quad (11)$$

Without loss of generality, we consider the case $d \equiv 0 \pmod{2}$, and the other case can be proved similarly. Denote that $\vec{\beta}_2 = \vec{\alpha}$ and $\vec{\beta}_0 = \vec{\alpha}$, we divide $2^d - 2$ remaining vectors $\vec{\beta}_i$ for $1 \leq i \leq 2^d - 2$ into two groups. If the number of coordinates in $\vec{\beta}_i$ that match with those in $\vec{\alpha}$ equals $j$ modulo 2, then $\vec{\beta}_i$ goes to $G_j$. Thus we have two groups $G_0$ and $G_1$. For $G \in \{G_0, G_1\}$, consider $V(G) = \bigcup_{\vec{\beta}_i \in G} V_k(\vec{\beta}_i)$ for some $i$ and let $V_G(\vec{v}) = \{\vec{\beta}_i \in G \mid \vec{v} \in V_k(\vec{\beta}_i)\}$ and $|V_G(\vec{v})|$ as the set’s cardinality. Then from equation 7

$$\prod_{\vec{\beta}_i \in G} \prod_{\vec{v} \in V_k(\vec{\beta}_i)} T_\vec{v} = 1 \iff \prod_{\vec{v} \in V(G)} T_\vec{v}^{\left|V_G(\vec{v})\right|} = 1. \quad (12)$$
Consider $\vec{v} \in \bigcup_{i=1}^{d-1} V_k(\vec{b}_i)$. Let’s say that $\vec{v}$ has $l_i$ coordinates that match with those in $\vec{\alpha}$, then the remaining $k-l_i$ elements of $\vec{v}$ match with those in $\vec{\alpha}'$. Consider case 1 when $0 < l_i < k$. For each $\vec{v}$, we form $\vec{\beta}_i$ by choosing additional $m \leq d-k$ elements among $d-k$ remaining elements of $\vec{\alpha}$ such that $m + l_i$ is the number of coordinates in vector $\vec{\beta}_i$ that match with $\vec{\alpha}$. For $j \in \{0, 1\}$, if $m + l_i \equiv j(\text{mod } 2)$, then $\vec{v}$ forms $(d^m_k)$ numbers of $\vec{\beta}_i$ that belongs to $G_j$. So when we sum between $0 \leq m \leq d-k$, $|V_{G_0}(\vec{v})| = \sum_{m+l_i \equiv 0(\text{mod } 2)} (d^m_k) = 2^{d-k-1}$ and $|V_{G_1}(\vec{v})| = \sum_{m+l_i \equiv 1(\text{mod } 2)} (d^m_k) = 2^{d-k-1}$. Thus,

$$\frac{T_{\vec{v}}^{|V_{G_1}(\vec{v})|}}{T_{\vec{v}}^{|V_{G_0}(\vec{v})|}} = 1. \quad (13)$$

If $l_i = k$, we encounter the vector $\vec{\beta}_i = \vec{\alpha}$ when forming $\vec{\beta}_i$. If $l_i = 0$, we encounter the vector $\vec{\beta}_i = \vec{\alpha}'$ when forming $\vec{\beta}_i$. Since we omit $\vec{\alpha}$ and $\vec{\alpha}'$ from $G_0$, $|V_{G_0}(\vec{v})| = 2^{d-k-1} - 1$ and $|V_{G_1}(\vec{v})| = 2^{d-k-1}$ in either case of $l_i$. Thus,

$$\frac{T_{\vec{v}}^{|V_{G_1}(\vec{v})|}}{T_{\vec{v}}^{|V_{G_0}(\vec{v})|}} = T_{\vec{v}} \quad (14)$$

and therefore

$$\prod_{\vec{v} \in V(G_1)} T_{\vec{v}}^{|V_{G_1}(\vec{v})|} \prod_{\vec{v} \in V(G_0)} T_{\vec{v}}^{|V_{G_0}(\vec{v})|} = 1 \Rightarrow \prod_{\vec{v} \in V(G_1)} T_{\vec{v}} \prod_{\vec{v} \in V(G_0)} T_{\vec{v}} = 1 \Rightarrow \prod_{\vec{v} \in V_k(\vec{\alpha})} T_{\vec{v}} = 1. \quad (15)$$

This equality implies that $A'$ is unchanged, and thus uniquely determined, regardless of whether there exist distinct scaling vectors $S_k$ and $S_k'$.

### 4.2 Unit Consistency

$A' = \text{CSA}(A, k)$ guarantees scale-invariance with respect to every $k$-dimensional subtensor of $A'$. It then remains to show that the computation result from $\text{TCA}(A, k)$ is unit-consistent.

**Theorem 4.2 (Unit-consistency)** Given a tensor $A$ and an arbitrary conorm-formant positive scaling vector $C_k, C_k *_k \text{TCA}(A, k) = \text{TCA}(C_k *_k A, k)$, where $C_k$ scales all $k$-dimensional subtensors of $A$ (with operator $*_k$ as defined in definition 3.3).

**Proof 4.2.1** Let $S_k \leftarrow \text{CSA}(A, k)$. It can be shown that $A' = \text{CSA}(C_k *_k A, k) = \text{CSA}(A, k)$ for all $A$. With an abuse of notation, we assume all unknown entries of $A'$ are assigned the value of 1, i.e., $A'(\vec{\alpha}) = 1$ for $\vec{\alpha} \in \mathfrak{F}(A)$. The complete $\text{TCA}$ process can then be defined as $\text{TCA}(A, k) = S_k^{(-1)} *_k A'$, where $S_k^{(-1)} \{S_k^{(-1)}\}$ is the inverse vector of $S_k$. Now, using the uniqueness theorem 4.7, we can subsume the scaling vector $C_k$ into $S_k$ and deduce that

$$C_k *_k \text{TCA}(A, k) = (S_k^{(-1)} \circ C_k) *_k A' = \text{TCA}(C_k *_k A, k) \quad (16)$$
The time complexity for CSA$(A, k)$ is clearly $O(|\sigma(A)|)$, and the same complexity applies to the TCA process. The details are in [1], and in brief we consider addition from each of the set of subtensors $(i)$ to acquire the time complexity.

### 4.3 Consensus-Ordering Consistency

When $k = d - 1$, TCA can be shown to respect an ordering relationship of the $(d-1)$-dimensional subtensors of a given $d$-dimensional tensor $A$.

**Definition 4.3** Given a tensor $A$, we define a permutation for list of indices $D = \{d_1, \ldots, d_D\} \subseteq [n_d]$ as $\gamma_D = \{\gamma(d_1), \ldots, \gamma(d_D)\}$. Define a vector $\bar{\alpha} \in \mathbb{Z}_{\geq 0}^{d-1}$ that preserves/follows ordering $\gamma_D$ in tensor $A$ if

1. (Known terms condition): $A(\bar{\alpha}, \gamma(d_l)) > 0$ for all $1 \leq l \leq |D|$.

2. (Ordering condition): $A(\bar{\alpha}, \gamma(d_1)) < \cdots < A(\bar{\alpha}, \gamma(d_l)) < \cdots < A(\bar{\alpha}, \gamma(d_D))$ for $1 < l < |D|$.

Note that tensor $A'$ from TCA$(A,d - 1)$ also has the vector $\bar{\alpha}$ following the same ordering $\gamma_D = \{\gamma(d_1), \ldots, \gamma(d_D)\}$ with respect to tensor $A$. We have the following theorem:

**Theorem 4.3** (Consensus-ordering) Given a fully-supported tensor $A$, the obtained result $A' = TCA(A,d - 1)$, the set $D = \{d_1, \ldots, d_D\} \subseteq [n_d]$ with permutation $\gamma_D = \{\gamma(d_1), \ldots, \gamma(d_D)\}$, and the set of known vectors on the subtensor as $NZ_l = \{\bar{\alpha} \in \mathbb{Z}_{\geq 0}^{d-1}|(\bar{\alpha}, l) \in \sigma(A)\}$ with label $l \in D$, with $NZ_l = NZ$ assumed fixed for all $p \in D$. Given that each vector in $NZ$ follows ordering $\gamma_D$, then any new vector $\bar{\alpha}' \in NZ^C$, where $NZ^C$ is the complement set of $NZ$, follows the ordering $\gamma_D$ in tensor $A'$ and the completion result in $A'$ is unique.

**Proof 4.3.1** For any vector $\bar{\alpha} = (\alpha_1, \ldots, \alpha_{d-1})$, the ordering condition from definition 4.3 gives:

$$A(\bar{\alpha}, \gamma(d_1)) < \cdots < A(\bar{\alpha}, \gamma(d_l)) < \cdots < A(\bar{\alpha}, \gamma(d_D))$$

(17)

Given that $A'$ from TCA is normalized via CSA, then without loss of generality we can omit the dimension $k$ when using notation (4) from section 2, since $k$ is specified as $k = d - 1$. Specifically, after the TCA process, with vector $S = S_{d-1}$, $S_i = S_{d-1,i}$ as the entry of vector $S$.

$$A(\bar{\alpha}, \gamma(d_i)) = \prod_{l=1}^{d-1} S_{\alpha_l}^{-1} \cdot S_{\gamma(d_i)}^{-1} \cdot A'(\bar{\alpha}, \gamma(d_i)) = S_{\bar{\alpha}}^{-1} \cdot S_{\gamma(d_i)}^{-1} \cdot A'(\bar{\alpha}, \gamma(d_i))$$

Substituting into the above inequality, we have:

$$A'(\bar{\alpha}, \gamma(d_1)) \cdot S_{\gamma(d_1)}^{-1} \cdot \cdots < A'(\bar{\alpha}, \gamma(d_i)) \cdot S_{\gamma(d_i)}^{-1} \cdots < \cdots < A'(\bar{\alpha}, \gamma(d_D)) \cdot S_{\gamma(d_D)}^{-1}$$

(18)
Using the known-terms condition from Definition 4.3, the assumption that \(A(\vec{\alpha}', p)\) is unknown for all \(p \in [n_d]\) and \(\vec{\alpha} \in NZ^C\), and the CSA result with respect to \((d - 1)\)-dimensional subtensor in direction \(\gamma(d_i)\):

\[
\prod_{\vec{\alpha} \in NZ} A'(\vec{\alpha}, \gamma(d_i)) = 1.
\] (19)

Substituting (19) into (18) gives

\[
S_{\gamma(d_1)}^{-1} < \cdots < S_{\gamma(d_i)}^{-1} < \cdots < S_{\gamma(d_{|D|})}^{-1}.
\] (20)

For any vector \(\vec{\alpha}' \in NZ^C\), the following entry is uniquely determined from theorem 4.7,

\[
A'(\vec{\alpha}', \gamma(d_i)) = S_{\vec{\alpha}'} \cdot S_{\gamma(d_i)}^{-1},
\] (21)

and we therefore deduce that

\[
A'(\vec{\alpha}', \gamma(d_1)) < \cdots < A'(\vec{\alpha}', \gamma(d_i)) < \cdots < A'(\vec{\alpha}', \gamma(d_{|D|})),
\] (22)

thus completing the proof.

5 UC Completion for Recommender Systems

In this section we examine applications of UC completion in the area of recommender systems. Because of its particular relevance to matrix-formulated recommender-system problems, we briefly discuss the special case of \(d = 2, k = 1\), for a given \(m \times n\) matrix \(A \in \mathbb{R}_{>0}^{m \times n}\) with full support. For notational convenience, we define the matrix completion function \(MCA(A)\) as a special case of \(TCA\):

\[
MCA(A) \equiv TCA(A, 1) \text{ for } d = 2.
\] (23)

In this case, unit-consistency property can be expressed as \(R \cdot MCA(A) \cdot C = MCA(R \cdot A \cdot C)\) for positive vectors \(R\) and \(C\), which in terms of matrix multiplication with positive diagonal matrices \(R\) and \(C\) implies \(R \cdot MCA(A) \cdot C = MCA(RAC)\). The time and space complexity for \(MCA(A)\) is \(O(|\sigma(A)|)\).

Both the theorem 4.7 and definition 4.3 allow us to quantify how we can look into the ordering of the recommender system through the following definition.

Definition 5.1 Denote \(RS(\vec{\alpha}) = A'(\vec{\alpha})\) as the recommendation result for vector position \(\vec{\alpha}\) from tensor \(A\) with \(A' = TCA(A, d - 1)\).

Because all results in this paper are transposition consistent, we implicitly assume without loss of generality that \(n \geq m\) purely to be consistent with our general use of \(n\) as the variable that functionally determines the time and space complexity of our algorithms.
5.1 Consensus-ordering

Following by definition 4.3 and theorem 4.3, we formally state consensus-ordering property in the context of matrix, 3-dimensional tensor, 4-dimensional tensor, and $d$-dimensional tensor.

**Corollary 5.0.1** (Consensus-ordering for matrix) Given a matrix $A$, $MCA(A)$, set of products $P \subseteq [n]$, and set of users $U \subseteq [m]$. We have the following statements:

1. Given all users $u \in U$ follows ordering $\gamma_P$. Then the recommendation result $RS(u',p)$ for any user-product $(u',p) \in U^C \times P$ is unique and follows ordering $\gamma_P$.

2. Given all products $p \in P$ follows $\gamma_U$. Then the recommendation result $RS(u,p')$ for any user-product $(u,p') \in U \times P^C$ is unique and follows ordering $\gamma_U$.

For 3D tensor, we have the following interpretation.

**Corollary 5.0.2** (Consensus ordering for 3-dimensional tensor) Given a 3-dimensional tensor $A \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Let the first, second, and third dimension be users, attributes, and products and the subsets of users, attributes, and products as $U \subseteq [n_1], \Gamma \subseteq [n_2], \text{ and } P \subseteq [n_3]$. Define on a slice of user $u$ the set of known entries by the vector of attribute and product $NZ_u = \{(\alpha,p) \in [n_2] \times [n_3] | (u,\alpha,p) \in \sigma(A)\}$. Similarly, we define the set of known entries for the attribute $\alpha$ slice as $NZ_\alpha$ and that for the product $p$ slice $NZ_p$. For the recommendation result from the TCA($A,2$) and fixed set $NZ$, we have the following statements:

1. Assuming that for all $u \in U$, $NZ_u = NZ$. If all the ratings from vector of attribute and product $(\alpha,p) \in NZ$ follows ordering $\gamma_U$, the recommendation result $RS(u,\alpha',p')$ for attribute-product vector $(\alpha',p') \in NZ^C$ and each user $u \in U$ is unique and also follows ordering $\gamma_U$.

2. Assuming that for all $p \in P$, $NZ_p = NZ$. If all the ratings from vector of user and attribute $(u,\alpha) \in NZ$ follows ordering $\gamma_P$, the recommendation result $RS(u',\alpha,p)$ for user-attribute vector $(u',\alpha) \in NZ^C$ and each product $p \in P$ is unique and follows ordering $\gamma_P$.

3. Assuming that for all $\alpha \in \Gamma$, $NZ_\alpha = NZ$. If all the scores based on vector of user and attribute $(u,\alpha) \in NZ$ follows ordering $\gamma_\Gamma$, the recommendation result $RS(u',\alpha,p')$ for user-product vector $(u',\alpha,p') \in NZ^C$ and each attribute $\alpha \in \Gamma$ is unique and follows ordering $\gamma_\Gamma$.

The strongest part of this corollary is the symmetric property with respect to the attributes, regardless of whether the attributes are all from the users or products. The cases from the matrix to the 4-dimensional tensor provide the intuition to reasoning from ordering with respect to the choice of spatial label.
on different coordinates.

For the $d$-dimensional tensor, we denote user as $u$, product as $p$, $\vec{\alpha}_u$ as the vector of user’s attributes, $\vec{\alpha}_p$ as the vector of product’s attributes. We can generalize all the above cases into a $d$-dimensional scaling by the following corollary, assuming the only features we focus on are user, user’s attributes, product’s attributes, and product, following theorem 4.3.

**Corollary 5.0.3 (Consensus-ordering for $d$-dimensional tensor)** Given a $d$-dimensional tensor $A \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and the recommendation outcome $A' = TCA(A, d - 1)$. Denote the subset of entries in the $d$ dimension is $D \subseteq [n_d]$, and the subset of vectors indicating known subtensor labeled as $l \in [n_d]$ as $NZ_l = \{\vec{\alpha} \in [n_1] \times \cdots \times [n_{d-1}] \mid (\vec{\alpha}, l) \in \sigma(A)\}$. Moreover, denote user as $u$, product as $p$, $\vec{\alpha}_u$ as the vector of user’s attribute, $\vec{\alpha}_p$ as the vector of user’s attribute. Assuming that $NZ_l = NZ$ for all $l \in D$ and every vector in $NZ$ follows ordering $\gamma_D$, we have the following cases:

1. The set of user-attributes vectors $\{\vec{\alpha} = (u, \vec{\alpha}_u, \vec{\alpha}_p)\}$ and the set of products $D = \{p_1, \ldots, p_{|D|}\}$.

2. The set of product-attributes vectors $\{\vec{\alpha} = (\vec{\alpha}_u, \vec{\alpha}_p, p)\}$ and the set of users $D = \{u_1, \ldots, u_{|D|}\}$.

3. The set of user-attributes-product vectors $\{\vec{\alpha} = (u, \vec{\alpha}_u, \vec{\alpha}_p, p)\}$ and the set of user’s last attribute $D = \{\alpha_{u_1}, \ldots, \alpha_{u_{|D|}}\}$.

4. The set of user-attributes-product vectors $\{\vec{\alpha} = (u, \vec{\alpha}_u, \vec{\alpha}_p, p)\}$ and the set of product’s last attribute $D = \{\alpha_{p_1}, \ldots, \alpha_{p_{|D|}}\}$.

Then all cases conclude that any new recommendation result $RS(\vec{\alpha}, u)$ for vector $(\vec{\alpha}', u) \in NZ^C \times D$ is unique and follows ordering $\gamma_D$ for $D$ is defined above.

This corollary sums up how the consensus-ordering property with respect to the same index label in the original tensor preserves in the output tensor. As long as there exists unanimity among attributes on the first $d - 1$ dimensions with respect to the last coordinate, the ordering follows trivially by theorem 4.3.

Both consensus-ordering and unit-consistent properties generalize to higher dimensions. For this paper, we can only deduce statements in the case of $(d-1)$-scaling. Assuming that any vector relationship, such as user-attribute-product, is on the first $(d-1)$ dimensions of the $d$-dimensional tensor and is uniformly unanimous over the subset of indices on the last coordinate. If all users and products have at least one shared coordinate, e.g., attribute, among the list of spatial indexing attributes, the ordering statement by that attribute coordinates with any new user, and the attribute vector follows by the permutation index from that indexing list. The same argument follows if we interchange the relationship of rating scores among users, product, and different attributes.
5.1.1 Time complexity

For the tensor completion process, the time complexity to make a user’s query is the number of diagonal matrices needed to give predictions for a user in tensor $A$. We only need to choose one term in each of such diagonal matrices and multiply them all together. As a recommender query only takes $O(d)$ time complexity for a user, we can omit this term in complexity analysis. As the recommendation process is $\text{TCA}(A, d - 1)$, we have the following theorem:

**Theorem 5.1 (Time complexity)** The time complexity for the RS procedure on tensor $A$ is $O(|\sigma(A)|)$.

**Proof 5.1.1** The $\text{TCA}(A, d - 1)$ process has time complexity $O(|\sigma(A)|)$.

As $d = 2$, the time complexity of the RS procedure is $O(|\sigma(A)|)$.

**Corollary 5.1.1 (Time complexity)** The time complexity for the recommendation procedure on matrix $A$ is $O(|\sigma(A)|)$.

6 Empirical Corroboration of Performance Quality

In this section we provide empirical results comparing our approach to state-of-the-art methods on standard datasets using the standard measure of performance: root mean-squared error (RMSE). As a measure of squared error, we naturally argue that RMSE is a poor choice because it is strongly scale-sensitive, e.g., users who give consistently higher ratings (larger magnitude) will have a greater influence on system recommendations than users who tend to give more moderate or lower ratings. However, as we believe that our unit-consistent (UC) approach is the only one that satisfies necessary conditions that are fundamental to the problem, we expect it to perform comparably to or better than other methods according to any metric.

Our first set of comparison results is for the MovieLens-1M benchmark dataset, shown in Figure 1. The results show that our UC method and the top-performing state-of-the-art methods are almost indistinguishable according RMSE. This is significant because the competing methods are implicitly or explicitly designed to minimize squared error. Our UC approach, by contrast, is not at all designed to minimize squared error, and yet it performs nearly identically to methods that are tailored to minimize this measure.

Our second set of comparison results is for the Jester-2 benchmark dataset, shown in Figure 2. As can be seen, relative order of performance is very different for this dataset compared to the previous one, which tends to suggest that RMSE is ineffective for robustly characterizing relative performance across

---

4Thanks to Nikki Hotrabhavananda and Will Starms for obtaining comparison results provided in this section.
Figure 1: The UC approach yields results comparable to state-of-the-art on the standard MovieLens-1m benchmark dataset according to RMSE. SVD and other unitarily-invariant optimization methods implicitly minimize RMSE, which is an essentially arbitrary measure of effectiveness. The fact that UC performs nearly identically according to this measure suggests that it is capturing something fundamental about the problem.

different datasets. Again, the UC method performs comparably to state-of-the-art methods. If the conclusions of our analyses are correct, we should expect the UC approach to be similarly competitive with methods that are tailored to minimize other measures of error – even according to those measures for which they are tailored to minimize.

The most significant conclusion to be drawn is that the results presented in Figures 1 and 2 are consistent with our hypothesis that the UC method should yield near-optimal performance according to any chosen measure of error. The results provided in this section are clearly too limited to be fully convincing, so we look forward to results from a more comprehensive battery of future tests involving larger and more diverse datasets.

7 Discussion and Future Work

We have discussed that it is possible to transform a matrix to a unique scale-canonical form in which the product of nonzero elements in each row and column
Figure 2: The UC approach also yields results comparable to state-of-the-art on the standard Jester-2 benchmark dataset. Variations in performance for the various methods across the MovieLens, Jester-2, and other standard datasets suggests that RMSE is a relatively low-accuracy measure for this problem. Despite the large variation in results from one benchmark dataset to another, the UC method consistently performs comparably to or better than state-of-the-art methods that are tailored to minimize RMSE.
is 1. We showed that this provides a means for performing unit-consistent matrix completion by replacing missing entries in the scale-canonical form with 1s, and then applying the inverse scaling to obtain the completed matrix. We then showed that this completion algorithm can be generalized for unit-consistent tensor completion.

We have argued that black-box AI systems, or generic methods to optimize an arbitrary metric (e.g., RMSE), are unnecessary for recommender system applications because a single criterion, unit consistency, is entirely sufficient to efficiently obtain unique solutions that have provably rigorous – and intuitively expected – properties. We have provided empirical evidence to support our hypothesis that our approach should be competitive with state-of-the-art alternatives according to any chosen metric – even methods that are explicitly designed to optimize that chosen metric.

It can be verified that the intermediate logspace solution we employed in our matrix and tensor scaling steps can be applied directly to achieve Translation Consistent (TC) matrix/tensor completions. Future work will more thoroughly examine its various properties and the problems to which it may be applicable.

Finally, our philosophical approach to the recommender system problem was to identify properties that should be expected of any reasonable solution. We identified unit consistency as being such a property, and we showed that it alone is sufficient to yield a unique solution without need for generic optimization and/or heuristic AI methods. This is surprising because the problem seems at first glance to demand unknowable information deriving from individual human psychology, thus appearing to be unavoidably within the domain of AI. This example shows that recognition of a few constraints on the solution space can sometimes replace profound mystery with simple clarity.

8 Acknowledgements

The first author would like to thank the Electrical Engineering and Computer Science (EECS) Department of the University of Missouri-Columbia for an Undergraduate Research Stipend that helped to support this work.

References

[1] T. Nguyen and J. Uhlmann, “Canonical tensor scaling,” Jou. Cie. Ing., vol. 13, no. 1, pp. 13-21, 2021.

[2] R. Sinkhorn, “A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices,” The Annals of Mathematical Statistics, Ann. Math. Statist. 35(2), 876-879, June, 1964.

[3] Y. Hu, Y. Koren and C. Volinsky, “Collaborative Filtering for Implicit Feedback Datasets,” 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 263-272.
[4] P. Drineas, I. Kerenidis, and P. Raghavan, “Competitive recommender systems.” In Proceedings of the thirty-fourth annual symposium on Theory of computing (STOC ’02).

[5] Moritz Hardt, “Understanding alternating minimization for matrix completion,” 2014 IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS), 2014.

[6] Ruoyu Sun and Zhi-Quan Luo, “Guaranteed Matrix Completion via Non-convex Factorization.” 2015 IEEE 56th Annual Symposium on Foundations of Computer Science, 2015.

[7] H. W. Kuhn, A. W. Tucker, “Nonlinear Programming.” Proceedings of the Second Berkeley Symposium on mathematical Statistics and Probability, Berkeley, U. of Calif. Press, 1950, 481–492.

[8] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. “Item-based collaborative filtering recommendation algorithms”. In Proceedings of the 10th conference on World Wide Web (WWW ’01).

[9] J. Pereira, P. Matuszyk, S. Krieter, M. Spiliopoulou, and G. Saake. “A feature-based personalized recommender system for product-line configuration”. In Proceedings of the 2016 International Conference on Generative Programming: Concepts and Experiences (GPCE 2016).

[10] E. Tang. 2019, “A quantum-inspired classical algorithm for recommendation systems”, In Proceedings of the 51st Annual Symposium on Theory of Computing (STOC 2019).

[11] Y. Zhang and X. Chen, “Explainable Recommendation: A Survey and New Perspectives”, Foundations and Trends in Information Retrieval, vol. 14, no. 1, pp. 1-101.

[12] U.G. Rothblum and S.A. Zenios, “Scalings of Matrices Satisfying Line-Product Constraints and Generalizations,” Linear Algebra and Its Applications, 175:159-175, 1992.

[13] Z. Song, D. P. Woodruff, and H. Zhang. “Sublinear time orthogonal tensor decomposition”. In Neural Information Processing Systems, 2016.

[14] E. Hazan, T. Koren, and N. Srebro. “Beating SGD: Learning SVMs in sublinear time”. In Neural Information Processing Systems, 2011.

[15] R. Kannan and S. Vempala. “Randomized algorithms in numerical linear algebra”. Acta Numerica, 26:95–135, 2017.

[16] Y. Koren, R. Bell, and C. Volinsky. “Matrix factorization techniques for recommender systems”. Computer, 42(8), 2009.

[17] R. Bell and Y. Koren, “Lessons from the Netflix prize challenge”. ACM SIGKDD Explorations Newsletter, 9(2):75–79, 2007.
[18] Jeffrey Uhlmann, “A Generalized Matrix Inverse that is Consistent with Respect to Diagonal Transformations,” *SIAM Journal on Matrix Analysis* (SIMAX), 2018.

[19] Jeffrey Uhlmann, “A Rank-Preserving Generalized Matrix Inverse for Consistency with Respect to Similarity,” *IEEE Control Systems Letters*, ISSN: 2475-1456, 2018.

[20] J.K. Uhlmann, “On the Relative Gain Array (RGA) with Singular and Rectangular Matrices,” *Applied Mathematics Letters*, Vol. 93, 2019.

[21] Bo Zhang and Jeffrey Uhlmann, “Applying a Unit-Consistent Generalized Matrix Inverse for Stable Control of Robotic Systems,” *ASME J. of Mechanisms and Robotics*, 11(3), 2019.

[22] Bo Zhang and Jeffrey Uhlmann, “A Generalized Matrix Inverse with Applications to Robotic Systems,” Bo Zhang and Jeffrey Uhlmann, [https://arxiv.org/pdf/1806.01776.pdf](https://arxiv.org/pdf/1806.01776.pdf), 2018.

[23] Jeffrey Uhlmann, “A Scale-Consistent Approach for Recommender Systems,” (arxiv.org/pdf/1905.00055.pdf), 2019.

[24] Benjamin Recht. “A Simpler Approach to Matrix Completion,” *J. Mach. Learn. Res.* 12, 3413–3430, 2011.

[25] Boaz Barak and Ankur Moitra. “Noisy Tensor Completion via the Sum-of-Squares Hierarchy.” *COLT 2016*: 417-445.

[26] Aaron Potechin and David Steurer. “Exact tensor completion with sum-of-squares.” *Proceedings of the 2017 Conference on Learning Theory*, PMLR 65:1619-1673, 2017.

[27] Daniel Lemire. “Scale and translation invariant collaborative filtering systems.” *Information Retrieval*, 8(1), 129–150, 2005.

[28] Guangxiang Zeng, Hengshu Zhu, Qi Liu, Ping Luo, Enhong Chen, and Tong Zhang. 2015. “Matrix factorization with scale-invariant parameters.” *In Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI’15)*. AAAI Press, 4017–4024.

[29] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints.” *International Journal of Computer Vision* 60, 91–110, 2004.

[30] Gihwi Kim, Ilyoung Choi, Qinglong Li, and Jackyeong Kim. “A CNN-Based Advertisement Recommendation through Real-Time User Face Recognition.” *Applied Science*, 2021.

[31] David Pennock, Eric Horvitz, and C. Giles. “Social Choice Theory and Recommender Systems: Analysis of the Axiomatic Foundations of Collaborative Filtering”. *Proceedings of AAAI Conference*, 2000.