# Measuring the Predictability of Recommender Systems using Structural Complexity Metrics

Alfonso Valderrama
Department of Computer Science, University of Chile
Center for Artificial Intelligence (CENIA)
alfonso.valderrama@ug.uchile.cl

Andrés Abeliuk
Department of Computer Science, University of Chile
Center for Artificial Intelligence (CENIA)
aabeliuk@dcc.uchile.cl

## ABSTRACT

Recommender systems (RS) are central to the filtering and curation of online content. These algorithms predict user ratings for unseen items based on past preferences. Despite their importance, the innate predictability of RS has received limited attention. This study introduces data-driven metrics to measure the predictability of RS based on the structural complexity of the user-item rating matrix. A low predictability score indicates complex and unpredictable user-item interactions, while a high predictability score reveals less complex patterns with predictive potential. We propose two strategies that use singular value decomposition (SVD) and matrix factorization (MF) to measure structural complexity. By perturbing the data and evaluating the prediction of the perturbed version, we explore the structural consistency indicated by the SVD singular vectors. The assumption is that a random perturbation of highly structured data does not change its structure. Empirical results show a high correlation between our metrics and the accuracy of the best-performing prediction algorithms on real data sets.

## KEYWORDS

Recommender systems, Collaborative filtering, Predictability, Data structure, Machine learning

**ACM Reference Format:**
Alfonso Valderrama and Andrés Abeliuk. 2023. Measuring the Predictability of Recommender Systems using Structural Complexity Metrics. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

## 1 INTRODUCTION

As the amount of information and content available to users continues to explode, recommender systems play an essential role in enhancing users' experience in areas ranging from e-commerce and entertainment to social media and personalized content delivery. These systems are designed to balance the huge amount of content available with the individual preferences of users to maximize the interaction-utility ratio of the users.

Among the various paradigms in recommendation systems, collaborative filtering (CF) stands out as a widely adopted approach known for its effectiveness in delivering valuable and personalized recommendations to users [17]. By leveraging the collective wisdom of users' preferences and behaviors, collaborative filtering recommends items based on the similarity of users' tastes and interactions. Despite its practical success, much of the knowledge surrounding collaborative filtering remains largely empirical, leaving a gap in our comprehensive understanding of the underlying characteristics of the filtering problem and the intricacies of this specific approach.

Unraveling the inner workings of collaborative filtering is a major challenge due to its inherent complexity. The interactions between users and items within a recommendation system generate large and intricate datasets, making extracting meaningful patterns and underlying mechanisms difficult. To address these challenges, researchers are increasingly turning to interdisciplinary approaches that combine insights from data science, machine learning, and the social sciences [4]. By integrating theories and methods from these diverse fields, they aim to gain a more holistic understanding of how users' social interactions, psychology, and preferences influence the collaborative filtering process.

In the context of collaborative filtering, data predictability is a critical aspect of understanding data complexity. Understanding the predictability of data in collaborative filtering recommender systems can provide valuable insight into the effectiveness of the algorithms. A low predictability score indicates the presence of highly complex and unpredictable user-item interactions, which could affect the accuracy and reliability of the recommendations generated. On the other hand, a high predictability score indicates that the system is effectively capturing the underlying patterns and user preferences, resulting in more reliable and personalized recommendations. While predictability has been extensively studied in various domains [1, 6, 12], its application to collaborative filtering recommender systems remains unexplored.

Our goal is to provide a data-driven metric to quantify the predictability of any instance of a recommender system (RS), defined as the possible maximum precision of a prediction algorithm. Quantifying the predictability of RS allows us to evaluate predictive algorithms; estimate the extent to which the structure of the RS is explicable; and monitor changes during the system's evolution.

This article proposes two strategies to measure predictability in collaborative filtering datasets. Each metric is comprehensively defined and contrasted with the accuracy of well-known CF algorithms to demonstrate its relationship to the predictability of the


*Conference acronym 'XX, June 03–05, 2018, Woodstock, NY*

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

data. To validate their efficacy, we conducted tests using both artificial and real-world datasets, revealing that the metrics have a high correlation with the accuracy of the best-performing algorithms.

The paper is structured as follows: In the Metrics section, we present and justify the two metrics studied in this paper. The Methodology section elaborates on the testing procedures employed to assess these methods using real datasets. The Results section showcases the correlations between the metrics and prediction accuracy of CF algorithms. Finally, in the Discussion section, the conclusions, limitations observed during the tests, and perspectives for future work are discussed.

## 2 RELATED WORK

To the best of our knowledge, this is the first study to explore predictability and complexity measures in the context of RS. Yet, prediction in RS can be seen as a specific case of link prediction in graphs. Our metrics borrow ideas from previous work studying structural properties in complex networks to reveal their intrinsic predictability.

### 2.1 Predictability in complex networks

The work on network link prediction by [18] presents an entropy-based metric for predictability in the context of network graphs. They introduce a measure of structural entropy using a lossless compression algorithm by [5] that converts the network structure into a binary string, whose complexity can be computed using Shannon entropy. Their results using several real networks show a linear relationship between the entropy measure and the performance of the best link prediction algorithm for each dataset.

In a similar context, [12] presents another predictability metric for network graphs. This method takes advantage of the representation of the graph as an adjacency matrix, which is symmetric and binary. Assuming that structural consistency for networks under small perturbations can be approximated by an eigenvalue change of the adjacency matrix, they compute the eigendecomposition of a perturbed matrix (where the matrix is the same as the original but with a fraction of links missing) and estimate the structural differences between the original matrix and the current one with an eigenvalue perturbation.

Both of these recent studies cannot be directly applied to RS, which are represented by user-item rating matrices that are non-square, and weighted. Consequently, any method that aims to assess the structural complexity of recommender systems must inherently account for these properties. These methods are effective in scenarios where binary values dominate the analysis, a situation characterized by unweighted graph edges where the representation matrix is also symmetric. In this work, we aim to measure these properties on non-symmetric, non-square, non-binary matrices, which is the general case for collaborative filtering RS representations.

### 2.2 Collaborative Filtering Algorithms

Typically, collaborative filtering algorithms are based on traditional machine learning methods such as decision trees, rule-based methods, Bayes classifiers, regression models, support vector machines, latent factor models, and neural networks [2]. Latent factor collaborative filtering is a popular technique for making personalized recommendations that are based on the idea of representing users and items as vectors in a latent space of lower dimensionality [10]. These latent factors are not directly observable but can be inferred from the ratings or interactions between users and items. There are many methods to learn the latent structure, such as non-negative matrix factorization [11], support vector machines [21], probabilistic matrix factorization [20], and more recently, deep neural networks [8].

### 2.3 Correlation between prediction performance and dataset meta-features

In the context of RS predictability, a paper that has made a significant contribution is that of McElfresh et al. [13]. In it, the authors present a machine learning method for selecting the best-performing RS prediction algorithm for a given dataset, based on various meta-characteristics of the dataset. Such meta-features include, but are not limited to: overall mean ratings, Shannon's entropy, mean rating distribution, Gini coefficient, among others. They test the algorithmic performance of algorithms belonging to five families: SlopeOne, Co-Clustering, Matrix Factorization, RP3Beta, and User KNN. For each of the aforementioned families, they determine which of these meta-features best correlates with its performance; the correlation results are based on plotting against 85 different datasets. Using these results, when given a new dataset, the algorithm calculates its meta-features and provides the algorithm that best correlates with the feature that obtained the highest value.

Our work is closely related to the context of McElfresh et al's work. However, the goal of our work is different: while the use of predictability in their study relates to the performance of specific algorithms, our goal is to establish a relationship between a metric and the best possible algorithm. Furthermore, our methods aim to provide a principled metric that captures the structure of an RS dataset, so that the causal relationship between the value of the metric and optimal performance is robust to new datasets.

## 3 METRICS

For a recommender system, the main goal is to predict which items a user is likely to be interested in based on their past interactions with the system.Formally, given a set of users $\mathcal{U}$ and a set of items $\mathcal{I}$, a CF recommender system can be represented as a matrix $M$ of size $|\mathcal{U}| \times |\mathcal{I}|$, where rows represent users, columns represent items, and cells contain the ratings given by the interaction of the user in the row with the item in the column. For most real-world datasets, this model leaves the vast majority of cells empty, since most users interact with only a handful of items. Our goal is to derive a complexity measure for the matrix $M$ that takes into account the limited information about the interactions.

In this study, we introduce predictability metrics that aim to rank recommender system datasets based on their degree of predictability, ranging from easier to harder to predict. To achieve this, we propose two main strategies that are based on the Singular Value Decomposition (SVD) of a matrix.

To avoid potential confusion, we make a slight distinction between two applications of SVD in our work. The first is the exact singular value decomposition of a matrix, which we will refer to as

SVD. The second is the (truncated) approximate factorization of a matrix done by least squares optimization, commonly used in CF recommendation algorithms; we will refer to it as TSVD henceforth.

## 3.1 Analytical Structural Consistency (ASC)

Our approach is inspired by the notion of structural consistency in graphs [12], which essentially performs a prediction from a perturbed version of the training set using eigendecomposition. Since eigendecomposition works only on square matrices, we adapt the notion of structural consistency to work for RS non-square matrices using singular value decomposition (SVD).

In the context of RS, structural consistency is based on the hypothesis that if the corresponding bipartite graph is perturbed by a permutation of its edges, and this perturbation does not significantly alter its structural properties, then the graph is highly predictable. Conversely, if the perturbation leads to substantial changes in the internal structure, the original data is less predictable.

We explain how we adapt the method provided by Lü et al [12] as follows. First, we describe their consistency method for adjacency matrices using matrix notation. Next, we describe our adaptation and the technical differences that must be taken into account for it to work.

*3.1.1 Adjacency matrix perturbation by diagonalization.* The method in Lü et al. [12] takes an adjacency matrix $A$ and removes a fraction $p$ of its links, represented by the matrix $\Delta A$. The resulting matrix with removed links, $A^R = A - \Delta A$, can be diagonalized since it is symmetric. Its diagonalization can be written as

$$A^R = XKX^T \tag{1}$$

where $X$ is an orthogonal matrix that has $A^R$'s normalized eigenvalues as columns, and $K$ is the eigenvalue matrix, with its columns sorted to keep consistency with $X$. They define $\tilde{A}$ as a structural perturbation of $A$, with the following formula:

$$\tilde{A} = A^R + X\Delta KX^T = X(K + \Delta K)X^T \tag{2}$$

where $\Delta K$ is the difference between $A$'s and $A^R$'s eigenvalue matrices, and is approximated using

$$\mathrm{Diag}(\Delta K) \approx \mathrm{Diag}(X^T \Delta A X) \tag{3}$$

Finally, the matrix $\tilde{A}$ is used as a prediction for $A$. The obtained root mean square error (RMSE) result from testing the prediction is used as the predictability metric.

*3.1.2 Structural perturbation by SVD.* Next, we explain our adaptation of the method to define structural perturbations for the case of SVD.

Let $M$ be a rating matrix, where rows represent users, columns represent items, and cells represent known ratings of the corresponding users and items (and 0's in the unknown ratings). We define the following matrices in an analogous manner to what was done with the adjacency matrix $A$.

Consider a matrix $M^P$ that results from permuting a fraction $p$ ($0 < p < 1$) of the known ratings present in matrix $M$. Thus, the original matrix $M$ can be expressed as

$$M = M^P + \Delta M \tag{4}$$

for an appropriate $\Delta M$. The SVD of $M^P$ is written as follows:

$$M^P = U\Sigma V^T. \tag{5}$$

We also define the matrices $\Delta U$, $\Delta V$, and $\Delta\Sigma$ such that

$$M = (U + \Delta U)(\Sigma + \Delta\Sigma)(V + \Delta V)^T \tag{6}$$

is a singular value decomposition of $M$. In this sense, $\Delta U$, $\Delta\Sigma$ and $\Delta V$ denote the difference between the SVD matrices present for $M$ and $M^P$.

We define $\tilde{M}$ as a structural perturbation for $M$. That is, we make an approximation of the matrix $M$, denoted as $\tilde{M}$ and defined as follows.

$$\tilde{M} = M^P + U\Delta\Sigma V^T = U(\Sigma + \Delta\Sigma)V^T \tag{7}$$

Note that $\tilde{M}$ is similar to $M$ in its definition, but the former does not take into account $\Delta U$ nor $\Delta V$. The objective is to find a value or approximation for $\Delta\Sigma$, since the rest of the values can be obtained from $M^P$ and its SVD. We cannot obtain the same approximation as to what is done for $\Delta K$, because the latter works thanks to the original matrix being symmetric. What can be done is to take advantage of the symmetry of the product between a matrix and its transpose. Considering that, we multiply Equation (5) by its transposed version to the left to obtain

$$(M^P)^T M^P = V\Sigma^T U^T U\Sigma V^T = V(\Sigma^T\Sigma)V^T. \tag{8}$$

Where we used $U^T U = I$, since $U$ is orthogonal. Noticing that $V$ is also orthogonal and that $\Sigma^T\Sigma$ is diagonal and square, the result to the right is a diagonalization of the matrix $(M^P)^T M^P$. Also, because of $\Sigma$ being diagonal, the first $\mathrm{Dim}(\mathrm{Diag}(\Sigma))$ values of $\Sigma^T\Sigma$ are equal to the squares of the values present in $\mathrm{Diag}(\Sigma)$ (the following, if any, are equal to 0). The reasoning described in this paragraph is commonly used to obtain an exact SVD for a matrix.

Let $\mathcal{M} := (M^P)^T M^P$. We follow analogous steps as done with matrix $A$ to reach the parallel to Equation (3):

$$\mathrm{Diag}(\Delta(\Sigma^T\Sigma)) \approx \mathrm{Diag}(V^T \Delta\mathcal{M} V) \tag{9}$$

where

$$\Delta(\Sigma^T\Sigma) = (\Sigma + \Delta\Sigma)^T(\Sigma + \Delta\Sigma) - \Sigma^T\Sigma \tag{10}$$

and

$$\Delta\mathcal{M} = M^T M - \mathcal{M} \tag{11}$$

In particular, the result obtained in Equation (9) can be used as an approximation in the point-to-point version of Equation (10), which can be written as follows:

$$\Delta(\Sigma^T\Sigma)_{ii} = (\Sigma_{ii} + \Delta\Sigma_{ii})^2 - \Sigma_{ii}^2 \tag{12}$$

where $i$ is an index between 1 and $\mathrm{Dim}(\mathrm{Diag}(\Sigma))$. This equation holds because both $\Sigma$ and $\Delta\Sigma$ are diagonal. Clearing the value for $\Delta\Sigma_{ii}$, we obtain

$$\Delta\Sigma_{ii} \approx \sqrt{(\Sigma^T\Sigma)_{ii} + \Delta(\Sigma^T\Sigma)_{ii}} - \Sigma_{ii} \tag{13}$$

where $\Delta(\Sigma^T\Sigma)$ is the diagonal matrix representing the eigenvalue perturbation for matrix $(M^P)^T M^P$.

Note that obtaining $\Delta\Sigma$ can also be done in terms of the matrix $M^P(M^P)^T$. The decision as to which is used in practice is made by considering algorithmic efficiency.

Once the matrix $\tilde{M}$ is obtained, we use it as a prediction for $M$. The root mean square error (RMSE) result obtained by only testing the values present on the perturbed cells is used as the predictability metric.

### 3.2 Empirical Structural Consistency (ESC)

This metric is a simpler version of the ASC method, does not require a closed-form approximation, and is less computationally expensive. Roughly speaking, the structural perturbation here is taken as $\tilde{M} = M^P$, but we approximate its factorization using TSVD instead of its exact value. The steps involved are described below:

(1) We define a fraction $0 < p_T < 1$.
(2) Given a rating matrix $M$, we select a random set $R_M$ of its ratings. The defined set has $p_T$ of the total known cells of $M$.
(3) We define the matrix $M^P$, which is the result of randomly permuting $R_M$'s ratings on $M$.
(4) We train an unbiased TSVD recommendation algorithm with 20 hidden attributes, using the permuted matrix $M^P$ as the training set.
(5) We predict the ratings from $R_M$ using the trained algorithm from the previous step, and take the RMSE value.
(6) We iterate over the previous steps a fixed number of times $n$ and average the resulting RMSE values. This last result is used as the predictability metric.

## 4 METHODOLOGY

In this section, we put through empirical testing the predictability metrics. To this end, we conduct experiments on a representative sample of 100,000 rated interactions from each of the 12 real-world datasets under study (see Table 1 for a description). These experiments are designed to evaluate the relationships between the predictability metric and the accuracy of the best-performing CF predictive algorithms. The algorithms tested are the following:

- Non-negative Matrix Factorization (NMF), using 20 hidden attributes.
- Biased matrix factorization (TSVD), using 20 hidden attributes.
- TSVD with implicit ratings (SVD++).
- K-Nearest Neighbors (KNN), defining a maximum of 5 neighbors.
- KNN Means, defining a maximum of 5 neighbors.
- Slope-One algorithm.
- Co-clustering algorithm, using 5 clusters.

For each of the datasets, we train the algorithms on a random selection of 50% of the interactions and then test them on the remaining 50%. We compute the root mean square error (RMSE) of the predictions and normalize them by dividing the result by the rating range (e.g., if the scores range from 1 to 5, we divide the corresponding RMSE by 4). This normalization is done to compare two or more data sets that use different rating scales.

The best-performing algorithm with the lowest error is selected. Table 1 shows and highlights these minima. We then compute the predictability metrics for the data set; once a measure has been computed for all eight data sets, we compute the Pearson and Spearman correlation coefficients between the minimum RMSE and the metric. We do this for all of the predictability metrics.

To obtain the results using the ESC method, we consider $p_T = 0.1$ and $n = 10$. For the ASC method, we consider $p = 0.1$.

Additionally, we perform predictability measurements on manually generated datasets designed to represent extreme cases with distinct structural complexities. A common characteristic of all generated datasets is that the rating resolution is five rating values, and all possible ratings have the same frequency. Thus, simpler metrics like those used in McElfresh et al. [13] do not capture the differences between datasets because they rely on the distribution of ratings and not the structure.

A simplified illustration of the generated sets is shown in Figure 1. There, we have ordered the generated sets in qualitative order of structural complexity. In this sense, the set represented by figure 1a is the easiest to predict in theory, given that each user only gives one rating, while the one represented by figure 1e is arguably the hardest to predict, due to its generation's randomness. Each of the generated matrices is of size $500 \times 500$; after the generation, we extract a random 95% of the ratings, so that the studied data set is incomplete (i.e. its resulting sparsity is 95%).

## 5 RESULTS

We present the plot of the real data in Figure 2. The correlation coefficients are presented in table 2. The experiments with generated data are plotted in figure 3; their correlation coefficients are presented in table 3.

### 5.1 Outcome for the real datasets

The ASC method shows the highest correlation values in Table 2, and both are above 0.9. This empirical evidence supports the suitability of the method as a metric of predictability. Furthermore, it underscores the viability of perturbation-based structural analysis in recommender systems as a means of extracting meaningful insights about the underlying data.

The ESC method shows a high Pearson correlation and also a noticeable Spearman correlation. The fact that this method is highly correlated means that the predictability of a CF dataset depends on how much its structure is altered when a perturbation is applied to it.

### 5.2 The case of synthetic data

To test the robustness of the metrics, we have generated matrices that have the same global statistical properties but different structural properties. Specifically, we are interested in cases where the ratings have a fixed frequency distribution, but are ordered differently in the user-rating interaction matrix.

The results for the manually generated sets are shown in Figure 3. The correlation coefficients for each metric against the best algorithmic performance are shown in Table 3. The plot shows that only the ESC metric succeeds in correlating properly with the prediction error. Conversely, ASC shows roughly the same predictability value for different prediction errors.

Although ASC seems to contain enough complexity to capture the structure of the data set, the manually generated experiments show otherwise. One explanation for this could be that the approximation term $U\Delta\Sigma V^T$ in Equation (7) is either too small or overfitting, and therefore unable to capture a broader sense of structure in

| Dataset | # Ratings | # Users | # Items | NMF | TSVD | SVD++ | KNN | KNNMeans | SlopeOne | Co-Clustering |
|---|---|---|---|---|---|---|---|---|---|---|
| Amazon Arts, Crafts & Sewing | 100,000 | 91,134 | 1,129 | 0.307 | **0.286** | 0.290 | 0.303 | 0.305 | 0.306 | 0.306 |
| Amazon Books | 100,000 | 86,893 | 411 | 0.277 | **0.266** | 0.269 | 0.275 | 0.274 | 0.275 | 0.276 |
| Amazon CDs & Vinyl | 100,000 | 76,929 | 2,897 | 0.239 | **0.2216** | 0.2224 | 0.235 | 0.235 | 0.237 | 0.235 |
| Amazon Digital Music | 100,000 | 68,916 | 5,210 | 0.199 | 0.185 | **0.184** | 0.193 | 0.189 | 0.190 | 0.193 |
| Amazon Gift Cards | 100,000 | 92,029 | 502 | 0.230 | **0.223** | 0.225 | 0.227 | 0.226 | 0.226 | 0.236 |
| Amazon Industrial and Scientific | 100,000 | 93,815 | 1,192 | 0.301 | **0.287** | 0.289 | 0.298 | 0.300 | 0.300 | 0.300 |
| Amazon Magazine Subscriptions | 89,688 | 72,097 | 2,428 | 0.348 | **0.339** | 0.342 | 0.350 | 0.347 | 0.348 | 0.347 |
| Amazon Video Games | 100,000 | 71,998 | 2,531 | 0.327 | **0.300** | 0.303 | 0.322 | 0.328 | 0.332 | 0.324 |
| MovieLens | 100,000 | 669 | 3,264 | 0.248 | 0.235 | **0.233** | 0.258 | 0.253 | 0.238 | 0.246 |
| Netflix Prize 1 | 100,000 | 81,490 | 30 | 0.290 | **0.262** | 0.271 | 0.280 | 0.282 | 0.287 | 0.289 |
| Netflix Prize 2 | 100,000 | 73,461 | 23 | 0.283 | **0.255** | 0.260 | 0.269 | 0.272 | 0.277 | 0.283 |
| Steam Australian Reviews | 59,305 | 25,458 | 3,682 | 0.366 | 0.302 | **0.301** | 0.330 | 0.355 | 0.361 | 0.351 |

**Table 1: Dataset sample details and RMSE values obtained by implementing the four CF algorithms: Non-Negative Matrix Factorization (NMF), Singular Value Decomposition (TSVD), Singular Value Decomposition Plus-plus (SVD++), Simple K-Nearest Neighbors (KNN), K-Nearest Neighbors with Means (KNNMeans), Slope One, and Co-Clustering. Minimum RMSE values are highlighted in bold; these values are used as a reference to test the prediction measures shown in this paper. Amazon datasets are provided by [15]; MovieLens dataset by [7]; Netflix datasets are obtained from [14]; and Steam dataset by [9, 16, 19].**

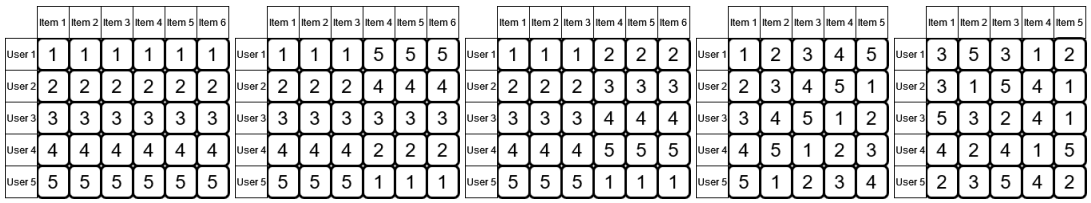


**Figure 1: Graphical examples of the generated cases studied for the predictability metrics. The five dataset types have the same numerical rating distribution, but they vary in the relative distribution among agents. Figure 1a has all users rating items with one value; figure 1b has users that rate at most two values, and items can be set into two groups according to the ratings given to them; figure 1c has users rating using consecutive ratings (5 is taken as consecutive to 1); figure 1d does something similar, but with a wider variety of users; and figure 1e has all ratings randomly and independently assigned. Intuitively, a suitable measurement for structural complexity $c$ should satisfy $c(1a) < c(1b) \approx c(1c) < c(1d) < c(1e)$. The matrices used to obtain the results in figure 3 follow this basic structure, but are of size $500 \times 500$.**

the points being evaluated. Note that when $\Delta\Sigma = 0$, the computed value is a root mean square difference between a subset of values and a permutation of them. This operation is only dependent on rating frequencies and therefore does not take into account any underlying structure.

## 6 DISCUSSION

### 6.1 Conclusions and Implications

In this article, we have presented two complexity metrics that prove valuable in determining the predictability of RS datasets. We have addressed the importance of considering the intrinsic structural properties of the data as a proxy for predictability. Their usefulness lies in a data-driven understanding of the limitations of CF recommendation algorithms, as well as having numerical values to contrast when training an algorithm. Of the two metrics analyzed, ESC is the most promising for general use cases, as its correlation coefficient against prediction errors is high for both real and generated data, thus demonstrating high robustness in measuring predictability.

We believe that our method is novel in three respects. First, the application of a principled measure of predictability in recommendation systems is novel. Second, from a technical point of view, the extension of structural consistency [12] to non-binary and non-symmetric matrices involved non-trivial analytical calculations to compute the approximation of the perturbed matrix using SVD decomposition (instead of eigenvalue decomposition); Third, we extended the perturbation method from removing links to changing the weights of the links; and we proposed a new empirical method based on structural consistency that is computationally more efficient and doesn't rely on first-order approximations.

Furthermore, we are confident that our proposed complexity metrics could be integrated into existing RS to improve their performance. First, the metrics can be used to evaluate the performance of an algorithm by comparing it to the proposed metrics, which quantify the intrinsic optimal performance that any algorithm could achieve. Second, the metrics can be used to capture changes as the system evolves. Previous work measuring the detrimental effects of feedback loops uses metrics such as homogenization and popularity bias. We argue that our proposed metric based on structural

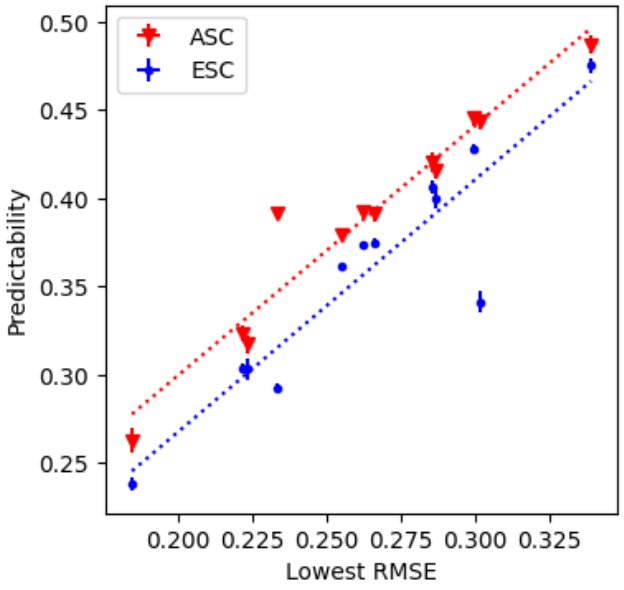


**Figure 2: Graphical correlation for real datasets between predictability metrics and the lowest RMSE from the tested recommendation techniques. Each point is obtained by iterating 10 times over each metric and algorithm and taking its mean; the error bars are taken from the standard deviation of the aforementioned iterations on each axis. Both metrics show a strong linear correlation with algorithmic performance, although ESC presents problems with one dataset (Steam reviews).**

| Metric | Pearson | Spearman |
|---|---|---|
| ASC | **0.968** | 0.821 |
| ESC | 0.924 | **0.825** |

**Table 2: Correlation coefficients using real data, for each of the predictability metrics, against the minimum RMSE values obtained by the recommendation algorithms. The top correlations are highlighted in bold.**

consistency captures both metrics because homogenization and popularity bias reduce the complexity of the system (making it more predictable), and can provide new insights into the effects of feedback loops [3].

## 6.2 Limitations and Future Work

The experimental results on real data show a significant correlation for the ASC method. This suggests that structural complexity can be both analyzed via perturbations and used as a strategy to measure predictability. The main problems identified are technical, mainly: the possibility of a computation in which the factorization values may be non-real numbers (see that Equation (13) contains a square root of an approximation); and the computational complexity required to compute large datasets (identified as $O(\max(\{m,n\})^3)$ for an $m \times n$ matrix), which makes it difficult to scale. Future work will address the cases where these problems are present.

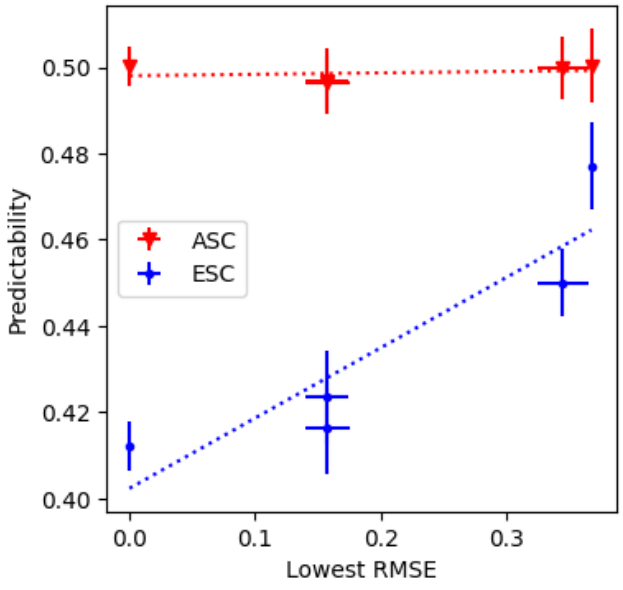


**Figure 3: Graphical correlation for generated data between predictability measurements and the lowest RMSE from recommendation techniques, taking into account 5% of their ratings as known. Each point is obtained by iterating 10 times over each metric and algorithm and taking its mean; the error bars are taken from the standard deviation of the aforementioned iterations on each axis. Only the ESC metric has an identifiable correlation with the prediction errors for these cases.**

| Metric | Pearson | Spearman |
|---|---|---|
| ASC | 0.257 | 0.300 |
| ESC | **0.905** | **0.900** |

**Table 3: Pearson and Spearman correlation coefficients for generated data between the prediction error and the studied predictability metrics, when contrasted against generated data. Only ESC shows a significant correlation.**

As the only metric that correlates with the error of the generated cases, the ESC method seems most likely to serve as a viable strategy for measuring predictability. Its results raise the question of the reason for its robustness. We formulate the hypothesis that the perturbation emphasizes the structural noise present in the data set; consequently, the approximation performed by the TSVD algorithm further emphasizes the structural noise. This hypothesis will be addressed in future work.

## REFERENCES

[1] Andrés Abeliuk, Zhishen Huang, Emilio Ferrara, and Kristina Lerman. 2020. Predictability limit of partially observed systems. *Scientific Reports* 10 (2020), 2045–2322. Issue 1.

[2] Charu C Aggarwal et al. 2016. *Recommender systems*. Vol. 1. Springer.

[3] Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhardt. 2018. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM conference on recommender systems*. 224–232.

[4] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), 1–39.

[5] Yongwook Choi and Wojciech Szpankowski. 2012. Compression of graphical structures: Fundamental limits, algorithms, and experiments. *IEEE Transactions on Information Theory* 58, 2 (2012), 620–638.

[6] Joshua Garland, Ryan James, and Elizabeth Bradley. 2014. Model-free quantification of time-series predictability. *Physical Review E* 90, 5 (2014), 052910.

[7] F. Maxwell Harper and Joseph A. Donstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, 4 (2015), 19 pages.

[8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web* (Perth, Australia) *(WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. https://doi.org/10.1145/3038912.3052569

[9] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. *ICDM* (2018).

[10] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[11] Daniel Lee and H Sebastian Seung. 2000. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems* 13 (2000).

[12] Linyuan Lü, Liming Pan, Tao Zhou, Yi-Cheng Zhang, and H Eugene Stanley. 2015. Toward link predictability of complex networks. *Proceedings of the National Academy of Sciences* 112, 8 (2015), 2325–2330.

[13] Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, John P. Dickerson, and Colin White. 2022. On the Generalizability and Predictability of Recommender Systems. *arXiv preprint arXiv:2206.11886* (2022).

[14] Netflix. 2019. Netflix Prize Dataset. https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data. Accessed: 2023-07-02.

[15] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fined-grained aspects. *Empirical Methods in Natural Language Processing (EMNLP)* (2019).

[16] Apurva Pathak, Kshitiz Gupta, and Julian McAuley. 2017. Generating and personalizing bundle recommendations on Steam. *SIGIR* (2017).

[17] Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)* 47, 1 (2014), 1–45.

[18] Jiachen Sun, Ling Feng, Jiarong Xie, Xiao Ma, Dashun Wang, and Yanqing Hu. 2020. Revealing the predictability of intrinsic structure in complex networks. *Nature communications* 11, 1 (2020), 574.

[19] Mengting Wan and Julian McAuley. 2018. Item recommendation on monotonic behavior chains. *RecSys* (2018).

[20] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 448–456.

[21] Zhonghang Xia, Yulin Dong, and Guangming Xing. 2006. Support vector machines for collaborative filtering. In *Proceedings of the 44th annual Southeast regional conference*. 169–174.