Journal of Information & Knowledge Management
Vol. 21, No. 3 (2022) 2250051 (28 pages)
© World Scientific Publishing Co.
DOI: 10.1142/S0219649222500514

Knowledge Discovery in a Recommender System: The Matrix Factorization Approach

Murchhana Tripathy*, Santilata Champati†
and Hemanta Kumar Bhuyan‡
Information Systems & Technology
T A Pai Management Institute
Manipal Academy of Higher Education
Manipal, India

Department of Mathematics, ITER
Siksha 'O' Anusandhan Deemed to be University, Odisha, India

Vignan's Foundation for Science
Technology & Research (Deemed to be University)
Guntur, AP, India
*murchhanatripathy@gmail.com
†santilatachampati@soa.ac.in
‡hmb.bhuyan@gmail.com

Published 22 June 2022

Abstract. Two famous matrix factorization techniques, the Singular Value Decomposition (SVD) and the Nonnegative Matrix Factorization (NMF), are popularly used by recommender system applications. Recommender system data matrices have many missing entries, and to make them suitable for factorization, the missing entries need to be filled. For matrix completion, we use mean, median and mode as three different cases of imputation. The natural clusters produced after factorization are used to formulate simple out-of-sample extension algorithms and methods to generate recommendation for a new user. Two cluster evaluation measures, Normalized Mutual Information (NMI) and Purity are used to evaluate the quality of clusters.

Keywords: Recommender system; knowledge discovery; Singular Value Decomposition (SVD); Nonnegative Matrix Factorization (NMF); NMI; purity; imputation; similarity measures; vector projection.

1. Introduction

Matrix factorization is a mathematical technique (Hubert *et al.*, 2000) in which a matrix can be expressed as a product of several smaller and simpler matrices. Also popularly known as matrix decomposition, they have their roots in linear algebra but have a huge application in data mining and machine learning. In data mining applications, datasets are expressed as matrices in which rows represent objects or cases, or subjects and columns represent variables or attributes. Each entry in the matrix representation of data typically describes the value of the feature for the object. Given this scenario, matrix factorization computes a lower rank approximation of a large data matrix to understand the structure of the data, the relationship between objects, the relationship between attributes, the influence of the attributes on the behaviour of the objects, and the most significant attributes for data analysis (Hubert *et al.*, 2000). In a broader sense, matrix factorization techniques help to understand and interpret the data. The other direct benefit of lower rank approximation using matrix factorization techniques is dimensionality reduction which allows representing data compactly with minimal loss of information in the original data matrix (Hubert *et al.*, 2000). Recommender systems are software systems that help in giving recommendations about a product or a thing to an interested user by using knowledge in the existing data. They involve large datasets of millions of rows and thousands of columns. They typically have been used to recommend movies, music, television programs, hotels, items to consumers in an e-commerce application, etc. to name a few (Sarwar *et al.*, 2000; Martinez *et al.*, 2009; Benzi *et al.*, 2016; Ebadi and Krzyzak, 2016; Weerasinghe and Rupasingha, 2021). Singular Value Decomposition (SVD) and Nonnegative Matrix Factorization (NMF) are two matrix-based techniques that play a significant role in recommender systems (Lee and Seung, 1999; Strang, 2006). They belong to the category of unsupervised matrix decomposition techniques (Ch *et al.*, 2015).

Being one of the popular machine learning applications in the digital age, recommender systems use SVD for dimensionality reduction in many applications (Sarwar *et al.*, 2000; Martinez *et al.*, 2009; Vozalis *et al.*, 2009; Zhou *et al.*, 2015). SVD has been used for movie recommendation (Vozalis and Margaritis, 2007; Kurucz *et al.*, 2007; Nilashi *et al.* 2018), TV program recommendation (Martinez *et al.*, 2009), hotel recommendation (Ahani *et al.*, 2019), web service recommendation (Weerasinghe and Rupasingha, 2021) and item recommendation in e-commerce applications (Sarwar *et al.*, 2002), etc. to name a few. SVD is helpful in finding a lower-dimensional feature space for a dataset in which information present in the dataset is captured with the help of a fewer number of dimensions. SVD of matrix A decomposes it into three matrices, the left singular vector matrix U, the right singular vector matrix V and a diagonal matrix $\sum$ containing the singular values. The singular values are arranged in descending order along the diagonal, and are always positive. If we decompose a $User \times Item$ matrix, where rows represent users and columns represent items, then the left singular vector matrix is the user-to-concept matrix, and the right singular vector matrix is the concept-to-item matrix. The singular values in the middle matrix represent the strength of each concept. Using SVD, a data matrix can be approximated only with the significant singular values, and the smaller singular values can be discarded. Since singular values are the square root of the eigenvalues of the data matrix and eigenvalues capture the characteristics of a data matrix, singular values also capture the characteristics of the data matrix. The significant singular values capture about 95% of the information in the dataset. That is why only the significant singular values are retained, and the smaller ones are discarded.

NMF is another MF technique that is mainly used for its clustering capability in data mining and machine learning applications (Kumar, 2009; Li and Ding, 2018). NMF has been used for movie recommendation (Zhang *et al.*, 2006), song recommendation (Benzi *et al.*, 2016), hotel recommendation (Ebadi and Krzyzak, 2016), on-line course recommendation (Campos *et al.*, 2020), drug repositioning (Sadeghi *et al.*, 2021), music recommendation (Su *et al.*, 2017), etc. Given a matrix A, NMF decomposes it into two matrices, W and X. The constraint on which NMF works is: A, W, and X are all nonnegative. The entries of a nonnegative matrix can be either zero or a positive quantity. Due to the nonnegative property, it is easy to interpret the results of the decomposition. For example, in a facial recognition application, the matrix W contains the basis images such as eyes, nose, ears, lips, etc., and H indicates the importance of each of the basis images. In a recommended system application, NMF is used for clustering. After decomposition of a rating matrix A, the matrix W gives the clustering of the users, and the matrix H gives the clustering of the items. The clustering remains the same in each run of NMF on a dataset, but values in W and H keep changing. Thus, W and H behave as two simple factored matrices that produce the original matrix upon multiplication.

The article identifies and compares the capabilities of SVD and NMF in discovering knowledge in a user-based collaborative filtering recommender system in the form of missing data imputation, clustering, and out-of-sample extension. For missing value imputation, we use the three measures of central tendencies, mean, median, and mode along with the SVD method. We evaluate the quality of clusters using two external evaluation measures Normalized Mutual Information (NMI), and purity. Further, for out-of-sample extension, we use distance-based similarity method and cosine similarity along with vector projection and formulate new algorithms.

The rest of the paper is given the following structure. Section 2 gives a background of SVD, NMF, and Recommender Systems. The literature review is presented in Sec. 3. Section 4 discusses missing data imputation in a recommender system. The use of SVD and NMF for knowledge discovery in a recommender system has been discussed in Secs. 5 and 6, respectively. Cluster quality evaluation using NMI and purity is presented in Sec. 7. Section 8 captures the contribution of our work. The paper is concluded in Sec. 9.

2. Background

In this section, we present a brief introduction of SVD (Strang, 2006) and NMF (Lee and Seung, 1999). Along with that, we also give an overview of recommender systems.

2.1. *Singular Value Decomposition (SVD)*

Given any $m \times n$ matrix A, SVD decomposes it into the product of three matrices such that

$$A = U \sum V^T, \qquad (1)$$

where U and V both are orthogonal matrices of dimension $m \times m$ and $n \times n$, respectively. The columns of U are the eigenvectors of AA^T, and that of V are the eigenvectors of A^TA. $\sum$ is an $m \times n$ diagonal matrix where entries in the diagonal represent the singular values. They are the square roots of the common, nonzero, distinct eigenvalues of AA^T and A^TA in descending order. If the $m \times n$ matrix has rank r, then the first r places will contain the singular values, and the rest of the entries of $\sum$ will be zero. SVD has a long history of theoretical development and has been independently studied by several mathematicians such as Beltrami, Jordan, Sylvester, Schmidt, and Weyl (Stewart, 1998).

2.2. *Nonnegative Matrix Factorization (NMF)*

Given any $m \times n$ nonnegative matrix A and a reduced rank k, where $k \leq \min(m, n)$, NMF factorizes A into two nonnegative matrices $W \in R^{m\times k}$ and $X = B^T \in R^{k\times n}$ such that

$$A = WX + E = WB^T + E, \tag{2}$$

where W is known as the matrix of feature vectors or basis vectors, and B is known as the encoding matrix or the component matrix. E represents the approximation error and $E \in R^{m\times n}$.

First investigated by Lee and Seung (1999), NMF has become popular for performing decomposition of non-negative datasets where the attributes have either zero or positive values and their corresponding hidden components are meaningful only when their values are nonnegative. W and X in Eq. (2) are solved by using the optimization problem given by

$$\min_{A,X} f_k(W, X) = \frac{1}{2}\|A - WX\|_F^2. \tag{3}$$

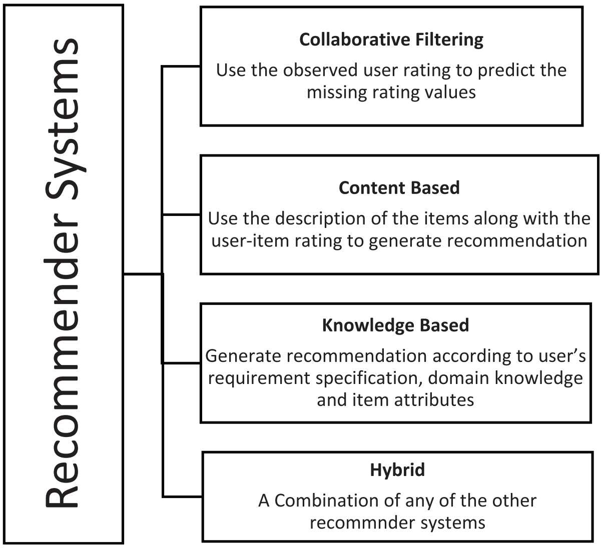
Equation (3) computes the difference between the given data matrix A and the estimated model WX using the Frobenius norm.

2.3. *Recommender systems*

With the advancement of computer hardware, software, storage, internet, and mobile technology, many new technological systems have been developed, and recommender systems are one among them. Recommender systems are a class of information filtering systems used to suggest products or services to users (Ricci *et al.*, 2011). The recommendations generated can be personalized or non-personalized (Ricci *et al.*, 2011). A personalized recommendation is generated in the form of a ranked list of items based on the user's past preference data, whereas a non-personalized recommendation gives information about the popular choices in a particular category of items. The proliferation of e-commerce and m-commerce has overwhelmed the users with a wide range of products and services, and recommender systems act as a guide to choosing the best options available for the users.

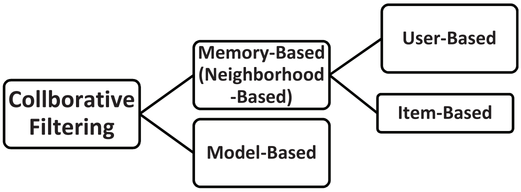
Recommender systems have proved to be very useful in coping with today's information overload problem. Especially in this pandemic situation, where people rely mostly on online purchases, recommender systems seem to be a very useful tool. They not only help users to buy products of their interest but also add business value to a company. There are several reasons for which many companies adopt this technology and build their own recommendation engine. Some of the benefits of having a recommender system in place are increasing the sales volume, selling diversified products, increasing customer satisfaction, understanding customer's purchase behaviour, targeted marketing, decreasing customer churn rate, increasing conversion rate that is, the number of customers that purchase the item in comparison to the number of simple visits to the website (Ricci *et al.*, 2011). Recommender systems are used in many industries such as entertainment, e-commerce, content recommendation, such as generating suggestions for news items, research articles, etc., and services such as hotel or restaurant recommendation, matchmaking services, etc., to name a few. Some popular recommender system applications are Netflix's movie recommendation, Google's new personalized system, Facebook's friends' suggestion, Youtube's video recommendation, Amazon's product recommendation, Trivago's hotel recommendation, etc. Recommender systems are broadly classified into four categories (Aggarwal, 2016). Figure 1 represents a classification of the recommender systems.

For this work, CF-based recommender systems have been considered, and suitable examples have been constructed in that line. Figure 2 gives a classification of the Collaborative Filtering recommender systems.


Recommender Systems
Collaborative Filtering
Use the observed user rating to predict the missing rating values
Content Based
Use the description of the items along with the user-item rating to generate recommendation
Knowledge Based
Generate recommendation according to user's requirement specification, domain knowledge and item attributes
Hybrid
A Combination of any of the other recommnder systems

(Ref.: Aggarwal (2016). Recommender systems: The Textbook. Cham: Springer International Publishing.)

Fig. 1. Classification of recommender systems.



(Ref.: Aggarwal (2016). Recommender systems: The Textbook. Cham: Springer International Publishing.)

Fig. 2. Classification of collaborative filtering recommender systems.

In user-based CF, ratings provided by similar users to that of the target user are used to predict the missing rating values, whereas in item-based CF, ratings given to a set of similar items to that of the target item by a specific user is used to predict the missing values of the target item. In a model-based method, a data mining algorithm is used to build a model and generate a recommendation.

3. Literature Review

This section captures notable contributions about the application of SVD and NMF in connection to recommender systems.

3.1. *SVD for recommender systems*

Koren *et al.* (2009) discussed matrix factorization techniques within the collaborative filtering framework as an alternative to user or item neighbourhood computation, which can improve the accuracy and efficiency of a recommender system. Vozalis and Margaritis (2007) applied SVD on demographic data from the MovieLens database in both user-based and item-based collaborative filtering settings and showed that the item-based CF method with SVD outperformed the generalized CF method and the user-based CF method. Martinez *et al.* (2009) developed a web-based television program recommendation system in which they used SVD as a dimensionality reduction technique. Sarwar *et al.* (2000) used SVD for dimensionality reduction and unknown rating prediction. Vozalis *et al.* (2009) used SVD as a dimensionality reduction technique and performed item-based filtering. Sarwar *et al.* (2002) proposed incremental SVD to resolve the scalability issue associated with recommender systems data. The idea of incremental SVD was further enhanced by Brand (2003) to construct an online SVD model where he used several rank-1 SVD rules for modifying the rows and columns without recomputing SVD each time. Kurucz *et al.* (2007) showed the application of SVD on the large-scale Netflix dataset for expectation maximization where they used zeros and item average for missing value imputation. Pan *et al.* (2010) proposed a transfer learning approach using SVD on heterogeneous datasets. Zhou *et al.* (2015) proposed an Incremental ApproSVD algorithm, which helped in the out-of-sample extension of a large number of new user ratings by adopting an incremental approach. Yuan *et al.* (2019) adapted the neighbourhood-based method and proposed an imputation-based SVD (ISVD) algorithm to improve the quality of SVD-based recommendations. Martinez *et al.* (2009) utilized SVD to reduce the dimensionality of the data and to implement item-based CF for TV program recommendations. Nilashi *et al.* (2018) developed a movie recommender system using SVD for dimensionality reduction and neighbourhood computation for both users and items. Ahani *et al.* (2019) developed a hybrid hotel recommender system for TripAdvisor in which SVD was used for dimensionality reduction and neighbourhood computation. Guo *et al.* (2019) proposed a method to solve the cold start problem in which SVD was used for dimensionality reduction. Lever *et al.* (2018) used the SVD-based CF method for knowledge discovery in biomedical data. Weerasinghe and Rupasingha (2021) proposed a web service recommendation approach using SVD.

3.2. *NMF for recommender systems*

Chen *et al.* (2009) applied Orthogonal NMF to cluster the users and the items of the $User \times Item$ data matrix simultaneously. Zhang *et al.* (2006) used two types of NMF, one based on the Expectation-Maximization (EM) algorithm and the other one the weighted NMF (WNMF) on MovieLens and Jester datasets, and found that the EM-based approach performed better than WNMF. Gu *et al.* (2010) proposed a collaborative filtering method by combining graph regularization techniques along with weighted NMF to produce a more accurate recommendation. Luo *et al.* (2014) developed a regularized single element based NMF(RSNMF) model for collaborative filtering in which they examined the nonnegative update process. Benzi *et al.* (2016) developed a song recommender system using an NMF-based collaborative filtering approach. Aghdam *et al.* (2015) used NMF with KL divergence to learn the hidden features of users and items in a recommender system and used them for missing value estimation. Ebadi and Krzyzak (2016) used NMF to factorize the $user \times hotel$ rating matrix in a hotel recommender system and provided recommendations about unseen hotels to the users. Bao *et al.* (2014) used NMF to factorize a matrix containing word frequency in the user reviews of products and discovered the latent factors. Desarkar *et al.* (2012) used NMF on the preference relations of users on items to generate a personalized recommendation of $top - k$ items. Hernando *et al.* (2016) proposed a probabilistic NMF method to determine k user groups and generate a recommendation. Li *et al.* (2009) proposed a privacy-preserving-based NMF method for the data created by adding random numbers to the user rating and then generate a recommendation. Bobadilla *et al.* (2017) proposed a Bayesian NMF algorithm for recommender system data clustering that improved performance and accuracy. Swaminathan *et al.* (2019) developed a recommender system for antimicrobial resistance using NMF. Badami and Nasraoui (2021) used NMF on rating data to design a polarization-aware recommender system that recommended relevant items as well as items covering opposite views. Campos *et al.* (2020) proposed a recommender system for Massive online open courses (MOOC) to help students find suitable courses in the platform in which they used NMF for topic modelling. Sadeghi *et al.* (2021) studied drug repositioning by adapting an NMF-based recommender system approach on various datasets containing information about drugs and diseases and found that the NMF-based method was superior to other existing methods in the field. Altulyan *et al.* (2019) used NMF to build a recommender system for elderly people, especially people suffering from Alzheimer's disease, to generate recommendations about various routine activities. Zhang *et al.* (2021) developed an incremental NMF to generate recommendations for online learning, video tracking, etc. Su *et al.* (2017) proposed a music recommender system where they used NMF to reduce the computational cost and improve the quality of prediction.

3.3. *Research gap*

Review of the relevant literature shows that SVD has been used for dimensionality reduction (Sarwar *et al.*, 2000; Vozalis and Margaritis, 2007; Martinez *et al.*, 2009; Nilashi *et al.*, 2018; Ahani *et al.*, 2019; Guo *et al.*, 2019), missing value imputation (Kurucz *et al.*, 2007; Yuan *et al.*, 2019), solving the scalability issue (Sarwar *et al.*, 2000; Brand, 2003) and out-of-sample extension (Zhou *et al.*, 2015). Similarly, NMF has been used for clustering (Chen *et al.*, 2009; Hernando *et al.*, 2016; Bobadilla *et al.*, 2017) and recommendation generation (Zhang *et al.*, 2006; Li *et al.*, 2009; Gu *et al.*, 2010; Desarkar *et al.*, 2012; Luo *et al.*, 2014; Benzi *et al.*, 2016; Ebadi and Krzyzak, 2016; Su *et al.*, 2017; Swaminathan *et al.*, 2019; Altulyan *et al.*, 2019; Campos *et al.*, 2020; Badami and Nasraoui, 2021; Sadeghi *et al.*, 2021; Zhang *et al.*, 2021). As given above these are the activities typically carried out by SVD and NMF in a recommender system application. We observe that SVD and NMF have been used with many other techniques where they are used to achieve a key objective. For example, only dimensionality reduction or out-of-sample extension by SVD and clustering by NMF. In these applications, the other techniques are used to accomplish the related objectives. No doubt, using multiple efficient methods produce optimal output and the concept of ensemble methods (Aggarwal, 2016) in machine learning is the best example of it. However, using too many methods sometimes makes the application complicated, lacking in simplicity and understandability which are the characteristics of a good software. Additionally, if a code or model is too complicated, it can only be modified by its creator. It becomes difficult for future developers to work on it and make further changes or upgradation to it. Keeping in view these factors, through this work, we show how SVD and NMF can be used to fulfil multiple objectives in a recommender system. We explore the different aspects of knowledge discovery by these methods. NMF has been used for clustering in recommender system applications, but SVD has not been used for the same. By taking suitable examples, we have done clustering using SVD and NMF and have shown how these clusters can help in out-of-sample extension and thereby solve the cold-start problem of recommender systems (Aggarwal, 2016). Also, SVD has been used to solve the missing data problem of a recommender system dataset. In summary, our analysis shows that SVD can perform dimensionality reduction, missing value imputation, clustering and out-of-sample extension in recommender systems. Similarly, NMF can perform dimensionality reduction, clustering and out-of-sample extension. The processes are more autonomous in case of SVD in comparison to NMF. Because in SVD the input is only the dataset, whereas in case of NMF, the rank of the matrix needs to be specified along with the input dataset. Further, for the same reason there can be biases in the results produced by NMF.

4. Missing Data Imputation in Recommender Systems

Missing data is a genuine problem in almost all studies where massive data is involved, and Recommender Systems are no exception. Recommender systems mostly use rating data. Data is collected through surveys by designing some questionnaires either manually or electronically. Data is missing due to several factors such as when respondents fail to understand the questions, lose their interest in answering each question, refuse to participate in the survey, do not know the answer to the question, encounter an irrelevant question, etc. Matrix factorization methods need a data matrix without any missing entries. This study uses two matrix factorization methods, SVD and NMF and having a matrix without any missing entries is a basic need.

Missing data can be broadly classified into three types: Missing Completely At Random (MCAR), Missing At Random (MAR), and Missing Not At Random (MNAR) (Bennett, 2001; Graham, 2009; Kang, 2013). When data is MAR, missing values depend on the observed values in the dataset but not on the unobserved ones. In MCAR, missing values do not depend on the observed values, and in MNAR, missing values depend on the unobserved values only. In recommender system applications, it is assumed that missing data is of type MAR, i.e. it is believed that the data exhibit some pattern, and the missing data can be inferred from the existing observed values. Various methods of handling missing data are available in the data mining literature, and imputation is one of them (Bennett, 2001; Graham, 2009; Kang, 2013). It is the process of filling missing data items with some values and is one of the easiest methods. It is of two types, single imputation and multiple imputation. In single imputation, each missing data is replaced by a single value. Several single imputation methods are available in the literature and mean substitution is one (Bennett, 2001; Graham, 2009; Kang, 2013). The mean substitution method encourages us to think about median and mode as potential options to use for imputation, given that there exists a strong corelation between the type of data used and the measure of central tendency that can be used to best represent the data.

4.1. *Data type and the appropriate statistical measure for imputation*

There is a close relationship between the type of data and the statistical measure used with them, and this has got equal importance while choosing the imputation method. Data can be broadly classified into two types: Numerical data and Categorical data.

4.1.1. *Numerical data and the corresponding imputation techniques*

In the case of numerical data, if the data is normally distributed, then the best statistic to use is the mean, although the value of the mean, median, and mode coincide, and any one of them can be used. The reason for this is that the computation of mean involves all the values present in the dataset, and change in any one of them will certainly affect the mean. This cannot be the case for median unless the middlemost element itself is changed when an odd number of elements are present, or one of the elements of the two middle elements is changed when there is an even number of elements. Also, the mode is not changed with an arbitrary change in the dataset. If, however, some skewness exists in the data, then the median should be preferred over the mean because the mean is pulled towards the direction of the skew, and the median better justifies the job of being a measure of central tendency in the skewed dataset. Further, when outliers are present in the data, the median can be chosen over the mean to give less importance to the outliers. Because mean by its nature gives equal importance to each data point.

4.1.2. *Categorical data and the corresponding imputation techniques*

Categorical data is of two types: Nominal and Ordinal. Nominal data means the names or labels of some category. e.g. marital status, blood group, gender, etc. Nominal data may use whole numbers to represent the labels or types of objects. e.g. the jersey numbers of football players will take the numbers from 1 to 11. Since the numbers used in the case of nominal data only represent labels, they do not possess any mathematical properties of numbers. Thus, it is wrong to add two such numbers. When the addition is not defined, it is inappropriate to compute the mean. Also, it is wrong to compute the median for nominal data because there is no natural ordering among the data points. When there is no order, data cannot be arranged in ascending or descending order, and hence we cannot find the median of the data points. However, the mode is the only statistic that is meaningful here. Because mode represents the highest frequency of occurrence of an object in a dataset and for nominal data, due to the presence of labels, the mode can very well be computed.

Ordinal data is ordered categorical data where an ordering structure exists among the successive data items. e.g. Stages of lung cancer (1, 2, 3, 4), rating scale data to measure the quality of service (Poor, okay, good, very good, excellent), to measure the feeling of a person (Unhappy, Okay, happy, Very happy), to measure the pain of a patient who has undergone surgery (No Pain, mild pain, severe pain), college grading system (O, E, A, B, C), etc. For Ordinal data, it is known that a greater than or less than relation exists among the objects, but this relation cannot be quantified. For example, we can say that stage-2 cancer is more severe than stage-1, but we cannot claim that the severity is exactly twice. Similarly, we cannot claim that the difference between stage-1 and stage-2 cancer is equivalent to the difference between stage-3 and stage-4. Just like, nominal scale, the numbers used here do not possess any mathematical properties. So, it is wrong to treat these labels as numbers and perform mathematical operations such as addition or subtraction. Since the addition operation is not defined here, the calculation of the mean is not meaningful. However, the median is a meaningful statistic for ordinal data because the greater than and the less than relationships are known, and data can be arranged either in ascending order or descending order. For Ordinal data, the mode is also a meaningful statistic because the frequency of items involved can be counted, and it makes sense. For some cases of ordinal data, the use of the mode is more meaningful than the use of the median as an imputation technique. Recommender Systems is one such case, and in the following section, we discuss the reason behind it. The method of imputing missing values using mode is named *mode-substitution.*

4.2. *Mode-substitution as a single-imputation method for recommender systems*

The data matrix involved in recommender systems applications and especially in a collaborative filtering setting is filled with rating values, the range of which normally varies between 1 to 5 or 1 to 10, and this is ordinal data or ordered categorical data. For this type of data, the mode should be preferred over the median due to the following reasons:

(1) Data in the data matrix varies in a fixed range, and that's why there is no chance of outliers being present in the data. So, no need to calculate the median.
(2) Some items may get equal rating values by a greater number of users, and this case is clearly represented by mode.
(3) Suppose an even number of rating values are present like the following.

1 2 3 3 3 4 4 4 4 4

Then the median should ideally become $(3+4)/2 = 3.5$. But the addition operation is not defined here because the rating values are nothing but labels. Also, 3.5 is not a valid rating. With the use of mode, such difficulties can be avoided.

4.3. *The proposed algorithms*

For a given data matrix, first, we construct a missing matrix by deleting some values at random and then impute missing values with mean, median, and mode, respectively, and then we perform SVD. Depending on the imputation technique used, the algorithms can be named as *SVD-mean/median/mode*. We present the pseudo-code for *SVD-mode* and highlight the line where the code can be modified for the others.

Our algorithms differ according to step-2. In *SVD-mode*, we replace the missing entries with column mode to

According to step-2, we form a matrix R on which we apply SVD. Then we compute $U_{m\times r} \times \Sigma_{r\times r} \times V^T_{r\times n}$ to get a new matrix N, where r is the rank of the matrix

Algorithm 1. *SVD-mode*

```
Svd-mode:
Input: A m × n rating matrix $o$.
Output: A new m × n rating matrix $N$ with predicted values.
1. Construct the missing data matrix by deleting some of the entries of the original matrix $o$.
2. Replace the missing entries by column mode and form a matrix $R$.
4. Do the Reduced SVD of $R$.
5. Construct the new m × n rating matrix $N$ by computing $U_{m\times r} \times \Sigma_{r\times r} \times U^T_{r\times n}$.
6. Compare $o$ and $N$ by measuring the mean absolute error
```

and then compare the new matrix with the original matrix by mean absolute error (MAE). In *SVD-median* and *SVD-mean*, the missing entries are replaced by median and mean, respectively, and the same steps are followed. Normally before applying SVD to a dataset, mean centreing is done to normalize the data because there is a chance that different attributes are measured using different scales. Otherwise, there is a chance that a variable whose values have a higher range can outweigh a variable whose values have a lower range. The purpose of normalization is to keep the data on one scale. In a recommender system application, data is in the form of rating, and it is already bounded by a scale of 0 to 5 or 0 to 10. Thus, in recommender system applications, an algorithm can be designed by excluding this step.

Like SVD, the NMF method cannot be used for missing data imputation because the decomposed matrices W, X are different in each run of NMF. NMF behaves like a typical multiplication operation in this way. For example, 36 can be obtained by the multiplication of $(1 \times 36), (36 \times 1), (4 \times 9), (9 \times 4), (3 \times 12), (12 \times 3)$ or (6×6). Similarly, matrix A can be decomposed into various (W, X) pairs, and computation of mean, median, and mode will vary for each (W, X) pair. In such a situation, NMF is not a reliable method for missing value imputation.

4.4. *Examples*

In this section, we consider three examples to verify the methods of imputation. We follow the steps of the algorithms proposed in Sec. 4. We evaluate their performance based on the MAE. The method that gives the minimum MAE is considered to be the preferable one. Herlocker *et al.* (2004) divided the evaluation metrics for recommender systems into three broad categories such as predictive accuracy metrics, classification accuracy metrics, and rank accuracy metrics. For our purpose, predictive accuracy metrics are suitable because we intend to measure how close the recommender system's predicted rating is to the original rating. In the literature of recommender system, both MSE (Mean Squared Error) and MAE are used as the evaluation measures. We prefer MAE to MSE because MSE tends to give greater importance to values that have a larger difference in comparison to values that have a small difference, whereas MAE gives equal importance to all types of value. MAE measures the average absolute deviation between a predicted rating and the original rating. It is given by

$$\text{MAE} = \sum_{i=1}^{N} \frac{|O_i - N_i|}{N}, \tag{4}$$

where O_i is the original matrix, N_i is the new predicted matrix, and N is the total number of data items.

Example 1

This example is based on real data about 17 crops cultivated across 12 regions of Kandhamal district, Odisha, India. This dataset was obtained from the thesis work titled "Optimization Approach for Water and Land Use Planning in Rainfed Agricultural Systems" (Mohanty *et al.*, 2012). The original data matrices were in a form where rows represented crop names and columns represented names of months. The data was aggregated and converted to a 12 × 17 matrix where rows represented names of regions and columns represented crop names. Each cell value represented the total duration of time for which the crop was cultivated in a region. Then, this matrix was converted to a rating matrix by using the following formula.

$$\text{rating} = \frac{\text{Total duration of time for which the crop was cultivated}}{\text{Number of months needed for the cultivation of the crop}}. \tag{5}$$

The rating values varied in the range 0–10. The data is given in Table 1.

This is an example of numeric data. We deleted 13 values randomly and formed a missing data matrix. Then we imputed the missing values using mean, median, and mode and performed SVD. The MAE, in this case, is found to be 0.0615, 0.0686, and 0.0686 for mean, median, and mode, respectively. We see that mean performs better and thus should be the imputation technique.

Example 2

Table 1. Normalized data of seventeen cultivated crops of 12 regions of Kandhamal district, Odisha.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	3	4	6	9	1	2	3	4	1	1	2	1	1	1	0	2	0
2	2	2	7	10	1	2	3	4	1	1	2	1	1	1	2	2	1
3	2	3	5	7	1	2	2	4	1	1	1	1	1	1	0	2	1
4	3	1	7	3	1	1	1	2	1	1	1	1	1	1	1	1	0
5	3	1	4	6	1	2	2	1	1	1	1	1	1	1	0	1	0
6	3	3	6	8	1	2	2	4	1	1	3	1	1	1	0	2	0
7	2	2	7	9	1	2	2	4	1	1	3	0	1	1	2	2	1
8	2	2	7	8	1	2	2	4	1	1	3	1	1	1	0	3	0
9	2	3	7	9	1	2	2	3	1	1	3	1	1	1	0	1	0
10	2	3	6	7	1	2	3	4	1	1	3	1	1	1	0	1	0
11	3	3	5	10	1	2	2	4	1	1	3	1	1	1	1	1	0
12	3	3	6	9	1	2	3	4	1	1	3	1	1	1	1	2	0

In this example, we show that the median performs better and hence should be preferred for imputation.

$$O = \begin{bmatrix} 4 & 1 & 2 & 5 & 4 \\ 5 & 4 & 3 & 5 & 4 \\ 5 & 1 & 2 & 5 & 4 \\ 4 & 3 & 2 & 5 & 1 \\ 5 & 1 & 3 & 4 & 2 \\ 5 & 1 & 2 & 4 & 3 \\ 4 & 2 & 3 & 5 & 1 \\ 5 & 1 & 2 & 4 & 3 \\ 4 & 5 & 1 & 2 & 2 \end{bmatrix}$$

We consider the matrix O as given above. We deleted 12 values from O at random to form the missing matrix. In this case, the MAE is 0.2180, 0.1852, 0.2000 for mean, median, and mode, respectively. We see that the median performs better and thus can be considered as the imputation technique.

Example 3

In this case, we consider categorical binary data and show that mode is the preferred method of imputation.

$$O = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Here, the missing matrix was formed by deleting three values at random. For SVD, the MAE obtained are 0.0700, 0.0600, and 0.0400 for mean, median, and mode, respectively. We see that mode performs better for SVD.

5. SVD for Knowledge Discovery in Recommender Systems

In the SVD technique, the singular values are computed by considering distinct eigenvalues. Because the eigenvalues are distinct, their corresponding eigenvectors are linearly independent. These linearly independent eigenvectors of U and V give the basis vectors for all the four fundamental subspaces of a vector space in the following way:

1. First r columns of U: Column space of A
2. Last $m - r$ columns of U: Left Null space of A

3. First r columns of V or first r rows of V^T: Row space of A
4. Last $n - r$ columns of V or $n - r$ rows of V^T: Null space of A

Out of these four subspaces, the column space and the row space of matrix A are of great importance for an SVD based Recommender system application. The basis vectors in the column space of A are sufficient to describe the information contained in all the item vectors of the $user \times item$ matrix A. Similarly, the vectors in the row space of A contain all the information about the user vectors of A. The maximum number of linearly independent vectors in the column space of a matrix is equal to the maximum number of linearly independent vectors in the row space of the matrix, and that refers to the rank of the matrix. If we do the SVD of any matrix using MATLAB, we see that $\sum$ contains the singular values in decreasing order on the diagonal. We select only the nonzero diagonal values and discard the zeros. The number of nonzero diagonal elements gives the rank r of the matrix under consideration. Now, U can be chosen to be of dimensionality $m \times r$ and V of dimensionality $n \times r$. The dimension of $\sum$ becomes $r \times r$. The singular values represent the strength of the individual category along which the behavior of the users and items vary.

5.1. *SVD to discover the natural clusters in a recommender systems*

In the decomposition given in Fig. 3, we see that there are three nonzero singular values. This indicates that the rank of the matrix A is three and the dimension of both row space and column space of A is also 3. That means, actually, there are three categories along which the behaviour of the users and items can vary, and the numeric value of each of these singular values represent the strength of each of the categories. Also, we observe that in the matrix $\sum$, the last singular value is negligible in comparison to the first two. In practical applications, normally, such singular

$$\overset{\boldsymbol{A}}{\begin{bmatrix} 1 & 2 & 3 & 0 & 0 \\ 2 & 4 & 6 & 0 & 0 \\ 3 & 6 & 9 & 2 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 6 & 3 \\ 0 & 0 & 0 & 10 & 5 \\ 0 & 0 & 0 & 10 & 5 \end{bmatrix}} =$$

$$\overset{U}{\begin{bmatrix} -0.05 & -0.26 & -0.03 \\ -0.10 & -0.53 & -0.05 \\ -0.27 & -0.75 & -0.08 \\ -0.02 & -0.09 & 1.00 \\ -0.37 & 0.11 & 0.00 \\ -0.62 & 0.18 & 0.01 \\ -0.62 & 0.18 & 0.01 \end{bmatrix}} \overset{\sum}{\begin{bmatrix} 17.49 & 0 & 0 \\ 0 & 13.85 & 0 \\ 0 & 0 & 0.46 \end{bmatrix}} \overset{V^T}{\begin{bmatrix} -0.06 & -0.12 & -0.18 & -0.87 & -0.44 \\ -0.26 & -0.52 & -0.78 & 0.20 & 0.10 \\ -0.77 & 0.62 & -0.16 & -0.00 & 0.00 \end{bmatrix}}$$

Fig. 3. SVD of the example rating matrix.

values are ignored, considering them as noise in the data. However, we keep this value in order to get a clear understanding of the importance of singular values. In SVD, the matrix U captures user behaviour along with distinct categories, and the matrix V captures item behavior along with the categories. The matrix $U \sum$ helps to find to which category a user belongs. If we carefully look at the matrix U,we find that the first three users belong to the category-2 represented by σ_2 (indicated by values 0.26, 0.53, 0.75 in the second column) and the last three users to category-1 represented by σ_1 (indicated by values 0.37, 0.62, 0.62 in the first column). The middle user, i.e. user-4, seems to belong to category-3, represented by σ_3 (indicated by value 1.00). Now, let us look at $U \times \sum$ and try to figure out whether the inference is right or wrong.

$$U \times \sum = \begin{bmatrix} -0.85 & -3.64 & -0.01 \\ -1.69 & -7.29 & -0.02 \\ -4.72 & -10.43 & -0.03 \\ -0.30 & -1.30 & 0.46 \\ -6.53 & 1.52 & 0.00 \\ -10.89 & 2.53 & 0.00 \\ -10.89 & 2.53 & 0.00 \end{bmatrix}$$

Here, we see that the first three users belong to category-2 (indicated by values 3.64, 7.29, 10.43), and the last three users belong to category-1 (indicated by values 6.53, 10.89, 10.89), which is the same as the above inference. But, user-4 belongs to category-2 (indicated by value 1.30). Now, it is clearly understood that although many cases are clear from U, there will be some cases about which we cannot correctly infer from U, and thus it is better to consider $U \times \sum$. Similar to U, the matrix V^T along with $\sum$ help to find to which category an item belongs.

$$\sum \times V^T = \begin{bmatrix} -1.05 & -2.12 & -3.17 & -15.24 & -7.62 \\ -3.57 & -7.24 & -10.81 & 2.80 & 1.40 \\ -0.36 & 0.28 & -0.07 & 0.00 & 0.00 \end{bmatrix}$$

We see that the first three items belong to category-2 (indicated by values 3.57, 7.24, 10.81) and the last two to category-1 (indicated by values 15.24, 7.62). Basically, we have two categories of items. Note that, while deciding the category of a user/item, only the absolute value of the ijth place is considered where i represents row number and j represents column number.

From the above example, it is clear that by doing the SVD of a given $user \times item$ matrix in a recommender system, we are able to cluster the users and the items based on their similar behaviour. Such knowledge can be used for target marketing and to increase the profitability of an organization.

5.2. *Out-of-sample extension using SVD*

SVD can be used to find to which cluster a new user/item belongs. This can be referred to as out-of-sample extension because we are trying to assign an appropriate cluster for a user/item which is not present in the original dataset.

Suppose we have the following unknown user data q. $q = \begin{bmatrix} 0 \\ 4 \\ 5 \\ 0 \\ 0 \end{bmatrix}$, and we wish to find a suitable cluster for this user. This can be done by first projecting q to the column space of V and then by using similarity computation methods.

5.2.1. *Projection of a new user vector q to the column space of V*

If $q \in C(V)$, where $C(V)$ represents the column space of V, q can be expressed as a linear combination of the vectors of V.

$$\therefore q = c_1 v_1 + c_2 v_2 + c_3 v_3, \tag{6}$$

$$\text{where} \quad c_1 = v_1^T q, \quad c_2 = v_2^T q \quad \text{and} \quad c_3 = v_3^T q. \tag{7}$$

If $q \notin C(V)$, then $q^* =$ the projection of q onto $C(V)$, plays the role of q.

Multiplying both sides of equation (6) by v_1^T, we get

$$v_1^T q = v_1^T c_1 v_1 + v_1^T c_2 v_2 + v_1^T c_3 v_3 = c_1 v_1^T v_1 + c_2 v_1^T v_2 + c_3 v_1^T v_3 = c_1$$

[$\because v_1^T v_1 = 1$ and $v_1^T v_2 = v_1^T v_3 = 0$, according to the orthonormality condition].

Now, for the unknown user vector q, we find the coordinates in the column space of V as follows:

$$c_1 = v_1^T q = [-0.06 \quad -0.12 \quad -0.18 \quad -0.87 \quad -0.44] \begin{bmatrix} 0 \\ 4 \\ 5 \\ 0 \\ 0 \end{bmatrix} = -1.39$$

$$c_2 = v_2^T q = [-0.26 \quad -0.52 \quad -0.78 \quad 0.20 \quad 0.10] \begin{bmatrix} 0 \\ 4 \\ 5 \\ 0 \\ 0 \end{bmatrix} = -5.98$$

$$c_3 = v_3^T q = [-0.77 \quad 0.62 \quad -0.16 \quad 0.00 \quad 0.00] \begin{bmatrix} 0 \\ 4 \\ 5 \\ 0 \\ 0 \end{bmatrix} = 1.68$$

Now the projection of q on the column space of V is given by the coordinates $(-1.39, -5.98, 1.68)$.

5.2.2. *Similarity test for the new user vector to determine its cluster*

To determine an appropriate cluster for q, the similarity of q is tested with the existing users. For this, the projection of q on the column space of V is considered and is compared with the vectors of $U \times \sum$. Two distance measures, the Manhattan distance, the Euclidean distance, and the cosine similarity method, are used for the similarity test. The output of the test is given in Table 2.

From Table 2, we find that q is at a minimum distance from U_2, and it has the highest similarity with U_4. This is represented using boldface numbers in Table 2. Both U_2 and U_4 belong to cluster-2. Hence, q also belongs to cluster-2. Also, it can be observed from Table 1 that q is at a larger distance to the users of cluster-1 than is to the users U_5, U_6, and U_7 compared to all the four users in cluster-2, and also it has zero similarity with the users of cluster-1.

5.2.3. *Out-of-sample extension when the dataset is large*

Distance/cosine similarity computation to each user vector in the dataset becomes expensive when the dataset is large. To solve the problem, we propose algorithm 2 for similarity computation and show its working with the example taken.

In step-2 of algorithm 2, instead of finding the cluster mean from the original dataset, the mean can be computed from the clusters of $U \times \sum$. In that case, we do not have to project the mean vectors once again, only the new user vector can be projected, and the rest of the steps can be followed to determine the cluster of q. We tried both the methods in our example data, and both yielded the same output. In Table 3, we give the results.

Table 2. Output of similarity test for the new user vector q.

Users	Manhattan distance	Euclidean distance	Cosine similarity
U_1	4.58	2.94	0.06
U_2	**3.30**	**2.16**	0.06
U_3	9.47	5.80	0.05
U_4	7.00	4.97	**1.00**
U_5	14.33	9.26	0.00
U_6	19.70	12.87	0.00
U_7	19.70	12.87	0.00

Algorithm 2. Algorithm for out-of-sample extension for a single new user in a Recommender System

```
1. Determine user clusters from $U \times \sum$.
2. Find the cluster means for each user cluster from the original dataset.
3. Project the mean vectors to the column space of $V$.
4. Project the new user vector q to the column space of $V$ for which the cluster has to be determined.
5. Find the distance/ cosine similarity between the projected mean vectors and the projected new user vector $q$.
6. Based on step-5, assign $q$ to that user cluster whose mean is at a minimum distance from $q$ or has the highest cosine similarity value with $q$.
```

Table 3. Output of similarity test for the new user vector q considering cluster mean.

	Manhattan distance	Euclidean distance	Cosine similarity
Mean of cluster-1 & q	17.91	11.60	0
Mean of cluster-2 & q	2.41	1.69	0.6

From Table 3, it is evident that the distance between cluster-2 and the new user vector q is less as compared to its distance from cluster-1 as per the calculation by both Manhattan distance and Euclidean distance. Also, the similarity between q and cluster-2 is more. Thus, we can conclude that the new user vector q belongs to cluster-2.

5.2.4. *Out-of-sample extension in case of a large dataset with more than one new user*

In a recommender system application, it is quite possible that more than one new user may surface. In that case, algorithm two can be slightly modified, and out-of-sample extension can be done for all the new users at a time. The modified algorithm is presented as Algorithm 3.

Algorithm 3. Algorithm for out-of-sample extension for more than one new user in a Recommender Systems

```
1. Form a matrix $Q$ comprising of all the new user vectors $q_1, q_2, \dots, q_n$.
2. Determine the user clusters from $U \times \sum$.
3. Find the cluster mean from the clusters of $U \times \sum$.
4. Compute $C = V^T Q$. Let the columns of $C$ be $c_1, c_2, \dots, c_n$.
5. Find the distance/ cosine similarity between each cluster mean and each $c_i$.
6. Based on step 5, assign $q_i$ to that user cluster whose mean is at a minimum distance from $q_i$ or has the highest cosine similarity value with $q_i$.
```

Step 4 of the algorithm is crucial. Each column c_i of $C = V^T Q$ represents the projection of q_i on the column space of $V$when individually considered. Step-5 of the algorithm computes the distance/cosine similarity between each c_i of step-4 with each cluster mean of step-3. In step 6, a cluster is assigned to each new user vector based on its minimum distance from a cluster centre or maximum similarity with a cluster.

From the discussion of Sec. 5.2, it is understood that once the SVD of a data matrix is done, cluster assignment for any number of new user vectors can be done with the help of vector projection, without recomputing the SVD again, which saves a lot of time and memory. The method and algorithms presented in this section for user clustering and out-of-sample extension for new users can also be applied in a similar manner to item clustering and out-of-sample extension for new items.

6. NMF for Knowledge Discovery in Recommender Systems

NMF is another suitable technique that can be used for low-rank approximation and clustering in a collaborative filtering-based recommender system application. The *user* × *item* matrix in a recommender system contains rating values that are normally nonnegative. Given a rating matrix Y and a rank value k, NMF decomposes Y into two nonnegative matrices, A and $X = B^T$. A is the matrix of basis vectors, and B is the component matrix. The user-specified rank k determines the number of categories or clusters. The NMF decomposition of our example rating matrix is given in Fig. 4. From the SVD of the example matrix, we had obtained the rank to be three. So we conduct NMF by considering k value to be three.

6.1. *NMF to discover the natural clusters in a recommender systems*

The NMF of the rating matrix helps in simultaneous clustering of the users as well as of the items. From the matrix W, we see that the first three users belong to cluster-2 (indicated by values 3.42, 6.84, 10.27), the last three users belong to cluster-1 (indicated by values 6.71, 11.18, 11.18), and the fourth user belongs to cluster-3. From matrix X, we see that the last two items belong to cluster-1, the first item and the third item belong to cluster-2, and the second item belongs to cluster-3.

We find that clustering using NMF is simpler compared to SVD. But some limitations are there with the NMF approach. First, the user has to specify the value of

$$\overset{\boldsymbol{A}}{\begin{bmatrix} 1 & 2 & 3 & 0 & 0 \\ 2 & 4 & 6 & 0 & 0 \\ 3 & 6 & 9 & 2 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 6 & 3 \\ 0 & 0 & 0 & 10 & 5 \\ 0 & 0 & 0 & 10 & 5 \end{bmatrix}} \approx \overset{\boldsymbol{W}}{\begin{bmatrix} 0 & 3.42 & 0.34 \\ 0 & 6.84 & 0.68 \\ 2.23 & 10.27 & 1.02 \\ 0 & 0 & 1.41 \\ 6.71 & 0 & 0 \\ 11.18 & 0 & 0 \\ 11.18 & 0 & 0 \end{bmatrix}} \overset{\boldsymbol{X = B^T}}{\begin{bmatrix} 0 & 0 & 0 & 0.89 & 0.45 \\ 0.29 & 0.51 & 0.81 & 0 & 0 \\ 0 & 0.71 & 0.71 & 0 & 0 \end{bmatrix}}$$

Fig. 4. NMF of the example rating matrix.

k, which will determine the number of clusters. The value can be chosen depending on the application's requirement, or some other method must be applied to determine k. k has to be strictly less than or equal to the rank of the matrix under consideration; otherwise, the program will give an error. Second, the decomposed matrices W and X are not unique. Each run of the program will generate different values, although clustering remains the same. On the other hand, SVD does not have such limitations and is a completely automated method of discovering natural clusters in the dataset. Due to this advantage of SVD, it can also be considered to solve the initialization problem for NMF. After the SVD of a large data matrix, the lower rank value is obtained by considering the prominent singular values only as the initial value for k in NMF.

6.2. *Out-of-sample extension using NMF*

If we want to find out the user cluster for a new user vector $q = \begin{bmatrix} 0 \\ 4 \\ 5 \\ 0 \\ 0 \end{bmatrix}$, it canbe done by projecting q on B. Let the projection of q on B is $\hat{q}$, then

$$\hat{q} = (B^T B)^{-1} B^T q \tag{8}$$

$$\therefore \hat{q} = \left(\begin{bmatrix} 0 & 0 & 0 & 0.89 & 0.45 \\ 0.29 & 0.51 & 0.81 & 0 & 0 \\ 0 & 0.71 & 0.71 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0.29 & 0 \\ 0 & 0.51 & .71 \\ 0 & 0.81 & .71 \\ 0.89 & 0 & 0 \\ 0.45 & 0 & 0 \end{bmatrix} \right)^{-1}$$

$$\times \begin{bmatrix} 0 & 0 & 0 & 0.89 & 0.45 \\ 0.29 & 0.51 & 0.81 & 0 & 0 \\ 0 & 0.71 & 0.71 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 4 \\ 5 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1.25 \\ 5.19 \end{bmatrix}$$

The projected q has the highest value in the third row. So, it belongs to the third user cluster. Similarly, when another user vector $q = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$ is projected on B, then the projected vector $\hat{q} = \begin{bmatrix} 5.81 \\ 3.74 \\ 0 \end{bmatrix}$. In this case, the first row contains the highest value. So, the new user belongs to the user cluster-1.

Suppose we want to find the user clusters for more than one new user then first we construct a matrix Q with all the new users such that $Q = [q_1 \; q_2 \ldots \; q_i]$ and use the formula given by Eq. (8) by replacing q with Q. After that, each column vector can be checked and the row having the highest value in a column vector can be considered as the cluster number.

7. Cluster Quality Evaluation Using NMI (Normalized Mutual Information) and Purity

We use two cluster evaluation measures NMI and purity, to evaluate the quality of the clusters formed by SVD and NMF. Both of them are external evaluation measures because, in order to use them, it is assumed that class label information is available. For our example matrix A, we assume that users are classified according to their age. There are three classes, class-I with users of age between 0 and 18, class-II with users of age between 18 and 30, class-III with users of age greater than or equal to 30.

7.1. *Normalized Mutual Information (NMI)*

NMI is defined by the following formula:

$$\mathrm{NMI}(Y, C) = \frac{2 * I(Y; C)}{H(Y) + H(C)}, \tag{9}$$

where $Y = (Y_1, Y_2, Y_3)$ = class labels, $C = (C_1, C_2, C_3)$ = cluster labels, $H(Y)$ = Entropy of Y, $H(C)$ = Entropy of C and I(Y; C) = Mutual Information between Y and C. Let number of classes be denoted as m and number of clusters be denoted as k.

$$H(Y) = -\sum_{i=1}^{m} P(Y_i)\log_2 P(Y_i), \tag{10}$$

$$H(C) = -\sum_{l=1}^{k} P(C_l)\log_2 P(C_l), \tag{11}$$

$$I(Y; C) = H(Y) - H(Y|C), \tag{12}$$

$$\text{where} \quad H(Y|C) = \sum_{l=1}^{k} H(Y|C_l) \tag{13}$$

$$\text{and} \quad H(Y|C_l) = -P(C_l)\sum_{i=1}^{m} P(Y_i|C_l)\log_2 P(Y_i|C_l) \tag{14}$$

7.1.1. *NMI for SVD clustering*

In this case, we have $m = 3$ & $k = 2$.

We assume that $Y_1 = \{U_1, U_2\}, Y_2 = \{U_3, U_4\}, Y_3 = \{U_5, U_6, U_7\}$.

From SVD clustering, we have obtained $C_1 = \{U_5, U_6, U_7\}, C_2 = \{U_1, U_2, U_3, U_4\}$

$$P(Y_1) = \frac{2}{7}, \quad P(Y_2) = \frac{2}{7}, \quad P(Y_3) = \frac{3}{7}$$

$$P(C_1) = \frac{3}{7}, \quad P(C_2) = \frac{4}{7}$$

According to Eq. (10)

$$H(Y) = -\frac{2}{7}\log_2\frac{2}{7} - \frac{2}{7}\log_2\frac{2}{7} - \frac{3}{7}\log_2\frac{3}{7} = 1.55386$$

According to Eq. (11)

$$H(C) = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7} = .983528$$

According to Eq. (14)

$$H(Y|C_1) = -\frac{3}{7}\left[0\log_2 0 + 0\log_2 0 + \frac{3}{3}\log_2\left(\frac{3}{3}\right)\right] = 0$$

$$H(Y|C_2) = -\frac{4}{7}\left[\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4} + 0\right] = 0.571$$

According to Eq. (13), $H(Y|C) = \sum_{l=1}^{2} H(Y|C_l) = H(Y|C_1) + H(Y|C_2) = 0 + .571 = .571$

According to Eq. (12), $I(Y;C) = H(Y) - H(Y|C) = 1.55386 - 0.571 = .98286$.

According to Eq. (9), $NMI(Y, C) = \frac{2*I(Y;C)}{H(Y)+H(C)} = \frac{2*.98286}{1.55386+.983528} = .774$.

7.1.2. *NMI for NMF clustering*

In this case, we have $m = 3$ & $k = 3$.

We assume that $Y_1 = \{U_1, U_2\} Y_2 = \{U_3, U_4\} Y_3 = \{U_5, U_6, U_7\}$

From NMF clustering, we have obtained $C_1 = \{U_5, U_6, U_7\}, C_2 = \{U_1, U_2, U_3\}, C_3 = \{U_4\}$

$$P(Y_1) = \frac{2}{7}, \quad P(Y_2) = \frac{2}{7}, \quad P(Y_3) = \frac{3}{7}$$

$$P(C_1) = \frac{3}{7}, \quad P(C_2) = \frac{3}{7}, \quad P(C_3) = \frac{1}{7}$$

According to Eq. (10)

$$H(Y) = -\frac{2}{7}\log_2\frac{2}{7} - \frac{2}{7}\log_2\frac{2}{7} - \frac{3}{7}\log_2\frac{3}{7} = 1.55386$$

According to Eq. (11)

$$H(C) = -\frac{3}{7}\log_2\frac{3}{7} - \frac{3}{7}\log_2\frac{3}{7} - \frac{1}{7}\log_2\frac{1}{7} = 1.44416$$

According to Eq. (14)

$$H(Y|C_1) = -\frac{3}{7}\left[0\log_2 0 + 0\log_2 0 + \frac{3}{3}\ \log_2\left(\frac{3}{3}\right)\right] = 0$$

$$H(Y|C_2) = -\frac{3}{7}\left[\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3} + 0\right] = 0.392$$

$$H(Y|C_3) = -\frac{1}{7}\left[0 + 1\log_2 1 + 0\right] = 0$$

According to Eq. (13), $H(Y|C) = \sum_{l=1}^{3} H(Y|C_l) = H(Y|C_1) + H(Y|C_2) + H(Y|C_3) = 0 + .392 + 0 = .392$

According to Eq. (12), $I(Y;C) = H(Y) - H(Y|C) = 1.55386 - 0.392 = 1.16186$

According to Eq. (9), $NMI(Y,C) = \frac{2*I(Y;C)}{H(Y)+H(C)} = \frac{2*1.16186}{1.55386+1.44416} = .775$

7.2. *Purity*

Computation of purity is based on the fact that a cluster is considered to be pure if it contains labelled objects of one and only one class, and it is impure if it contains labelled objects from many different classes.

The formula for purity is given by:

$$\text{purity}(C,Y) = \frac{1}{N}\sum_k \max_j|C_k \cap Y_j| \tag{15}$$

where $C = \{C_1, C_2, \ldots, C_k\}$ is the set of clusters and $Y = \{Y_1, Y_2, \ldots, Y_j\}$ is the set of classes.

To compute purity, first, the majority class and the number of members of the majority class in each cluster is determined. Then the sum of the number of majority class members in each cluster is divided by the total number of objects to determine the purity of the clustering.

7.2.1. *Purity for SVD clustering*

$$\text{purity}(C,Y) = \frac{1}{7}\{\ \max_j|C_1 \cap Y_j| +\ \max_j|C_2 \cap Y_j|\} = \frac{1}{7}\{3+2\} = \frac{5}{7} = .714$$

7.2.2. *Purity for NMF clustering*

$$\begin{aligned}\text{purity}(C,Y) &= \frac{1}{7}\{\max_j|C_1 \cap Y_j| + \max_j|C_2 \cap Y_j| + \max_j|C_3 \cap Y_j|\}\\ &= \frac{1}{7}\{3+2+1\} = \frac{6}{7} = .857\end{aligned}$$

It is observed that NMI of both SVD-clustering and NMF-clustering are approximately the same, but the purity of NMF-clustering is higher than that of SVD-clustering. Thus, in this case, NMF-clustering can be preferred over SVD-clustering. However, the clusters were automatically discovered in the case of SVD, but in the case of NMF, the number of clusters was predefined like the K-means or K-medoid clustering algorithms. Thus, depending on the values of NMI and purity and the requirement of the application, one can choose between SVD-clustering and NMF-clustering.

8. Contribution of Work

Through this work, we have explored the different aspects of knowledge discovery by using two matrix factorization techniques, SVD and NMF in the context of a recommender system. Through our algorithms and examples, we have shown that SVD and NMF can be used for clustering and out-of-sample extension. We have demonstrated the clustering capabilities of SVD and NMF by constructing suitable rating matrices. For out-of-sample extension, in the case of SVD, we have developed two new algorithms, one for single new user and another for more than one new user and have shown their working through examples. Similarly, for NMF, we have done out-of-sample extension using vector projection and have given a suitable example for that. For multiple users, we have explained the steps to be followed for out-of-sample extension. Further, we have explored the capability of SVD in doing missing data imputation. In this context, we have given a detailed analysis about the type of data used and the method of imputation that can be considered for that. We have given appropriate examples and shown that along with mean, median and mode can also be used for imputation. We have also given reasoning about the mode-substitution method as an imputation technique for recommender system rating matrices. In this line, we have proposed the SVD-mode algorithm. In continuation to this, we have given reasoning about why NMF cannot be used for missing data imputation. In the end, we have evaluated the quality of clusters using two external evaluation measures NMI and purity and have reached at the conclusion that depending on the need of the application, we can go for any of these methods for clustering. However, SVD is a more autonomous method than NMF because in case of NMF, the number of clusters need to be specified, but SVD does not need anything other than the dataset.

9. Conclusion

This paper gives a comparative study and analysis of two famous matrix factorization techniques, the SVD and the NMF, for knowledge discovery in a recommender system. Recommender systems are very useful for mankind as well as for business organizations. Thus Recommender systems data is analysed using many mathematical tools, and MF techniques have become quite popular in that. MF needs a matrix with no missing values and recommender systems rating matrices have a genuine problem of missing data. Algorithms for missing data imputation using mean/median/mode have been proposed to combat this problem. The natural clustering ability of SVD and NMF has been used for knowledge discovery in the dataset. Algorithms and methods for out-of-sample extension for both SVD and NMF have been presented for two different cases, one for the case of a single user and the other for the case of more than one user. The clusters obtained from SVD-clustering and NMF-clustering can help in target marketing and generate profit for business organizations. On the other hand, with the help of an out-of-sample extension, clusters for new users can be obtained, and more appropriate recommendations can be made. NMI and purity have been used to measure the cluster quality, and it is found that NMF produces better clusters than SVD. But the number of clusters needs to be specified in the case of NMF, whereas SVD can find the number of clusters automatically. The number of clusters obtained by SVD clustering can be used for NMF initialization. Thus, both the matrix factorization techniques can be used in a recommender system in their own right.

References

Aggarwal, CC (2016). *Recommender Systems: The Textbook.* Cham: Springer International Publishing.

Aghdam, MH, M Analoui and P Kabiri (2015). A novel nonnegative matrix factorization method for recommender systems. *Applied Mathematics & Information Sciences*, 9(5), 2721–2732.

Ahani, A, M Nilashi, O Ibrahim, L Sanzogni and S Weaven (2019). Market segmentation and travel choice prediction in Spa hotels through TripAdvisor's online reviews. *International Journal of Hospitality Management*, 80, 52–77.

Altulyan, MS, C Huang, L Yao, X Wang, S Kanhere and Y Cao (2019, November). Reminder care system: An activity-aware cross-device recommendation system. In *International Conference on Advanced Data Mining and Applications*, (pp. 207–220). Springer, Cham.

Badami, M and O Nasraoui (2021). PaRIS: Polarization-aware recommender interactive system. In *Proceedings of the 2nd Workshop on Online Misinformation-and Harm-Aware Recommender Systems (OHARS 2021)*, Amsterdam, Netherlands.

Bao, Y, H Fang and J Zhang (2014). Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 2–8.

Bennett, DA (2001). How can I deal with missing data in my study? *Australian and New Zealand Journal of Public Health*, 25(5), 464–469.

Benzi, K, V Kalofolias, X Bresson and P Vandergheynst (2016). Song recommendation with nonnegative matrix factorization and graph total variation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2439–2443.

Bobadilla, J, R Bojorque, AH Esteban and R Hurtado (2017). Recommender systems clustering using Bayesian nonnegative matrix factorization. *IEEE Access*, 6, 3549–3564.

Brand, M. (2003). Fast online svd revisions for lightweight recommender systems. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pp. 37–46.

Campos, R, RP dos Santos and J Oliveira (2020). A recommendation system based on knowledge gap identification in MOOCs ecosystems. In *XVI Brazilian Symposium on Information Systems*, pp. 1–8.

Ch, AK, SM Dias and NJ Vieira (2015). Knowledge reduction in formal contexts using non-negative matrix factorization. *Mathematics and Computers in Simulation*, 109, 46–63.

Chen, G, F Wang and C Zhang (2009). Collaborative filtering using orthogonal nonnegative matrix tri-factorization. *Information Processing & Management*, 45(3), 368–379.

Desarkar, MS, R Saxena and S Sarkar (2012). Preference relation based matrix factorization for recommender systems. In *International Conference on User Modelling, Adaptation, and Personalization*, pp. 63–75.

Ebadi, A and A Krzyzak (2016). A hybrid multi-criteria hotel recommender system using explicit and implicit feedbacks. *International Journal of Computer and Information Engineering*, 10(8), 1377–1385.

Graham, J. W. (2009). Missing data analysis: Making it work in the real world. *Annual Review of Psychology*, 60, 549–576.

Gu, Q, J Zhou and C Ding (2010). Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pp. 199–210.

Guo, X, SC Yin, YW Zhang, W Li and Q He (2019). Cold start recommendation based on attribute-fused singular value decomposition. *IEEE Access*, 7, 11349–11359.

Herlocker, JL, JA Konstan, LG Terveen and JT Riedl (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 5–53.

Hernando, A, J Bobadilla and F Ortega (2016). A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. *Knowledge-Based Systems*, 97, 188–202.

Hubert, L, J Meulman and W Heiser (2000). Two purposes for matrix factorization: A historical appraisal. *SIAM Review*, 42(1), 68–82.

Kang, H (2013). The prevention and handling of the missing data. *Korean Journal of Anesthesiology*, 64(5), 402–406.

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

Kumar, AC (2009). Analysis of unsupervised dimensionality reduction techniques. *Computer Science and Information Systems*, 6(2), 217–227.

Kurucz, M., AA Benczúr and K Csalogány (2007). Methods for large scale SVD with missing values. In *Proceedings of KDD Cup and Workshop*, pp. 31–38.

Lee, DD and HS Seung (1999). Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401(6755), 788–791.

Lever, J, S Gakkhar, M Gottlieb, T Rashnavadi, S Lin, C Siu, ... and SJ Jones (2018). A collaborative filtering-based approach to biomedical knowledge discovery. *Bioinformatics*, 34(4), 652–659.

Li, T, C Gao and J Du (2009). A NMF-based privacy-preserving recommendation algorithm. In *2009 First International Conference on Information Science and Engineering*, pp. 754–757.

Li, T and CC Ding (2018). Nonnegative matrix factorizations for clustering: A survey. In *Data Clustering*, pp. 149–176. Chapman and Hall/CRC.

Luo, X, M Zhou, Y Xia and Q Zhu (2014). An efficient nonnegative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, 10(2), 1273–1284.

Martinez, ABB, JJP Arias, AF Vilas, JG Duque and ML Nores (2009). What's on TV tonight? An efficient and effective personalized recommender system of TV programs. *IEEE Transactions on Consumer Electronics*, 55(1), 286–294.

Mohanty PK, D Panigrahi and M Acharya (2012): Optimization approach for water and land use planning in rain fed agricultural systems: PhD Thesis. Siksha 'O' Anusandhan University, Odisha, India.

Nilashi, M, O Ibrahim and K Bagherifard (2018). A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications*, 92, 507–520.

Pan, W, E Xiang, N Liu and Q Yang (2010). Transfer learning in collaborative filtering for sparsity reduction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 230–235.

Ricci, F, L Rokach and B Shapira (2011). *Recommender Systems Handbook.* Boston, MA, Springer.

Sarwar, B, G Karypis, J Konstan and J Riedl (2000). Application of dimensionality reduction in recommender system–a case study: Technical Report, Minnesota Univ. Minneapolis Dept. of Computer Science.

Sadeghi, S, J Lu and A Ngom (2021). A network-based drug repurposing method via nonnegative matrix factorization. *Bioinformatics* (Oxford, England), btab826.

Sarwar, B, G Karypis, J Konstan and J Riedl (2000). Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, pp. 158–167.

Sarwar, B, G Karypis, J Konstan and J Riedl (2002). Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Fifth International Conference on Computer and Information Science*, pp. 27–28.

Stewart, GW (1998). On the early history of the singular value decomposition.

Strang, G (2006) *Linear Algebra and its Applications, Cengage Learning*, Harcourt Brace-Jovanovich, San Diego.

Su, JH, WY Chang and VS Tseng (2017). Effective social content-based collaborative filtering for music recommendation. *Intelligent Data Analysis*, 21(S1), S195–S216.

Swaminathan, S, DY Zubarev and M Kunitomi (2019, November). A recommender system for antimicrobial resistance. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1373–1379. IEEE.

Vozalis, MG and KG Margaritis (2007). Using SVD and demographic data for the enhancement of generalized collaborative filtering. *Information Sciences*, 177(15), 3017–3037.

Vozalis, M, A Markos and K Margaritis (2009). Evaluation of standard SVD-based techniques for Collaborative Filtering. *Proceeding of the 9th Hellenic European Research on Computer Mathematics and its Applications.*

Weerasinghe, AM and RA Rupasingha (2021). Improving web service recommendation using clustering with K-NN and SVD algorithms. *KSII Transactions on Internet & Information Systems*, 15(5), 1708–1727.

Yuan, X, L Han, S Qian, G Xu and H Yan (2019). Singular value decomposition based recommendation using imputed data. *Knowledge-Based Systems*, 163, 485–494.

Zhang, S, W Wang, J Ford and F Makedon (2006). Learning from incomplete ratings using nonnegative matrix factorization. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pp. 549–553.

Zhang, X, L Chen, Y Wang and G Wang (2021). Improving incremental nonnegative matrix factorization method for recommendations based on three-way decision making. *Cognitive Computation*, 1–19.

Zhou, X, J He, G Huang and Y Zhang (2015). SVD-based incremental approaches for recommender systems. *Journal of Computer and System Sciences*, 81(4), 717–733.