SVD-initialised K-means clustering for collaborative filtering recommender systems

Murchhana Tripathy*

Information Systems and Technology,
T A Pai Management Institute,
Manipal, Karnataka, India
Email: murchhanatripathy@gmail.com
*Corresponding author

Santilata Champati

Department of Mathematics,
Institute of Technical Education and Research,
Siksha 'O' Anusandhan Deemed to be University,
Bhubaneswar, Odisha, India
Email: santilatachampati@soa.ac.in

Srikanta Patnaik

Department of Computer Science and Engineering,
Institute of Technical Education and Research,
Siksha 'O' Anusandhan Deemed to be University,
Bhubaneswar, Odisha, India
Email: srikantapatnaik@soa.ac.in

Abstract: K-means is a popular partitional clustering algorithm used by collaborative filtering recommender systems. However, the clustering quality depends on the value of K and the initial centroid points and consequently research efforts have instituted many new methods and algorithms to address this problem. Singular value decomposition (SVD) is a popular matrix factorisation technique that can discover natural clusters in a data matrix. We use this potential of SVD to solve the K-means initialisation problem. After finding the clusters, they are further refined by using the rank of the matrix and the within-cluster distance. The use of SVD based initialisation for K-means helps to retain the cluster quality and the cluster initialisation process gets automated.

Keywords: recommender systems; collaborative filtering; singular value decomposition; SVD; K-means initialisation; within-cluster distance; rank of the matrix.

Reference to this paper should be made as follows: Tripathy, M. Champati, S. and Patnaik, S. (2022) 'SVD-initialised K-means clustering for collaborative filtering recommender systems', *Int. J. Management and Decision Making*, Vol. 21, No. 1, pp.71–91.

Biographical notes: Murchhana Tripathy is working as a Faculty in the Information Systems and Technology area in T A Pai Management Institute, Manipal, Karnataka, India. She received her PhD in Computer Science and Engineering from the Institute of Technical Education and Research, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha, India. Her research interests include data mining, machine learning and artificial intelligence.

Santilata Champati is an Associate Professor in the Department of Mathematics, Institute of Technical Education and Research, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha, India. She received her PhD in Mathematics from Berhampur University, Odisha, India. Her research interests include data mining, machine learning and prediction of dynamic behaviour of data spaces using fractional difference analysis.

Srikanta Patnaik is a Professor in the Department of Computer Science and Engineering, Institute of Technical Education and Research, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha, India. He received his PhD (Engineering) on Computational Intelligence from Jadavpur University, India in 1999 and supervised 25 PhD theses and more than 60 master theses in the area of computational intelligence, soft computing applications and re-engineering. He has published around hundred research papers in international journals and conference proceedings. He is author of three text books and 52 edited volumes and few invited book chapters, published by leading international publisher like Springer-Verlag, Kluwer Academic, etc.

1 Introduction

Recommender systems are a class of popular software systems used by many business organisations. They came into the limelight since their inception in the mid-nineties through the work of the GroupLens research group, the University of Minnesota which designed a news recommender system for Usenet news. Later MovieLens was created by the same research group which was used to generate personalised movie recommendations. Recommender systems were presumed to be very important by 2006 through the announcement of the Netflix prize competition and its subsequent editions. They have been popularly used by many e-commerce giants such as Amazon, Netflix, Spotify, Facebook, LinkedIn, Google Scholar, YouTube, Flipkart, Trivago, etc. to name a few. There are four categories of Recommender systems which are used these days and they are collaborative filtering (CF) based, content-based, knowledge-based and hybrid recommender systems (Manouselis, 2008; Aggarwal, 2016) All of them use the rating information provided by users for the items to generate recommendations because the data for any recommender systems are in the form of ratings only. But besides CF methods rest others need some additional information along with the rating data to generate a recommendation. For example, content-based methods use the description of the items along with the user-item rating and knowledge-based systems generate recommendations according to user's requirement specification, domain knowledge and item attributes. Similarly, a hybrid method is a combination of any of the other three

methods. For this reason, collaborative filtering recommender systems (CFRS) are very simple to implement and the most preferred ones.

In this article, we propose a singular value decomposition (SVD)-initialised K-means clustering method for a CFRS. The motivation behind this work comes from the fact that clustering in CF helps to improve the recommendation generation process. Clustering creates groups of users or items such that the most similar users or items are always in a group. The members in a group share the same interest and considering this information helps to generate better recommendation. Also, it is well known that the clustering result produced by K-means is affected by the K value that is the number of centroids and the initial centroid points. Through this work, we attempt to solve the K-means initialisation problem. Further, in a recommender system application, large rating matrices are involved, and dimensionality reduction is performed before doing any analysis. SVD is a popular dimensionality reduction technique and in recommender system applications, where we intend to use K-means clustering, it can be first used for dimensionality reduction and then the output of SVD can be used to solve the initialisation problem of K-means. We propose a novel K-means initialisation algorithm for CFRS by using the group patterns obtained from SVD, where we do not have to specify the value of K or the initial centroid points. Our contribution is an automated centroid initialisation method for K-means using SVD which also yields quality clusters and more accurate recommendations. This is achieved by doing the following. Quality in a recommender system is assessed by similar choice or taste and the user/item groups formed by applying SVD on a dataset retain this property. Singular values are the square root of the eigenvalues of the associated Gram matrix A^TA of any rating matrix A. The eigenvalues capture the hidden characteristics of a dataset and hence the singular values. Therefore, the formation of user groups or item groups using SVD on a rating matrix is pertaining to the characteristics of the users and items. This knowledge is used to enhance the K-means clustering quality for CFRS. In addition to this, the rank of a matrix and the within cluster distance are used to take care of two conflicting interests that is restricting the number of clusters and avoiding unnecessary cluster splitting by the algorithm. This ensures that the actual number of clusters are found in the dataset which are based on the characteristics of the dataset. In this work we have used a user-based CF setting but this can also be applied to item-based CF.

The rest of the article is arranged as follows. Section 2 gives the background about CF, K-means and SVD. In Section 3, related work regarding several K-means initialisation methods and the use of K-means in CF has been presented. Section 4 explains the methodology followed and presents an object process diagram. Section-5 is the result section and in Section 6 we present the SVD-initialised K-means clustering algorithm. In Section 7, a discussion about the working of the algorithm and the challenges are presented. Section 8 concludes the article.

2 Background

2.1 CFRS

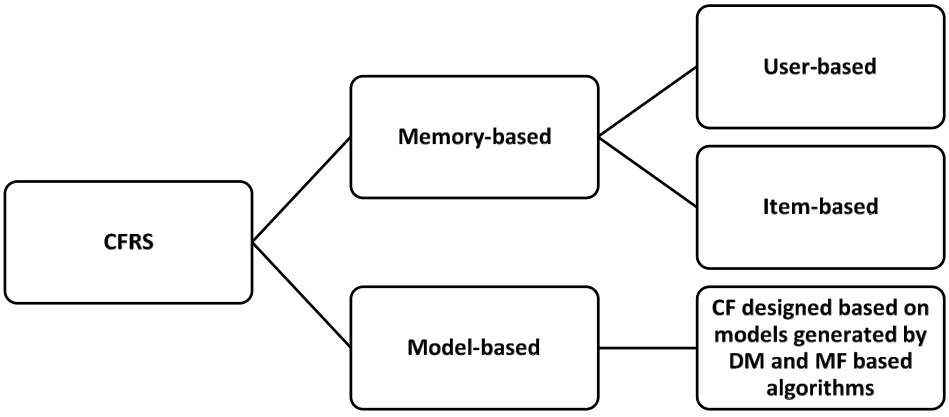
CFRS is very popularly used by many e-commerce giants such as Amazon, YouTube, Netflix, etc. According to an article published by McKinsey, 35% of the purchase on Amazon are due to the implementation of CF recommendation engines (MacKenzie

et al., 2013) Similarly, 70% of the time what people watch on YouTube and 75% of the time what people watch on Netflix are due to the use of a CF recommender system (MacKenzie et al., 2013) CF algorithms are popular because their implementation is very simple. They directly use information about customers' interest in products that are available in the form of ratings. Ratings are normally specified by using a five-point scale, a seven-point scale or a ten-point scale. For example, Amazon and Netflix use a five-point scale. Sometimes binary rating method can be adapted where zero represents a disliking for an item and one represents liking for the item. CFRS work on the theory that there exists a higher degree of interconnection between the ratings given by the users and they exploit this collaborative power to generate a recommendation.

2.2 Classification of CFRS

The classification of CFRS is presented in Figure 1. It is of two types: memory-based and model-based. Further memory-based methods can be either user-based or item-based and model-based CF are designed by models generated by either data mining (DM) methods or matrix factorisation (MF) techniques.

Figure 1 Classification of CFRS



2.2.1 Memory-based methods

Memory-based CF methods predict unknown ratings or generate a recommendation from the neighbourhood of users and items. They are of two types, user-based CF and item-based CF. In user-based CF, the rating provided by similar users of an individual user is used to generate a recommendation for that user. In item-based CF, the rating of a desired item *i* by a specific user u is predicted by using the ratings given by *u* for a set of similar items *I* to that of *i*. In this work, a user-based CF setting is used. Thus, we explain all the related concepts concerning a user-based CF. The concepts are the same for item-based CF. To determine the neighbourhood of a user, a similarity measure such as Euclidean distance, cosine similarity or Pearson correlation coefficient is used. Then the ratings of the neighbours are used to make item prediction for an active user. Determining the neighbourhood of a user and making a prediction for an active user are the offline and the online phase of the CF respectively. After the offline phase is over, the neighbourhood of

each user is stored. In the online phase, the ratings of the neighbourhood is used to predict unknown ratings (Aggarwal, 2016) For a *user* × *item* matrix of size $m \times n$, let n_1 specifies the highest count of ratings for a user. The time for similarity computation between two individual users is n_1 and the time required to determine the neighbourhood of one user is $O(m.n_1)$ Thus for m users, the time required to determine the neighbourhood is $O(m^2.n_1)$ For item-based methods, the time required to determine the neighbourhood of an item is $O(n^2.m_1)$ where m_1 is the maximum number of ratings specified for a user. The space complexity of the user-based and item-based methods is $O(m^2)$ and $O(n^2)$ respectively because the distance between each pair of users or each pair of item needs to be calculated and stored. Here it is observed that both the time and space requirement of the neighbourhood-based method is quadratic. In this situation, clustering comes as a rescue. In clustering-based methods, clustering becomes the offline phase replacing the offline nearest neighbour phase. In the clustering approach, an algorithm is used to discover the desired number of clusters. Also, a vector that summarises or represents each of these clusters is found out (Linden et al., 2003) When a new user arrives, the similarity of the user is checked with these representative vectors and the most similar vector's cluster is considered to be the new user's cluster. Then rating prediction is done by considering users within that cluster only. In this process, the pairwise distance computation reduces significantly and efficiency increases. Also, clustering methods have better scalability (Linden et al., 2003) But the disadvantage of the clustering approach is that accuracy gets compromised because neighbours of an active user in a single cluster are of inferior quality than the neighbours found from the entire dataset (Aggarwal, 2016; Linden et al., 2003) This issue needs to be addressed so that the advantage of clustering can be utilised without compromising on the quality of the clusters.

2.2.2 Model-based methods

In Model-based CF, DM algorithms such as KNN, SVM etc. and MF methods such as SVD and NMF are used on the training data to build a model and the prediction for the test data is done by using these models.

2.3 K-means clustering

Clustering refers to a DM task, where the data points in a data set are divided into groups such that the data points in a group are more like each other and the data points of two different groups are dissimilar. Clustering is also referred to as an unsupervised classification scheme (Nath et al., 2014; Dey, 2016). Jabeen et al. (2018) describes clustering to be partitional, hierarchical, density-based, graph theoretic-based or soft computing-based. K-means is a partitional clustering algorithm used in numerous DM and machine learning applications and recommender systems are one of them. K-means find K mutually exclusive clusters from the dataset. The pseudocode for classical K-means clustering is given below.

Algorithm: K-means

Input: Dataset X with n points and K

Output: K clusters

1 Select any K points from X as centroids.

2 Assign each point to its nearest centroid, each centroid representing an individual cluster.

3 Calculate mean of each cluster formed at step-2 and update centroid to be the calculated mean.

4 Repeat steps 2 and 3 until clusters remain unchanged.

5 Yield K clusters.

The classical K-means algorithm selects K centroids randomly from a set of given n points and then assigns the remaining points to its nearest centroid and forms the initial K clusters. Then it calculates the mean of each of these clusters and updates the centroids. This process continues until the clusters are stable. In each iteration, K-means try to minimise the within-cluster distance D_{wc} given by equation (1)

$$D_{wc} = \sum_{i=1}^{K} \sum_{x \in C_i} d(x, C_i)^2 \tag{1}$$

where C_i is the ith centroid and x is a data point that belongs to C_i. $d(x, C_i)$ is the distance between x and C_i.

K-means is simple, easy to apply, converges for sure, works on small as well as large datasets. But the quality of clusters depends highly on the value of K and the selected initial centroids. Different values of K and the initial centroids produce different clustering and hence clustering is not unique. Due to these drawbacks, many attempts have been made to solve the initial centroid selection problem. In the related work section, we present some of the popularly used centroid selection methods and also the application of K-means in CFRS.

2.4 SVD

Given any $m \times n$ matrix A, SVD decomposes it into the product of three matrices (Strang, 2006) such that,

$$A = USV^T \tag{2}$$

where U and V both are orthogonal matrices of dimension $m \times m$ and $n \times n$ respectively. U and V being orthogonal matrices are square matrices with orthonormal columns, which means the following:

Let U_i and U_j be any two columns of U, then

$$U_i^T U_j = \begin{cases} 0, \textit{when } i \neq j \\ 1, \textit{when } i = j \end{cases} \tag{3}$$

The same type of relation holds good for the columns of V as well. The columns of U are the eigenvectors of AA^T and that of V are the eigenvectors of A^TA. S 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 S will be zero. SVD has a very long history of theoretical

development and has been independently studied by several mathematicians. However, for rectangular matrices, Eckart and Young first developed the SVD. Later, many practical methods of computing SVD were developed and subsequently many efficient algorithms were proposed.

In recommender systems, SVD has been mostly used for dimensionality reduction and out-of-sample extension. In this article, we attempt to use the natural clustering ability of SVD for K-means initialisation. We take this approach to retain the quality and accuracy of the clusters formed by the clustering algorithm.

3 Related work

Many researchers have tried to solve the K-means initialisation problems in the past and K-means has been popularly used in a CFRS. The related work in connection to the same have been presented in Subsections 3.1 and 3.2 respectively. 'Solving the K-means initialisation problem' is a very popular problem because the quality of clusters produced depend highly on the initial centroid points and outliers may become part of clusters or may form their own clusters. It is found that various methods of K-means initialisation based on distance, density, soft-computing, evolutionary algorithm, variance and subsampling etc. are available in general and the same have been used in case of a CFRS as well. Few instances of the use of SVD is found in connection to K-means initialisation (Maitra, 2009; Zarzour et al., 2018a), but in that SVD is used as a dimensionality reduction method only and further other techniques have been used to find the initial cluster centres. In our approach, we use the result obtained from applying SVD to a rating matrix to determine the number of clusters and the initial centroid points. Thus, assuming that SVD is used for dimensionality reduction, a single method can be used for two purposes which makes the approach more simple, relatable, and understandable.

3.1 K-means initialisation methods

MacQueen (1967) first proposed the K-means algorithm in which he used three parameters, K for the number of clusters, C for partition coarsening or as an upper limit for inter-cluster distance and R for partition refinement or as an upper limit for the intra-cluster distance. The first K points were used as the initial means and it was checked whether the means satisfied the between cluster criteria (C) Any two mean that did not satisfy the criteria were averaged to form a single mean. Then the remaining points were assigned to their closest means such that they satisfied the within-cluster criteria (R) The assignment of points and mean computation continued till stable clusters were formed. Bradley and Fayyad (1998) chose n subsamples from a dataset and run K-means for each of the subsamples. Clusters obtained thus were stored in a set S. Then they applied K-means on S by considering the centroids of each of the subsamples individually. The final clustering was based on the centroids that produced the smallest sum-of-squared error (SSE). Mirkin (2016) adapted a threshold-based approach to find the number of initial centroids. He first computed the distance between each pair of points and selected two farthest-distant points as the first two centroids C_1 and C_2. Then the distance between the rest of the points and C_1 and C_2 were computed and the minimum distant point for each centroid was taken. Between those two points, the point which had the highest magnitude was added to the set of the centroids. This process continued till the number of centroids

was equal to a user-specified threshold value of K. Arthur and Vassilvitskii (2007) devised the K-means++ algorithm, considered to be one of the most efficient K-means method. In their approach first, a point x_i from a dataset X was randomly selected as the first centroid. The next centroid was chosen with a probability of $\frac{D(x_j)^2}{\sum x_j \in (X - x_i)^{D(x_j)^2}}$ where $D(x_j)$ refers to the shortest distance from point x_j to the centroid that have already been chosen. The process continued till K number of points were selected as centroids. Su and Dy (2007) proposed a PCA-based K-means, in which the largest eigenvalue of the covariance matrix of a dataset X was chosen to project X. Then the projected X was split into two clusters at the mean and the within-cluster SSE was calculated. The cluster with the largest SSE was considered for further partitioning. The process continued till K clusters were found. Onoda et al. (2012) proposed an independent component analysis (ICA)-based K-means clustering where a constant c was calculated by using the formula $c = \frac{IC_j * x_i'}{|IC_j||*x_i'|}$ where IC_j was the j^{th} independent component and x_i' was the transpose of the i^{th} data vector x_i. K centroids were chosen in ascending order of c. Katsavounidis et al. (1994) proposed a method known as the KKZ method for K-means. In their formulation, they first calculated the $L_2 - norm$ of all the data vectors and selected the vector with the maximum norm as the first centroid. Then iteratively they computed the distance between the remaining vectors and the selected centroids and chose the vector which is the farthest from the existing centroids as the next centroid till K clusters were found. Hartigan and Wong (1979) proposed a method for initial centroid selection where first they computed the mean of the dataset and the distance between mean and each of the data points. The points were arranged in descending order of their distance from the mean and the i^{th} centroid was chosen as the $\left(1 + \frac{(i-1)n}{K}\right)^{\text{th}}$ point for n number of data points and k clusters. Al-Daoud (2005) came up with a variance-based technique where they first computed the variance of each column in the data matrix. Then they selected the column with the maximum variance and sorted its components. The sorted vector was divided into K groups and data points corresponding to the median of each of the groups were considered as initial centroids. Astrahan (1970) proposed a density-based K-means method where he first computed the average pair-wise distance d between the points. Then the density of each of the points was found by calculating the number of points that lie within a distance of d from the points. Points were sorted in descending order of density and a point having the highest density was chosen as the first centroid. The rest of the centroids were selected iteratively from the sorted list of points in such a way that they were at a minimum distance of d from the existing centroids. Milligan (1980) proposed a centroid selection method in which he first used a hierarchical clustering method to find partitions and then computed the centre of each of these partitions and used them as initial centroids. Steinley (2003) formulated a method that divided a dataset into K clusters and computed the SSE for a user-specified number of iterations. Clustering that minimised the SSE was finally considered, and its centroids were used as the initial centroids. Maitra (2009) proposed an algorithm for K-means initialisation in which he used SVD for dimensionality reduction and considered m significant singular values and the corresponding left singular vectors of U for further processing. Then one-

dimensional K-means was used for each vector of U and an appropriate number of modes were determined for initialisation. Then a product set of the one-dimensional local modes was formed and the candidates which were not close to any observation in U were eliminated. The rest of the modes were used for K-means initialisation. Further single-linkage algorithm on K modes was used to reduce the number of modes to the desired value of K. Huang (1998) extends the classical K-means algorithm and proposed K-modes and K-prototype algorithms for categorical data and mixed data consisting of both categorical and numerical data. The cost function of the K-mode algorithm was designed by using a dissimilarity measure which counts the total number of mismatching values of the corresponding attributes of two objects. The K-Prototype algorithm used a cost function that combined the dissimilarity measure with the Euclidean distance measure. The working of both the algorithms was similar to that of classical K-means. Kettani et al. (2015) proposed a K-means initialisation method where they consider $K = floor\left((n)^{\frac{1}{2}}\right)$ as the number of initial clusters and n is the number of objects. With the selected value of K, K-means is executed and the smallest cluster is removed while the CH cluster validity index of the partition is stored. This process is repeated until K = 2. Finally, partition with the maximum CH index and the corresponding K is generated. Yuan et al. (2004) proposed an algorithm where they first computed the pairwise distance of points and found two points with the least distance. By taking these two points, they form a data point set A_S and delete them from the main dataset. In the next step, the closest point to S is found and is added to S and deleted from the main dataset. The process continued till $|S| = \alpha \times \frac{n}{k}$, where $0 < \propto \leq 1$. If $S < K$, then S is incremented to $S +$ 1 and the entire process continues. For each S, the mean of the points is calculated and taken as initial centroids. Nayak et al. (2017) proposed a centroid rearrangement scheme by giving equal importance to both inter-cluster and intra-cluster distance to achieve consistent clustering output. Nayak et al. (2018) proposed an evolutionary algorithm-based clustering method where the clusters in a dataset are found automatically. Gupta and Chandra (2019) proposed the modified partition based cluster initialisation method for K-means, where each dimension of data is partitioned into k equal-sized partitions and further k partitions are chosen arbitrarily one from each dimension to choose the centroid. Basar et al. (2020) proposed a K-means cluster initialisation method for colour images by using a histogram method and the modes of the histogram. Chowdhury et al. (2020) propose an entropy-based K-means initialisation method my maximising the Shannon entropy. Nidheesh et al. (2017) proposed a density-based K-means clustering, in which initial centroids are selected from dense well-separated regions.

3.2 K-means in CF

Al Mamunur Rashid et al. (2006) used bisecting K-means on the user rating data to build a prediction model for new users. Bisecting K-means started by considering all users as a single cluster and at each iteration pick the largest user cluster to split till K clusters are found. Xue et al. (2005) proposed a scalable CF algorithm in which they used classical K-means by choosing the first K users as the initial centroids. The K clusters thus formed were used for neighbourhood selection of an active user and prediction generation. Ungar and Foster (1998) proposed a repeated clustering for a user-movie rating data matrix

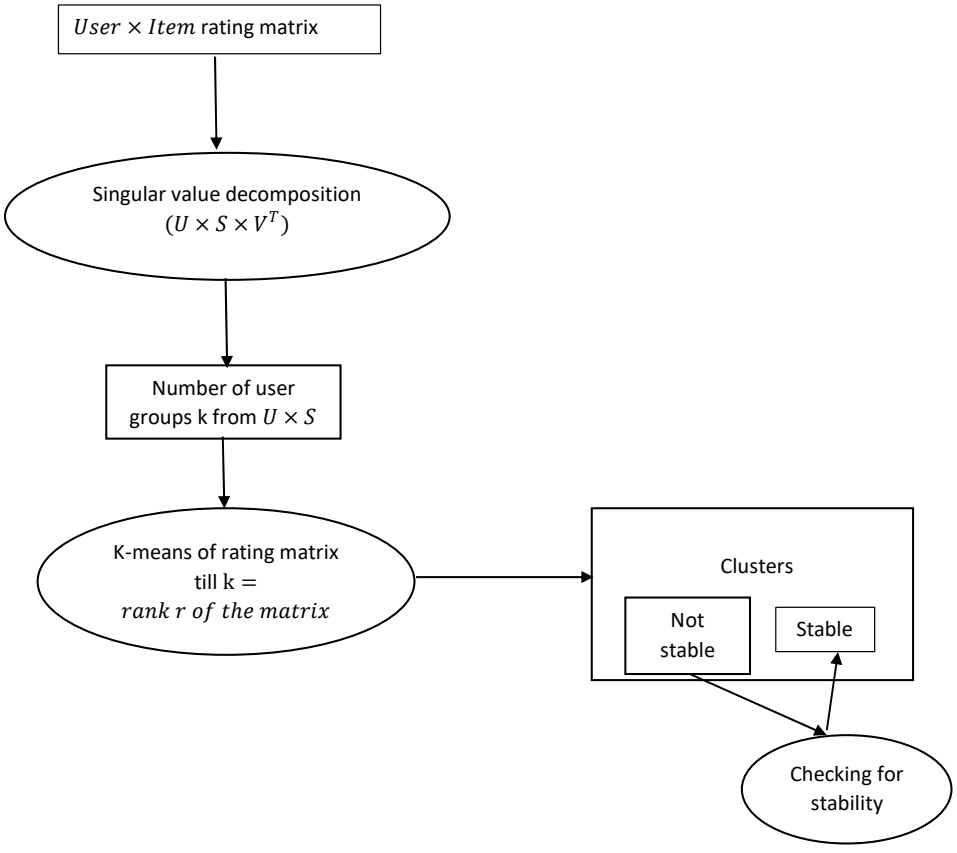
where they used K-means with soft clustering. In the first iteration users were clustered based on their preferences for movies and movies were clustered based on users' preferences. From the second iteration onwards users were clustered based on movie clusters and movies based on user clusters. Considering each user cluster and a movie cluster as a class, they used a soft-clustering approach where each user was assigned to a class with a certain degree of membership proportional to the similarity between a user and the mean of the class. Dakhel and Mahdavi (2011) used K-means to cluster users where they used Minkowski distance as a similarity measure. Then these clusters and a weighted neighbourhood-based approach were used to make predictions for an active user. Chee et al. (2001) proposed a CF method RecTree where they used K-means to partition the data so that the execution time of the collaborative filter can be improved. Ding and Lee (2005) used K-means to cluster similar items for their proposed algorithm that assigned time weight to different items. Kim and Ahn (2008) used a genetic algorithm-based K-means known as GA K-means to cluster customers of a diet shopping mall. They used GA to optimise the initial centroids. Sarwar et al. (2001) used bisecting K-means to cluster the user-item rating matrix and predicted rating by determining the neighbourhood of an active user from the generated clusters in a CF framework. Zhang et al. (2013) proposed a cloud-based CF method known as BiFu where they used K-means to cluster similar users as well as items simultaneously. Zhang et al. (2014) used classical K-means by randomly choosing K training data points as initial seeds to generate user clusters for the MovieLens dataset. Kim et al. (2008) proposed a preference ranking based method and a 'deviation from the average' preference method as pre-processing for user-item data and then applied K-means to cluster users. Zarzour et al. (2018a) used K-means to create user clusters and then for each cluster applied SVD to reduce the dimensionality. Further, they calculated the similarity of users in the reduced space using cosine similarity and generated a recommendation model. Zahra et al. (2015) randomly selected a set of centroids and used K-means clustering to cluster users. Then they used the Pearson correlation coefficient to find similarity between an active user and the centroids and generated recommendation. Li et al. (2019) proposed a canopy-K-means clustering algorithm for CF in which they considered the user-item category preferred data to cluster users and thereby solve the problems of sparsity and scalability. Yassine
et al. (2021) used K-means with demographic attributes for CF and clustered items and users to generate recommendation. Chen et al. (2021) proposed a differentially private user-based CF using K-means address the degradation in recommendation generation. Zarzour et al. (2018b) used K-means with principal component analysis to enhance the performance of traditional CF algorithms. Kumar and Prabhu (2020) developed a hybrid algorithm using K-means and Ant colony optimisation for movie recommender systems. Yuan and Luo (2019) used K-means and a user-based CF approach to generate personalised diet recommendation. Phorasim and Yu (2016) used user-based CF and K-means for movie recommendation. Rahul et al. (2021) proposed a movie recommender system using K-means clustering where they used SVD for dimensionality reduction. Jinchao and Jigang (2020) proposed a CF algorithm based on the SVD and binary K-means where SVD was used as a dimensionality reduction technique and then binary K-means was used to form user clusters. Ifada et al. (2020) studied the working of three MF techniques such as SVD, principal component analysis (PCA) and non-negative matrix factorisation (NMF) with two clustering techniques, K-means and fuzzy C-means

for item recommendation and found that the combination of K-means with any of the MF techniques performed better.

4 Methodology

Figure 2 represents the object process diagram for SVD-initialised K-means clustering for a CFRS.

Figure 2 Object process diagram of SVD-initialised K-means clustering for a CFRS



A *User* × *Item* rating matrix is taken as input and the SVD of the same is done. With the decomposition, three matrices U, S, V^T are obtained. Multiplication of U and S give information about the user groups k and the number of nonzero singular values in the matrix S give the rank of the matrix. Then the K-means of the rating matrix is done till the number of user groups is equal to rank of the matrix. At each step, clustering is checked if it is stable or not. If clustering is stable, then the k value for which the clustering is obtained can be considered as the number of clusters and if not then the previous k value for which stable clusters are formed is considered as the final value for the number of clusters to perform K-means. The definition for a stable cluster is given below.

Stable cluster: If X is a rating matrix and x_i and x_j are any two users belonging to one cluster then for $\forall$ x_i and x_j, they must always belong to the same cluster for any number of iterations of K-means.

5 Result

This section presents the results obtained from the experiments done with some example rating matrices using the MATLAB software. Steps 1 to 6 details the procedure followed in the experiment. Followed by this, the result obtained is presented in detail.

1 Do the SVD of the given rating matrix X by using the MATLAB command [U, S, V] = *svd*(X).

2 Compute $U \times S$ and determine user groups from that. Let the number of user groups found be K.

3 Use K found in step-2 to perform K-means of X using MATLAB command [*idx*, C, *sumd*] = *Kmeans* (X), where *idx* represents the cluster number, C is a matrix containing the centroids and *sumd* is a vector containing the within-cluster distance that is the sum of the distance between a point and its centroid within a cluster, for all the points belonging to that cluster only.

4 Check if the user groups found by SVD is similar to the user clusters found by K-means. A user group is considered similar to a user cluster when the same users that constituted a group in SVD also constitute a cluster in K-means. If both are similar, then SVD can be used to initialise K-means. Also, check if the clusters are stable.

5 But K-means can run for any value of K. So determining a suitable value of K is important.

6 Check the rank of matrix X. If K is less than X then slowly start increasing K up to the rank of the matrix and do K-means for each K value. Check if clustering is stable. If clustering is stable, then this K can be considered as the number of clusters and if not then go back to the previous K value where stable clusters were formed and consider that K as the final value for the number of clusters to perform K-means.

To explain the procedure followed, we present two examples.

Example 1: Let $A = \begin{bmatrix} 5 & 4 & 3 & 0 & 0 \\ 4 & 3 & 2 & 0 & 0 \\ 3 & 4 & 5 & 0 & 4 \\ 0 & 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 2 & 2 \end{bmatrix}$ be the rating matrix.

After doing the SVD of A, find $U \times S$ to be

Figure 3 $U \times S$ matrix of the rating matrix A

$$U \times S = \begin{bmatrix} -6.7490 & -1.3828 & 1.5914 & -0.0771 & 0.0000 \\ -5.0778 & -1.0839 & 1.4126 & 0.2153 & -0.0000 \\ -7.6447 & 1.2143 & -2.4647 & -0.0947 & -0.0000 \\ -0.06629 & 4.3211 & 0.9411 & -0.0541 & -0.0000 \\ -0.3692 & 2.9629 & 0.9939 & -0.3115 & -0.0000 \\ -0.5874 & 2.7164 & -0.1056 & 0.5149 & 0.0000 \end{bmatrix}$$

From the matrix of Figure 3, the user groups are found out by considering the highest value in a row. It is observed that the first three users belong to group-1, indicated by the highest value in column-1 and the last three users belong to group-2 indicated by the highest value in column-2. This indicates that the *user* × *item* rating matrix A has two user groups and therefore we consider K = 2 for K-means. The clustering result obtained by using MATLAB's in-built command [*idx*, *C*, *sumd*] = *Kmeans* (*A*) is shown in Table 1.

Table 1 K-means clustering result for the rating matrix A

Idx					
Users			*Cluster id*		
u_1			2		
u_2			2		
u_3			2		
u_4			2		
u_5			1		
u_6			1		
Centroid: C					
C_1	4	3.6667	3.3333	0	1.1333
C_2	0	0	0	3	1.6667
Sumd					
Cluster-1 (centroid C_1 and users u_1, u_2, u_3)				18.0000	
Cluster-2 (centroid C_2 and users u_4, u_5, u_6,)				2.6667	

With K = 2, K-means was performed multiple numbers of time and in each run, the first three users always belonged to one cluster and the last three users always belonged to another cluster and also the mapping with the centroids and *sumd* remained intact for each user cluster. Thus the clustering result obtained was stable.

Next, the rank of the matrix was checked and it was found to be four. So we considered K = 3 and performed K-means. It was observed that the cluster having users u_1, u_2, u_3 which also has the larger *sumd* got split into two clusters and so with K = 3, we got u_1, u_2 forming a cluster, u_4, u_5, u_6 forming a cluster and u_3 forming a cluster. Most of the time, i.e., in forty runs of the code, 38 times this result was found and twice it was observed that u_1, u_2, u_3 formed one cluster, u_4, u_5 formed a cluster and u_6 formed a cluster.

So the clustering result obtained was stable with an accuracy of 95%. Here. Accuracy is calculated by using the formula in equation (4)

$$Accuracy = \frac{Number\ of\ runs\ for\ which\ the\ clustering\ was\ same}{Total\ number\ of\ runs} \tag{4}$$

Then we considered K = 4 and performed K-means. It was observed that the cluster structure kept on changing very frequently and hence the clustering was unstable.

After running K-means for different K values, it was found that K = 3 can be considered as the number of initial centroids with 95% accuracy and K = 2 can be considered with 100% accuracy.

Example 2: Let $B = \begin{bmatrix} 1 & 2 & 3 & 0 & 0 \\ 2 & 4 & 6 & 0 & 0 \\ 3 & 6 & 9 & 2 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 6 & 3 \\ 0 & 0 & 0 & 10 & 5 \\ 0 & 0 & 0 & 10 & 5 \end{bmatrix}$ be another rating matrix, where a rating scale of 1 to 10 is used, unlike the previous example where a rating scale of 1 to 5 is used. After doing the SVD of B, we find $U \times S$ to be

Figure 4 $U \times S$ matrix of the rating matrix B

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

In the matrix of Figure 4, it is observed that the first four users belong to group-2, indicated by the highest value in column-2 and the last three users belong to group-1 indicated by the highest value in column-1. This indicates that the *user* × *item* rating matrix B has two user groups and therefore we consider K = 2 for K-means. The clustering result obtained by using MATLAB's in-built command [*idx*, *C*, *sumd*] = *K-means* (*B*) is shown in Table 2.

Table 2 K-means clustering result for the rating matrix *B*

Idx					
Users			*Cluster id*		
u_1			2		
u_2			2		
u_3			2		
u_4			2		
u_5			1		
u_6			1		
u_7			1		
Centroid: C					
C_1	0	0	0	8.6667	4.3333
C_2	1.5	3.25	4.75	0.5	0.25
Sumd					
Cluster-1 (centroid C_1 and users u_5, u_6, u_7)				13.3333	
Cluster-2 (centroid C_2 and users u_1, u_2, u_3, u_4)				60.2500	

With K = 2, K-means was performed multiple numbers of time and in each run, the first four users always belonged to one cluster and the last three users always belonged to another cluster and also the mapping with the centroids and *sumd* remained intact for each user cluster. Thus, the clustering result obtained was stable.

Next, the rank of the matrix was checked, and it was found to be three. So, we considered K = 3 and performed K-means. It was observed that the cluster having users u_1, u_2, u_3, u_4, which also has the larger *sumd* got split into two clusters and so with K = 3, we got u_1,u_4 forming a cluster, u_2, u_3 forming a cluster and u_5, u_6, u_7 forming a cluster. In multiple runs of K-means, the result obtained was the same and also the mapping with the centroids and *sumd* remained intact for each user cluster. Thus, the clustering result obtained was stable. Although the rank of the matrix is three, still we run K-means with K = 4 to investigate the behaviour of K-means and we found that the clustering obtained was unstable. Since there are seven users K can be increased up to seven and K-means can be performed but as we keep increasing K, more and more numbers of trivial clusters get formed and clustering also is unstable. For example, with K = 7, each point forms one cluster. With K value greater than the number of users, the program gives an error.

After running K-means for different K values, it was found that K = 3 produced stable clustering and hence for the rating matrix *B*, K = 3 can be considered as the number of initial centroids with 100% accuracy.

6 Proposed SVD-initialised K-means clustering algorithm

Based on the discussions in Sections 4 and 5, we propose the SVD-initialised K-means clustering algorithm.

6.1 Algorithm: SVD-initialised K-means

Input: A rating matrix X with m users and n items.

Output: User clusters for the m users.

1 Do the SVD of the rating matrix X: $X = USV^T$

2 Compute $U \times S$.

3 Determine the number of user groups from $U \times S$ and let it be the number of initial centroids K for K-means. Compute the mean of each of these groups and let them be initial centroid points stored in the set C.

4

a Run classical K-means until clustering does not change.

b Update C, the set of cluster centroids. Save clustering results.

5 While (number of clusters < rank of the rating matrix) {

if (*sumd* of a cluster ≥ l smallest *sumd*)

a Split the cluster into two parts and select the mean of each part as a centroid.

b Update K: K = number of clusters unsplit + 2 × number of clusters split and C.

else

Go to step-8.

6 Run classical K-means and go to step 4(b) to update C and save clustering results.

7 Go to step-5.

}

8 Consider the clustering of step 4 (b) as final and yield clusters.

7 Discussion

CFRS is popularly used by many e-commerce companies and online merchants because they help in profit generation and customer satisfaction through their recommendations. Clustering methods make a recommender system efficient because in clustering a new user is always compared with the representative vectors of a cluster instead of the entire dataset for recommendation generation. Among many available clustering algorithms, K-means is most popular because it is simple to implement and converges for sure. The related work presented in Section 3 reflects the popularity of K-means for CFRS applications. However, K-means suffers from the centroid initialisation problem and many algorithms have been formulated to address this issue. In this work, we propose a new algorithm for K-means initialisation in a CFRS setting using SVD which we discuss briefly. After finding the SVD of the data matrix, the product matrix $U \times S$ is computed and user groups are determined. The number of user groups is considered as the initial value of K and their respective means are taken as the initial centroid points for K-means clustering. Then classical K-means is run till clustering does not change and the set of

centroids is updated, and the clustering result is saved at step 4(b) of algorithm 6.1. The while loop continues to run until the loop condition is satisfied. If the loop condition is satisfied, control moves to the following if…else block. Inside the if…else block, first, the if condition is checked and upon satisfaction the statements 5(a) and 5(b) are executed. Then statement 6 is executed where classical K-means is run and the program control moves to step 4(b) where C is updated and the clustering result gets saved. After that control again moves to step-5 and the condition is checked. At any stage whenever the while loop and the if condition both are satisfied, the algorithm works in the above-discussed manner. In case the while loop condition satisfies but the if condition does not satisfy, control moves to else statement where it goes to step-8. Similarly, if the while loop condition becomes false then control comes out of the loop and goes to step-8. Next we discuss about the significance of considering the rank of a rating matrix to restrict the number of clusters and the significance of considering the within cluster distance or *sumd* for cluster splitting which are important considerations and challenges in the algorithm formulation. The rank of a matrix is equal to the number of nonzero singular values in the SVD of a rating matrix. The nonzero singular values determine the number of categories along which the behaviour of the users and items can vary. Singular values are the square root of the eigenvalues of the associated Gram matrix A^TA of any rating matrix A. The eigenvalues are also known as the characteristics values because they capture the hidden characteristics of a dataset. Since the singular values are closely associated with the eigenvalues, they also capture the hidden characteristics of a data matrix and their categorisation is based on the characteristics of a dataset. Further, in a CF setting, users are recommended items based on the behaviour of the '*likeminded user*'. Likeminded users are those who have same kind of choice or similar type of taste, and this represents a character of the users which is captured by the singular values. Since rank and singular values are the same and SVD is used to solve the K-means initialisation problem, the maximum number of clusters that can be formed can be equal to the rank of the matrix. Hence we incorporated this constraint in the algorithm design. In the sample rating matrices when K-means was performed with an increased value of K after considering K as the number of user groups from the SVD, it was observed that the clusters with bigger *sumd* got split. However, in some of the matrices, unnecessary splitting was happening and, in those matrices, the difference between *sumd* of clusters was not that higher. The example of such a rating matrix is $C = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix}$ whose rank is three and SVD produces two user groups and K-means with K = 2 produces two clusters with *sumd* 26.25 and 11.33 respectively. The nature of K-means clustering is to minimise the within-cluster distance (Dey et al., 2017), but it is also true that K-means can run for any number of K provided K < n, where n is the number of observations. The real challenge is to be able to find clusters such that the user clusters retain the characteristics of the users as much as possible because it is the basis of recommendation generation. A restriction on *sumd* helps to take care of these conflicting interests so that unnecessary splitting of

clusters as well as unnecessary mixing of users from two different user groups does not take place and more refined clusters are obtained. Based on our observation, we formulated the condition for the if statement as (*sumd* of a cluster ≥ $l \times$ smallest *sumd*) in Algorithm 6.1, where l is a user-defined constant.

The proposed algorithm performs SVD once. The time complexity of performing SVD for an $m \times n$ data matrix is $O(mn \times mim(m, n))$ that is it is either $O(mn^2)$ or $O(m^2n)$ Further, the algorithm performs K-means for a maximum number of T times where $T = 1 +$ (*rank – number of initial clusters*) The time complexity of K-means is $O(IKmn)$ where I is the number of iterations required for convergence. Since performing SVD takes quadratic time and K-means takes linear time, the complexity of the proposed algorithm is quadratic.

8 Conclusions

In this article, we have attempted to use SVD for K-means initialisation in a CFRS data matrix. The purpose of clustering in a CFRS is to cluster users or items so that neighbourhood formation becomes easy and based on the clusters formed recommendation can be generated efficiently. But this efficiency comes by compromising with quality and accuracy. The use of SVD for K-means initialisation helps to retain the cluster quality and accuracy. SVD forms user groups where users having similar taste form a group and thus retains the inherent characteristics of the users. When this knowledge is used to initialise K-means not only the quality of clusters is retained but also the cluster initialisation process gets automated. In our proposed method it is not required to specify the number of clusters K and the initial centroid points as input. To accomplish the task a new algorithm has been formulated. The algorithm eventually discovers the actual number of user clusters from the knowledge obtained from user groups in SVD and further refines the clustering result by using the rank of the matrix and within-cluster distance criteria traditionally used by the K-means clustering algorithm. The proposed approach discovers clusters based on the characteristics of the dataset which is most important for recommendation generation and takes advantage of the distance criteria for refinement and hence is expected to generate a better recommendation. However, every method has its own limitation. SVD being a MF technique can be applied to matrices which have no missing entries. But recommender system matrices suffer from the problem of having too many missing values in a rating matrix. Thus, before applying a MF technique like SVD, the missing values need to be handled and the rating matrices need to be completed. Another limitation of this approach is the quadratic time complexity because a simple application of K-means with random centroids can produce clusters in linear time. However, for accurate recommendation generation SVD is used along with K-means. Thus, the proposed method is most suitable for applications where large rating matrices are involved and dimensionality reduction is essential before clustering. The algorithm that is proposed here is for user-based CF, however the same approach can also be applied to item based CFRS. In the future, we want to investigate the working of the method on other partitional clustering algorithms such as K-medoid or partitioning around medoid (PAM) and NMF which is a MF method but is very famous for its clustering capabilities. Also, the method can be applied to some of the benchmark datasets used in recommender systems and its performance can be studied.

References

Aggarwal, C.C. (2016) *Recommender Systems*, Vol. 1, Springer International Publishing, Cham.

Al Mamunur Rashid, S.K.L., Karypis, G. and Riedl, J. (2006) 'ClustKNN: a highly scalable hybrid model-& memory-based CF algorithm', *Proceeding of webKDD*.

Al-Daoud, M.D.B. (2005) 'A new algorithm for cluster initialization', in *WEC: The 2nd World Enformatika Conference*.

Arthur, D. and Vassilvitskii, S. (2007) 'K-means++: the advantages of careful seeding', in *SODA'07: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, pp.1027–1035.

Astrahan, M.M. (1970) *Speech Analysis by Clustering, or the Hyperphoneme Method*, No. AIM-124, Stanford Univ. Dept. of Computer Science, CA.

Basar, S., Ali, M., Ochoa-Ruiz, G., Zareei, M., Waheed, A. and Adnan, A. (2020) Unsupervised color image segmentation: a case of RGB histogram based K-means clustering initialization', *Plos One*, Vol. 15, No. 10, p.e0240015.

Bradley, P.S. and Fayyad, U.M. (1998) 'Refining initial points for K-means clustering', in *ICML*, July, Vol. 98, pp.91–99.

Chee, S.H.S., Han, J. and Wang, K. (2001) 'Rectree: an efficient collaborative filtering method', in *International Conference on Data Warehousing and Knowledge Discovery*, September, pp.141–151, Springer, Berlin, Heidelberg.

Chen, Z., Wang, Y., Zhang, S., Zhong, H. and Chen, L. (2021) 'Differentially private user-based collaborative filtering recommendation based on K-means clustering', *Expert Systems with Applications*, Vol. 168, p.114366.

Chowdhury, K., Chaudhuri, D. and Pal, A.K. (2020) 'An entropy-based initialization method of K-means clustering on the optimal number of clusters', *Neural Computing and Applications*, pp.1–18.

Dakhel, G.M. and Mahdavi, M. (2011) 'A new collaborative filtering algorithm using K-means clustering and neighbors 'voting', in *11th International Conference on Hybrid Intelligent Systems (HIS)*, IEEE, December, pp.179–184.

Dey, N. (2016) (Ed.) *Classification and Clustering in Biomedical Signal Processing*, IGI Global, Hershey, PA.

Dey, N., Ashour, A.S. and Borra, S. (2017) (Eds.): *Classification in BioApps: Automation of Decision Making*, Vol. 26, Springer, Cham.

Ding, Y. and Li, X. (2005) 'Time weight collaborative filtering', in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, October, pp.485–492.

Gupta, M.K. and Chandra, P. (2019) 'MP-K-means: modified partition based cluster initialization method for K-means algorithm', *Int. J. Recent Technol. Eng.*, Vol. 8, No. 4, pp.1140–1148.

Hartigan, J.A. and Wong, M.A. (1979) 'Algorithm AS 136: a K-means clustering algorithm', *Journal of the Royal Statistical Society.Series C (Applied Statistics),* Vol. 28, No. 1, 100–108.

Huang, Z. (1998) 'Extensions to the K-means algorithm for clustering large data sets with categorical values', *Data Mining and Knowledge Discovery*, Vol. 2, No. 3, 283–304.

Ifada, N., Sophan, M.K., Fitriantama, M.N. and Wahyuni, S. (2020) 'Collaborative filtering item recommendation methods based on matrix factorization and clustering approaches', in *10th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*, IEEE, August, pp.226–230.

Jabeen, A., Ahmad, N. and Raza, K. (2018) 'Machine learning-based state-of-the-art methods for the classification of RNA-seq data', in *Classification in BioApps*, pp.133–172, Springer, Cham.

Jinchao, G.U.O. and Jigang, Y.A.N.G. (2020) 'Collaborative filtering algorithm based on the improved SVD algorithm and binary K-means clustering algorithm', *Journal of Light Industry*, Vol. 35, No. 4, pp.88–95.

Katsavounidis, I., Kuo, C.C.J. and Zhang, Z. (1994) 'A new initialization technique for generalized Lloyd iteration', *IEEE Signal Processing Letters*, Vol. 1, No. 10, pp.144–146.

Kettani, O., Ramdani, F. and Tadili, B. (2015) 'AK-means: an automatic clustering algorithm based on K-means', *Journal of Advanced Computer Science & Technology*, Vol. 4, No. 2, p.231.

Kim, K.J. and Ahn, H. (2008) 'A recommender system using GA K-means clustering in an online shopping market', *Expert Systems with Applications*, Vol. 34, No. 2, pp.1200–1209.

Kim, T.H., Park, S.I. and Yang, S.B. (2008) 'Improving prediction quality in collaborative filtering based on clustering', in *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, IEEE, December, Vol. 1, pp.704–710.

Kumar, M.S. and Prabhu, J. (2020) 'A hybrid model collaborative movie recommendation system using K-means clustering with ant colony optimisation', *International Journal of Internet Technology and Secured Transactions*, Vol. 10, No. 3, pp.337–354.

Li, J., Zhang, K., Yang, X., Wei, P., Wang, J., Mitra, K. and Ranjan, R. (2019) 'Category preferred canopy–K-means based collaborative filtering algorithm', *Future Generation Computer Systems*, Vol. 93, pp.1046–1054.

Linden, G., Smith, B. and York, J. (2003) 'Amazon.com recommendations: Item-to-item collaborative filtering', *IEEE Internet Computing*, Vol. 7, No. 1, pp.76–80.

MacKenzie, I., Meyer, C. and Noble, S. (2013) *How Retailers Can Keep Up with Consumers*, Vol. 18, McKinsey & Company, USA.

MacQueen, J. (1967) 'Some methods for classification and analysis of multivariate observations', in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, June, Vol. 1, No. 14, pp.281–297.

Maitra, R. (2009) 'Initializing partition-optimization algorithms', *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 6, No. 1, pp.144–157.

Manouselis, N. (2008) 'Deploying and evaluating multiattribute product recommendation in e-markets', *International Journal of Management and Decision Making*, Vol. 9, No. 1, pp.43–61.

Milligan, G.W. and Isaac, P.D. (1980) 'The validation of four ultrametric clustering algorithms', *Pattern Recognition*, Vol. 12, No. 2, pp.41–50.

Mirkin, B. (2016) *Clustering: A Data Recovery Approach*, Chapman and Hall/CRC.Boca Raton FL, USA.

Nath, S.S., Mishra, G., Kar, J., Chakraborty, S. and Dey, N. (2014) 'A survey of image classification methods and techniques', in *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT),* IEEE, July, pp.554–557.

Nayak, S.K., Rout, P.K. and Jagadev, A.K. (2018) 'Automatic clustering by elitism-based multi-objective differential evolution', *International Journal of Management and Decision Making*, Vol. 17, No. 1, pp.50–74.

Nayak, S.K., Rout, P.K., Jagadev, A.K. and Patnaik, S. (2017) 'A modified differential evolution-based fuzzy multi-objective approach for clustering', *International Journal of Management and Decision Making*, Vol. 16, No. 1, pp.24–49.

Nidheesh, N., Nazeer, K.A. and Ameer, P.M. (2017) 'An enhanced deterministic K-means clustering algorithm for cancer subtype prediction from gene expression data', *Computers in Biology and Medicine*, Vol. 91, pp.213–221.

Onoda, T., Sakai, M. and Yamada, S. (2012) 'Careful seeding method based on independent components analysis for K-means clustering', *Journal of Emerging Technologies in Web Intelligence*, Vol. 4, No. 1, pp.51–59.

Phorasim, P. and Yu, L. (2017) 'Movies recommendation system using collaborative filtering and K-means', *International Journal of Advanced Computer Research*, Vol. 7, No. 29, p.52.

Rahul, M., Kumar, V. and Yadav, V. (2021) 'Movie recommender system using single value decomposition and K-means clustering', in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, Vol. 1022, No. 1, p.012100.

Sarwar, B.M., Karypis, G., Konstan, J. and Riedl, J. (2002) 'Recommender systems for large-scale e-commerce: scalable neighborhood formation using clustering', in *Proceedings of the 5th International Conference on Computer and Information Technology*, December, Vol. 1, pp.291–324.

Steinley, D. (2003) 'Local optima in K-means clustering: what you don't know may hurt you', *Psychological Methods*, Vol. 8, No. 3, p.294.

Strang, G. (2006) *Linear Algebra and its Applications (1988),* Cengage Learning, Harcourt Brace Jovanovich, San Diego.

Su, T. and Dy, J.G. (2007) 'In search of deterministic methods for initializing K-means and Gaussian mixture clustering', *Intelligent Data Analysis*, Vol. 11, No. 4, pp.319–338.

Ungar, L.H. and Foster, D.P. (1998) 'Clustering methods for collaborative filtering', in *AAAI Workshop on Recommendation Systems*, July, Vol. 1, pp.114–129.

Xue, G.R., Lin, C., Yang, Q., Xi, W., Zeng, H.J., Yu, Y. and Chen, Z. (2005) 'Scalable collaborative filtering using cluster-based smoothing', in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, August, pp.114–121.

Yassine, A.F.O.U.D.I., Mohamed, L.A.Z.A.A.R. and Al Achhab, M. (2021) 'Intelligent recommender system based on unsupervised machine learning and demographic attributes', *Simulation Modelling Practice and Theory*, Vol. 107, p.102198.

Yuan, F., Meng, Z.H., Zhang, H.X. and Dong, C.R. (2004) 'A new algorithm to get the initial centroids', in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, IEEE, August, Vol. 2, Cat. No. 04EX826, pp.1191–1193.

Yuan, Z. and Luo, F. (2019) 'Personalized diet recommendation based on K-means and collaborative filtering algorithm', in J*ournal of Physics: Conference Series*, IOP Publishing, June, Vol. 1213, No. 3, p.032013.

Zahra, S., Ghazanfar, M.A., Khalid, A., Azam, M.A., Naeem, U. and Prugel-Bennett, A. (2015) 'Novel centroid selection approaches for KMeans-clustering based recommender systems', *Information Sciences*, Vol. 320, pp.156–189.

Zarzour, H., Al-Sharif, Z., Al-Ayyoub, M. and Jararweh, Y. (2018a) 'A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques', in *9th International Conference on Information and Communication Systems (ICICS),* IEEE, April, pp.102–106.

Zarzour, H., Maazouzi, F., Soltani, M. and Chemam, C. (2018b) 'An improved collaborative filtering recommendation algorithm for big data', in *IFIP International Conference on Computational Intelligence and Its Applications*, Springer, Cham, May, pp.660–668.

Zhang, C.X., Zhang, Z.K., Yu, L., Liu, C., Liu, H. and Yan, X.Y. (2014) 'Information filtering via collaborative user clustering modeling', *Physica A: Statistical Mechanics and its Applications*, Vol. 396, pp.195–203.

Zhang, D., Hsu, C.H., Chen, M., Chen, Q., Xiong, N. and Lloret, J. (2013) 'Cold-start recommendation using bi-clustering and fusion for large-scale social recommender systems', *IEEE Transactions on Emerging Topics in Computing*, Vol. 2, No. 2, pp.239–250.