Journal of Ambient Intelligence and Humanized Computing (2023) 14:15875–15883
https://doi.org/10.1007/s12652-021-03128-6

ORIGINAL RESEARCH

Collaborative filtering with q-divergence-based fuzzy clustering for spherical data

Yuchi Kanzawa[1] · **Tadafumi Kondo**[2]

Received: 15 February 2020 / Accepted: 8 March 2021 / Published online: 28 March 2021

Abstract
Although recommendation systems are the most powerful tool to help people choose items, a higher recommendation accuracy is required to satisfy the needs of the people. Motivated by this requirement, this study proposes a novel collaborative filtering (CF) algorithm, which is the underlying technology of a recommendation system. It filters items for a target user based on the reactions of similar users. Cluster analysis helps detect similar users by grouping a set of users such that users in the same group are more similar to each other than to those in other groups. However, in most representative CF algorithms such as GroupLens algorithm, users are considered as spherical data, and as categorical multivariate data in the clustering phase of a previous study. This study overcomes this logic gap by proposing a novel CF method using fuzzy clustering for spherical data based on q-divergence as both the clustering phase and the GroupLens algorithm consistently deal with users as spherical data. Experiments were conducted on six real datasets—BookCrossing, Epinions, Jester, LibimSeTi, MovieLens, and SUSHI, to compare the performance of the proposed method with GroupLens and the method using fuzzy clustering for categorical multivariate data based on q-divergence, which are conventional methods, where the performance is measured by the area under the receiver operating curve. The results of the experiments indicate that the proposed algorithm outperforms the others in terms of recommendation accuracy.

Keywords Recommendation system · Collaborative filtering · Fuzzy clustering for spherical data · Q-divergence

1 Introduction

Currently, there exists a considerable amount of information on digital platforms, thus, it is extremely difficult to select information that is truly relevant to each user. Recommender systems are the most powerful tool to help people in choosing products, activities, and friends from many representative options. Although recommendation systems such as Amazon.com have been ubiquitous, their recommendation accuracy is not sufficient to satisfy the growing needs of the people. This study is motivated by the requirement of higher accuracy of recommendation system.

Among many techniques combined in recommender systems, collaborative filtering (CF) is the most fundamental technique (Paul et al. 1994; Sarwar et al. 2001), which can filter items (products, activities, or friends) that a user may like based on the preferences of similar users. The most representative CF method is GroupLens (Herlocker et al. 1999), which is simple and time efficient. However, the similarity of "similar users" is heuristically defined. An adequate definition of similarity can help CF suggest more appropriate items to users. We consider that users implicitly belong to a latent group, where users have similar preferences in the same group. If we can determine such groups, we can determine users similar to a target user, and then, CF can help suggest items to the target user based on the preferences of similar users.

Clustering is only a technique to detect latent groups. Many clustering methods have been proposed and applied based on the type of given data. Honda (2016) suggested applying fuzzy clustering for categorical multivariate data induced by multinomial mixture models (FCCMM), which is based on a cluster-wise bag-of-words concept. Kondo and

✉ Yuchi Kanzawa
kanzawa@sic.shibaura-it.ac.jp

Tadafumi Kondo
ma17049@shibaura-it.ac.jp

[1] Shibaura Institute of Technology Toyosu Campus, 3-7-5 Toyosu, Koto, Tokyo, Japan

[2] Mitsubishi Electric Corporation, 2-7-3 Marunouchi, Chiyoda, Tokyo, Japan

Kanzawa (2018) modified the FCCMM algorithm, referred to as q-divergence-based fuzzy clustering for categorical multivariate data induced by multinomial mixture models (QFCCMM). The q-divergence was focused because it is not only a generalization of the standard Kullback-Leibler divergence used in FCCMM but also the divergence discussed in Tsallis statistics with which the predictions and consequences in a wide spectrum of complex systems were confirmed (Tsallis 2009). Utilizing the q-divergence instead of the Kullback-Leibler one in clustering task, there was a potential that clusters could be captured adequately, and actually QFCCMM achieved higher clustering accuracy than FCCMM (Kondo and Kanzawa 2018). Furthermore, in a previous study (Kondo and Kanzawa 2019), we proposed applying the QFCCMM as a preparatory step of GroupLens for CF tasks and indicated that the QFCCMM-based CF algorithm outperforms not only the GroupLens algorithm but also the FCCMM-based CF algorithm.

A clustering method should be applied based on the given data type. FCCMM (Honda et al. 2015) and QFCCMM (Kondo and Kanzawa 2018) were proposed originally for categorical multivariate data, such as document data. In the case of applying FCCMM or QFCCMM for the CF task, we consider the vector of rating of items given by a user (rating vector) as categorical multivariate data. On the other hand, GroupLens does not deal with the rating vector as categorical multivariate data. Pearson's coefficient used in GroupLens focuses on the directions of the users' items rating vectors instead of their magnitudes. Since the users' items rating vectors are made of uniform magnitude, they are on the unit hypersphere with the dimension of items. In other words, GroupLens deals with the rating vector as spherical data. Therefore, there is a logic gap that users are considered as categorical multivariate data in the QFCCMM-based CF algorithm, whereas they are considered as spherical data in the GroupLens algorithm. There is a need to design a clustering method for spherical data that has the potential to solve this logic gap. In a previous study, Higashi et al. proposed q-divergence-based fuzzy clustering for spherical data, referred to as QFCS (Higashi et al. 2019), and demonstrated that the proposed clustering algorithm achieved higher clustering accuracy using several document datasets. Although QFCS is worth applying to not only clustering documents but also to clustering rating vectors for CF tasks, it was not applied in the literature.

In this study, we propose a CF algorithm with the help of QFCS clustering algorithm. First, for all unevaluated elements in the given rating matrix, the lowest value among all already evaluated values is tentatively set. Subsequently, all values are normalized such that all users' items rating vectors are on the unit hypersphere. Second, the QFCS algorithm segments the users' items rating vectors into some clusters. Third, the GroupLens algorithm is applied for each users' items rating cluster. Finally, every item is recommended if the corresponding estimated rating value is higher than the predefined cut-off value. Through numerical experiments using six real datasets, the results of the proposed method are compared with those of two counter candidates (GroupLens and the QFCCMM-based algorithm). The experimental results indicate that the proposed algorithm performs better than the two in terms of recommendation accuracy.

The remainder of this paper is organized as follows: Sect. 2 introduces a representative CF algorithm, GroupLens; a clustering-based CF algorithm from our previous work, QFCCMM-based CF algorithm; and fuzzy clustering algorithm for spherical data, QFCS algorithm. Section 3 presents the proposed CF algorithm. Section 4 presents numerical experiments, and the conclusion is presented in Sect. 5.

2 Preliminaries

2.1 Conventional collaborative filtering method: GroupLens

The most frequently used CF algorithms are based on the concept of "neighborhood" (Herlocker et al. 1999), wherein a user's neighbor is selected based on the preference of the target user, and then, the latent preferences of the target user are estimated from the preferences of the target users' neighbor.

Let N and M be the number of users and items, respectively. Let $x_{k,\ell}(\geq 0)$ $(k \in \{1, \dots, N\}, \ell \in \{1, \dots, M\})$ be the rating value that the user #k evaluated the item #ℓ. The matrix whose (k, ℓ)-th element value is $x_{k,\ell}$ is denoted by X. Since all users do not always evaluate all items, some elements of X are missing. Then, the goal of CF is to estimate such missing values. Let $y_{k,\ell} \in \{0, 1\}$ be the indicator whether the user #k evaluated the item #ℓ, and it is defined as

$$y_{k,\ell} = \begin{cases} 1 & (\text{The user } \#k \text{ evaluated the item } \#\ell), \\ 0 & (\text{The user } \#k \text{ has not evaluated the item } \#\ell). \end{cases} \tag{1}$$

Y denotes the matrix for which the (k, ℓ)-th element value is $y_{k,\ell}$. Let $\Omega_{\text{item}}(k)$ be the set of items which the users #k evaluated. Let $\text{sim}(k, k')$ be the similarity measure between the target user #k and the user #k' neighbor to the target user. The similarity measure $\text{sim}(k, k')$ is defined by Pearson's correlation coefficient using the rating values of items that both users #k and #k' have evaluated, as described by

$$\text{sim}(k, k') = \frac{\sum_{\ell \in \Omega_{\text{item}}(k) \cap \Omega_{\text{item}}(k')} \left(x_{k,\ell} - \overline{x_{k,\cdot}^{k'}}\right)\left(x_{k',\ell} - \overline{x_{k',\cdot}^{k}}\right)}{\sqrt{\sum_{\ell \in \Omega_{\text{item}}(k) \cap \Omega_{\text{item}}(k')} \left(x_{k,\ell} - \overline{x_{k,\cdot}^{k'}}\right)^2} \sqrt{\sum_{\ell \in \Omega_{\text{item}}(k) \cap \Omega_{\text{item}}(k')} \left(x_{k',\ell} - \overline{x_{k',\cdot}^{k}}\right)^2}}, \tag{2}$$

where $\overline{x_{k,\cdot}^{k'}}$ is the mean rating value of the user #k for items that both users #k and #k' evaluated, described as

$$\overline{x_{k,\cdot}^{k'}} = \frac{\sum\limits_{\ell \in \Omega_{\mathrm{item}}(k) \cap \Omega_{\mathrm{item}}(k')} x_{k,\ell}}{|\Omega_{\mathrm{item}}(k) \cap \Omega_{\mathrm{item}}(k')|}. \tag{3}$$

If $\Omega_{\mathrm{item}}(k) \cap \Omega_{\mathrm{item}}(k')$ is empty, $\mathrm{sim}(k,k')$ is set to zero. Let $\hat{x}_{k,\ell}$ be the missing value for the item #ℓ, which the target user #k has not evaluated, and let $\bar{x}_{k,\cdot}$ be the mean rating value of the user #k for items that the user #k evaluated, as

$$\bar{x}_{k,\cdot} = \frac{\sum_{\ell \in \Omega_{\mathrm{item}}(k)} x_{k,\ell}}{|\Omega_{\mathrm{item}}(k)|}.$$

The GroupLens method (Herlocker et al. 1999) estimates the unknown rating value $\hat{x}_{k,\ell}$ of the target user #k such that the deviance between $\hat{x}_{k,\ell}$ and $\bar{x}_{k,\cdot}$ is Pearson's correlation coefficient-weighted mean of the deviance between $\hat{x}_{k',\ell}$ and $\bar{x}_{k',\cdot}$, where #k' represents every user with a positive correlation for the target user #k. Then, the estimated rating value $\hat{x}_{k,\ell}$ of the target user #k is described as

$$\hat{x}_{k,\ell} = \bar{x}_{k,\cdot} + \frac{\sum_{\substack{k' \in \Omega_{\mathrm{user}}(\ell) \\ \mathrm{sim}(k,k') \geq 0}} \mathrm{sim}(k,k')(x_{k',\ell} - \bar{x}_{k',\cdot})}{\sum_{\substack{k' \in \Omega_{\mathrm{user}}(\ell) \\ \mathrm{sim}(k,k') \geq 0}} \mathrm{sim}(k,k')}, \tag{4}$$

where $\Omega_{\mathrm{user}}(\ell)$ is the set of users who evaluated the item #ℓ. If there is no user #k' satisfying both $k' \in \Omega_{\mathrm{user}}(\ell)$ and $\mathrm{sim}(k,k') \geq 0$ for the target user #k, $\hat{x}_{k,\ell}$ in Eq. (4) is just $\bar{x}_{k,\cdot}$.

The GroupLens algorithm is summarized as [GroupLens]

Step 1. Obtain the similarities among users according to their preferences as Eq. (2).

Step 2. Estimate the missing values $\hat{x}_{k,\ell}$ ($k \in \{1,\dots,N\}$, $\ell \in \{1,\dots,M\}$) if $y_{k,\ell} = 0$, as Eq. (4). □

2.2 QFCCMM-based CF (Kondo and Kanzawa 2019)

In the GroupLens method, similar users #k' ($k' \in \{1,\dots,N\}$) to the target user #k are heuristically defined as those satisfying $\mathrm{sim}(k,k') \geq 0$, in Eq. (4). Note that there is theoretical basis for this definition, and there exist many ways to define similar users to the target user. We focus on clustering users based on their preferences. Kondo and Kanzawa proposed the QFCCMM (Kondo and Kanzawa 2018) algorithm, as follows. Let $X = \{x_k \in \mathbb{R}^M \mid k \in \{1,\dots,N\}, x_{k,\ell} \geq 0, \ell \in \{1,\dots,M\}\}$ be a categorical multivariate dataset, where $x_{k,\ell}$ represents co-occurrence relations between the k-th user and the ℓ-th item. The membership of x_k to the i-th cluster is denoted by $u_{i,k}$ ($i \in \{1,\dots,C\}, k \in \{1,\dots,N\}$), and the set of $u_{i,k}$ is denoted by U. U obeys the constraint

$$\sum_{i=1}^{C} u_{i,k} = 1, \quad (k \in \{1,\dots,N\}). \tag{5}$$

The typicality of the ℓ-th item for the i-th cluster is denoted by $w_{i,\ell}$ ($i \in \{1,\dots,C\}, \ell \in \{1,\dots,M\}$); the set of $w_{i,\ell}$ is denoted by w, which obeys the constraint

$$\sum_{\ell=1}^{M} w_{i,\ell} = 1 \text{ and } w_{i,\ell} \in [0,1], (i \in \{1,\dots,C\}). \tag{6}$$

The variable controlling the i-th cluster size is denoted by π_i. The i-th element of vector π is denoted by π_i, which obeys the following constraint:

$$\sum_{i=1}^{C} \pi_i = 1. \tag{7}$$

The QFCCMM algorithm is obtained by solving the optimization problem

$$\begin{aligned} &\underset{U,w,\pi}{\mathrm{maximize}} \sum_{i=1}^{C}\sum_{k=1}^{N}\sum_{\ell=1}^{M} (\pi_i)^{1-q}(u_{i,k})^q \frac{1}{t}\left(\left(w_{i,\ell}\right)^t - 1\right)x_{k,\ell} \\ &-\frac{\lambda^{-1}}{q-1}\left(\sum_{i=1}^{C}\sum_{k=1}^{N}(\pi_i)^{1-q}(u_{i,k})^q - 1\right) \end{aligned} \tag{8}$$

subject to Eqs. (5), (6), and (7), where (q, λ, t) are the fuzzification parameters satisfying $q > 1$, $\lambda > 0$, and $t > 0$. This method is named "q-divergence-based fuzzy clustering for categorical multivariate data" because the second term of the objective function is the q-divergence. The algorithm is presented below (Kondo and Kanzawa 2018).

Step 1. Set fuzzification parameters $q > 1$, $\lambda > 0$ and $t > 0$, the number of clusters C. Initialize typicalities w, and initial variables controlling the cluster size π.

Step 2. Calculate s as

$$s_{i,k} = \frac{1}{t}\sum_{\ell=1}^{M}\left(\left(w_{i,\ell}\right)^t - 1\right)x_{k,\ell} \tag{9}$$

for all $i \in \{1,\dots,C\}$ and $k \in \{1,\dots,N\}$.

Step 3. Calculate U as

$$u_{i,k} = \frac{\pi_i(1 + \lambda(1-q)s_{i,k})^{1/(1-q)}}{\sum_{j=1}^{C}\pi_j(1 + \lambda(1-q)s_{j,k})^{1/(1-q)}} \tag{10}$$

for all $i \in \{1,\dots,C\}$ and $k \in \{1,\dots,N\}$.

Step 4. Calculate w as

$$w_{i,\ell} = \frac{\left(\sum_{k=1}^{N}(u_{i,k})^q x_{k,\ell}\right)^{1/(1-t)}}{\sum_{r=1}^{M}\left(\sum_{k=1}^{N}(u_{i,k})^q x_{k,r}\right)^{1/(1-t)}} \tag{11}$$

for all $i \in \{1, \dots, C\}$ and $\ell \in \{1, \dots, M\}$.

Step 5. Calculate π as

$$\pi_i = \frac{\left(\sum_{k=1}^{N} (u_{i,k})^q (1 + \lambda(1-q)s_{i,k})\right)^{1/q}}{\sum_{j=1}^{C} \left(\sum_{k=1}^{N} (u_{j,k})^q (1 + \lambda(1-q)s_{j,k})\right)^{1/q}} \tag{12}$$

for all $i \in \{1, \dots, C\}$.

Step 6. Check the limiting criterion for (U, w, π). If the criterion is not satisfied, go to Step 2.

The cluster index $i \in \{1, \dots, C\}$ for the user #k, $f(x_k)$ is determined by

$$f(x_k) = \arg \max_{1 \le j \le C} \{u_{j,k}\}.$$

Furthermore, Kondo and Kanzawa proposed using the above QFCCMM algorithm for CF tasks as follows (Kondo and Kanzawa 2019):

Step 1. Define a cut-off value, $\check{x}$.

Step 2. Replace each missing value with the lowest value among all the ratings values.

Step 3. Process Algorithm 2.2.

Step 4. Calculate $\hat{x}$ using

$$\hat{x}_{k,\ell} = \bar{x}_{k,\cdot} + \frac{\sum_{\substack{k' \in \Omega_{\text{user}}(\ell) \\ f(x_{k'}) \equiv f(x_k)}} \text{sim}(k,k')(x_{k',\ell} - \bar{x}_{k',\cdot})}{\sum_{\substack{k' \in \Omega_{\text{user}}(\ell) \\ f(x_{k'}) \equiv f(x_k)}} \text{sim}(k,k')} \tag{13}$$

for all $i \in \{1, \dots, C\}$ and $\ell \in \{1, \dots, M\}$ if $y_{k,\ell} = 0$. If there is no user #k' satisfying both $k' \in \Omega_{\text{user}}(\ell)$ and $f(x_{k'}) \equiv f(x_k)$ for the target user #k, set $\hat{x}_{k,\ell} = \bar{x}_{k,\cdot}$.

Step 5. Recommend all items to the target user #k with $\hat{x}_{k,\ell} \ge \check{x}$ and $y_{k,\ell} = 0$. □

It was shown through some numerical experiments that this algorithm is better than the GroupLens algorithm in terms of recommendation accuracy (Kondo and Kanzawa 2019).

2.3 Fuzzy clustering for spherical data based on *q*-divergence (Higashi et al. 2019)

Higashi et al. (2019) proposed a fuzzy clustering method for spherical data based on q-Divergence (QFCS), defined as

$$\begin{aligned} \underset{U,w,\pi}{\text{minimize}} & \sum_{i=1}^{C} \sum_{k=1}^{N} (\pi_i)^{1-q} (u_{i,k})^q \left(1 - x_k^{\mathsf{T}} v_i\right) \\ & + \frac{\lambda^{-1}}{q-1} \left(\sum_{i=1}^{C} \sum_{k=1}^{N} (\pi_i)^{1-q} (u_{i,k})^q - 1 \right) \end{aligned} \tag{14}$$

which is subject to the constraints in Eqs. (5), (7), and

$$\|v_i\|_2 = 1 \text{ for all } i \in \{1, \dots, C\}, \tag{15}$$

where x_k is on the $M-1$-dimensional unit sphere, and (q, λ) are the fuzzification parameters satisfying $q > 1$ and $\lambda > 0$. This method is named as "q-divergence-based fuzzy clustering for spherical data" because the second term of the objective function is the q-divergence. Both QFCCMM and QFCS methods are based on q-divergence, and the difference between them is the target data type; the QFCCMM method is, as in the name, for categorical multivariate data, and the QFCS method is, as in the name, for spherical data. The QFCS algorithm is described as (Higashi et al. 2019).

Step 1. Fix $q > 1$, $\lambda > 0$. Assume initial cluster centers v and initial variable controlling cluster sizes π.

Step 2. Update U as

$$u_{i,k} = \frac{\pi_i \left(1 - \lambda(1-q)\left(1 - x_k^{\mathsf{T}} v_i\right)\right)^{1/(1-q)}}{\sum_{j=1}^{C} \pi_j \left(1 - \lambda(1-q)\left(1 - x_k^{\mathsf{T}} v_j\right)\right)^{1/(1-q)}} \tag{16}$$

for all $i \in \{1, \dots, C\}$ and $k \in \{1, \dots, N\}$.

Step 3. Update π as

$$\pi_i = \frac{\left(\sum_{k=1}^{N} (u_{i,k})^q \left(1 - (1-q)\lambda\left(1 - x_k^{\mathsf{T}} v_i\right)\right)\right)^{1/q}}{\sum_{j=1}^{C} \left(\sum_{k=1}^{N} (u_{j,k})^q \left(1 - (1-q)\lambda\left(1 - x_k^{\mathsf{T}} v_j\right)\right)\right)^{1/q}} \tag{17}$$

for all $i \in \{1, \dots, C\}$.

Step 4. Calculate v_i as

$$v_i = \frac{\sum_{k=1}^{N} (u_{i,k})^q x_k}{\left\| \sum_{k=1}^{N} (u_{i,k})^q x_k \right\|_2} \tag{18}$$

for all $i \in \{1, \dots, C\}$.

Step 5. Check the limiting criterion for (U, v, π). If the criterion is not satisfied, go to step 2.

Higashi et al. (2019) showed using numerical experiments using 16 real document datasets that QFCS outperformed the conventional methods in terms of clustering accuracy.

3 Proposed method

In a previous work (Kondo and Kanzawa 2019), the neighborhood for the target users was defined using the QFCCMM clustering algorithm.

QFCCMM (Kondo and Kanzawa 2018) was proposed originally for categorical multivariate data, such as document data. In the case of applying QFCCMM for the CF task, we consider the users' items rating vector as categorical multivariate data. On the other hand, GroupLens does not deal with users' items rating vector as categorical multivariate data. For Pearson's coefficient used in GroupLens,

Table 1 Example of initial rating matrix: $N = 5$, $M = 4$, and $\{x_{k,\ell}\}_{(k,\ell)=(1,1)}^{(5,4)}$ are actual rating values from the users, and $x_{1,4}$ needs to be predicted

Item				
User	#1	#2	#3	#4
#1	$x_{1,1} = 1$	$x_{1,2} = 1$	$x_{1,3} = 5$	N/A
#2	$x_{2,1} = 5$	$x_{2,2} = 5$	$x_{2,3} = 1$	$x_{2,4} = 1$
#3	$x_{3,1} = 2$	$x_{3,2} = 2$	$x_{3,3} = 4$	$x_{3,4} = 4$
#4	$x_{4,1} = 5$	$x_{4,2} = 5$	$x_{4,3} = 1$	$x_{4,4} = 1$
#5	$x_{5,1} = 1$	$x_{5,2} = 1$	$x_{5,3} = 5$	$x_{5,4} = 5$

given in Eq. (2), all rating vectors have uniform magnitude, and they are on the unit hypersphere with the dimension of items. In other words, GroupLens deals with user's items rating vectors as spherical data.

Thus, we propose adopting QFCS instead of QFCCMM to segment users' items rating vectors, and we apply GroupLens to the users segment that the target user belongs to. Incorporating Algorithm 2.3, we propose the following algorithm for estimating the missing values:

Step 1. Define a cut-off value, $\check{x}$.

Step 2. Replace each missing value with the lowest value among all ratings' values.

Step 3. Normalize $\{x_{k,\ell}\}_{\ell=1}^{M}$ $(k \in \{1,\dots,N\})$ into $\{\tilde{x}_{k,\ell}\}_{\ell=1}^{M}$ $(k \in \{1,\dots,N\})$, as

$$\tilde{x}_{k,\ell} = \frac{x_{k,\ell}}{\sqrt{\sum_{\ell=1}^{M} x_{k,\ell}}}. \tag{19}$$

Step 4. Process Algorithm 2.3 for $\tilde{x}$.

Step 5. Estimate the missing values $\hat{x}_{k,\ell}$ $(k \in \{1,\dots,N\}$, $\ell \in \{1,\dots,M\})$ if $y_{k,\ell} = 0$, as Eq. (13).

Step 6. Recommend all items to the target user $\#k$ with $\hat{x}_{k,\ell} \geq \check{x}$ and $y_{k,\ell} = 0$. □

The flow of Algorithm 3 is described using Tables 1–6. Table 1 shows an initial rating matrix, for five users versus four items, where the user #1 has not evaluated the item #4 yet, and it is denoted by "N/A." On applying Step 2 of Algorithm 3 to Table 1, we obtain the rating matrix as shown in Table 2. Thus, $x_{1,4}$, denoted by "N/A", is replaced with $\min_{\substack{1\leq k\leq 5\\ 1\leq \ell\leq 4\\ (k,\ell)\notin\{(1,4)\}}} x_{k,\ell} = 1$. On applying Step 3 of Algorithm 3 to Table 2, we obtain the rating matrix as shown in Table 3. Thus, the rating values are normalized for each user, which is a preparation for applying clustering for spherical data. Applying Step 4 of Algorithm 3 to Table 3, we obtain the rating matrix as shown in Table 4, where the user #1 is placed in cluster #1. Immediately before Step 5 of Algorithm 3 is applied to cluster #1 in Table 3, the value $x_{1,4}$ is restored to "N/A", to be predicted, as shown in Table 5. Applying Step 4 of Algorithm 3 to cluster #1 in Table 5, the restored "N/A" is replaced with the predicted rating value, as shown in Table 6. If the estimated value is higher than a given cut-off value $\check{x}$, the corresponding item is recommended to the target user.

4 Numerical experiments

Numerical experiments were conducted to compare the CF accuracy of the following three algorithms: Algorithm 2.1, Algorithm 2.2, and Algorithm 3, using six real datasets: "BookCrossing" (Ziegler et al. 2005), "Epinions" (Massa et al. 2008), "Jester" (Goldberg et al. 2001), "LibimSeTi" (Brozovsky and Petricek 2007), "MovieLens" (Harper and Konstan 2015), and "SUSHI" (Kamishima and Akaho 2009).

4.1 Datasets

The "BookCrossing" dataset was compiled by Cai-Nicolas Ziegler in a four-week crawl of the BookCrossing community with the kind permission of Ron Hornbaker, CTO of Humankind Systems. It contains 1,149,780 ratings for approximately 271,379 books provided by 278,858 users

Table 2 Example of rating matrix after Step 2 of Algorithm 3: $N = 5$ and $M = 4$

Item				
User	#1	#2	#3	#4
#1	$x_{1,1} = 1$	$x_{1,2} = 1$	$x_{1,3} = 5$	$x_{1,4} = \min_{\substack{1\leq k\leq 5\\ 1\leq \ell\leq 4\\ (k,\ell)\notin\{(1,4)\}}} x_{k,\ell} = 1$
#2	$x_{2,1} = 5$	$x_{2,2} = 5$	$x_{2,3} = 1$	$x_{2,4} = 1$
#3	$x_{3,1} = 2$	$x_{3,2} = 2$	$x_{3,3} = 4$	$x_{3,4} = 4$
#4	$x_{4,1} = 5$	$x_{4,2} = 5$	$x_{4,3} = 1$	$x_{4,4} = 1$
#5	$x_{5,1} = 1$	$x_{5,2} = 1$	$x_{5,3} = 5$	$x_{5,4} = 5$

$x_{1,4}$ is set as the minimal value of $\{x_{k,\ell}\}_{(k,\ell)=(1,1)}^{(5,4)}$

Table 3 Example of rating matrix after Step 3 of Algorithm 3: $N = 5$ and $M = 4$

Item				
User	#1	#2	#3	#4
#1	$\tilde{x}_{1,1} = \frac{x_{1,1}}{\sqrt{\sum_{\ell=1}^{4}(x_{1,\ell})^2}} \simeq 0.19$	$\tilde{x}_{1,2} = \frac{x_{1,2}}{\sqrt{\sum_{\ell=1}^{4}(x_{1,\ell})^2}} \simeq 0.19$	$\tilde{x}_{1,3} = \frac{x_{1,3}}{\sqrt{\sum_{\ell=1}^{4}(x_{1,\ell})^2}} \simeq 0.94$	$\tilde{x}_{1,4} = \frac{x_{1,4}}{\sqrt{\sum_{\ell=1}^{4}(x_{1,\ell})^2}} \simeq 0.19$
#2	$\tilde{x}_{2,1} = \frac{x_{2,1}}{\sqrt{\sum_{\ell=1}^{4}(x_{2,\ell})^2}} \simeq 0.69$	$\tilde{x}_{2,2} = \frac{x_{2,2}}{\sqrt{\sum_{\ell=1}^{4}(x_{2,\ell})^2}} \simeq 0.69$	$\tilde{x}_{2,3} = \frac{x_{2,3}}{\sqrt{\sum_{\ell=1}^{4}(x_{2,\ell})^2}} \simeq 0.14$	$\tilde{x}_{2,4} = \frac{x_{2,4}}{\sqrt{\sum_{\ell=1}^{4}(x_{2,\ell})^2}} \simeq 0.14$
#3	$\tilde{x}_{3,1} = \frac{x_{3,1}}{\sqrt{\sum_{\ell=1}^{4}(x_{3,\ell})^2}} \simeq 0.33$	$\tilde{x}_{3,2} = \frac{x_{3,2}}{\sqrt{\sum_{\ell=1}^{4}(x_{3,\ell})^2}} \simeq 0.33$	$\tilde{x}_{3,3} = \frac{x_{3,3}}{\sqrt{\sum_{\ell=1}^{4}(x_{3,\ell})^2}} \simeq 0.67$	$\tilde{x}_{3,4} = \frac{x_{3,4}}{\sqrt{\sum_{\ell=1}^{4}(x_{3,\ell})^2}} \simeq 0.67$
#4	$\tilde{x}_{4,1} = \frac{x_{4,1}}{\sqrt{\sum_{\ell=1}^{4}(x_{4,\ell})^2}} \simeq 0.69$	$\tilde{x}_{4,2} = \frac{x_{4,2}}{\sqrt{\sum_{\ell=1}^{4}(x_{4,\ell})^2}} \simeq 0.69$	$\tilde{x}_{4,3} = \frac{x_{4,3}}{\sqrt{\sum_{\ell=1}^{4}(x_{4,\ell})^2}} \simeq 0.14$	$\tilde{x}_{4,4} = \frac{x_{4,4}}{\sqrt{\sum_{\ell=1}^{4}(x_{4,\ell})^2}} \simeq 0.14$
#5	$\tilde{x}_{5,1} = \frac{x_{5,1}}{\sqrt{\sum_{\ell=1}^{4}(x_{5,\ell})^2}} \simeq 0.14$	$\tilde{x}_{5,2} = \frac{x_{5,2}}{\sqrt{\sum_{\ell=1}^{4}(x_{5,\ell})^2}} \simeq 0.14$	$\tilde{x}_{5,3} = \frac{x_{5,3}}{\sqrt{\sum_{\ell=1}^{4}(x_{5,\ell})^2}} \simeq 0.69$	$\tilde{x}_{5,4} = \frac{x_{5,4}}{\sqrt{\sum_{\ell=1}^{4}(x_{5,\ell})^2}} \simeq 0.69$

Rating values are normalized and are on the unit hypersphere

Table 4 Example of rating matrix after Step 4 of Algorithm 3: $N = 5$, $M = 4$, and $C = 2$

Item					
Cluster	User	#1	#2	#3	#4
#1	#1	$\tilde{x}_{1,1} \simeq 0.19$	$\tilde{x}_{1,2} \simeq 0.19$	$\tilde{x}_{1,3} \simeq 0.94$	$\tilde{x}_{1,4} \simeq 0.19$
	#3	$\tilde{x}_{3,1} \simeq 0.33$	$\tilde{x}_{3,2} \simeq 0.33$	$\tilde{x}_{3,3} \simeq 0.67$	$\tilde{x}_{3,4} \simeq 0.67$
	#5	$\tilde{x}_{5,1} \simeq 0.14$	$\tilde{x}_{5,2} \simeq 0.14$	$\tilde{x}_{5,3} \simeq 0.69$	$\tilde{x}_{5,4} \simeq 0.69$
#2	#2	$\tilde{x}_{2,1} \simeq 0.69$	$\tilde{x}_{2,2} \simeq 0.69$	$\tilde{x}_{2,3} \simeq 0.14$	$\tilde{x}_{2,4} \simeq 0.14$
	#4	$\tilde{x}_{4,1} \simeq 0.69$	$\tilde{x}_{4,2} \simeq 0.69$	$\tilde{x}_{4,3} \simeq 0.14$	$\tilde{x}_{4,4} \simeq 0.14$

Table 5 Example of the rating matrix immediately before Step 5 of Algorithm 3: $N = 3$ and $M = 4$

Item					
Cluster	User	#1	#2	#3	#4
#1	#1	$x_{1,1} = 1$	$x_{1,2} = 1$	$x_{1,3} = 5$	N/A
	#3	$x_{3,1} = 2$	$x_{3,2} = 2$	$x_{3,3} = 4$	$x_{3,4} = 4$
	#5	$x_{5,1} = 1$	$x_{5,2} = 1$	$x_{5,3} = 5$	$x_{5,4} = 5$

$x_{1,4}$ is predicted from the user ratings in cluster #1

Table 6 Example of the rating matrix after Step 5 of Algorithm 3: $N = 3$ and $M = 4$

Item					
Cluster	User	#1	#2	#3	#4
#1	#1	$x_{1,1} = 1$	$x_{1,2} = 1$	$x_{1,3} = 5$	$\hat{x}_{1,4} \simeq 3.83$
	#3	$x_{3,1} = 2$	$x_{3,2} = 2$	$x_{3,3} = 4$	$x_{3,4} = 4$
	#5	$x_{5,1} = 1$	$x_{5,2} = 1$	$x_{5,3} = 5$	$x_{5,4} = 5$

$x_{1,4}$ is replaced with the predicted values, $\hat{x}_{1,4} \simeq 3.83$. If the predicted value is higher than a predefined cut-off value $\check{x}$, then the corresponding item is recommended to the corresponding user

(Ziegler et al. 2005). However, only 35,179 ratings from 1091 users for 2248 books were used for this experiment. Therefore, each book was evaluated by more than 8 users, with each user rating over 15 books. In this case, the ratings were scaled from 1 to 10, with 10 being the best score. The “Epinions” dataset (Massa et al. 2008) was collected by Paolo Massa in a 5-week crawl from the Epinions.com web site, and it contains the rating of users for products such as software, music, television shows, and so on. In “Epinions”, 49,290 users recorded 664,824 ratings for 139,738 products; however, we used 42,808 ratings from 1022 users for 835 products in our experiment. Further, the ratings were scaled from 1 to 5, with 5 being the best score. The “Jester” dataset (Goldberg et al. 2001) was collected by Ken Goldberg from the Jester Online Joke website, and it contains the rating of users for jokes. In “Jester”, 59,132 users recorded around 1.7 million ratings for 150 jokes; however, we used 373,338 ratings from 2916 users for 140 products in our experiment. Further, the ratings were scaled from − 10 to 10, with 10 being the best score. The “LibimSeTi” profile dataset (Brozovsky and Petricek 2007) was released by Vaclav Petricek of eHarmony.com. This dataset includes 17,359,346 anonymous ratings of 168,791 profiles created by 135,359 LibimSeTi users on April 4th, 2006. The ratings were scaled from 1 to 10, with 10 being the best score. Thus, each profile was evaluated by at least 230 users, and each user evaluated at least 230 profiles. In our experiment, only 400,955 ratings from 866 users for 1156 profiles were used. The "MovieLens" dataset was compiled through the "MovieLens" website (Harper and Konstan 2015). This dataset contains the ratings of users for kinds of movies. In "MovieLens", 6040 users recorded 1,000,000 ratings for 3900 movie titles, but we used 277,546 ratings from 905 users for 684 movies in our experiment. Therefore, each movie was evaluated by more than 240 people, and each user rated over 200 movies. Further, the ratings were scaled from 1 to 5, with 5 being the best score. The "SUSHI" dataset (Kamishima and Akaho 2009) was compiled by Toshihiro Kamishima, and contains the rating of users for kinds of sushi. In "SUSHI", 5000 users recorded 50,000 ratings for 100 kinds of sushi. Further, the ratings were scaled from 1 to 5, with 5 being the best score.

4.2 Experimental setting

Algorithm 2.1 did not contain parameter settings. In Algorithm 2.2, the cluster numbers and fuzzification parameters were set as $C \in \{2, 3, \dots, 20\}$, $q \in \{1.0001, 1.0004, 1.0007, 1.001, 1.01, 1.1\}$, $\lambda \in \{10^0, \dots, 10^6\}$, and $t \in \{10^{-6}, \dots, 10^{-2}\}$. In Step 1 of Algorithms 2.2, all the variables controlling cluster sizes were initialized with the reciprocal of the cluster number, and the item typicality values were initialized at random. For the 10 initial settings, the clustering result with the maximal objective function value was selected for Step 3 in Algorithm 2.2. In Algorithm 3, the cluster number and fuzzification parameters were set as the same as in Algorithm 2.2 except for t, which was not needed. In Step 1 of Algorithms 2.3, all the variables controlling cluster sizes were initialized with the reciprocal of the cluster number, and the cluster center values were initialized at random. For the 10 initial settings, the clustering result with the minimal objective function value was selected for Step 3 in Algorithm 3.

The experiment was performed as follows. First, 10,000 rating values in the "BookCrossing" dataset, 20,000 rating values in the "Epinions" dataset, 20,000 rating values in the "Jester" dataset, 20,000 rating values in the "LibimSeTi" dataset, 20,000 rating values in the "MovieLens" dataset, and 10,000 rating values in the "SUSHI" dataset, were randomly selected to be missing from originally evaluated values. It is because the originally evaluated values were used for evaluating the recommendation accuracy of algorithms. Note that the originally missing values were not used. After these true rating values were hidden from the original datasets, Algorithms 2.1, 2.2, and 3 predicted these hidden rating values. Then, the predicted rating values and the true rating values were used for calculating an evaluation measure of recommendation accuracy of algorithms, which is mentioned in the next subsection. These experiments were executed for five settings of selecting missing values.

4.3 Evaluation measure

We applied the three algorithms (Algorithms 2.1, 2.2, and 3) to these six real datasets, and then compared the obtained recommendation accuracy using the area underneath the receiver operating characteristic (ROC) curve (AUROC) (Swets 1979; Hanley and McNeil 1982), defined as follows.

All algorithms recommend items if the corresponding estimation of the rating value is higher than the predefined cut-off value $\check{x}$. If the true rating value is higher than $\check{x}$, the item should be recommended. Here, the following four numbers are considered:

- True positive (TP) is the number of items the algorithm recommended when such the items should be recommended.
- True negative (TN) is the number of items the algorithm did not recommend when such the items should not be recommended.
- False positive (FP) is the number of items the algorithm recommended when such the items should not be recommended.
- False negative (FN) is the number of items the algorithm did not recommend when such the items should be recommended.

True positive rate (TPR) is the percentage of TP in TP and TN. False positive rate (FPR) is the percentage of FP in FP and FN. TPR and FPR, including TP, TN, FP, and FN, change according to the cut-off $\check{x}$. Then, the ROC curve is drawn by connecting several pairs of the FPR and TPR obtained from different cut-off $\check{x}$, and AUROC is the area under the ROC curve. The higher the AUROC value, the more accurate the result of the CF algorithm. In this experiment, the AUROC was calculated using the discrete cut-off values from 0.1 to the maximal rating value in increments of 0.1.

4.4 Results and discussion

Tables 7, 8, 9, 10, 11, 12 show the highest AUROC value for each method and the parameter value at which the highest AUROC value was achieved. Table 13 shows their summary, where the highest AUROC value among the three methods is underlined.

Table 13 indicates that all algorithms produced the same AUROC values for two datasets: Epinions and SUSHI;

Table 7 Highest AUROC value for each method and the corresponding parameter values for the "BookCrossing" dataset

Method	AUROC	Parameter value			
		q	λ	t	C
Algorithm 2.1	0.704500				
Algorithm 2.2	0.720019	1.0004	10^2	10^{-6}	15
Algorithm 3	0.723660	1.0010	10^2		35

Table 8 The highest AUROC value for each method and the corresponding parameter values for the "Epinions" dataset

Method	AUROC	Parameter value			
		q	λ	t	C
Algorithm 2.1	0.733662				
Algorithm 2.2	0.733662	1.0001	1.0	10^{-4}	2
Algorithm 3	0.733662	1.0001	10		2

Table 9 The highest AUROC value for each method and the corresponding parameter values for the "Jester" dataset

Method	AUROC	Parameter value			
		q	λ	t	C
Algorithm 2.1	0.830666				
Algorithm 2.2	0.840505	1.0001	10^3	10^{-5}	10
Algorithm 3	0.841978	1.0001	10^3		10

Table 10 Highest AUROC value for each method and the corresponding parameter values for the "LibimSeTi" dataset

Method	AUROC	Parameter value			
		q	λ	C	
Algorithm 2.1	0.913046				
Algorithm 2.2	0.927429	1.0007	10	10^{-4}	15
Algorithm 3	0.929594	1.0004	10^3		20

Table 11 The highest AUROC value for each method and the corresponding parameter values for the "MovieLens" dataset

Method	AUROC	Parameter value			
		q	λ	C	
Algorithm 2.1	0.787796				
Algorithm 2.2	0.792885	1.0007	10^3	10^{-3}	7
Algorithm 3	0.792885	1.0100	10^5		5

Algorithm 2.2 and Algorithm 3 produced the same AUROC values for one dataset: MovieLens, which are higher than those obtained by Algorithm 2.1; and Algorithm 3 produced the highest AUROC values than those obtained from others for the Epinions, Jester, and LibimSeTi datasets.

Table 12 The highest AUROC value for each method and the corresponding parameter values for the "SUSHI" dataset

Method	AUROC	Parameter value			
		q	λ	t	C
Algorithm 2.1	0.723708				
Algorithm 2.2	0.723708	1.0001	1	10^{-4}	2
Algorithm 3	0.723708	1.0001	10		2

Table 13 Summary of the highest AUROC values for all real datasets

method			
data	Algorithm 2.1	Algorithm 2.2	Algorithm 3
BookCrossing	0.704500	0.720019	0.723660
Epinions	0.733662	0.733662	0.733662
Jester	0.830666	0.840505	0.841978
LibimSeTi	0.913046	0.927429	0.929594
MovieLens	0.787796	0.792885	0.792885
SUSHI	0.723708	0.723708	0.723708

Table 13 shows that the AUROC value obtained from Algorithm 3 is higher than or the same as those obtained from the other methods for all datasets. Therefore, the proposed algorithm is better than the others in terms of recommendation accuracy. The better recommendation accuracy of the proposed method is attributed to the fact that clustering for spherical data allows segmenting users more accurately than clustering for categorical multivariate data.

5 Conclusion

In this study, we proposed a CF algorithm based on q-divergence-based fuzzy clustering for spherical data. The experiment was conducted on six datasets using three different algorithms. The results of the experiment indicate that the proposed algorithm outperforms the conventional methods in terms of recommendation accuracy, and this is attributed to the fact that clustering for spherical data enables a more accurate segmentation of users in comparison with clustering for categorical multivariate data. The results thus indicate that users' items rating vector should be considered as spherical data than categorical multivariate data to better recommendation accuracy.

There is a major limitation in this study. The proposed algorithm must be applied with a predefined cluster number and two fuzzification parameter values. The experiment was conducted through several cluster numbers and fuzzification parameters, and the best AUROC value was compared with conventional methods. This means that the proposed method achieves high recommendation accuracy provided that the predefined cluster number and fuzzification parameters were set adequately. However, if they are not set adequately, the recommendation accuracy would degrade, and it would possibly be worse than that of conventional methods.

To overcome this limitation, future research aims to select an appropriate cluster number and fuzzification parameter values for the proposed method; for example, adopting cluster validity indices (Dunn 1974; Gath and Geva 1989; Xie and Beni 1991; Wang and Zhang 2007) and conducting cross validation.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Brozovsky L, Petricek V (2007) Recommender system for online dating service. Eprint arXiv: cs/0703042 [online]. https://arxiv.org/abs/cs/0703042. Accessed 15 Oct 2015

Dunn JC (1974) Well separated clusters and optimal fuzzy partitions. J Cybern 4:95–104. https://doi.org/10.1080/01969727408546059

Gath I, Geva AB (1989) Unsupervised optimal fuzzy clustering. IEEE Trans Pattern Anal Mach Intell 11(7):773–780. https://doi.org/10.1109/34.192473

Goldberg K, Roeder T, Gupta D, Perkins C (2001) Eigentaste: a constant time collaborative filtering algorithm. Inf Retriev 4(2):133–151. https://doi.org/10.1023/A:1011419012209

Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143:29–36. https://doi.org/10.1148/radiology.143.1.7063747

Harper FM, Konstan JA (2015) The MovieLens datasets: history and context. ACM Trans Interact Intell Syst 5(4):19. https://doi.org/10.1145/2827872

Herlocker JL, Konstan JA, Borchers A, Riedl J (1999) An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp 230–237. https://doi.org/10.1145/312624.312682

Higashi M, Kondo T, Kanzawa Y (2019) Fuzzy clustering method for spherical data based on q-divergence. J Adv Comput Intell Intell Inform 23(3):561–570. https://doi.org/10.20965/jaciii.2019.p0561

Honda K (2016) Fuzzy co-clustering and application to collaborative filtering. In: Huynh VN et al (eds) IUKM 2016, LNAI, vol 9978. Springer, Berlin, pp 16–23

Honda K, Oshio S, Notsu A (2015) Fuzzy co-clustering induced by multinomial mixture models. J Adv Comput Intell Intell Inform 19(6):717–726. https://doi.org/10.20965/jaciii.2015.p0717

Kamishima T, Akaho S (2009) Efficient clustering for orders. In: Zighed DA et al (eds) Mining complex data, vol. 165 of studies in computational intelligence chapter 15. Springer, pp 261–280.

Kondo T, Kanzawa Y (2018) Fuzzy clustering methods for categorical multivariate data based on q-divergence. J Adv Comput Intell Intell Inform 22(4):524–536. https://doi.org/10.20965/jaciii.2019.p0493

Kondo T, Kanzawa Y (2019) Collaborative filtering using fuzzy clustering for categorical multivariate data based on q-divergence. J Adv Comput Intell Intell Inform 23(3):493–501. https://doi.org/10.20965/jaciii.2019.p0493

Massa P, Souren K, Salvetti M, Tomasoni D (2008) Trustlet, open research on trust metrics. Scal Comput Pract Exp 9(4):341–351

Paul R, Neophytos I, Mitesh S, Peter S, Jhon R (1994) GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on Computer supported cooperative work, pp 175–186. https://doi.org/10.1145/192844.192905

Sarwar B, Karypis G, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, pp 285–295. https://doi.org/10.1145/371920.372071

Swets JA (1979) ROC analysis applied to the evaluation of medical imaging techniques. Investig Radiol 14:109–121. https://doi.org/10.1097/00004424-197903000-00002

Tsallis C (2009) Introduction to nonextensive statistical mechanics : approaching a complex world. Springer, Berlin

Wang W, Zhang Y (2007) On fuzzy cluster validity indices. Fuzzy Sets Syst 158(19):2095–2117. https://doi.org/10.1016/j.fss.2007.03.004

Xie XL, Beni G (1991) A validity measure for fuzzy clustering. IEEE Trans Pattern Anal Mach Intell 13(8):841–847. https://doi.org/10.1109/34.85677

Ziegler C, McNee SM, Konstan JA, Lausen G (2005) Improving recommendation lists through topic diversification. In: Proceedings of the 14th international conference on World Wide Web, pp 22–32. https://doi.org/10.1145/1060745.1060754

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.