Neural Computing and Applications (2023) 35:177–189
https://doi.org/10.1007/s00521-020-05613-z

S.I. : INFORMATION, INTELLIGENCE, SYSTEMS AND APPLICATIONS

Improvement of similarity–diversity trade-off in recommender systems based on a facility location model

Antiopi Panteli[1] · Basilis Boutsinas[1]

Received: 22 January 2020 / Accepted: 11 December 2020 / Published online: 2 January 2021

Abstract

There is a growing interest in the offering of novel alternative choices to users of recommender systems. These recommendations should match the target query while at the same time they should be diverse with each other in order to provide useful alternatives to the user, i.e., novel recommendations. In this paper, the problem of extracting novel recommendations, under the similarity–diversity trade-off, is modeled as a facility location problem. The results from tests in the benchmark Travel Case Base were satisfactory when compared to well-known recommender techniques, in terms of both similarity and diversity. It is shown that the proposed method is flexible enough, since a parameter of the adopted facility location model constitutes a regulator for the trade-off between similarity and diversity. Also, our work can broaden the perspectives of the interaction and combination of different scientific fields in order to achieve the best possible results.

Keywords Recommender Systems · Similarity–diversity trade-off · Facility location model · Biclustering

1 Introduction

The development of the Internet has resulted in an overload of data and users often find it difficult to extract information that best corresponds to their preferences or needs. Thus, recommender systems became part of life since they can manage and process the available information in order to filter the redundant part and extract useful knowledge [57]. Recommender systems aim to reduce complexity in human life through selecting from a very large amount of information the part that is relevant to the active user [44]. Thus, their applications can be found in different aspects of everyday life such as health [71], music [19], movies [27], travel [25, 67], and e-learning [12].

There are some major types of recommender systems that have been studied in the literature, such as the content-based systems (CBS), the collaborative filtering (CF) approaches, the knowledge-based systems (KBS), as well as hybrid approaches. Case-based systems that are studied in this paper constitute a subclass of knowledge-based systems (KBS). More specifically, the primary components of case-based systems are:

- a data base of previously solved problems along with their solutions
- the user query which specifies the needs and preferences of the user in the form of attribute-value pairs
- a similarity function which estimates how the user's query matches the cases of the case base and thus, the cases of the retrieval set [63].

Case-based recommender systems have been used in the literature in order to provide recommendations concerning different aspects of everyday life. For instance, case-based recommender systems have been used in heart disease diagnosis [58], wealth management services [50], education [14], music services [40] and so on.

The recommendations have to be quite similar to the inserted query, but also they have to be different to each other in order to provide diverse options to the user [35, 68, 77, 78], since users may get bored after receiving many recommended objects under the same topic. In [17] the diversity in the recommendation procedure was described as the opposite of similarity. In fact, the diversity

✉ Antiopi Panteli
apanteli@upatras.gr

Basilis Boutsinas
vutsinas@upatras.gr

[1] MISBILAB, Department of Business Administration, University of Patras, Patras, Greece

of a recommendation list provides significant value to the user, by offering solution to over-fitting. It has been shown that the diversity of proposed results increases the level of satisfaction of the target user. This derives from the fact that different objects which match the user's requirements and preferences offer a great range of options. However, the most common problem in the study of recommender systems is the trade-off between similarity and diversity.

In this work, we apply a facility location model to improve the similarity–diversity trade-off in case-based recommender systems. The basic components of a facility location model are the demand points, the facilities that can serve these demand points and a distance function that represents the distance between each facility and each demand point. More specifically, this model solves the multiple p-median problem, which takes as input a distance matrix and aims to select p facilities in such a way that they can serve each demand point at least mc times and the total distance between these facilities and the demand points is the minimum possible. To the best of our knowledge, the application of this location model to case-based recommender systems has never been studied in the literature before. In that sense, our work can broaden the perspectives of the interaction and combination of different scientific fields in order to achieve the best possible results. We evaluate the proposed approach by using a recommender system that offers alternative choices in order to propose diverse travel recommendation plans to users, by selecting a specific number of cases from the Case Base that match the user's requirements and preferences expressed in the target query. An important characteristic of the proposed approach is that a specific parameter of the facility location model constitutes a regulator for the trade-off between similarity and diversity of the recommendation set.

2 Background

2.1 Trade-off between similarity and diversity

Usually accuracy constitutes an evaluation measure of recommendation algorithms. However, it has been stated [33, 66] that accuracy is not enough and user satisfaction in terms of the recommendation list depends on other factors as well. This derives probably from the fact that when the accuracy and similarity of the recommended objects to a given query are high, it is possible that the objects are similar with each other as well and thus diverse choices are not really offered to the user.

Therefore, the recommendation list should contain items that are similar to the user inserted query, but are also significantly diverse to each other at the same time. The diversity is a relatively new term and was first described in [17]. In a survey about query result diversification it is stated that diversity contributes to less simplistic results and can bring new information not previously mentioned [76]. Also, since the users' queries are sometimes ambiguous, diversity can be used in order to provide results with varying information that may satisfy the users' true intentions.

Even though there are many studies on the trade-off between similarity and diversity in the field of collaborative filtering recommender systems, the corresponding studies in case-based recommender systems are quite a few [17, 35, 46, 51, 68]. In [17] a greedy selection algorithm is proposed to address the problem of diversity in case-based recommender systems. However, this method is inefficient and in [68], a bounded version of the algorithm was proposed. This improved version first selects a certain number, say b, of items that are the most similar to the target query and then applies the initial greedy selection algorithm to these b items rather than to the entire initial set of items. However, as b approaches the size of the original set of items, the complexity of this version of the algorithm approaches the complexity of the greedy selection method. In [46] the authors present a retrieval method that offers similarity-preserving increases in diversity. In small retrieval sets the increase in diversity is slightly less than in [68] where the similarity though is not fully maintained. They concluded that the similarity-preserving ability of the algorithm in the increases of diversity, depends on the similarity measure that is used. Another more recent algorithm was proposed in [35]. The trade-off between similarity and diversity is represented as a quadratic programming problem. The results were obtained with a slightly worse computational time complexity than in [68]. In [51], the proposed recommender system evaluates the users using specific demographical and financial attributes and proposes diverse and personalized investment portfolios, by combining case-based reasoning along with a diversification strategy.

2.1.1 Optimization methods applied in recommender systems, in general

Most works in the literature related to diversity enhancement in recommendation lists refer to collaborative filtering approaches [1, 2, 6, 7, 13, 22, 38, 41, 74, 79] rather than to case-based approaches. In collaborative filtering the preferences of users about several items are expressed in the form of ratings. The input data matrix consists of users and items with their corresponding ratings. The CF systems recommend a list of items to a user based on a prediction about her possible preference on these items by taking into account the ratings of other users with similar preferences, i.e., expressed by items that both liked in the past. The trade-off between similarity and diversity is also a problem in collaborative filtering recommender systems. In [74] the competing objectives of diversity maximization in the retrieved recommendation set and the maintenance of adequate levels of similarity to the user query are modeled as a binary optimization problem. In [60] a ranking method was proposed that improves the overall recommendation diversity by taking into account the diversity impact of each item on the final recommendation set. In [22] more diverse recommendations were offered through the calculation of category correlations and in [1] the diversity was increased by avoiding showing items of the same category. The authors in [26] considered the effect of recommender systems on the diversity of sales and used Gini coefficient to measure sales diversity.

Optimization techniques can find an optimal or near-optimal solution with low computational effort. Heuristic optimization methods have been used in recommender systems since they can explore and analyze large quantities of data. There are many data mining techniques used in recommender systems since they can handle large databases, discover patterns and provide personalized suggestions to the users based on their preferences. Such techniques are the association rule mining [21] and classification methods based on decision trees [37]. In both studies, customer data regarding product purchases have been used as input tables. Other data mining techniques employed in the recommendation process are the k-nearest neighbor or a weighted version of it which selects only significant nearest neighbors [7] with an input table of users' ratings about movies, clustering and so on. The use of clustering in recommender systems can be found in many recent works, since it can improve the diversity of the proposed results. All these works focus on the collaborative filtering recommender systems where the input data consist of a user that has rated (or selected) a set of items and the system tries to recommend a series of items that the user might be interested in because other users with similar interests have already selected them. In [75] and [43] the input table, deriving from the well-known MovieLens database, and in [13] from the Netflix database [9], which are both widely used in collaborative filtering, contains users that have rated items. In [75], the items that define the user's profile are grouped based on their similarity and the recommendation is executed not on the entire user profile but on the clusters. In [13] a set of items that are similar to the representative items of the clusters is retrieved. In [6] where the input table contains ratings about items, the recommendation list consists of items that are selected from different clusters in order that the diversity is maximized without decreasing accuracy. A tunable parameter is employed that regulates diversity levels of the recommendation list. The nearest-neighbor algorithm that is presented in [43] improves the aggregate diversity of the recommendation list since it uses multi-dimensional clustering in order to propose clusters of items to the user.

The application of neural network techniques in recommender systems is also proposed in the literature. A neural network, which is characterized by a parallel distributed architecture, contains entities that are connected, processes information, has the capacity of learning and performs difficult computational tasks [36]. In [32] where the input table consists of users and ratings from MovieLens database and Pinterest, a nonlinear neural network model is applied in the field of item-based collaborative filtering which can distinguish the selected or rated items in the user's profile that are more useful for making a prediction. Some other examples of the incorporation of neural networks in the field of recommender systems can be found in [23, 34, 42, 54].

In [5], the authors use for the offline evaluation of their proposed method input tables from the MovieLens database consisting of users and ratings. A greedy algorithm tries to maximize a diversity-weighted utility objective function. In [62], through the use of the same database as in [5], the recommendation process is treated as a multi-objective problem where several recommendation methods are combined in order that accuracy and diversity are optimized.

In [3], the authors refer to the application of multi-criteria optimization approaches to recommender systems where a decision maker is directed to choose the best option by taking into account competing criteria. The authors suggest that Data Envelopment Analysis (DEA), which is widely used in operational research, might be applied to multi-criteria recommender systems. This method can suggest to the user a reduced set of items that have the best ratings across all criteria among the candidates. However, this method has not been studied yet in real or synthetic data, so its effectiveness is unknown.

In [30] the authors employ the maximum coverage problem from the facility location field in order to make the recommendation. They use an existing greedy algorithm in logged sales data in order to solve the location problem and they try to show that their maximum coverage list which is further enhanced by the use of product associated recommendations is better than the best-seller list which contains the most popular products of an e-commerce site. However, the experiments gave a slight difference of around 10% between the two lists. They also measured the diversity of the proposed recommendation list in terms of categories and not in explicit values. Their input data are a list of customers' ID, a time stamp and a list of products for each of them. Thus, their type of recommendation cannot be categorized as case-based. The input data are similar to an input table of collaborative filtering recommendation although the evaluation methodology is different.

As a result, to the best of our knowledge, the application of a location problem to case-based recommender systems in order to control the trade-off between diversity and similarity, has not yet been proposed in the literature. The multiple p-median problem is characterized by a specific parameter which can control the level of diversity in the recommendation list and thus handle the aforementioned trade-off.

2.2 Multiple p-median problem

In this subsection, we give a brief overview of the facility location problem which we applied to recommender systems, namely the multiple p-median problem (MPMP). It is an extension of the well-known p-median problem, which was introduced in [29].

In general, the basic components of a facility location problem are the demand points, the facilities that aim to serve these demand points and a distance function that represents the distance between each demand point and each facility. Given a specific distance threshold, a demand point can be served by a specific facility if the corresponding distance of this demand point from the facility is less than the threshold.

The MPMP focuses on the possibility of service of a demand point more than one times, expressed by a parameter (*mc*). It can be applied in various situations where it may be necessary to provide backup facilities that can cover the demand in case the primary facility assigned to a demand point becomes unavailable.

For example, since uncertainty is inherent in real-life applications, the primary facility, due to weather, labor actions, electricity problems and other factors, may not be able to satisfy the demand. Thus, it is important to have alternative facilities that can cover the demand in a way that minimizes the cost while also hedging against failures

The objective of the problem is to select *p* facilities represented by columns in order to serve each demand point at least *mc* times, each time from a different facility. The facilities have to be selected in such a way that the total distance between demand points and facilities is the minimum possible. The formulation of the MPMP requires defining first some variables and parameters.

$$x_{ij} = \begin{cases} 1, & \text{if demand point } j \text{ is served by a facility at site } i \\ 0, & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{if a facility is located at candidate site } i \\ 0, & \text{otherwise} \end{cases}$$

The available facilities are defined by parameter *p* and *mc* represents the number of times a demand point should be served. The value d_{ij} indicates the distance between the facility *i* and the demand point *j*.

Thus, the formulation of the MPMP is the following:

$$\text{Minimize } \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij} \tag{1}$$

$$\text{subject to } \sum_{i \in I} x_{ij} \geq mc, \ \forall j \in J \tag{2}$$

$$\sum_{i \in I} y_i = p \tag{3}$$

$$x_{ij} - y_i \leq 0, \ \forall i \in I; j \in J \tag{4}$$

$$y_i \in \{0, 1\}, \ \forall i \in I \tag{5}$$

$$x_{ij} \in \{0, 1\}, \ \forall i \in I; j \in J \tag{6}$$

The objective function (1) minimizes the total distance between the demand points and the selected facilities. Constraint (2) ensures that each demand point is served at least mc times. Note that partial coverage of a demand point by a facility is not considered. Constraint (3) indicates that p facilities should be located in order to satisfy the demand. Constraint (4) ensures that no demand point is assigned to a location unless there is an open facility at that location. Constraints (5) and (6) refer to the nature of the decision variables.

Since this problem is NP-hard, exact methods can be used efficiently for smaller instances but for the larger ones, e.g., including 1000 nodes or more, exact methods become inefficient, since computational time increases rapidly with instance size. As a result, we used a heuristic method to solve the MPMP, which is presented in [53]. This method integrates a specific biclustering algorithm [16] which is based on the idea of association rule mining.

2.2.1 Biclustering

Clustering is the partition of a set of objects into clusters with respect to a set of features (attributes) that characterize these objects. Objects within the same cluster are more similar to each other, when compared over this set of features, with respect to objects belonging to other clusters [11]. Biclustering techniques perform simultaneous clustering on rows and columns of the input data matrix. The term biclustering was first introduced by [20] for the simultaneous clustering of gene expression data in DNA microarray analysis. A bicluster represented a type of joint behavior of a set of genes in a corresponding set of samples. In general, biclustering refers to a distinct class of clustering algorithms that perform simultaneous row–column clustering.

Biclustering technique has been extensively studied in the literature, and it has been used in various domains such as DNA microarray analysis [20], machine-part cell formation [15], text mining [52], nutritional data analysis [39], patient data [72], agriculture [48], and target marketing [24].

Most interesting biclustering problems are proved to be NP-complete [8, 20, 70] either searching for a minimum set of overlapping (or mutually exclusive) biclusters or searching for one "large" bicluster. As a result, most of the developed biclustering algorithms are based on heuristics [18, 45]. Two different categories of heuristics for detecting biclusters are mainly considered in the literature. The first one is adopted in algorithms trying to detect "good" biclusters using an objective function to measure the quality of the biclusters. They start with a set of initial (usually randomly selected) biclusters [10, 49, 73], or one large bicluster [39] which is usually the whole input data matrix [20, 31], and then try to improve the quality of these biclusters by altering them (e.g., removing, adding, or permuting rows and columns). The algorithms adopting this approach do not guarantee the detection of the best biclusters. They only provide an approximate solution, offering nevertheless a low time complexity [15].

The second approach is adopted by algorithms trying to exhaustively enumerate all candidate biclusters in order to guarantee the detection of the best ones [4, 16, 56, 59, 70]. Since there are $(2^{|R|-1}) * (2^{|C|-1})$ possible biclusters in an input data matrix $D(R, C)$, each such algorithm is based on some principle in order to reduce the search space.

The adopted, in this paper, biclustering algorithm [16] incorporates the second approach, the exhaustive bicluster enumeration of possible biclusters, in order to guarantee the detection of the best biclusters. In order to reduce the search space, it is based on the key idea of association rule mining, which states that every subset of a frequent item set must also be frequent. Therefore, the minimum support measure of Apriori like association rule mining algorithms is used in order to control the size of the biclusters. The minimum support measure is given as the minimum accepted percentage of the number of rows in the whole dataset and defines the minimum number of rows of a bicluster. More specifically, the greater the minimum support (*ms*) is, the higher the number of rows of the extracted biclusters.

Of course, any of the methods that can solve effectively the multiple p-median facility location model can be also used instead.

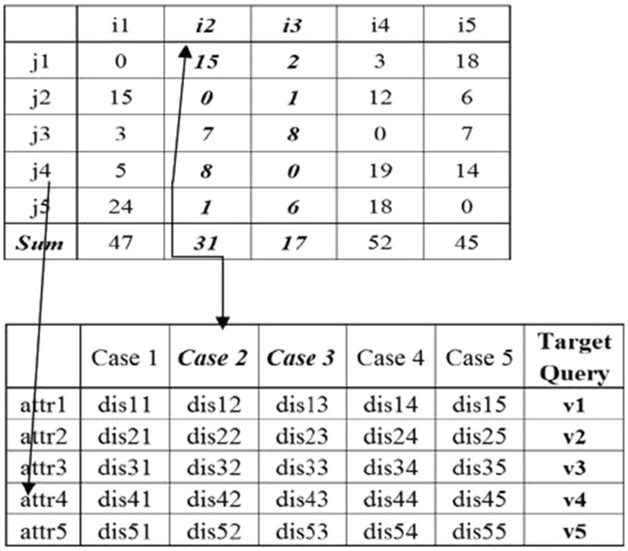
3 Application of multiple p-median problem to recommender systems

In this section we present the application of the MPMP to recommender systems as described in Fig. 1. Based on the formulation presented in the previous section, the MPMP concerns facilities that serve the demand points. Each facility i can be represented by a column of a matrix and each demand point j by a row or vice versa.

As for recommender systems, they contain a huge number of cases with specific attributes. Each $Case_i$ can be represented with a column and each attribute $attr_j$ with a row of a matrix. Given a target query that a user inserts to the recommender system, all the cases of the database indicate a specific level of dissimilarity with the target query. More specifically, for each attribute $attr_j$ of the $Case_i$ under observation, the cell value dis_{ij} indicates a specific dissimilarity level of the $Case_i$ with the target query for the attribute $attr_j$, given its value v_j. As a result, a matrix can be created containing all the dissimilarity ratios between the target query and all the remaining cases of the database. Also, for each $Case_i$, the total dissimilarity ratio can be measured through the aggregation of the dissimilarity levels for each individual attribute for this case.

It is obvious from Fig. 1 that the MPMP model can be easily applied to recommender systems. More specifically, the decision variable x_{ij} determines whether the $attr_j$ of the $Case_i$ has similar value as that of the target query or not and the decision variable y_i determines whether the $Case_i$ is selected to be included in the recommendation set or not.

The objective function of the minimization of the total distance between facilities and demand points can be



	i1	*i2*	*i3*	i4	i5
j1	0	*15*	*2*	3	18
j2	15	*0*	*1*	12	6
j3	3	*7*	*8*	0	7
j4	5	*8*	*0*	19	14
j5	24	*1*	*6*	18	0
Sum	47	*31*	*17*	52	45

	Case 1	*Case 2*	*Case 3*	Case 4	Case 5	**Target Query**
attr1	dis11	dis12	dis13	dis14	dis15	**v1**
attr2	dis21	dis22	dis23	dis24	dis25	**v2**
attr3	dis31	dis32	dis33	dis34	dis35	**v3**
attr4	dis41	dis42	dis43	dis44	dis45	**v4**
attr5	dis51	dis52	dis53	dis54	dis55	**v5**

Fig. 1 An example showing how a facility location problem can be applied to recommender systems

adapted to a recommender system, where the aim is to minimize the dissimilarity level between the cases of the recommendation set and the target query, since the aim of the system is to propose cases similar to the target query.

In fact, the selected solutions (biclusters) must contain in total p columns and each row must be served at least by mc columns. The mc parameter can impose a higher or lower similarity to diversity ratio according to the different values it can take. Higher values of the parameter mc lead to higher similarity to diversity ratios of the recommended cases when compared to the target query.

4 Time complexity

Exact methods that solve facility location problems may fail to provide a query result in reasonable time. It was shown theoretically that exponential cases exist within the class of network problems [47] during solving linear programming models (e.g., variants of Simplex).

On the other hand, most interesting proposed biclustering problems are proved to be NP-complete [8, 20, 70]. Thus, the proposed methodology is based on a heuristic method that reduces the search space providing results close to optimal ones within a shorter time limit.

The adopted heuristic method for solving the multiple p-median facility location model exhibits an average gap from optimal 2.54% while the average speedup (speedup is calculated by dividing the CPLEX solution time by the solution time of the proposed method) is 106, compared to CPLEX for the benchmark instances of OR library. It must be noted that for many instances, the gap and the speed up took values up to 0.72%, and 1865.892, respectively [53].

Also, for large-scale problems up to 1200 nodes, the gap from the optimal solution was up to 0.84% while speed up took values up to 8824. For even larger problems (up to 2000 nodes), given a time limit of 3600 seconds, the proposed method outperformed the best integer solution provided by CPLEX by up to 49%. The time complexity of the adopted heuristic method is dominated by the time complexity of the used heuristic biclustering algorithm [16].

It is $O(|R| * |C| * |C_1 \cup C_2 \cup \ldots \cup C_k|)$ where R is the number of characteristics used to describe cases, C is the number of cases and $C_1, \ldots, C_k$ are the examined sets of candidate frequent itemsets.

The running time of the proposed method is in the range of 0.01 to 850 seconds depending on the number of frequent itemsets that are extracted and evaluated from the Apriori algorithm and the number of recommended cases to the user. Our approach was implemented in Java Environment and the experiments were carried out on a PC Intel Core TM i7-4700 CPU (2,40GHz) with 8GB RAM.

5 Experimental results

5.1 Case base

The proposed method is applied in case-based recommender systems. In case-based recommender systems, a database consisting of cases along with their values in specific attributes is needed. However, most works in recommender systems are related to collaborative filtering and thus there is a great availability of databases that contain ratings of users about items. In contrast to collaborative filtering, the literature in case-based recommendation is too limited and the standard benchmark case library that is used in research studies is the Travel Case Base (https://ai-cbr.cs.auckland.ac.nz/cases.html) which contains categorical and numerical data. In order to further evaluate the proposed methodology, we also used the dataset proposed in [51]. This financial dataset contains the demographical and financial attributes of 1173 users along with their corresponding portfolios. Each user is characterized by 8 attributes and each portfolio presents the user's capital allocation in 20 asset classes.

5.2 Data transformation

Table 1 describes the two data sets. Each case of the Travel Case Base, which refers to a specific travel proposal, is represented by a column of the Data Matrix, whereas each attribute (such as type, price and number of people) is represented by a row of the Data Matrix. Each case is characterized by a dissimilarity ratio—when compared to a query as a whole—and a dissimilarity score for each of the attributes. The dissimilarity measure that is used is defined by the following equation:

$$d_{\chi^2}(A,B) = \sum_{j=1}^{m} \frac{n_{a_j} + n_{b_j}}{n_{a_j} \times n_{b_j}} \times \delta(a_j, b_j) \tag{7}$$

where n_{a_j},n_{b_j} are the numbers of input objects that have values a_j and b_j for attribute j and

$$\delta(a_j, b_j) \begin{cases} 0, & \text{if } a_j = b_j \\ 1, & \text{if } a_j \neq b_j \end{cases}$$

To facilitate matching on the numeric attributes (price, duration), it is a common approach either to discretize values in some ranges [35]. Based on the Sturges measure [69] and the specific problem and data, we discretized values of numerical attributes into 3 intervals of equal length. In categorical attributes, the comparison is exact between values of the case under observation and the target query.

Table 1 Data Sets

Data Set	Columns	Rows	Data Type	Dissimilarity Measure
Travel Case Base	Travel Package	Attributes such as number of people, price etc.	Both categorical and numerical	(7)
Financial Database	Users	Users' attributes such as sex, age, risk, investment goals	Both categorical and numerical	(7)
	Users' Portfolios	Asset classes (1–20)	Numerical	(7)

As for the financial data set, we followed the procedure that the authors realized in [51]. When a user inserts a query, the comparison is performed among the users of the database regarding a series of demographical and financial attributes with the use of the dissimilarity measure of equation (7). The 50 more similar users to the one that has inserted a query are extracted along with their corresponding portfolios. The algorithm is then executed on the portfolios and not on the users. Each portfolio is represented by a column of a matrix and each asset class is represented by a row.

5.3 Experimental tests

We performed sets of experimental tests on the Travel Case Base for different values of the parameter mc $(mc = 3, 4, 5, 7, 10, 15)$ in order to show the effectiveness and flexibility of the proposed methodology. As we have already stated, this parameter refers to the alternative choices which are recommended to the user by a recommender system. Table 2 presents the experimental results of the application of the multiple p-median model to recommender systems for the Travel Case Base. The diversity ratio is calculated as in [68] by the following equation and is defined as the average dissimilarity between all pairs of cases in the case-set:

$$Div(c_1, \ldots, c_n) = \frac{\sum_{i=1}^{p} \sum_{j=1}^{p} (1 - Sim(c_i, c_j))}{\frac{p}{2}(p - 1)} \tag{8}$$

where p is the number of cases retrieved by the case base in order to find their diversity and similarity, respectively.

Similarity is obtained through the following equation:

$$Sim(t, c) = \frac{\sum_{i=1}^{p} w_i \times Sim(t_i, c_i)}{\sum_{i=1}^{p} w_i} \tag{9}$$

where t is the target query and c is the case with which the comparison is done. We computed as well, the standard deviation for both the diversity and similarity measures, as it is shown in Table 2, in order to find the variability of the reported results. The Travel Case Base contains 1024 cases but no real user queries. Thus, we used for our experiments

Table 2 Experimental Results

p	mc	Diversity	Standard Deviation (div)	Similarity	Standard Deviation (div)	Similarity to Diversity Ratio
5	3	0.33783	0.09705	0.73169	0.08650	2.166
5	4	0.31407	0.09754	0.73295	0.09062	2.334
5	5	0.25923	0.08276	0.73321	0.00222	2.828
10	3	0.36108	0.08866	0.70065	0.08439	1.940
10	5	0.35329	0.09139	0.70068	0.08704	1.983
10	7	0.33663	0.09409	0.69950	0.09239	2.078
10	10	0.30170	0.09532	0.69990	0.09721	2.320
15	5	0.37412	0.09435	0.67896	0.09383	1.815
15	10	0.35160	0.11819	0.66820	0.11721	1.900
15	15	0.34350	0.12819	0.66370	0.12629	1.932
20	5	0.42599	0.10370	0.63301	0.10277	1.486
20	10	0.40553	0.13861	0.63494	0.14408	1.566
20	20	0.38848	0.13833	0.62245	0.14258	1.602

the Leave-One-Out (L-O-O) approach in which each case is removed from the Case Base in turn, and the values of its attributes are considered as a query to the proposed approach.

For each target query, we performed experimental tests for different values of the p parameter. In fact, experimental tests were performed for $p = 5, 10, 15$ and 20. A total mc out of the cases (columns) should match the attribute values (rows) of the target query.

As the value of mc increases, we expect higher similarity to diversity ratios of the recommended cases when compared to the target query. When the recommendation set is limited and consequently the parameter mc is small since $mc \leq p$, the existence of similar cases to the query is more probable. Thus, when mc is small, as it increases (i.e., from 3 to 4 or 5), it forces the selected cases to serve each row (attribute) more times and hence forces the cases to be more similar to each other. Therefore, the increase in similarity is high and thus the similarity to diversity ratio increases. This is represented in Table 2 and in Fig. 2 that illustrates the results of similarity and diversity measures and the ratios of similarity to diversity when the recommendation set is small ($p = 5$). In fact, the average similarity to diversity ratio of recommended cases when $mc = 5$ is higher than the average ratio when $mc = 4$ and the corresponding ratio when $mc = 4$ is greater than the one when $mc = 3$. The average similarity is high enough as we expected. When the value of mc increases from 3 to 5, the average similarity also increases (from 0.73169 to 0.73295 and 0.73321 correspondingly) as shown in Table 2. This high value of average similarity, due to the trade-off between similarity and diversity, implies a lower value of diversity.

In contrast, when the size of the recommendation list is large ($p = 10, 15, 20$), the existence of similar cases to the query is less probable and thus the diversity is high. In general, only a limited number of cases present a high similarity ratio when compared to a specific query.

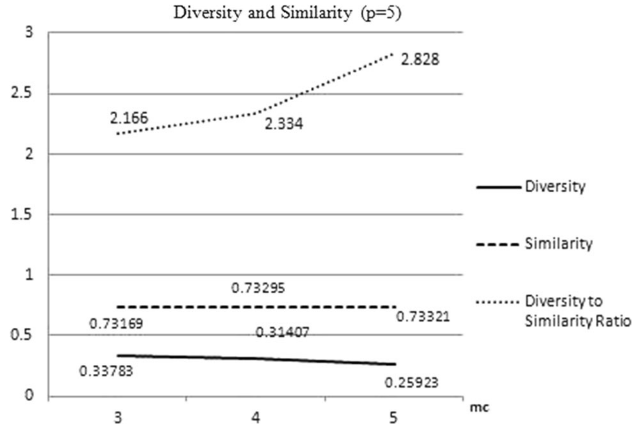


Fig. 2 Graph representation for Similarity and Diversity results ($p = 5$)

As a result, when the recommendation list is large, increasing mc (e.g., from 3 to 10) tends to force the selected cases to serve each row (attribute) more times which affects the diversity measure. More specifically, the diversity decreases and consequently the similarity to diversity ratio increases. This is shown in Table 2 for the corresponding values of $p = 10, 15, 20$ and $mc = 3, 5, 7, 10, 15, 20$ and in Figs. 3, 4 and 5.

The low values of the standard deviation (close to zero) for both the diversity and similarity measures indicate that the corresponding results of each query are close to the average ones and they do not fluctuate around wide intervals.

5.4 Comparison with other methods

As mentioned earlier, the leave-one-out method is adopted and constitutes our basic method for the experimental tests. However, in order to perform a comparison analysis with the reported results in [68], we also performed tests by randomly selecting a subset of cases as queries, as in [68]. More specifically, 400 cases were randomly selected out of the whole set of 1024 cases in order to form queries for $p = 5$ and the remaining cases forming the Case Base. We repeated this procedure 50 times for different sets of 400 queries and from the similarity and diversity results, we computed an overall average similarity and diversity value. The results for both approaches are presented in Table 3.

The average diversity (Table 3) of the selected cases, calculated by (8) with $mc = 3$, is 0.338 and thus it is quite satisfactory when compared to the relative results of diversity presented in [68]. As shown in Table 3, our implementation outperforms the Standard and Random technique in terms of diversity, whose values range from 0.289 to 0.326 [68], respectively, and performs well against Bounded Greedy (0.375) and Greedy technique (0.458) for

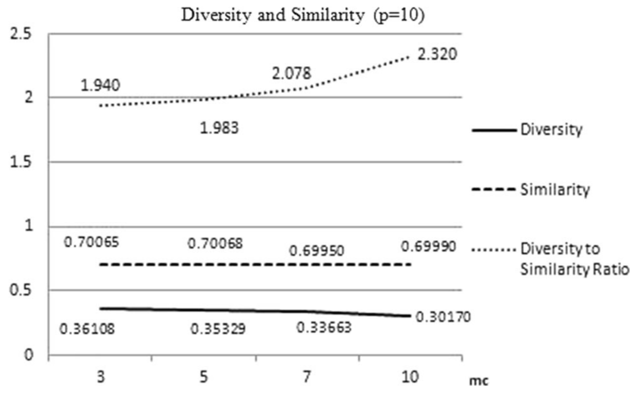


Fig. 3 Graph representation for Similarity and Diversity results ($p = 10$)

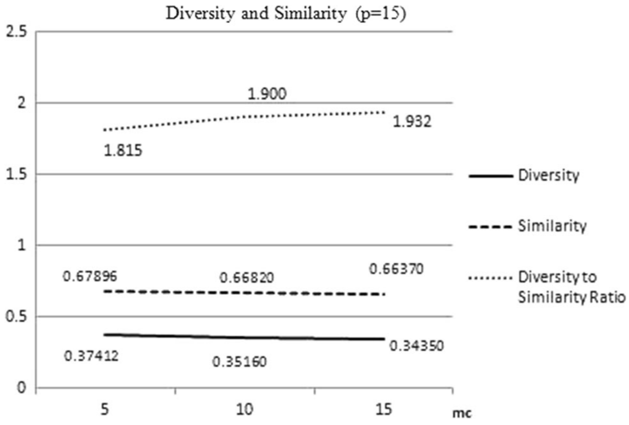


Fig. 4 Graph representation for Similarity and Diversity results ($p = 15$)

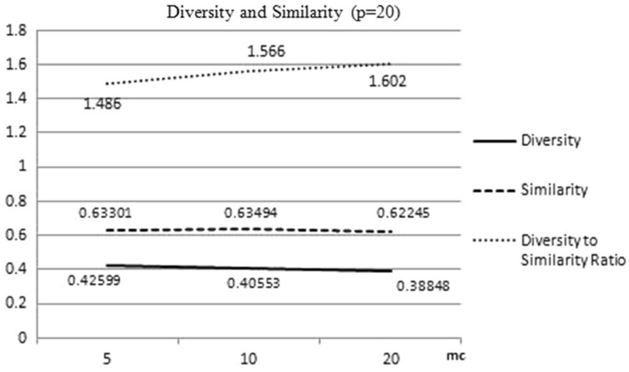


Fig. 5 Graph representation for Similarity and Diversity results ($p = 20$)

$p = 5$, where p is the retrieval set size. However, when $mc = 4$ and $mc = 5$ and the similarity is higher, due to the trade-off between the two measures, the diversity ratio decreases to 0.31407 and 0.25923, respectively (Table 2).

Moreover, the proposed method reports even better results in terms of diversity when tests are performed by randomly selecting 400 cases as queries, as in [68]. As shown in Table 3, the similarity value in this case is 0.692 and the diversity value is 0.391, given that $p = 5$ and $mc = 3$. The results of this method (400Q) are only indicative and the corresponding tests were performed in order to make the exact comparison with the approach of [68].

Here, it should be mentioned that the comparison among all methods is performed for $p = 5$, which refers to the number of recommended cases to the user. This is due to the fact that the exact values of diversity and similarity for the alternative diversity preserving techniques and for different values of p are not available in the literature. However, in [68], it has been reported that the value of 0.4 for the diversity measure is achieved for the Bounded Greedy method when $p = 10$, for the Random method when $p = 23$ and for the Standard method when $p = 46$, whereas for the proposed method it is achieved when $p = 20$, as shown in Table 2.

We also used the financial data set proposed in [51] in order to further evaluate the proposed method. To compare the results of the proposed method to the results of [51] we extract $p = 5$ recommended cases to the user. The experiments showed that our method gives an average diversity level of 0.435 which outperforms the user-match and cosine similarity retrieval methods that are used in [51] exhibiting average diversity values of 0.37 and 0.43, respectively. In [51], the average diversity level could be increased to almost 0.7 if a case revision phase is used (the authors used several revise techniques) where the cases are transformed. For example, they group cases into clusters, and then they consider their centroids. Of course, such transformations introduce a different definition of the similarity–diversity trade-off problem in recommender systems.

5.5 Evaluation of the proposed method

5.5.1 Gap metric

Since there is a trade-off between similarity and diversity, which means that the increase in diversity is usually achieved at the expense of similarity, it would be useful to propose an approach that fully preserves the degree of similarity while at the same time achieves higher levels of diversity.

Table 3 Gap metric results of the proposed method in comparison to alternative techniques

p=5	Standard	Bounded Greedy	Random	Greedy	Proposed Method (L-O-O)	Proposed Method (400 Q.)
	S	BG	R	G	PM	PM
Similarity	0.780	0.753	0.748	0.700	0.732	0.692
Diversity	0.289	0.375	0.326	0.458	0.338	0.390
Gap Metric	0.630	0.502	0.564	0.346	0.538	0.420

In this paper, we introduce the Gap metric that can be adopted to compare any set of such methods since it measures the trade-off directly. It aims to measure the gap between similarity and diversity with respect to similarity. It is calculated by dividing the difference between the value of similarity and diversity by the similarity. Thus, a method is good if the Gap metric is low, i.e., if the difference between similarity and diversity is low while similarity is high. Concerning the proposed method for both the L-O-O and 400Q approaches, we first computed the *Gap metric* for each query separately and then we computed the overall average *Gap metric* that is reported in Table 3.

Based on the Gap metric, we evaluated the alternative methods. The corresponding results of the metric that are presented in Table 3 show that the proposed algorithm (when the leave-one-out method is used) outperforms both the Standard and the Random method and is quite close to the Bounded Greedy technique. However, when the method of the randomly selected 400 queries is adopted, as in [68], the evaluation metric results show that the proposed method outperforms all the alternative methods apart from the Greedy method which however, exhibits a very high time complexity [68]. Thus, in general, the proposed algorithm performs well when it is compared to the alternative diversity preserving techniques, while at the same time it has a lower time complexity.

We also report the similarity level produced by our proposed method for the financial data set proposed in [51] to be 0.76 and thus the Gap metric is 0.428. Note that the corresponding value for the proposed method in [51] cannot be calculated since the authors do not report the corresponding average levels of similarity of recommended cases.

5.5.2 Precision analysis

In order to further evaluate the proposed method, we performed a precision analysis based on the method used in [35]. Precision is widely used in information retrieval in order to evaluate a system's accuracy. It consists of retaining a set T_u of 'ground-truth' cases that are known to be relevant to the user inserted query [35] and evaluating the system's ability to retrieve these items.

However, in contrast to collaborative filtering techniques, in case-based recommendation we do not have information on which set of cases best matches a particular query [35]. Thus, we need to find a set of cases that are relevant to each query. In order to achieve this, we used a query-case similarity function, as proposed in [35]:

$$\epsilon sim_{(q,c)} \tag{10}$$

where ϵ is a small scaling parameter $\epsilon \in [0, 1]$, since only a limited number of cases match each given query. The $sim_{(q,c)}$ is a similarity function of the case to the given query. We used this function in the same way we used it in Table 1 for numeric and categorical attributes. Thus, by applying Equation (7) a relevant set T_u was obtained for each case of the case-base. The proposed recommendation strategy was then tested for $p = 5$ against the relevant cases for each case in turn, on a set of over 500 randomly generated queries.

The adopted metric for precision evaluation is defined in [64] and was also used in [35]:

$$\text{Precision} = \frac{T_u \cap R_u}{p} \tag{11}$$

where T_u is a set of items known to be relevant to the user, R_u is the recommended set of items for user u by the proposed retrieval strategy and p is the number of recommended cases to the user by the retrieval strategy.

The precision was normalized by dividing it by the scaling parameter ϵ. The experimental tests indicated that the precision of the proposed method is almost 80%, a value approaching the precision of the algorithm that constructs the recommendation set only based on the similarity of cases to the given query and better than the precision obtained by the methods tested in [35] which varied from 60–75%.

We have also conducted experimental tests regarding the precision for greater values of p. However, users are more likely to be overwhelmed and confused by recommendation lists that contain a large number of items, as stated in [61]. Also, the limited size of the recommendation list is essential for small display devices such as mobile phones. Moreover, precision deteriorates by the increase in the number of recommended items, p, because of less matches [28, 55, 65]

Indeed, the precision measure indicated that when the value of p increases, the percentage cases that are similar to the query decrease. Therefore, it is not really useful to calculate the precision measure for larger values of p.

6 Conclusions

The application of a facility location model to recommender systems, first studied in this paper, provides effectiveness and flexibility in terms of similarity of the recommending set when compared to the target query and diversity between the recommended cases with each other.

The *mc* parameter constitutes a regulator for the trade-off between similarity and diversity of the recommendation set. When the size of the recommendation set is small, the existence of similar cases to the target query is more probable, thus the increase in the *mc* parameter forces the selection of cases that serve each row (attribute) more times and hence these cases are more similar to each other, resulting in a similarity increase and consequently in the increase in similarity to diversity ratio. When the size of the recommendation set is large, the existence of similar cases to the target query is less probable and thus diversity is high. However, as the *mc* becomes higher, it tends to force the selected cases to serve each row (attribute) more times and this leads to diversity decrease and to increase in the similarity to diversity ratio. Moreover, the proposed approach performs well when compared to the alternative diversity preserving techniques, while at the same time it exhibits very fast computation times with respect to these approaches.

Acknowledgements The authors wish to thank Aristotelis Kompothrekas, PhD candidate in Patras University, for his valuable contribution in performing the experimental tests. The publication of this article has been financed by the Research Committee of the University of Patras.

Compliance with ethical standards

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

References

1. Abbassi Z, Mirrokni VS, Thakur M (2013) Diversity maximization under matroid constraints. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 32–40
2. Adomavicius G, Kwon Y (2011) Improving aggregate recommendation diversity using ranking-based techniques. IEEE Trans Knowl Data Eng 24(5):896–911
3. Adomavicius G, Manouselis N, Kwon Y (2011) Multi-criteria recommender systems. In: Ricci F, Rokach L, Shapira B, Kantor P (eds) Recommender systems handbook. Springer, pp 769–803
4. Agrawal R, Mannila H, Srikant R, Toivonen H, Verkamo AI et al (1996) Fast discovery of association rules. Adv Knowl Discov Data Min 12(1):307–328
5. Ashkan A, Kveton B, Berkovsky S, Wen Z (2015) Optimal greedy diversity for recommendation. IJCAI 15:1742–1748
6. Aytekin T, Karakaya MÖ (2014) Clustering-based diversity improvement in top-n recommendation. J Intell Inf Syst 42(1):1–18
7. Bag S, Ghadge A, Tiwari MK (2019) An integrated recommender system for improved accuracy and aggregate diversity. Comput Ind Eng 130:187–197
8. Ben-Dor A, Chor B, Karp R, Yakhini Z (2003) Discovering local structure in gene expression data: the order-preserving submatrix problem. J Comput Biol 10(3–4):373–384
9. Bennett J, Lanning S et al (2007) The netflix prize. In: Proceedings of KDD cup and workshop, vol. 2007, p 35. New York
10. Bergmann S, Ihmels J, Barkai N (2003) Iterative signature algorithm for the analysis of large-scale gene expression data. Phys Rev E 67(3):031902
11. Berkhin P (2006) A survey of clustering data mining techniques. In: Kogan J, Nicholas C, Teboulle M (eds) Grouping multidimensional data. Springer, pp 25–71
12. Bobadilla J, Serradilla F, Hernando A et al (2009) Collaborative filtering adapted to recommender systems of e-learning. Knowl Based Syst 22(4):261–265
13. Boim R, Milo T, Novgorodov S (2011) Diversification and refinement in collaborative filtering recommender. In: Proceedings of the 20th ACM international conference on information and knowledge management, pp 739–744
14. Bousbahi F, Chorfi H (2015) Mooc-rec: a case based recommender system for moocs. Procedia Soc Behav Sci 195:1813–1822
15. Boutsinas B (2013) Machine-part cell formation using biclustering. Eur J Oper Research 230(3):563–572
16. Boutsinas B (2013) A new biclustering algorithm based on association rule mining. Int J Artif Intell Tools 22(03):1350017
17. Bradley K, Smyth B (2001) Improving recommendation diversity. In: Proceedings of the twelfth Irish conference on artificial intelligence and cognitive science, Maynooth, Ireland, pp 85–94
18. Busygin S, Prokopyev O, Pardalos PM (2008) Biclustering in data mining. Comput Oper Res 35(9):2964–2987
19. Chen HC, Chen AL (2005) A music recommendation system based on music and user grouping. J Intell Inf Syst 24(2–3):113–132
20. Cheng Y, Church G (2000) Biclustering of expression data. In: Ismb, vol 8, pp 93–103
21. Cho YH, Kim JK, Kim SH (2002) A personalized recommender system based on web usage mining and decision tree induction. Expert Syst Appl 23(3):329–342
22. Choi SM, Han YS (2010) A content recommendation system based on category correlations. In: 2010 Fifth international multiconference on computing in the global information technology, IEEE, pp 66–70
23. Devi MK, Samy RT, Kumar SV, Venkatesh P (2010) Probabilistic neural network approach to alleviate sparsity and cold start problems in collaborative recommender systems. In: 2010 IEEE international conference on computational intelligence and computing Research, IEEE, pp 1–4
24. Dolnicar S, Kaiser S, Lazarevski K, Leisch F (2012) Biclustering: Overcoming data dimensionality problems in market segmentation. J Travel Res 51(1):41–49
25. Fesenmaier DR, Ricci F, Schaumlechner E, Wöber K, Zanella C et al (2003)DIETORECS: Travel advisory for multiple decision styles. na
26. Fleder DM, Hosanagar K (2007) Recommender systems and their impact on sales diversity. In: Proceedings of the 8th ACM conference on Electronic commerce, pp 192–199
27. Golbeck J (2006) Generating predictive movie recommendations from trust in social networks. In: International conference on trust management, Springer, pp 93–104

28. Gunawardana A, Shani G (2009) A survey of accuracy evaluation metrics of recommendation tasks. J Mach Learn Res 10:2935–2962
29. Hakimi SL (1964) Optimum locations of switching centers and the absolute centers and medians of a graph. Oper Res 12(3):450–459
30. Hammar M, Karlsson R, Nilsson BJ (2013) Using maximum coverage to optimize recommendation systems in e-commerce. In: Proceedings of the 7th ACM conference on Recommender systems, pp 265–272
31. Hartigan JA (1972) Direct clustering of a data matrix. J Am Stat Assoc 67(337):123–129
32. He X, He Z, Song J, Liu Z, Jiang YG, Chua TS (2018) Nais: Neural attentive item similarity model for recommendation. IEEE Trans Knowl Data Eng 30(12):2354–2366
33. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22(1):5–53
34. Huang Z, Shan G, Cheng J, Sun J (2019) Trec: An efficient recommendation system for hunting passengers with deep neural networks. Neural Comput Appl 31(1):209–222
35. Hurley N, Zhang M (2011) Novelty and diversity in top-n recommendation-analysis and evaluation. ACM Trans Internet Technol (TOIT) 10(4):14
36. Ibnkahla M (2000) Applications of neural networks to digital communications-a survey. Signal processing 80(7):1185–1215
37. Kim JK, Cho YH, Kim WJ, Kim JR, Suh JH (2002) A personalized recommendation procedure for internet shopping support. Electron Commer Res Appl 1(3–4):301–313
38. Lathia N, Hailes S, Capra L, Amatriain X (2010) Temporal diversity in recommender systems. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pp 210–217
39. Lazzeroni L, Owen A (2002) Plaid models for gene expression data. Statistica Sinica 12(1):61–86
40. Lee JS, Lee JC (2007) Context awareness by case-based reasoning in a music recommendation system. In: International symposium on ubiquitious computing systems, Springer, pp 45–58
41. Lee K, Lee K (2015) Escaping your comfort zone: a graph-based recommender system for finding novel recommendations among relevant items. Expert Syst Appl 42(10):4851–4858
42. Lee M, Choi P, Woo Y (2002) A hybrid recommender system combining collaborative filtering with neural network. In: International conference on adaptive hypermedia and adaptive web-based systems, Springer, pp 531–534
43. Li X, Murata T (2012) Multidimensional clustering based collaborative filtering approach for diversified recommendation. In: 2012 7th International conference on computer science & education (ICCSE), IEEE, pp 905–910
44. Lorenzi F, Ricci F (2003) Case-based recommender systems: A unifying view. In: IJCAI workshop on intelligent techniques for web personalization, Springer, pp 89–113
45. Madeira SC, Oliveira AL (2004) Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans Comput Biol Bioinform (TCBB) 1(1):24–45
46. McSherry D (2002) Diversity-conscious retrieval. In: European conference on case-based reasoning, Springer, pp 219–233
47. Megiddo N et al. (1986) On the complexity of linear programming. IBM Thomas J, Watson Research Division
48. Mucherino A, Papajorgji P, Pardalos PM (2009) A survey of data mining techniques applied to agriculture. Oper Res 9(2):121–140
49. Murali T, Kasif S (2002) Extracting conserved gene expression motifs from gene expression data. In: Biocomputing 2003, World Scientific, pp 77–88
50. Musto C, Semeraro G (2015) Case-based recommender systems for personalized finance advisory. In: FINREC, pp 35–36
51. Musto C, Semeraro G, Lops P, De Gemmis M, Lekkas G (2015) Personalized finance advisory through case-based recommender systems and diversification strategies. Decis Support Syst 77:100–111
52. Orzechowski P, Boryczko K (2016) Text mining with hybrid biclustering algorithms. In: International conference on artificial intelligence and soft computing, Springer, pp 102–113
53. Panteli A, Boutsinas B, Giannikos I (2019) On solving the multiple p-median problem based on biclustering. Oper Res Int J. https://doi.org/10.1007/s12351-019-00461-9
54. Paradarami TK, Bastian ND, Wightman JL (2017) A hybrid recommender system using artificial neural networks. Expert Syst Appl 83:300–313
55. Peker S, Kocyigit A (2016) An adjusted recommendation list size approach for users' multiple item preferences. In: International conference on artificial intelligence: methodology, systems, and applications, Springer, pp 310–319
56. Pensa RG, Robardet C, Boulicaut JF (2005) A bi-clustering framework for categorical data. In: European conference on principles of data mining and knowledge discovery, Springer, pp 643–650
57. Perugini S, Gonçalves MA, Fox EA (2004) Recommender systems research: a connection-centric survey. J Intell Inf Syst 23(2):107–143
58. Prakash P (2015) Decision support system in heart disease diagnosis by case based recommendation. Int J Sci Technol Res 4:51–55
59. Prelić A, Bleuler S, Zimmermann P, Wille A, Bühlmann P, Gruissem W, Hennig L, Thiele L, Zitzler E (2006) A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22(9):1122–1129
60. Premchaiswadi W, Poompuang P, Jongswat N, Premchaiswadi N (2013) Enhancing diversity-accuracy technique on user-based top-n recommendation algorithms. In: 2013 IEEE 37th annual computer software and applications conference workshops, IEEE, pp 403–408
61. Pu P, Faltings B, Chen L, Zhang J, Viappiani P (2011) Usability guidelines for product recommenders based on example critiquing research. In: Recommender systems handbook, Springer, pp 511–545
62. Ribeiro MT, Ziviani N, Moura ESD, Hata I, Lacerda A, Veloso A (2014) Multiobjective pareto-efficient approaches for recommender systems. ACM Trans Intell Syst Technol (TIST) 5(4):1–20
63. Ricci F, Rokach L, Shapira B (2011) Introduction to recommender systems handbook. In: Recommender systems handbook, Springer, pp 1–35
64. Sarwar B, Karypis G, Konstan J, Riedl J (2000) Application of dimensionality reduction in recommender system-a case study. Minnesota Univ Minneapolis Dept of Computer Science, Tech. rep
65. Sarwar B, Karypis G, Konstan J, Riedl J et al (2000) Analysis of recommendation algorithms for e-commerce. In: EC, pp 158–167
66. Shi L (2013) Trading-off among accuracy, similarity, diversity, and long-tail: A graph-based recommendation approach. RecSys '13, p. 57–64. Association for Computing Machinery, New York, NY, USA
67. Shih DH, Yen DC, Lin HC, Shih MH (2011) An implementation and evaluation of recommender systems for traveling abroad. Expert Syst Appl 38(12):15344–15355
68. Smyth B, McClave P (2001) Similarity vs. diversity. In: International conference on case-based reasoning, Springer, pp 347–361

69. Sturges HA (1926) The choice of a class interval. J Am Stat Assoc 21(153):65–66
70. Tanay A, Sharan R, Shamir R (2002) Discovering statistically significant biclusters in gene expression data. Bioinformatics 18(suppl-1):S136–S144
71. Tran T, Atas M, Felfernig A, Stettinger M (2018) An overview of recommender systems in the healthy food domain. J Intell Inf Syst 50(3):501–526
72. Vandromme M, Jacques J, Taillard J, Jourdan L, Dhaenens C (2020) A biclustering method for heterogeneous and temporal medical data. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2020.2983692
73. Yang J, Wang W, Wang H, Yu P (2002) d-clusters: Capturing subspace correlation in a large data set. In: icde, IEEE, p. 0517.
74. Zhang M, Hurley N (2008) Avoiding monotony: improving the diversity of recommendation lists. In: Proceedings of the 2008 ACM conference on recommender systems, pp 123–130
75. Zhang M, Hurley N (2009) Novel item recommendation by user profile partitioning. In: 2009 IEEE/WIC/ACM international joint conference on web intelligence and intelligent agent technology, vol. 1. IEEE, pp 508–515
76. Zheng K, Wang H, Qi Z, Li J, Gao H (2017) A survey of query result diversification. Knowl Inf Syst 51(1):1–36
77. Zhou T, Kuscsik Z, Liu JG, Medo M, Wakeling JR, Zhang YC (2010) Solving the apparent diversity-accuracy dilemma of recommender systems. Proc Natl Acad Sci 107(10):4511–4515
78. Ziegler CN, McNee SM, Konstan JA, Lausen G (2005) Improving recommendation lists through topic diversification. In: Proceedings of the 14th international conference on world wide web, WWW '05, pp 22–32. ACM, New York, NY, USA
79. Ziegler CN, 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

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