

Collaborative Recommendation with Multi-Criteria Ratings

Wei-Guang Teng* Hsin-Hsien Lee

Department of Engineering Science

National Cheng Kung University

Tainan 701, Taiwan, ROC

Email: wgteng@mail.ncku.edu.tw

Received June 15, 2007 ; Accepted June 30, 2007

Abstract. Recommendation systems utilize data analysis techniques to the problem of helping users find the items they would like. Example applications include the recommendation systems for movies, books, CDs and many others. As recommendation systems emerge as an independent research area, the rating structure plays a critical role in recent studies. Among many alternatives, the collaborative filtering algorithms are generally accepted to be successful to estimate user ratings of unseen items and then to derive proper recommendations. In this paper, we extend the concept of single criterion ratings to multi-criteria ones, i.e., an item can be evaluated in many different aspects. For example, the goodness of a restaurant can be evaluated in terms of its food, decor, service and cost. Since there are usually conflicts among different criteria, the recommendation problem cannot be formulated as an optimization problem any more. Instead, we propose in this paper to use data query techniques to solve this multi-criteria recommendation problem. Empirical studies show that our approach is of both theoretical and practical values.

Keywords: recommendation system, collaborative filtering, multi-criteria rating

1 Introduction

As the information technology advances, people usually encounter the problem of information overloading. Also, people always have their own preference when choosing what they need. To connect such personal opinions with available data items of a huge amount imposes a great challenge for researchers. Prior studies have shown that recommendation systems can help to resolve difficulties by providing users recommendations on choosing data items. Specifically, recommendation systems apply data analysis techniques to the problem of helping users find the items they would like to favor the most. Examples of such applications include recommending books, CDs, movies, news articles to users in many e-commerce websites.

Recommendations can be based on demographics of the users, overall top selling items, or past buying habit of users as a predictor of future items. Nevertheless, as recommendation systems emerge as an independent research area in recent years, researchers started focusing on recommendation methods explicitly on rating structures. Specifically, each user can provide a rating for a specific item to express the degree of preference. In prior studies, many techniques are developed to estimate ratings of unseen items for the user who is to be recommended and thus to recommend items with the highest estimated rating. Among many alternatives, the collaborative filtering algorithms are generally accepted to be the most successful recommendation techniques to date. The basic idea of collaborative filtering algorithms is to identify similarities among users or items, and then to generate rating estimations by utilizing the similarity information.

In this paper, we focus on extending the concept of single criterion ratings to multi-criteria ones, i.e., an item can be evaluated in many different aspects. For example, the goodness of a restaurant can be evaluated in terms of its food, decor, service and cost. This extension of multi-criteria ratings has significant impacts on existing recommendation techniques. First of all, in our opinion, the recommendation problem cannot be formulated as an optimization problem any more since there are usually conflicts among different criteria. In other words, since people may usually hesitate to make decisions when there is no overwhelmingly good item among all the candidates, it is impractical to have such an optimal item as the recommendation. Secondly, a high dimensional space is required to represent all of the candidate items in terms of multiple rating criteria. Lastly but not the least, more efforts have to be done in addition to the rating estimations for a feasible recommendation technique.

Consequently, we propose in this paper to use data query techniques to solve this multi-criteria recommendation problem. The idea is that instead of choose the optimal candidates as recommendations, our approach filters

* Correspondence author

out relatively poor candidates which are dominated by some other candidates. Therefore, the multi-criteria recommendation problem can be formulated as a skyline query problem. To illustrate the usefulness and to evaluate the feasibility of our approach, some real datasets generated in practical applications are collected and used in our examples.

The rest of the paper is organized as follows. An overview of the recommendation systems and relevant recommendation techniques are explored in Section 2. The multi-criteria recommendation problem and our approach for solving this problem are developed in Section 3. Empirical studies of utilizing our approach on practical applications are conducted in Section 4. This paper concludes with Section 5.

2 Related Works on Recommendation Techniques

Recommendation systems have attracted an increasing amount of research interests in recent years. To explore the usage of recommendation systems, an overview is provided in Section 2.1. To focus on our discussion on collaborative recommendation systems, we further investigate the two popular techniques, i.e., the user-based and the item-based collaborative filtering, in Section 2.2.

2.1 Overview of Recommendation Systems

A recommendation system is a system that assists a user in evaluating items by providing predictions and opinions as a recommendation [1][2]. Examples of such applications include recommending books, CDs, movies, news articles to users in many e-commerce websites. Note that recommendation systems range from manually operated ones where users can recommend items to other users by writing reviews, to automated ones where a list of recommended items can be generated automatically for individual users [3]. Moreover, techniques on improving the effectiveness of a recommendation system constitute a problem-rich research area since they help users to deal with information overload and offer personalized recommendations, content, and services.

A formal definition of the recommendation problem [4] can be illustrated as follows. Let C be the set of all users and let S be the set of all items to be recommended, such as books, movies, or restaurants. Then, the recommendation problem is to choose items $s_{c_i} \in S$ which maximize the utility for each user $c_i \in C$. Specifically,

$$\forall c_i \in C, s_{c_i} = \arg \max_{s \in S} u(c_i, s) \quad (1)$$

where u be an utility function that measures the usefulness of item s to user c , that is, $u: C \times S \rightarrow R$, where R is a totally ordered set. Consequently, to design an efficient and effective approach to resolve this recommendation problem poses a great challenge to researchers.

In relevant studies, recommendation systems can be usually classified into the following two categories, based on how recommendations are made [5].

- (1) Content-based recommendations: the user will be recommended items similar to the ones the user preferred in the past; and
- (2) Collaborative recommendations: the user will be recommended items that people with similar tastes and preferences liked in the past.

The main idea of the content-based methods is that the recommendation systems recommend the items based on what the user has looked at recently or has purchased in the past. Specifically, the utility $u(c_i, s)$ of item s for user c_i is estimated based on the utilities $u(c_i, s_j)$ assigned by user c_i to items $s_j \in S$ which are *similar* to item s . For example, to recommend movies to user c_i , the content-based recommendation system tries to discover the commonalities (e.g., specific actors, directors, or subject matters) among the movies which user c_i has rated highly in the past. Then, only the unseen movies which have a high degree of similarity to the preferences of user c_i would be recommended.

A major drawback of the content-based methods is that there is no interaction among users. When a new item which is not similar to any of the preferred items of a specific user, it cannot be recommended to that user by the content-based methods. However, if this new item is popular with many other people, a recommendation is likely to be made in practice. To generalize this idea, the collaborative methods recommend items to a specific user based on the opinions of other like-minded users. Specifically, the utility $u(c_i, s)$ of item s for user c_i is estimated based on the utilities $u(c_j, s)$ assigned by those users $c_j \in C$ who are *similar* to user c . For example, to recommend movies to user c_i , the collaborative recommendation system tries to find *peers* of user c_i (i.e., other users who have similar tastes in movies and thus rate the same movies similarly). Then, only the unseen movies that are most highly rated by the peers of user c_i would be recommended.

In prior studies, several recommendation systems are developed using a hybrid approach [4] by combining

collaborative and content-based methods. This hybrid approach reduces certain limitations of either content-based or collaborative methods, and is used for recommending books, CDs, and news articles in practice [6].

2.2 Collaborative Filtering Recommendation System

The collaborative recommendation method or more specifically, the collaborative filtering (abbreviated as CF) technique is generally accepted to be a successful recommendation technique to date. We thus focus on extending the capabilities of CF recommendation systems in this paper. In addition, as recommendation systems emerge as an independent research area in recent years, researchers started focusing on recommendation methods explicitly on rating structures. As shown in Fig.1, the ratings collected from users can be represented as a User \times Item matrix, with an entry r_{ui} representing either the rating user u gave to item i , if he rated it, or *null* otherwise.

$$\begin{array}{c} \text{User} \\ \left[\begin{array}{cccc} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & & & \\ \vdots & & \ddots & \\ r_{m1} & & & r_{mn} \end{array} \right] \end{array}$$

Fig.1. The User \times Item matrix ($m \times n$)

Intuitively, once the ratings for the yet unrated items are estimated, the user can be recommended with the item(s) having the highest estimated rating. Consequently, different recommendation techniques use different methods for estimating ratings of unrated items. In prior works, two most popular types of CF algorithms are developed:

- (1) User-based CF algorithms build for each user a neighborhood of users (i.e., peers) with similar ratings in the User \times Item matrix. Ratings from these users are then employed to generate estimations.
- (2) Item-based CF algorithms compute a set of similar items for each item according to the contents of the User \times Item matrix and use these similarities to generate estimations.

Since either user-based or item-based algorithms rely on similarity computations, the k-nearest neighbors (kNN) algorithm is then commonly used [7] to estimate item ratings for the target user. The entire User \times Item matrix is used to generate such estimations.

It is noted that there are a few alternatives of similarity calculations. Two most popular ones are presented here, i.e., the correlation-based similarity and the cosine-based similarity. Firstly, the Pearson correlation formula in statistics can be used for computing similarities. For example, when measuring the similarity between user i and user j , we can have the following equation.

$$\text{corr}(i, j) = W_{ij} = \frac{\sum_{k \in I} (r_{ik} - \bar{r}_i) * (r_{jk} - \bar{r}_j)}{\sqrt{\sum_{k \in I} (r_{ik} - \bar{r}_i)^2 \sum_{k \in I} (r_{jk} - \bar{r}_j)^2}} \quad (2)$$

where I is the set of items user i and j both rated, r_{ik} is the rating user i gave to item k , and \bar{r}_i is the average rating of user i . Secondly, another similarity metric can also be used by computing the cosine value between two vectors. For example, when computing similarities between items, each item is thought of as a vector in the $|users|$ multidimensional space, and the similarity between item i and j can be computed using the cosine-based similarity.

$$\text{cos}(i, j) = W_{ij} = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\| \cdot \|\vec{j}\|} \quad (3)$$

It is noted that W_{ij} is shown in both equations (2) and (3) since the calculated similarities are taken as weighting values when performing kNN in the next step for estimating ratings as weighted sums. Specifically, the top k items (or users) which are most similar to the target item (or user) are selected, and finally the predictions for user i and item i are computed using the formula of weighted sum. Note that when utilizing the user-based CF

algorithms, the previous ratings given by the target user are also referenced. In general, the goal is to predict the utility of a certain item for a particular user based on his previous likings and the opinions of other like-minded users. As a result, we have the following equations for both user-based and item-based CF algorithms.

$$P_{i,a} = r_i + \frac{\sum_{j=1}^k W_{ij}(r_{ja} - \bar{r}_j)}{\sum_{j=1}^k W_{ij}} \quad \text{for user-based CF algorithms} \quad (4a)$$

$$P_{u,i} = \frac{\sum_{\text{all_similar_items}, a} (W_{ia} \times r_{ua})}{\sum_{\text{all_similar_items}, a} |W_{ia}|} \quad \text{for item-based CF algorithms} \quad (4b)$$

where $P_{u,a}$ stands for the predicted rating of item a for user u .

To further illustrate the concepts of these two CF algorithms, the overall process of generating recommendations using either user-based or item-based CF algorithms can be best understood in the following examples.

Example 1 (User-based CF recommendations): Assume that ratings on some movies are collected as shown in Table 1. Each entry r_{ij} is ranged from 0 to 10 where a higher value means a better perception of user $_i$ toward the j -th movie.

Table 1. An example of movie ratings given by some users

	Star Wars	Hoop Dream	Contact	Titanic	Matrix	007	Face/Off
User ₁	10	2	9	8	null	null	8
User ₂	10	4	null	2	6	8	10
User ₃	8	null	6	10	3	null	4
User ₄	null	null	4	8	null	10	10
User ₅	6	8	null	null	null	10	8
User ₆	10	4	4	?	?	?	6

To recommend a movie to user₆, we need to estimate the ratings of the three unseen movies, i.e., Titanic, Matrix and 007. If the user-based CF algorithm is utilized and the Pearson correlation is adopted as the similarity metric, we can calculate the similarity value for user₁ and user₆ using equation (2). To facilitate next steps, the mean values are calculated based on co-rated items first, i.e., the movies are rated by both user₆ and user $_i$ for $i=1\sim5$ respectively.

$$\bar{r}_6 = \frac{10+4+4+6}{4} = 6, \bar{r}_1 = \frac{10+2+9+8}{4} = 7.25, \bar{r}_2 = \frac{10+4+10}{3} = 8, \\ \bar{r}_3 = \frac{8+6+4}{3} = 6, \bar{r}_4 = \frac{4+10}{2} = 7, \text{ and } \bar{r}_5 = \frac{6+8+8}{3} = 7.33.$$

Thus,

$$\text{corr}(user_1, user_6) = \frac{(10-6)(10-7.25) + (4-6)(2-7.25) + (4-6)(9-7.25) + (6-6)(8-7.25)}{\sqrt{(10-6)^2 + (4-6)^2 + (4-6)^2 + (6-6)^2} \sqrt{(10-7.25)^2 + (2-7.25)^2 + (9-7.25)^2 + (8-7.25)^2}} \\ = 0.52 = W_{16}.$$

Similarly, we can have $W_{26} = 0.90$, $W_{36} = 0.56$, $W_{46} = 1$ and $W_{56} = -0.81$. Moreover, if $k = 3$ is selected for the kNN algorithm, it is obviously that user₄, user₂ and user₃ are the top-3 users who have the most similar opinions with user₆. Next, we can estimate the rating of the movie ‘‘Titanic’’ for user₆ using equation (4a) as

$$P_{user_6, \text{Titanic}} = 6 + \frac{1 \times (8-7) + 0.90 \times (2-8) + 0.56 \times (10-6)}{1 + 0.90 + 0.56} = 5.12.$$

Similarly, we can have $P_{user_6, \text{Matrix}} = 3.62$ and $P_{user_6, \text{007}} = 7.58$. Thus, the movie ‘‘007’’ is recommended to user₆ since it has the highest estimated rating among three candidates.

Example 2 (Item-based CF recommendations): As in Example 1, the problem of recommending user₆ a good movie is to be solved here using the item-based CF algorithm. If the cosine-based similarity is adopted, we can calculate the similarity value between two movies using equation (3). Note that since there are three candidate movies, each of them is examined respectively. Firstly, we can have the similarity of “Titanic” and “Star Wars” as

$$\cos(\text{"Star Wars"}, \text{"Titanic"}) = W_{14} = \frac{(10 \times 8 + 10 \times 2 + 8 \times 10)}{\sqrt{10^2 + 10^2 + 8^2} * \sqrt{8^2 + 2^2 + 10^2}} = 0.85.$$

Note that this item-item similarity is computed by looking into *co-rated items* only, i.e., *null* terms are ignored when using equation (2). Similarly, we can have $W_{24} = 0.65$, $W_{34} = 0.94$ and $W_{47} = 0.80$. Thus, the top-3 movies whose ratings are most similar to those of “Titanic” are “Contact”, “Star Wars” and “Face/Off”. Next, we can estimate the rating of the movie “Titanic” for user₆ using equation (4b) as

$$P_{\text{user}_6, \text{"Titanic"}} = \frac{4 \times 0.94 + 10 \times 0.85 + 6 \times 0.80}{0.94 + 0.85 + 0.80} = 6.59.$$

Similarly, we can have $P_{\text{user}_6, \text{Matrix}} = 4.67$ and $P_{\text{user}_6, 007} = 4.66$. Thus, the movie “Titanic” is recommended to user₆ since it has the highest estimated rating.

One may notice that the results of Example 1 and Example 2 are not identical. This is because different algorithms are adopted to generate recommendations. In practice, there are usually different opinions from different recommendation systems which suggest that more than one item could be good candidate to a specific user. More insights of this issue will be explored in following sections.

In addition to the above techniques which operates on the entire User \times Item matrix, some other techniques generate recommendations by utilizing different models, including Bayesian networks [8], clustering [9], horting [10], and association rules [11][12]. Bayesian networks create a model based on a training set with a decision tree at each node and edges representing user information. This approach may prove practical for environments in which user preferences change slowly with respect to the time needed to build the model, but is not suitable for environments in which user preferences must be updated frequently. Clustering techniques identify groups of users who appear to have similar preferences. Therefore, recommendations for a target user can be made by averaging the opinions of the other users in that cluster once the clusters are created. However, clustering techniques usually produce less-personal recommendations than other methods. Horting is a graph-based technique in which nodes are users, and edges between nodes indicate degree of similarity between two users [10]. Predictions are produced by walking the graph to nearby nodes and combining the opinions of the nearby users. Algorithms to discover association rule are also used in some prior works to find associations between co-purchased items and then generates item recommendation based on the strength of the association between items [12].

3 Incorporating Multi-Criteria Ratings in Recommendation Systems

As mentioned in Section 2, the rating structure plays a crucial role in developing recommendation systems. However, if this rating structure is generally extended to be a multi-dimensional one, recommendation algorithms developed in prior works may encounter some conflict. To further investigate the problem of multi-criteria ratings, corresponding issues are generally explored in Section 3.1. The techniques used for multi-criteria decision analysis are also discussed in Section 3.1. Moreover, we develop in Section 3.2 an extensive approach to solve the multi-criteria recommendation problem by utilizing the concept of skyline queries.

3.1 From Single Criterion to Multi-Criteria Ratings

Most of the current recommendation systems deal with single criterion ratings, such as ratings of movies and books, as described in Section 2. Nevertheless, in some applications, such as recommendation systems for restaurants, it is crucial to incorporate multi-criteria ratings into recommendation methods [4]. For example, many restaurant guides, such as Zagat Survey [13], provide four criteria for restaurant ratings: food, decor, service and cost. To extend the rating structure with such multi-criteria ratings, an example is provided in Table 2. One may notice that when the concept of multi-criteria ratings is introduced, each user may give two or more ratings to an item to express his degree of preference on different aspects. For simplicity, only two criteria, i.e., food and decor of the restaurants, are listed in Table 2. Nevertheless, more than two criteria can be generally considered in practical applications.

The recommendation problem becomes more complicated with this structure of multi-criteria ratings. The intrinsic difficulty comes from the possibly conflicting criteria, e.g., the food of a restaurant is delicious while the decor is primitive. Moreover, when there is no overwhelmingly good item among all the candidates, one may usually hesitate to make decisions. Consequently, it is a great challenge to develop a proper recommendation technique to guide people to make choices.

Table 2. An example of restaurant ratings given by some users on (a) food; and (b) decor

Food	R ₁	R ₂	R ₃	R ₄	R ₅	R ₆
User ₁	6	8	null	4	8	8
User ₂	null	4	6	null	null	8
User ₃	6	null	4	6	4	null
User ₄	null	2	8	null	2	2
User ₅	null	8	4	8	2	6
User ₆	6	8	4	?	?	?

(a)

Decor	R ₁	R ₂	R ₃	R ₄	R ₅	R ₆
User ₁	10	8	null	10	8	6
User ₂	null	2	4	null	null	2
User ₃	10	null	6	10	2	null
User ₄	null	6	4	null	2	6
User ₅	null	8	6	4	10	8
User ₆	10	8	6	?	?	?

(b)

To the best of our knowledge, although multi-criteria ratings have not yet been examined in prior works on recommendation systems, they have been extensively studied in the *operations research* community [14][15]. Specifically, there are several methods proposed in prior works to resolve the issue of multi-criteria decision analysis (abbreviated as MCDA). A typical problem in MCDA [16] is concerned with the task of ranking a finite number of decision alternatives, each of which is explicitly described in terms of different attributes (or decision criteria) which have to be taken into account simultaneously. In practice, an alternative might be better than another alternative in terms of one criterion, and worse in terms of another criterion. As one can imagine, a real-life engineering problem can usually involve the need to evaluate alternatives in terms of conflicting criteria.

To remedy this problem, the possible alternatives (or options) and corresponding criteria are represented as a two-dimensional *decision matrix*. As the rows and the columns of a decision matrix represent decision alternatives and criteria respectively, a value found at the intersection of a row and a column represents a criterion outcome. Note that the different weights for each criterion may be provided to reflect the relative importance when making decisions among all the alternatives. Thus, typical solutions to the multi-criteria optimization problem in MCDA take a linear (weighted) combination of multiple criteria, and thus reduce the problem to a single-criterion optimization problem.

In our opinion, the above approach used in MCDA cannot be directly applied in the multi-criteria recommendation problem as discussed in this paper. The major difficulty is that the weights of different criteria vary with different user and vary with time. For example, some people may think that price is the most important criterion when choosing restaurants, while others may not think so. Also, one usually can afford higher prices to dine in a restaurant with wonderful decor in certain anniversaries than usual days. Consequently, a proper recommendation technique should not assume that the weights of all criteria are time-invariant and exactly known. Moreover, the naive approach of taking the weighted average of all criteria as a summarized score results in less information and fails to provide proper recommendations. Therefore, we shall develop in this paper a novel view and an innovative approach for solving the multi-criteria recommendation problem.

3.2 Extending CF Techniques for Multi-Criteria Recommendations

As mentioned in Section 2, the CF techniques are accepted to be successful to solve the recommendation problem with single criterion. When this recommendation problem is extended to be with multiple conflicting criteria, the CF techniques can still be utilized to estimate the ratings for each criterion, respectively. However, equation

(1) may become questionable with the problem of multi-criteria recommendation, since it suggests the candidate with the maximum utility, i.e., estimated rating, should be recommended. In the case of multi-criteria recommendation, there is usually no such optimal candidate which is with the highest (estimated) ratings in terms of all criteria. For a better illustration, the multi-dimensional space is utilized for representing all the candidates. Specifically, each criterion forms a dimension while each candidate maps to a point whose coordinates are determined by its estimated ratings. This idea can be best understood with the following example.

Example 3: Suppose that a restaurant can be evaluated using two criteria, i.e., food and decor. User ratings are collected as shown in Table 2. To recommend restaurants for user₆, ratings of the three unvisited restaurants, i.e., R₄, R₅ and R₆, are estimated. The user-based CF algorithm is used for rating estimation in this example, and the estimated results are {R₄: 6.2, R₅: 4.9, R₆: 6.5} for food, and {R₄: 8.5, R₅: 7.1, R₆: 7.5} for decor. Note that the ratings are estimated on each criterion, respectively. As a result, these estimated ratings of the three restaurants can be mapped to three points in the 2-dimensional space as shown in Fig.2.

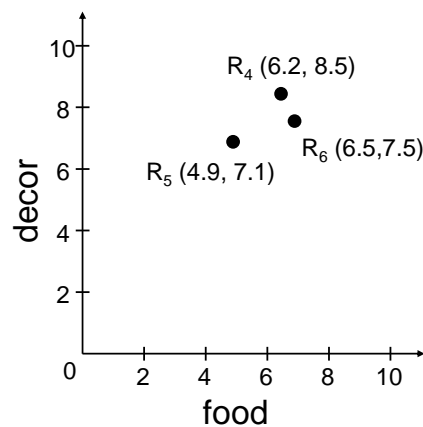


Fig.2. Transforming multi-criteria ratings into points in a multi-dimensional space

In Example 3, when making recommendations to user₆, the restaurant R₅ is obviously a poor candidate since it is dominated by either R₄ or R₆. Nevertheless, R₄ and R₆ are both good candidates since R₄ is with better decor and R₆ is with better food. Consequently, in our opinion, recommendations with multiple conflicting criteria are no longer an optimization problem, but a *data query* problem. In general, we propose in this paper the concept of regarding the recommendation problem as discovering a few good items among a large number of candidates. In other words, a proper technique should be developed to filter out relatively poor candidates in terms of all criteria.

Note that our idea can be formulated as the problem of skyline queries [17] in the database field. Formally speaking, given a d -dimensional dataset, a point p is said to dominate another point q if it is better than or equal to q in all dimensions and better than q in at least one. A skyline is a subset of points that are not dominated by any others. Skyline queries, which return skyline points, are thus useful in many decision making applications that involve high dimensional datasets. In this paper, the problem of multi-criteria recommendations can be taken as a skyline query problem when the ratings are estimated.

Generally speaking, all the candidate items can be represented as points in the multi-dimensional space. For simplicity, we assume that *as a user likes an item more, the corresponding rating gets higher for all criteria*. Thus, to make proper recommendations, it is necessary to identify skyline points among numerous candidates. A simple example with two criteria is shown in Fig. 3. Items E, F, G and H are considered poor candidates since each of these points is dominated by some other points. Thus, items A, B, C and D are considered good candidates, i.e., skyline points, to be recommended to the user. Moreover, the skyline is formed by connecting these four points A, B, C and D. Note that similar concepts can be easily extended to a k -dimensional space where k stands for the number of criterion.

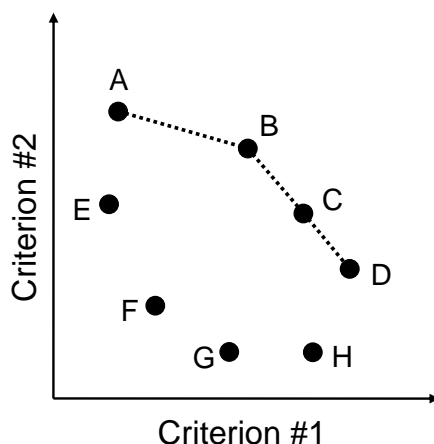


Fig.3. An example of identifying skyline points from numerous candidates

4 Empirical Studies

Many real datasets of user ratings on different topics can be found on the Internet, including the MovieLens movie ratings [18], the Jester joke ratings [19], the Book-Crossing book ratings [20], the Parliament voting [21] and many others. However, most of the available real datasets are with single criterion, i.e., each user can only have one rating for a specific item. To further investigate the problem of multi-criteria recommendations, some closely relevant datasets can be found at the Zagat Survey [13]. It is a worldwide provider of consumer survey on dining, travel and leisure information. As what we need, Zagat Survey separately rates the distinct qualities of a restaurant (Food, Decor, Service and Cost), hotel (Rooms, Service, Dining and Public Facilities) and other leisure categories, e.g., nightlife and attractions, based on a 30-point scale. Some data entries are collected as shown in Table 3.

Table 3. Multi-criteria ratings of some restaurants from the Zagat Survey

Restaurant	Food	Decor	Service	Cost
B's Bar-B-Q	19	7	14	\$9
Amura Japanese	23	19	21	\$23
Vivaldi	22	20	20	\$28
Panera Bread & Café	22	19	17	\$11
Daybreak Diner	24	18	25	\$9
Globe	23	25	18	\$17
Chatham's Place	28	24	25	\$52
Vito's ChopHouse	28	25	26	\$33
Butcher Shop	22	22	19	\$39
Everglades	23	22	21	\$41
...

Note that in Table 3, each restaurant is given numerical ratings for food, decor and service and typical cost is given in US dollars. Also, the approach adopted by Zagat Survey is to summarize the ratings and opinions of thousands of users. Although this allows people to search for their needs based on a variety of criteria, one may still need a recommendation system to precisely obtain their personalized recommendations by filtering out poor candidates. This makes our approach feasible in practical applications.

To facilitate next steps of our approach developed in Section 3, data preprocessing is required to be executed on the original dataset, e.g., the dataset shown in Table 3. Without loss of generality, we conduct normalization on the ratings of each criterion. Specifically, the ratings for food, decor and service in Table 3 are proportionally normalized to be within the range of 0 to 10. Moreover, since it is more beneficial for people to have a meal at a cheaper price, we give higher ratings to restaurants with low costs and vice versa. Table 4 shows the normalized results of data entries in Table 3.

Table 4. Normalized ratings of some restaurants from the Zagat Survey

Restaurant	Food	Decor	Service	Cost
B's Bar-B-Q	6.3	2.3	4.7	9.8
Amura Japanese	7.7	6.3	7.0	7.0
Vivaldi	7.3	6.7	6.7	6.0
Panera Bread & Cafe	7.3	6.3	5.7	9.4
Daybreak Diner	8.0	6.0	8.3	9.8
Globe	7.7	8.3	6.0	8.2
Chatham's Place	9.3	8.0	8.3	1.2
Vito's ChopHouse	9.3	8.3	8.7	5.0
Butcher Shop	7.3	7.3	6.3	3.8
Everglades	7.7	7.3	7.0	3.4
...

Once the collected dataset is in a proper format, we can then perform the skyline query to choose better restaurants as recommendations for users, especially for the users who have never visited those restaurants.

5 Conclusions

As recommendation systems attract increasing research interests in recent years, many practical applications start to adopt data analysis techniques to derive recommendations for their users. In this paper, we have extended the concept of single criterion ratings to multi-criteria ones to meet the requirements of more practical recommendation systems. Moreover, we have considered the multi-criteria recommendation problem as a data query problem, instead of the optimization problem from the traditional view. Specifically, the techniques of handling skyline queries have been utilized when there are conflicts among item ratings in terms of multiple criteria. To evaluate the effectiveness of our approach, empirical studies have been conducted to show that our approach is feasible in practical applications.

References

- [1] P. Resnick and H. R. Varian, "Recommender Systems," *Communications of the ACM*, Vol.40, No.3, pp.56-58, March 1997.
- [2] J. B. Schafer, J. A. Konstan, and J. Riedl, "Meta-Recommendation Systems: User-controlled Integration of Diverse Recommendations," *Proceedings of the 11th International Conference on Information and Knowledge Management*, pp.43-51, November 2002.
- [3] S. K. Lam and J. Riedl, "Shilling Recommender Systems for Fun and Profit," *Proceedings of the 13th International Conference on World Wide Web*, pp. 393-402, May 2004.
- [4] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, Vol.17, No.6, pp.734-749, June 2005.
- [5] M. Balabanovic and Y. Shoham, "Fab: Content-Based, Collaborative Recommendation," *Communications of the ACM*, Vol.40, No.3, pp.66-72, March 1997.
- [6] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing*, Vol.7, No.1, pp.76-80, January 2003.
- [7] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," *Proceedings of ACM Conference on Computer Supported Cooperative Work*, pp.175-186, October 1994.

- [8] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, pp.43-52, July 1998.
- [9] L. H. Ungar and D. P. Foster, "Clustering Methods for Collaborative Filtering," *Proceedings of the Workshop on Recommendation Systems at the 15th National Conference on Artificial Intelligence*, July 1998.
- [10] C. C. Aggarwal, J. L. Wolf, K. Wu, and P. S Yu, "Horting Hatches An Egg: A New Graph-Theoretic Approach to Collaborative Filtering," *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.201-212, August 1999.
- [11] W. Lin, S. A. Alvarez, and C. Ruiz, "Efficient Adaptive-Support Association Rule Mining for Recommender Systems," *Journal of the Data Mining and Knowledge Discovery*, Vol.6, No.1, pp.83-105, January 2002.
- [12] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Analysis of Recommendation Algorithms for E-commerce," *Proceedings of the 2nd ACM Conference on Electronic Commerce*, pp.158-167, October 2000.
- [13] Zagat Survey, <http://www.zagat.com/>.
- [14] M. Ehrgott, *Multicriteria Optimization*, Springer Verlag, August 2004.
- [15] R.B. Statnikov and J. B. Matusov, *Multicriteria Optimization and Engineering*, Springer Verlag, March 1995.
- [16] G. R. Hjaltason and H. Samet, "Preference Disaggregation: 20 Years of MCDA Experience," *Journal of the European of Operational Research*, Vol.130, No.2, pp.233-245, April 2001.
- [17] S. Börzsönyi, D. Kossmann, and K.Stocker, "The Skyline Operator," *Proceedings of the 17th International Conference on Data Engineering*, pp.421-430, April 2000.
- [18] GroupLens Research, <http://www.grouplens.org/>.
- [19] Jester Joke Dataset, <http://www.ieor.berkeley.edu/~goldberg/jester-data/>.
- [20] Book-Crossing Dataset, <http://www.informatik.uni-freiburg.de/~cziegler/BX/>.
- [21] VoteWorld, http://ucdata.berkeley.edu:7101/new_web/VoteWorld/voteworld/datasets.html.