generate recommendations based on user preferences. However, the CF method is vulnerable if no user preferences are available under the cold start situation.

The *cold start* problem refers to a situation where no historical ratings are available for users or no preferences could be obtained, thus no recommendations could be provided [15]. For a hotel recommendation system, two situations will cause the cold start problem. First, when a user is newly registered in a website, no preferences are given to the recommendation system. Second, according to Wu et al [21], a large percentage of users only have one rating record, hence it will not be accurate to identify user preferences based on only one piece of historical data. The percentage accounts for nearly one hundred percent in a newly formed recommendation system. These users are also regarded as new users in our case.

Previous recommendation systems are developed mainly on movie rating or music rating and produce recommendations based on the only rating-the overall rating from each user. Since hotel websites collect a large amount of user ratings, it is interesting to investigate the hotel rating data and make use of it to generate targeted recommendations for hotel booking users. Meanwhile, the hotel rating data not only contains the overall ratings, but also consists of other criteria ratings. This difference of the hotel rating data helps us to generate specific recommendations according to users' requirement on each criterion.

Facing the above problem, we propose an approach which combines collaborative filtering with data classification technology to generate recommendations. The main idea of the approach is to find a new user a group in which previous similar users' preferences are available and recommend hotels according to similar users' preferences. The approach has three steps: find a group for the new user based on three criteria and process hotel average ratings in that group, compute the Euclidean similarity between hotel average ratings and the user requirement and finally generate hotel recommendation list based on the final ranking scores for each hotel.

This method functions properly under the most severe cold start situation, so it may give a hint to other recommendation systems to alleviate the cold start problem. The method not only addresses the problem with the overall rating, but also multiple hotel criteria ratings into consideration so that the recommendation is more specific. This may show a way of designing recommendation systems on data which contains multiple user ratings. By using this method, hotel websites can provide new users with reliable and targeted recommendations based on previous user ratings.

2. Literature review

"Recommender Systems (RSs) are software tools and techniques providing suggestions for items to be of use to a user" [5]. The aim of recommendation system is to provide users with personalized and useful recommendations of products and services. Many companies have done works on recommendation systems, for example Amazon.com [11] and Netflix [19] are two popular applications of recommendation systems.

Collaborative filtering (CF) is a method widely used in recommendation systems and is one of the most successful recommendation tools [9, 5]. The idea of CF is to make predictions on one user's preference by collecting his or her similar users' tastes. It makes automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating) [20]. For example, collaborative filtering could make predictions about which movie a user would like and how many scores a user will rate given similar users' preferences (likes or dislikes) when used in movie recommendation systems.

There are three kinds of CF methods: memory-based, model-based and hybrid. The memory-based CF computes similarity between items or users to make predictions. Different similarity measurement will lead to different results [5]. Probabilistic active learning method incorporated into memory-based collaborative filtering can reduce the computational cost and keep high accuracy as well [22]. The model-based algorithm was compared with memory-based algorithm in the face of profile injection attacks by [13] and the result showed a comparable accuracy. The hybrid CF combines the above two methods in the whole

recommendation process. A novel hybrid collaborative filtering method is proposed on personalized web service and got considerable good results [3]. Other collaborative filtering methods are also provided by researchers, such as cluster-based collaborative filtering [7]. However, one major challenge of collaborative filtering is the cold start problem [21]. Several approaches have been adopted in previous systems to cope with this challenge. Park, S. T. *et al* [15] constructed predictive models in regression based on user preferences to feature cold start problem. In their paper [10], the authors propose an approach to identify representative user preferences as an indication of new user preferences. In their method, users' preference can be obtained after users rated five to ten representative items.

Regarding hotel recommendations, two previous approaches have been proposed. A trust-based collaborative filtering method was proposed by Wu et al [21]. They incorporated CF with trust models and concluded that trust really influence the prediction accuracy, even though the improvement is not very significant. One of the difficulties in their research is that the rating values are too similar among users so that it is not easy to recognize a suitable number of similar users' group for a user using K nearest neighbor method. Huming, G. et al [7] proposed a recommendation system based on clustering and rankboost algorithm. They found a cluster center of k users based on historical ratings and generate recommendations according to the cluster's user preferences. The result shows when user number is greater than eight hundred, the result reserves a high accuracy.

Many systems use *cosine similarity* and *Pearson correlation similarity* to compute the similarity between two ratings [18]. They both perform well on finding user preferences but a problem occurs when measuring similarity between two vectors like (1,1,1,1,1) and (4,4,4,4,4) and (2,1,2,1,2) and (5,4,5,4,5) [23]. The results of the two groups of vectors are both 1 using the two measurements, which show that the two groups of vectors are similar. However, these two groups of ratings are not similar for the hotel rating data at all. The similarities computed by the Euclidean distance method for the above two groups are both 0.1297, which shows two vectors are not that similar. To avoid the above problem, we use Euclidean similarity computation. Meanwhile, Euclidean similarity shows a good result when compared with other similarity measures.

However, considering all the related work discussed above, few of them focus on the hotel rating data. Recommendations are all based on users' overall ratings. We aim at recommending hotels not only based on overall rating, but also on five other ratings. A combined approach is proposed for providing hotel recommendations.

3. Methodology

The combined method we proposed consists of three steps: data classification, similarity computation and ranking computation. The first step, Data Classification, is to divide data into smaller clusters from the whole dataset. In each small cluster, data share similar attributes. The averages in terms of the six criteria – the hotel website, where we collected source data, provides us with these attributes (criteria): general-score, cleanness, comfort, satisfaction, closeness-to-downtown, and service – for each hotel are computed according to the ratings from the users in that group. The second step, Similarity Computation, is to calculate the Euclidean similarity between the hotel averages and the user requirements, as an indication of degrees that how much a hotel is close to the expectation of the user requirement. A user requirement is expressed as $R'_s = (R'_1, R'_2, R'_3, R'_4, R'_5)$. The third step is to compute the ranking score of each hotel and rank the hotels according to the ranking scores to make a final recommendation in that group for the user. An algorithm works as follows, see Fig. 1:

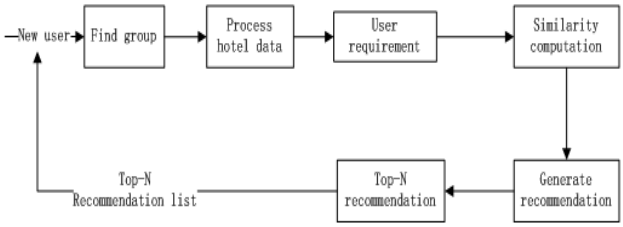

Fig. 1. Flow chart of Recommendation process.

In the dataset, each user provides one rating record for one hotel. Each rating record consists of six criteria ratings. A rating function R in a recommender system is defined as $R(u,m) \rightarrow R0 \times R1 \times \cdots \times Rs$, where R_0 denotes the set of possible overall ratings and R_s denotes the set of possible ratings for each s criterion ($1 \leq s \leq 5$), see Table 2. The rating set of R in the system could be expressed as a matrix:

Table 1. The rating set of R in the recommendation system.

Hotel # / User #	1	2	3	⋯	m-2	m-1	m
1	0	0	1		0	0	0
2	1	0	0	⋯	0	0	0
3	0	0	0		1	0	0
⋮		⋮		⋱		⋮	
u-1	0	1	0		0	0	0
u	0	0	0	⋯	0	1	0

where u denotes the number of users while m indicates the number of hotels. "1" means a certain rating record exists for a user and a hotel. "0" means no rating record for a user-hotel pair. Each user only has one rating record in this dataset, which means each row could only have one record "1".

3.1. Data classification

It is assumed that people will be more likely to listen to the advice of those who share something in common with them. For a new user, he or she would be more willing to accept the recommendations from the users who share similar interests with him or her. In order to find a similar user group to which a new user could refer, we use the data classification approach, to classify the users into smaller groups (called clusters) based on their similar interests. This approach divides a large dataset into smaller clusters so that the users in one cluster share more similar interests than in other parts. Hotel information, such as location and star level, rating information like rating date, and user information like travel purposes, are used as characteristics for data classification. We call these pieces of information the "attributes" of the dataset [17]. If we regard each of these attributes as a dimension and each value of that attribute covers a segment, then the whole dataset is divided into many small boxes. For example, three attributes each with three values will divide the dataset into 27 parts.

Regarding the hotel rating data, we choose three dimensions: User's Traveling Purpose (TA), Traveling Destination (City) and Hotel Star Level (SL) to classify the data. The information from these three dimensions is used as the main criteria for a new user to search for a hotel. New users will be concerned about why to go there, where to go and where to stay. For the user's traveling purpose dimension, five segments (Business, Romance, Friends, Family and Others) are provided. For the traveling destination dimension, five destinations (Amsterdam, Barcelona, Berlin, Paris and Rome) are gathered in this thesis. For the hotel star level dimension, it also has five levels from one star to five stars rated by official

organizations. In this way, we have 125 small groups and people share some common information in each group. It can be presented as Fig. 5:

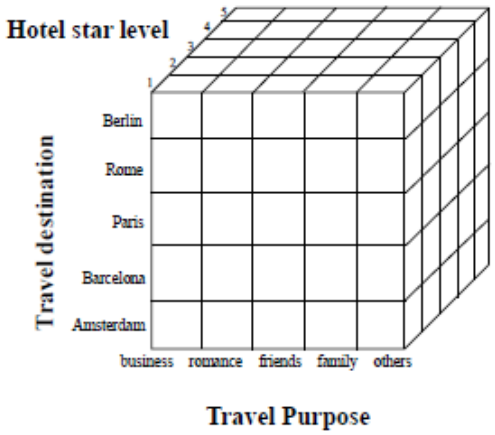

Fig. 2. Data Classification on hotel rating data.

A group, denoted as G_{ijk}→*Group {i=TA, j=City, k=SL}*, contains a set of user rating pairs R. For example, G_{ijk} represents a group of ratings R which are provided by users for the travel purpose *i*, the traveled city *j* and the hotels with *k* stars. In this way, we find a referencing group for the new user where many previous rating data can be referred to in this group.

For the group a new user belongs to, we compute the average scores in terms of all the criteria of each hotel in that group to indicate the group users' attitude towards the hotel. For each average rating, it is computed by:

$$\overline{R^{s}_{ijk,m}} = \frac{\sum_{1}^{N_{ijk,m}} R^{s}_{ijk,m}}{N_{ijk,m}} \tag{1}$$

where $R^{s}_{ijk,m}$ denotes ratings toward hotel m of users in group G_{ijk}. $N_{ijk,m}$ is the number of ratings of hotel m in group G_{ijk}.

The average ratings of overall rating and five criteria for each hotel, together with hotel id number, in group G_{ijk} are expressed as:

$$H^{m}_{ijk}: (\overline{R^{0}_{ijk,m}}, \overline{R^{1}_{ijk,m}}, \overline{R^{2}_{ijk,m}}, \overline{R^{3}_{ijk,m}}, \overline{R^{4}_{ijk,m}}, \overline{R^{5}_{ijk,m}}, H_m) \tag{2}$$

where $\overline{R^{0}_{ijk,m}}$ denotes the average overall rating which people in group G_{ijk} rated for hotel *m*, $\overline{R^{s}_{ijk,m}}$ ($1 \leq s \leq 5$) the average rating of the criteria "*s*" the hotel "*m*" get from group G_{ijk} and H_m the hotel id number.

3.2. Similarity Calculation

To provide a ranking base for generating a recommendation list, we calculate the similarity between the user requirement R'_s and the hotel average ratings H^{m}_{ijk}. The more similar the two ratings are, the closer a hotel is to the expectation of the new user. Many systems use *Cosine similarity* and *Pearson correlation similarity* to compute the similarity between two ratings. Cosine similarity measures the angle between two vectors as their similarity and Pearson correlation reveals linearity degree between two vectors [18]. They both perform well on finding user preferences but a problem occurs when measuring similarity between two vectors like (1,1,1,1,1) and (4,4,4,4,4) and (2,1,2,1,2) and (5,4,5,4,5) [23]. The results of the both methods are 1, which shows two vectors are similar. However, these two groups of ratings are not similar for the hotel rating data at all. The similarities computed by the Euclidean method

for the above two groups are both 0.1297, which shows two vectors are not that similar. To avoid the above problems, we use Euclidean similarity computation for measuring similarity.

3.2.1 Euclidean similarity

Euclidean similarity is based on the calculation of Euclidean distance. Euclidean distance computes the distance between two space vectors in n-dimensions [2]. The relationship between Euclidean distance and Euclidean similarity is as follows: smaller distance yields larger similarity. In the following, we discuss how to obtain Euclidean distance and similarity.

A user requirement $R_s^{'}$ is provided by the new user. $R_s^{'}$ can be denoted as $R_{ijk}^{s'}$ after data classification process for the new user. For example, a user may rate a hotel's cleanliness 4, comfort 4, condition 5, neighborhood 5 and service 4, then $R_{ijk}^{s'}$ is (4,4,5,5,4). After we get $R_{ijk}^{s'}$, we measure the Euclidean similarity between the hotel average ratings H_{ijk}^{m} and the user request $R_{ijk}^{s'}$. The distance between the hotel average ratings and the user request in one group can be defined by the following formula:

$$d(H_{ijk}^{m}, R_{ijk}^{s'}) = \sqrt{\sum_{s=1}^{5}(\overline{R_{ijk,m}^{s}} - R_{ijk}^{s'})^2} \tag{3}$$

where m is the hotel id and s ranges from 1 to 5 for the ratings on five criteria. The smaller the distance, the more similar the two ratings are. The reciprocal of the Euclidean distance plus one is commonly referred as Euclidean similarity [16]. The Euclidean similarity is described as:

$$Sim(H_{ijk}^{m}, R_{ijk}^{s'}) = \frac{1}{1+ d(H_{ijk}^{m}, R_{ijk}^{s'})} \tag{4}$$

where its value ranges from (0,1]. The larger the value is, the more similar the two subjects are. In this way, we know the similarity between each hotel and user's desired ratings in the group which the new user belongs to. The hotels with the similarity below the average are excluded from recommendation. In this way, we get a similarity value of each hotel in the group as a basis for the next ranking process.

3.3. Ranking Score computation

After computing the similarity, we need a ranking method for generating the recommendation list. Ranking score provides a ranking basis for the final recommendation within one group. The score can be derived as follows.

Ranking score F equals to the production of the hotel averaged overall rating and the similarity score in one group. It focuses on the similarity and the averaged overall ratings of hotels. It can be formulated as:

$$F(Hijk, m) = \overline{R_{ijk,m}^{0}} * Sim(H_{ijk}^{m}, R_{ijk}^{s'}) \tag{5}$$

The intuition behind this computation is that hotels which have greater similarities to the user requirement should have a higher rank in the recommendation list. For example, if two hotels in the same group have the same averaged overall rating, the larger similarity score one hotel has, the higher the ranking place it should have in the recommendation list.

3.4. Evaluation methods

In order to determine whether the combined method functions and to discover N in top-N recommendation, evaluation methods Normalized Mean Averaged Error (NMAE), 10-fold

Cross-Validation and Receiver Operating Characteristic (ROC) curves are used. They are widely used to measure the accuracy of recommendation systems.

3.4.1 Normalized Mean Averaged Error (NMAE)

To measure the error between the user requirement and the actual rating, Normalized Mean Averaged Error (NMAE) is adopted. It is a Mean Absolute Error (MAE) normalized with respect to the range of rating values [6]. It takes the rating range into consideration and modifies the computation of traditional mean average error. The actual user rating is selected from the testing group and expressed as $C^{s}_{ijk,m} = \{C^{1}_{ijk,m}, C^{2}_{ijk,m}, C^{3}_{ijk,m}, C^{4}_{ijk,m}, C^{5}_{ijk,m}\}$. NMAE is computed using:

$$NMAE = \frac{MAE}{r_{max} - r_{min}} = \frac{\sum_{s=1}^{n} \left| R^{s'}_{ijk} - C^{s}_{ijk,m} \right|}{(r_{max} - r_{min}) * n} \quad (6)$$

where n=5, r_{max} is the maximum value of rating 5, and r_{min} is the minimum value of rating 1. Generally, a NMAE value ranges from [0,1]. The smaller the NMAE is, the better the accuracy.

3.4.2 10-fold Cross-Validation

A single partition of training and testing is not trustable for measuring the error rate. Therefore, repeated training and testing should be done to get reliable results. Revealing the repeated validation methods, a leave-one-out cross validation is computationally expensive, 10-fold cross-validation is a method adopted [8]. 10-fold cross-validation divides a dataset into ten subsets. Each subset will be used as testing data once while the remaining will be used as training data. The cross validation ends until all the subsets have been tested. The average of the ten testing results is regarded as the final accuracy of the system.

3.4.3 Receiver Operating Characteristic (ROC) curves

To evaluate how effectively a recommendation list can help a new user to select satisfied hotels, Receiver Operating Characteristic (ROC) sensitivity measurement can be used [1]. It has a basis that filtering is a binary process. We calculate the average value of user requirement and the average of actual user ratings. In our data, we use sensitivity to show the probability of a "satisfied" hotel being recommended by the method and define specificity to indicate an "unsatisfied" hotel not being recommended [14]. ROC curve lots the sensitivity and the 1-specificity of the test.

We regard the average of user requirement $R^{s'}_{ijk}$ and the average of actual user rating $\overline{C^{s}_{ijk,m}}$ as the predicted average value and the actual average value. If both of the predicted average value and the actual average value are greater than a given threshold, we regard that a satisfied hotel is correctly recommended.

Let APR, AAR, and T denote the predicted average value, the actual average value, and a quality threshold respectively. The following possible cases are defined by the method for each hotel in the recommendation list:

- True Positive (TP) when APR ≥ T ∧ AAR ≥ T
- False Positive (FP) when APR ≥ T ∧ AAR < T
- True Negative (TN) when APR < T ∧ AAR < T
- False Negative (FN) when APR < T ∧AAR ≥T

For a set of recommendations, the sensitivity is defined as the True Positive Fraction (TPF) [12]:

$$sensitivity = TPF = \frac{N_{TP}}{N_{TP}+N_{FN}} \quad (7)$$

where N_{TP} and N_{FN} are the numbers of the true positive and the false negative occurrences over the set of recommendations respectively. This fraction shows the ratio of the correct recommendations in all recommendation lists.

The 1-specificity as the False Positive Fraction (FPF) is defined as

$$1\text{- }specificity = FPF = \frac{N_{FP}}{N_{FP}+N_{TN}}, \quad (8)$$

where N_{FP} and N_{TN} are the numbers of the true negative and the false positive occurrences over the set of recommendations respectively. This fraction shows the ratio of incorrect recommendations in all recommendation lists.

4. Results and Evaluation

4.1. Recommendation list

One user group is randomly selected and the above methods are applied. The random number generated is {2,2,3} which stands for the group G_{223}: {Romance, Barcelona,3}. To generate a recommendation list, (4,4,4,4,4) is randomly selected as the new user's requirements. By applying the method proposed in section 4, we obtain 32 hotels for recommendation to the new user in this group. The Top-10 Recommendation list is shown in Table 2.

Table 2. Top-10 Recommendation list.

HID	CScore	Similarity	Rank
65	4.272727	0.6439240	1
44	4.000000	0.6057470	2
52	4.333333	0.5557701	3
89	4.000000	0.6000000	4
99	4.000000	0.5958648	5
94	3.809524	0.6141984	6
158	4.333333	0.5313730	7
12	4.055556	0.5663218	8
161	4.000000	0.5584816	9
190	3.888889	0.5505103	10

The final Top-3 recommendation hotel list in group {Romance, Barcelona, 3} is presented in Table 3.

Table 3. Top-3 recommendation hotel list.

HID	Hotel name	Rank
65	Park Hotel Barcelona	1
44	Dalia Ramblas	2
52	Residencia Melondistrict Marina	3

4.2. Evaluation Result

4.2.1 10-fold cross-validation

The 10-fold cross-validation is adopted in to determine the error rate of the combined method. The data for each selected group is divided into two sets: training data and testing data. Training data is used for generating a recommendation list. Testing data is used to measure

the overall performance. Considering the sample size, the groups with the number of records greater than 300 can be selected. Five groups are selected from the whole dataset. User requirement is simulated by generating all permutations of five ratings and each rating value is greater or equal to 4. For each validation, one random user requirement is selected to do training and testing. The performance is measured by NMAE between the actual user ratings and the user requirement on each criterion. Five groups are tested under one user requirement (4,5,4,4,5) in one 10-fold cross-validation process. The result is shown in Table 4.

Table 4. Results of 10-fold cross validation

Group number	Top-3 NMAE	Top-10 NMAE
G_{143}	0.12	0.19
G_{423}	0.17	0.18
G_{235}	0.14	0.13
G_{354}	0.16	0.16
G_{512}	0.19	0.15

To avoid selecting extreme ratings from the testing part, five benchmark ratings are randomly picked from the permutation list and put into the 10-fold cross-validation process to test the performance. Totally fifty recommendations are generated for both the Top-3 and Top-10 lists. The mean NMAEs for both lists are computed and the result is shown in Table 5.

Table 5. Results of 10-fold cross validation under five user requirements

Group number	Number of records	Top-3 NMAE	Top-10 NAME
G_{143}	2,096	0.16	0.20
G_{423}	1,242	0.15	0.16
G_{235}	641	0.13	0.12
G_{354}	924	0.15	0.15
G_{512}	3,065	0.13	0.17

For the hotel recommendation, the expected range of NMAE lies in [0, 0.25]. If the NMAE equals zero, it means the real user rating is the same as the user requirements on the five criteria. If the NMAE equals 0.25, it shows that the absolute difference between two ratings equals 5, which supposes a large difference to the user requirement. From the table above, we observe that the number of Top-3 NMAE less than Top-10 NMAE in group $G_{143,}$ G_{423} and G_{512}. For group G_{354}, the NMAEs of the two lists are the same. The NMAE of Top-10 list is smaller than the NMAE of Top-3 list in group G_{235}. However, the NMAE of Top-3 list is lower than that of Top-10 list in average. Mean NMAEs of the two lists under the five group experiments is shown Fig. 3.

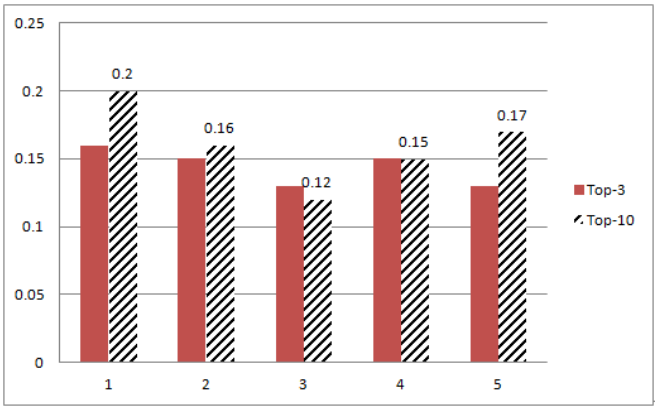

Fig. 3. Mean NMAEs of five groups of two top recommendation lists.

It is clear that the Top-3 recommendation approach has a comparatively lower or equal NMAE for the five groups than that of the Top-10 recommendation method in the three out of five cases. As the size of group increases, the NMAE of Top-3 list tends to be lower than that of Top-10 recommendation list.

4.2.2 ROC curves

The ROC curves are computed for the Top-3 list and the Top-10 list on the thresholds ranging from 4 to 4.5. Two groups G_{423}: {Family, Berlin, 3} and G_{514}: {Others, Barcelona, 4} are selected and five random user requirements are picked. Ten values for TPF and FPF are produced for each list, and we calculate the mean values of the ten values for the two top lists. With the thresholds 4.1 and 4.2, 4.3 and 4.4, TPF and FPF are the same for each recommendation of both lists. Results are illustrated in Fig. 4.

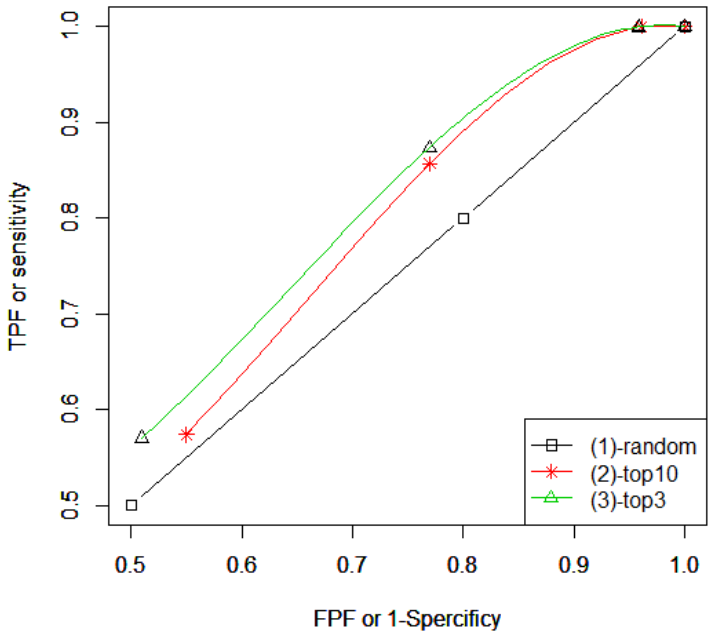

Fig. 4. ROC curves for Top-3 and Top-10 recommendation.

Taking the accuracy of the recommendation system into consideration, Fig.4 suggests that the two recommendation lists both have a relatively acceptable accuracy. Whereas the Top-3 recommendation list provides a better result than the Top-10 recommendation because under a given sensitivity level, Top-3 recommendation list has a lower 1-Spercificity even if the difference is not that significant.

5. Conclusion and Discussion

Cold start is a major problem to be dealt with in the field of the collaborative filtering (CF) researches. The problem is severe in the case of data analysis of hotel ratings because it is very hard to form user preferences as one user provides only one rating record for one hotel. In other words, we need quite a number of rating records from one user to identify his or her preference. Considering this difficulty, we propose a recommending method which combines the collaborative filtering with data classification method to alleviate the cold start problem. Five groups of experiments and two evaluations have been conducted. The results indicate that our methods functions and performs properly under the cold start situation. It is important that Top-N recommendations should be accurate and targeted so that users can book hotels efficiently. Concerning about the accuracy, we compare the NMAE of Top-3 recommendation list and that of Top-10 recommendation list. The result shows a better performance of Top-3 recommendation list using the combined approach.

Through alleviating the cold start problem, we propose a combined method of collaborative filtering and data classification to provide an acceptable hotel recommendations for users based not only on the rating records provided by a user (hotel guest) to a hotel, but also on the information about the user as well as the hotel. We focus on hotel rating data, especially on user requirements of the five specific criteria. By applying this method on hotel rating data, hotel recommendations that mostly to the expectation of new users could be provided through a Top-3 recommendation list. The methodology described is general and may be easily installed on systems, especially for newly formed hotel booking websites, to alleviate the cold start problem for hotel recommendations.

Despite of the work accomplished in this paper, further investigation is still necessary. Firstly, other dimensions such as Date and Nationality of users can be used in data classification. In this classification, we may observe how the guests' ratings are influenced by the travel seasons and the nationalities, which may help to make more accurate and focused recommendations. Secondly, we can include more factors in computing the ranking scores. Our ranking method takes two values, overall rating and similarity score into consideration. Other factors such as the total review number of a hotel could also try to be added to the ranking process to get higher recommendation accuracy. Thirdly, our major problem in this study is that we can only obtain one rating by a single guest, which makes it difficult to identify the guest's preference. A usable step is to suggest a membership system for the booking websites which can help the hotels to provide better services to the guests who in turn may become loyal customers via a trustable recommendation system.

References

1. Bradley, A. P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7), 1145-1159.
2. Borgefors, G., 1984. Distance transformations in arbitrary dimensions. *Computer vision, graphics, and image processing*, 27(3), 321-345.
3. Chen, X., Liu, X., Huang, Z., & Sun, H., 2010. Regionknn: A scalable hybrid collaborative filtering algorithm for personalized web service recommendation. In *Web Services (ICWS), 2010 IEEE International Conference*, pp. 9-16.
4. Francesco, R., Lior, R., & Bracha, S., 2011. Introduction to Recommender Systems Handbook, Recommender Systems Handbook.
5. Herlocker, J., Jung, S., & Webster, J. G., 2012. Collaborative filtering for digital libraries.
6. Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T., 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1), 5-53.
7. Huming, G., & Weili, L., 2010. A Hotel Recommendation System Based on Collaborative Filtering and Rankboost Algorithm. In *2010 Second International Conference on Multimedia* and Information Technology, Vol. 1, pp. 317-320.
8. Kohavi, R. (1995, August). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, Vol. 14, No. 2, pp. 1137-1145.
9. Koren, Y., & Bell, R., 2011. Advances in collaborative filtering. In *Recommender systems handbook*, pp. 145-186.
10. Liu, N. N., Meng, X., Liu, C., & Yang, Q., 2011. Wisdom of the better few: cold start recommendation via representative based rating elicitation. In *Proceedings of the fifth ACM conference on Recommender systems*, pp. 37-44.
11. Linden, G., Smith, B., & York, J., 2003. Amazon. com recommendations: Item-to-item collaborative filtering. Internet Computing, *IEEE*, 7(1), 76-80.
12. Metz, C. E., 1978. Basic principles of ROC analysis. In *Seminars in nuclear medicine*m, Vol. 8, No. 4, pp. 283-298.
13. Mobasher, B., Burke, R., & Sandvig, J. J., 2006. Model-based collaborative filtering as a defense against profile injection attacks. In *AAAI*, Vol. 6, pp. 1388.

14. Papagelis, M., Plexousakis, D., & Kutsuras, T., 2005. Alleviating the sparsity problem of collaborative filtering using trust inferences. In *Trust management* , pp. 224-239.
15. Park, S. T., & Chu, W., 2009. Pairwise preference regression for cold-start recommendation. In *Proceedings of the third ACM conference on Recommender systems*. pp. 21-28.
16. Qian, G., Sural, S., Gu, Y., & Pramanik, S., 2004. Similarity between Euclidean and cosine angle distance for nearest neighbor queries. *In Proceedings of the 2004 ACM symposium on Applied computing*, pp. 1232-1237.
17. Ramakrishnan, R., & Gehrke, J., 2000. Database management systems. Osborne/McGraw-Hill.
18. Strehl, A., Ghosh, J., & Mooney, R. (2000, July). Impact of similarity measures on web-page clustering. In *Workshop on Artificial Intelligence for Web Search*, pp. 58-64.
19. Su, X., & Khoshgoftaar, T. M., 2009. A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009, 4.
20. Terveen, L., & Hill, W., 2001. Beyond recommender systems: Helping people help each other. HCI in *the New Millennium*, 1, 487-509.
21. Wu, Q., Forsman, A., Yu, Z., & Song, W. W., 2014. January. A Computational Model for Trust-Based Collaborative Filtering. In *Web Information Systems Engineering–WISE 2013 Workshops*, pp. 266-279.
22. Yu, K., Schwaighofer, A., Tresp, V., Xu, X., & Kriegel, H. P., 2004. Probabilistic memory-based collaborative filtering. Knowledge and Data Engineering, *IEEE Transactions on*, 16(1), 56-69.
23. Zhu, W., 2014. Research on user similarity function of recommendation systems. College of Computer Science of Chongqing University. URL: < http://www.docin.com/p-976286686.html>. 2015.May. 26.