A mission location recommender system to missioner by using clustering based collaborative filtering

Razieh Qiasi[1], Seyyed Hassan Hani-Zavarei[2], Behrooz Minaei-Bidgoli[3]
[1]Department of Computer, Sama technical and vocational training collage, Islamic Azad University, Qom Branch, Qom, Iran
raziehghiasi@gmail.com
[2]Department of Information Technology, University of Qom
Qom, Iran
Hasanhani@yahoo.com
[3]Department of Computer Engineering, University of Science and Technology
Qom, Iran
minaeibi@cse.mcu.ed

Abstract

By expansion of religion mission boards to further parts of Iran, and also many different mission needs and increasing number of missions and mission locations, traditional and manual methods of missioner dispatch are not fast and accurate enough for dispatching manager's needs anymore. So, there is a need for an intelligent system which can improve dispatching programs by assisting the missioners in selecting the suitable location. Application of recommender systems is a suitable solution to this problem. Collaborative filtering is the most commonly used and effective recommendation technique among different types of recommender systems.

This paper presents a mission location recommender system based on collaborative filtering method. Traditional CF method is not scalable for the increasing number of missioners. To address this issue, this paper proposes developing a mission location recommender system based on clustering techniques followed by collaborative filtering. The experimental results show that the cluster based collaborative filtering has acceptable performance and it is the most accurate and scalable user based CF.

Keywords: Dispatching missioner, Recommender system, Collaborative filtering, clustering.

1. INTRODUCTION

Religion propaganda has an effective role in Iranians cultural and Islamic knowledge improvements and also it is important in exploitation of religion sciences scholars for the purpose of publication and spreading of religious learning. Because of religion propaganda's importance, there are several centers responsible of dispatching scholars and clerics to missions in Iran.

Suitable mission locations' selection is affected by several parameters, such as spreading of religion mission board in further parts of Iran, many different missions' needs and increasing number of missioners and mission locations, but traditional and manual methods of mission dispatch are not fast and accurate enough for dispatching managers' needs. Therefore there is a need for an intelligent system which can assist the missioners in selecting the suitable mission location and so improve dispatching programs. Recommender systems are a suitable solution for this problem.

Recommender systems are decision supports which present information about items with respect to user preferences by analyzing users' prior behavior. Generally three kinds of methods are applied in recommender systems: content filtering, collaborative filtering and hybrid method. Collaborative filtering is the most popular one among these methods. A collaborative filtering system's basic idea is to generate recommendations based on similar past users' experiences [20]. Collaborative filtering method can be memory based or model based.

Memory based collaborative filtering make recommendations on all of the gathered data. In this method the newly generated data can also be taken into account for recommendation, so its recommendations can be highly accurate. In model based collaborative filtering, first a model on all of the offline data must be constructed and then this model will be loaded to the memory to generate online recommendation results. While making some concessions on accuracy, this method significantly improves system's scalability.

Memory based techniques are quite successful in real world applications, because they are easy to understand, implement and work well in many real world situations. However, there are some problems that limit the application of memory based techniques like user-item rating matrix which will result in a scarcity matrix, especially in large scale applications, that each user only rates a small set of a large database items.

To overcome the weaknesses of memory-based techniques, researchers has focused on hybrid memory-based and model-based approaches with the aim of seeking more accurate, yet more efficient methods [3,5,12,15].

This paper proposes a hybrid memory and model based approach for building a mission location recommender system. This approach uses clustering techniques to identify the communities of similar missioners based on their rating data and uses these communities as a mechanism to make the recommendations.

Our efforts in this paper are aimed towards applying existing recommendation methods in propaganda domain, which is a new domain of issues. So, we are not going to propose a new method in recommendation systems.

The rest of this paper is consisted of the following sections. Section 2 summarizes the related works. The research methodology used in this study is reported in sections 3. Evaluation metrics used in this study are discussed in section4. Section 5 presents empirical results. Finally, Section 6 summarizes our conclusions of this work and suggests future research directions.

2. RELATED WORK

In case of having user clusters, traditional CF algorithms can be operated on the clusters instead of the whole user-item matrix. By reducing the dimensions of user-item rating matrix and therefore avoiding the data sparsely problem, this approach can provide better recommendation results in terms of accuracy and can improve the online performance of CF algorithms. So far, many researchers have used clustering to improve the scalability and sparsely problem. In the following, we'll describe more some of these researches.

A. Kohrs et al. [1] presents a novel algorithm for collaborative filtering based on hierarchical clustering which tries to balance robustness and accuracy of predictions and experimentally show that it is especially efficient in dealing with *bootstrapping* and *new user* situations.

S.H.S. Chee et al. [18] developed an efficient collaborative filtering method called RecTree that applies clustering techniques which create cohesive cliques economically. RecTree achieves better scale-up in comparison to other memory based collaborative filters by seeking advisors only within a clique rather than the entire database.

Sarwar et al. [3] presented a clustering-based algorithm that is suitable for a large data set.

Bridge et al. [4] generalized an existing clustering technique and applied it to a collaborative recommender's dataset to reduce cardinality and sparsely. They systematically tested several variations, by exploring the value of partitioning and grouping the data.

In Kelleher et al. [11] a collaborative recommender is presented that uses a user-based model to predict user ratings for specified items. The model comprises summary rating information derived from a hierarchical clustering of the users. They compare their algorithm with several others and show that its accuracy is good and its coverage is maximal. They also showed that the proposed algorithm is very efficient. prediction time in this method grows independently of the number of ratings and items and only grows logarithmically with respect to the number of users.

Xue et al. [8] presented a novel approach that combines the advantages of memory based collaborative filtering and model based collaborative filtering approaches by introducing a smoothing-based method. In this approach, clusters generated from the training data provide the basis for data smoothing and neighborhood selection. As a result, they provide higher accuracy as well as increased efficiency in recommendations. Their empirical studies on EachMovie and MovieLens datasets shows that the proposed approach consistently outperforms other user based traditional collaborative filtering algorithms.

Rashid, A.M. et al. [2] proposed ClustKnn, a simple and intuitive algorithm that is well suited for large data sets. First, by building a straightforward but efficient clustering model, this method tremendously compresses data. Recommendations are then generated quickly by using a simple Nearest Neighbor-based approach. The feasibility of ClustKnn has been demonstrated both analytically and empirically. They have done a comparison with a number of other popular CF algorithms which shows that apart from being highly scalable and intuitive ClustKnn provides very good recommender accuracy as well.

Mittal et al. [14] proposed a framework based on data partitioning/clustering algorithm application on ratings dataset followed by collaborative filtering for developing a movie recommender system. This system reduces the computation time considerably and increases prediction accuracy.

3. METHODOLOGY

This section provides details of the purposed method for constructing mission location recommender system. This method has two phases: offline phase (user clustering) and online phase (generation of prediction and recommendation).

3.1 User Clustering

User clustering techniques work by identifying groups of users who appear to have similar ratings (see Fig.1). Once the clusters are created, predictions for a target user can be made by averaging the opinions of the other users in that cluster. There are many algorithms that can be used to create clusters. In this paper, a *TwoStep* algorithm is selected.

TwoStep Clustering is a two-step clustering method. The first step compresses the raw input data into a manageable set of sub-clusters by making a single pass through the data. The second step uses a hierarchical clustering method to progressively merge the sub-clusters into larger and larger clusters, without requiring another pass through the data. Hierarchical clustering does not require the number of to-be-selected-clusters ahead of time.

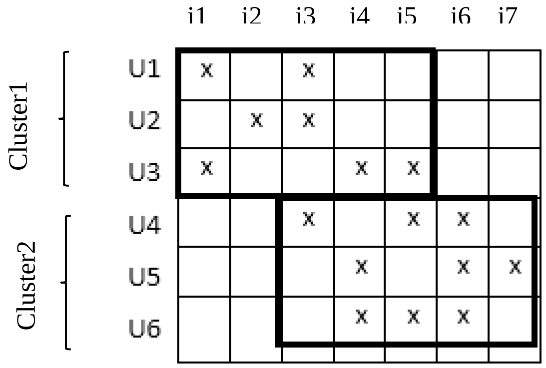

Fig. 1: User Clustering

This method uses a log -likelihood distance measure, to accommodate both symbolic and range fields. It is a probability-based distance. The distance between two clusters is related to the decrease in log-likelihood as they are combined into one cluster. We have assumed log-likelihood normal distributions for range fields and multinomial distributions for symbolic fields in our calculations. It is also assumed that the fields are independent of each other and so are the records. The distance between clusters i and j is defined as:

$$d(i,j) = \xi + \xi_j - \xi_{\langle i,j \rangle} \quad (1)$$

Where

$$\xi_u = -N_u\left(\sum_{k=1}^{k^A} \frac{1}{2}\log(\hat{\sigma}_k^2 + \hat{\sigma}_{uk}^2) + \sum_{k=1}^{k^B} \hat{E}_{uk}\right) \quad (2)$$

and

$$\hat{E}_{uk} = -\sum_{l=1}^{L_k} \frac{N_{ukl}}{N_u} log \frac{N_{ukl}}{N_u} \quad (3)$$

In the above equations, k^A is the number of range type input fields, k^B is the number of symbolic type input fields, L_k is the number of categories for the kth symbolic field, $N_u$is the number of records in cluster u, N_{ukl} is the number of records in cluster u which belongs to the lth category of the kth symbolic field, $\hat{\sigma}_k^2$ is the estimated variance of the kth continuous variable for all records, $\hat{\sigma}_{uk}^2$ is the estimated variance of the kth continuous variable for records in the vth cluster, and $\langle i,j \rangle$ is an index representing the cluster formed by combining clusters i and j.
If we ignore $\hat{\sigma}_k^2$ in the equations for ξ_u, the distance between clusters i and j would be exactly the decrease in log-likelihood when the two clusters are combined. The $\hat{\sigma}_k^2$ term is added to solve the problem caused by $\hat{\sigma}_{uk}^2 = 0$, which results in natural logarithm being undefined. (This would occur, for example, when a cluster has only one case)[19].

3.2 Generation of Prediction and Recommendation

The main task of rating prediction is finding nearest neighbors for active users. Finding the nearest neighbors requires computing the similarity between users. Therefore, this section includes these three main steps: similarity computation, neighborhood selection and the processes involved in rating prediction.

3.2.1 Similarity Computation

The notion of similarity is used to identify users that have common “preferences”. As mentioned above, traditional memory based collaborative filtering searches the whole ratings database to find the most similar users. Whereas in the method used in this paper, similarity of active user is computed by the members of the cluster which it blongs to. Therefore, execution time is reduced and scalability and sparsely problems are resolved too.

There are several methods to measure similarity among which Pearson’s correlation and cosine vector similarity are widely used in collaborative filtering [6,7].

- Pearson correlation coefficient (PC): This metric measures the degree of association between ratings’ patterns using a value between -1 and +1. A positive value is the evidence of a general trend where high ratings of user U are associated with high ratings of V and low ratings of U tend to be associated with low ratings of V (a negative value for the correlation implies the inverse of this association). PC can be computed by:

$$PC(u,v) = \frac{\sum_i (r_{ui} - \overline{r_i})(r_{vi} - \overline{r_i})}{\sqrt{\sum_i (r_{ui} - \overline{r_i})^2}\sqrt{\sum_i (r_{vi} - \overline{r_v})^2}} \quad (4)$$

Here $r_{u,i}$ ($r_{v,i}$) denotes the rating of user u (v) on item i, $\overline{r_i}$ is the average rating of the i-th item.

- Cosine measure: This metric defines the similarity between two users as the cosine of the angle between the rating vectors, with values between 0 and 1. A larger value means a higher similarity for the ratings (the two vectors are closer). The cosine similarity of users U and V is defined as:

$$Cosine\ (u,v) = \frac{\sum_i r_{ui} r_{vi}}{\sqrt{\sum_i {r_{ui}}^2}\sqrt{\sum_i {r_{vi}}^2}} \quad (5)$$

3.2.2 Neighborhood Selection

The next stage is selection of the neighbors who will serve as recommenders. Here, the entire cluster that user belongs to, can be selected as user's neighborhood. For a fair comparison we have recorded the number of neighbors used for prediction computation for each user and forced our basic CF algorithm to use same number of neighbors for prediction generation. Selection of the neighbors is normally done in two steps [9,17]:

- Threshold-based selection, in which the users whose similarity exceeds a certain threshold value are considered as neighbors of the target user.
- The top-n technique, in which n-best neighbors are selected; n's value is given ahead.

3.2.3 Prediction Rating

When a subset of the nearest neighbors of the active user are selected, predictions are generated based on a weighted aggregate of their ratings. Most used aggregating functions are weighted sum and simple weighted average. To make the prediction for the active user *u* on an item *i*, weighted sum is computed using all the ratings of the neighbors on that item by the following formula:

$$Pr_{u,i} = \overline{r_u} + \frac{\sum_{k\epsilon K} Sim(u,k)(r_{ki} - \overline{r_k})}{\sum_{k\epsilon K} |sim(u,k)|} \quad (6)$$

4. Evaluation

Evaluation is one of the key aspects in recommender systems. There has been considerable research in the area of recommender system's evaluation focused on accuracy and performance [6,10]. These researches introduce several metrics for assessing the accuracy of collaborative filtering methods [13]. These metrics are divided into two main categories: statistical accuracy metrics and decision-support accuracy metrics.

Statistical accuracy metrics: Statistical accuracy metrics evaluate the accuracy of a prediction algorithm by comparing the numerical deviation of the predicted ratings from the respective actual user ratings. Some of the frequently used metrics are mean absolute error (MAE), root mean squared error (RMSE).

Mean absolute error (MAE) is a quantity used to measure how close predicted ratings are to the actual rating as shown in (Eq.7). Root mean squared error (RMSE) amplifies the contributions of the absolute errors between the predictions and the true values as shown in Eq. (8).

$$MAE = \frac{\sum_{u,i} |Pr_{u,i} - r_{u,i}|}{U*I} \quad (7)$$

$$RMSE = \sqrt{\frac{\sum_{u.i}(Pr_{u,i} - r_{u,i})^2}{U*I}} \quad (8)$$

Where number of users, number of items, predicted rating and true rating are represented by U, I, $Pr_{u,i}$ & $r_{u,i}$. As lower as the MAE or RMSE becomes, the more accurate the predictions would be, thus formulating better recommendations will be possible.

Decision-support accuracy metrics: Decision-support accuracy metrics evaluate how effectively predictions help a user to select high-quality items. Some of the frequently used metrics are recall, precision and F-Measure.

The Precision metric measures the share of successful recommendations from the total number of computed recommendations (Eq. 9), while the Recall metric is the ratio of the number of ratings correctly predicted over the total test data (Eq.10).

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

$$Recall = \frac{TP}{TP+FN} \quad (10)$$

Here, TP is an interesting item that is recommended to the user, FN is an interesting item that is not recommended to the user and FP is an uninteresting item that is recommended to the user.

There are some drawbacks with using only two metrics of recall and precision. For example, with increasing the size of recommendation list, recall will increment, while precision will decrement. The F-measure has been used to alleviate these problems through applying the harmonic average of precision and recall which is defined as follows:

$$F - Measure = \frac{2*Precision*Recall}{Precision+Recall} \quad (11)$$

5. Experimental Result

5.1 Dataset

The data for this study is drawn from a dataset of Islamic center in Iran. In order to construct missioner rate matrix, we used the overall dispatch of a missioner to a specific location as the rate of missioner for that location. All ratings are integer values between one as the lowest value and twenty two as the highest value. Also, this data includes all missioners who dispatch at least one location. Therefore, 11850 ratings is accessible for 31 locations (31 state of Iran) and 8030 missioners. The sparsely (percentage of zero values in the missioner-location matrix) is 95.24%.

5.2 Experimental Procedure

The first stage in this study is creation of user clusters. But before that, in order to improve the quality of clustering, missioner-location matrix is normalized by using Gaussian normalization. In this method, the normalized rating for item *i* by user u,$\hat{r}_{u,i}$is computed as follows [16]:

$$\hat{r}_{u,i} = \frac{r_{u,i} - \overline{r_u}}{\sqrt{\sum_i (r_{u,i} - \overline{r_u})^2}} \quad (12)$$

Where $r_{u,i}$ stands for the rating of item i by user u, and $\overline{r_u}$ stands for the average rating for user u.

Then missioners have been partitioned into 18 clusters by using *TwoStep* method and based on missioner rate data. Figure2 shows distribution of cluster proportion for *TwoStep* Clustering.

ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 6, No.18 , November 2015
ISSN : 2322-5157
www.ACSIJ.org

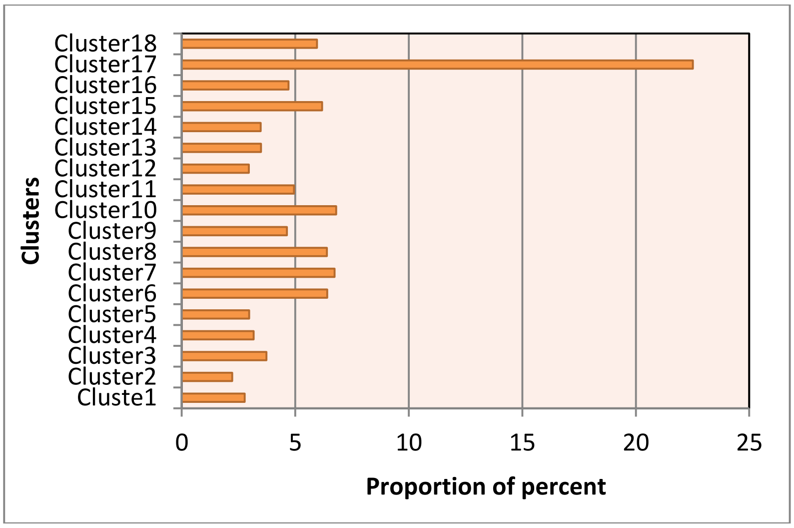

Fig.2: Distribution of cluster proportion for *TwoStep* Clustering.

As Fig. 2 shows, the *TwoStep* model tends to keep the size of different clusters balanced, which will create a better interpretation, capturing wider variations in the missioner's behavior.

In order to simplify the interpretation of the clusters, distribution of mission location in clusters' diagram is also shown in figure 3. As Fig. 3 shows, each cluster includes a location that has been requested and dispatched more than other locations. For example, in cluster1, many missioners dispatched to Gilan.

In the next phase, to conduct prediction and create recommendation, firstly the data set splits into two training and testing sets. We have chosen randomly 1606 missioners (20%) as the test set and the rest of the missioners as the training set (80%). Finally, with respect to the presented methodology in section 3, the recommender system is generated and the suitable mission location is recommended to the missioner.

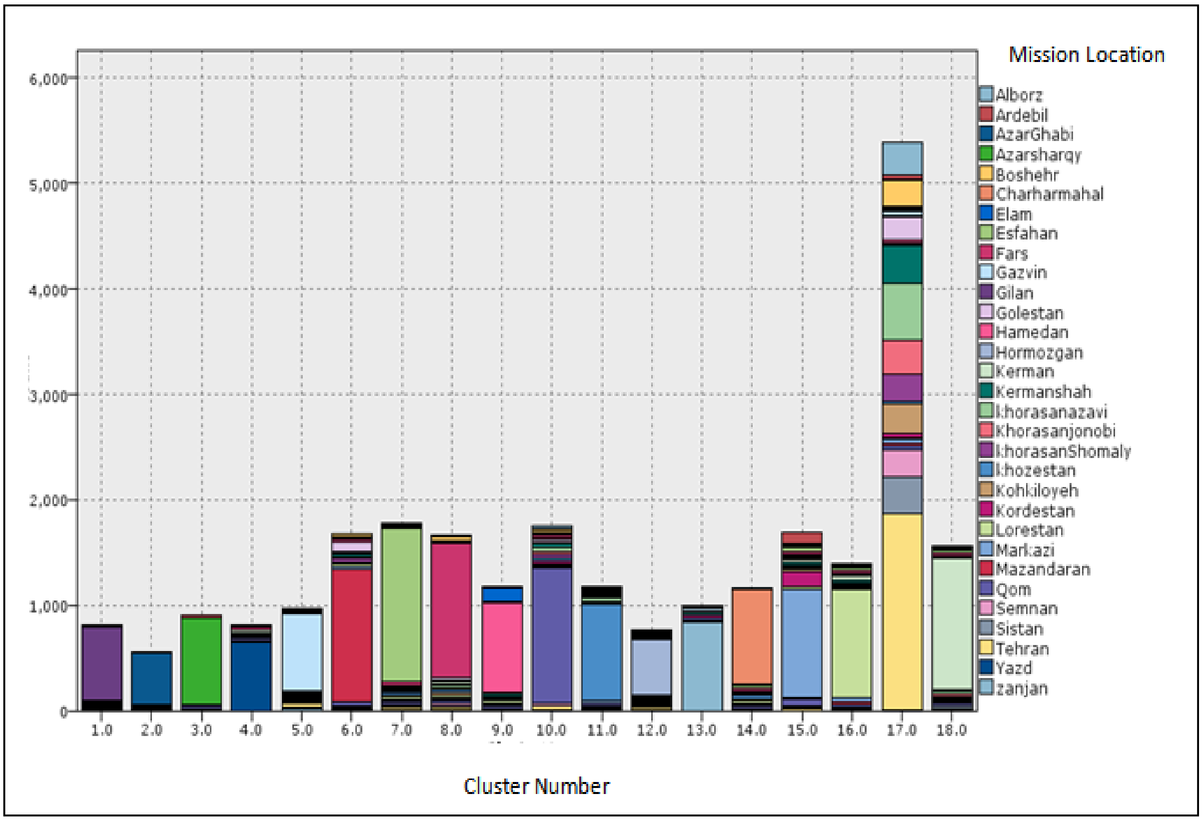

Fig. 3: Distribution of mission location in clusters.

ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 6, No.18 , November 2015
ISSN : 2322-5157
www.ACSIJ.org

5.3 Results Analysis

The size of neighborhood can have a significant impact on prediction quality; we built up our experiment by varying the neighborhood size from 5 to 30 and validating the predictions' efficiency by computing the MAE and RMSE metrics. Figure4 illustrates the sensitivity of the algorithms in relation to the different numbers of neighbors.

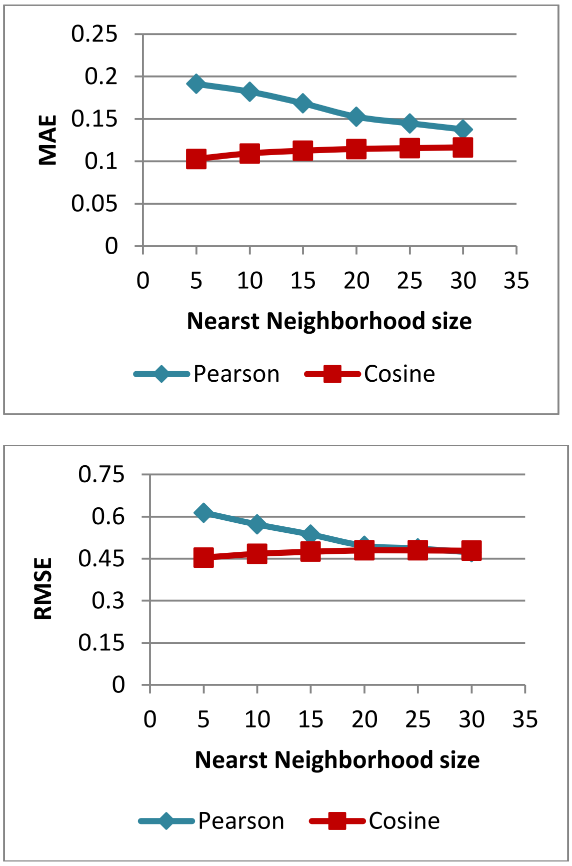

Fig. 4: Impact of neighborhood size on MAE and RMSE

As Fig. 4 shows, Pearson based and cosine based models show different types of sensitivity .In Pearson based model, as the neighborhood's size is increased the error decreases (prediction quality increases) and when this value reaches 30, the error will be minimized. Therefore, we can say that the optimal size of neighborhood is 30.

On the other hand, about cosine-based model, the prediction quality decreases by increasing the neighborhood size. Based on this observation, we select k=5 as optimal value for cosine based model.

Another important factor that affects the prediction quality is similarity measure. This study has used Pearson correlation and cosine similarity to find the similar users. For each similarity measure, we implemented the proposed algorithm to generate the prediction. Fig. 5 shows the results of the two different similarity measure on a given test set. Results shows that cosine similarity measure has better performance than Pearson similarity. Therefore, we chose cosine similarity for the rest of our experiments.

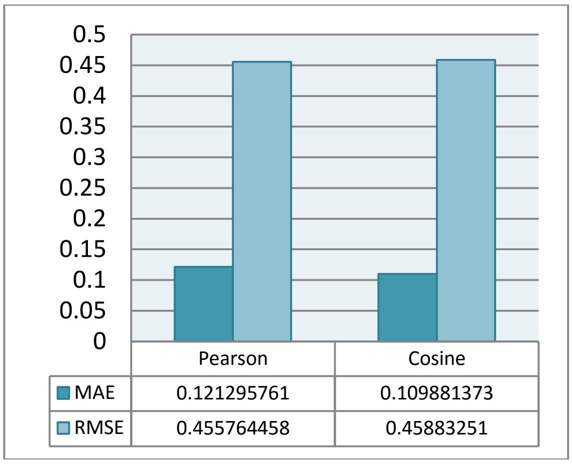

Fig. 5: Impact of the similarity computation measure on cluster based CF

We also surveyed the quality of the produced recommendations by using recall, precision and F-measure measures. Figure 6 shows the quality of recommendations with respect to different recommended locations. Figure 6 clearly specifies that precision has a reverse relationship with the number of recommended locations. So, precision decreases by increasing the number of recommended locations. On the other hand, recall has a direct relationship with the number of recommended locations. Thus, recall is increased by increasing the number of recommended locations.

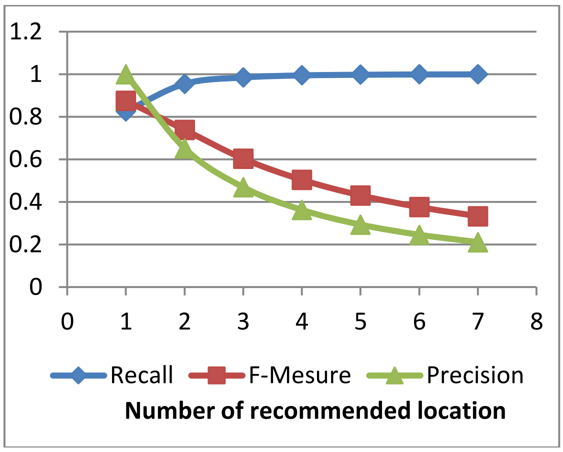

Fig. 6: Decision support measure comparison of mission location recommender system

Our purpose is to provide a ranked list of 3 recommendations. The results of observing recommendation evaluation metrics for 3 recommendations are shown in Table1. According to Table 1 we observed that considering the number of recommendations, precision metric has an acceptable rate. This means that most of the missioners can benefit from at least two of these three provided recommendations. Also high recall rate means that the system is capable of recommending most of the locations that the missioner is interested in.

Table 1: Comparison of Recall, Precision, F-measure for 3 top recommendations.

Precision (%)	*Recall (%)*	*F-Measure (%)*
47.1	98.6	60.6

Finally, to compare the performance of the proposed cluster-based recommendation algorithm with the performance achieved by user-based algorithm (memory based), we performed an experiment which calculated user-based recommendation algorithms with optimal neighborhood size of 50 and used cosine measure to compute similarity between users.

We also compared the proposed method with the clustering method which is a model based method. In clustering method three locations which have the highest scores among that cluster's members are recommended to that cluster's users.

These results are shown in Table 2. We observe that cluster based CF outperforms user based CF by the tradeoff made between Recall, Precision and F-Measure.

Although recall in the model based method is better than cluster based CF, but as it is shown in Table 2, this method operates poorly according to precision and F-measure.

Altogether, as expected, we found that cluster based CF performs better than memory based and model based techniques.

Table 2: results of comparing the performance of different algorithms.

Evaluation metric / *Method*	*Recall (%)*	*Precision (%)*	*F-Measure (%)*
Memory Based (User Based CF)	97	45.4	59
Model Based (Clustering)	100	4.9	9.2
(hybrid) Cluster Based CF	98	47.1	60.6

6. CONCLUSION AND FUTURE WORK

This paper proposes an intelligent system that can help both the missioners in selection of the suited mission location, and the dispatching manager in allocating locations to missioner and improvement of dispatching programs. This study suggests recommender systems as the suitable solution for the mentioned goals.

As collaborative filtering is a common and successful method, this paper has also used this method to recommend location to missioners. By increase in the number of missioners traditional collaborative filtering will not be able to solve the scalability problem. Therefore, we have used cluster based CF. Our experimental results proved suitable performance of this approach. We also showed that cluster based CF is more accurate and scalable than user based CF and clustering technique.

In future, we plan to use user profile's data to overcome the new user problem. Additionally, we plan to engage content-based algorithm which takes into account the content of location, to improve the quality of further recommendations.

Acknowledgments

The authors would like to thank Mr. Saadatmand for providing the experimental data and helping with data realization.

Reference

[1] A. Kohrs, and B. Merialdo, "Clustering for Collaborative Filtering Applications". In Proceedings of CIMCA'99. IOS Press, (1999).

[2] A.M. Rashid, S.K. Lam, G. Karypis, and J. Riedl, "ClustKNN: A Highly Scalable Hybrid Model- & Memory-Based CF Algorithm". WEBKDD, (2006).

[3] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Recommender Systems for Large-scale E-Commerce: Scalable Neighborhood Formation Using Clustering", the Fifth International Conference on Computer and Information Technology (ICCIT), (2002).

[4] D. Bridge, and J. Kelleher, "Experiments in sparsity reduction: Using clustering in collaborative recommenders", in Procs. Of the Thirteenth Irish Conference on Artificial Intelligence and Cognitive Science, (2002), pp. 144–149.

[5] D. M. Pennock, E. Horvitz, S. Lawrence, C. L. Giles, "Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach", in Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI), (2000).

[6] E. Vozalis, and K. G. Margaritis, "Analysis of Recommender Systems' Algorithms", In The Sixth Hellenic European Conference on Computer Mathematics and its Applications (HERCMA 2003), (2003).

[7] G. Adomavicius and A Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions", IEEE Trans. Knowl. Data Eng, (2005), Vol.17, No. 6, pp. 734-749.

[8] G. Xue, C. Lin, and Q. Yang, "Scalable collaborative filtering using cluster-based smoothing". In Proceedings of the ACM SIGIR Conference, (2005), pp.114-121.

[9] H. Jon, A. K. Joseph, R. John, "An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms", (2002) Information Retrieval, Vol. 5, pp. 287-310.

[10] J. Herlocker, J. L. Konstan, G. Tervin and J. Riedl, "Evaluating collaborative filtering recommender systems". ACM Transactions on Information Systems, (2004), Vol. 22, No.1, pp. 5-53.

[11] J. Kelleher and D. Bridge, "Rectree centroid: An accurate, calable collaborative recommender". In Procs. Of the Fourteenth Irish Conference on Artificial Intelligence and Cognitive Science, (2003), pp. 89–94.

[12] M. C. Pham, Y. Cao, R. Klamma, M. Jarke, "A Clustering Approach for Collaborative Filtering Recommendation Using Social Network Analysis", Journal of Universal Computer Science, (2011) , Vol. 17, No.4, pp. 583-604.

[13] M. Papagelis, D. Plexousakis, "Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents", Engineering Application of Artificial Intelligence, (2005), Vol.18, pp. 781-789.

[14] N. Mittal, R. Nayak, MC Govil and KC Jain, "Recommender System Framework using Clustering and Collaborative Filtering". 978-0-7695-4246-1/10- IEEE, Third International Conference on Emerging Trends in Engineering and Technology, (2010).

[15] Q. Yang, W. Xi, H. J. Zeng, Y. Yu, and Z. Chen, "Scalable Collaborative Filtering Using Cluster-based Smoothing", Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, (2005), Vol. 28,pp. 114-121.

[16] R. Jin and L. Si , "A Study of Methods for Normalizing User Ratings in Collaborative Filtering", Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, (2004), pp.568-569.

[17] S. Gong, "A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering", Journal Of Software, (2010), VOL. 5, NO. 7.

[18] S.H.S. Chee, J. Han and K. Wang, "Rectree: An efficient collaborative filtering method", Lecture Notes in Computer Science 2114, (2001).

[19] SPSS INC. , Clementine ®12.0 Algorithms Guide, available from: http://www.spss.com

[20] Y. L. Chen, and L. C. Cheng, "A novel collaborative filtering approach for recommending ranked items". Expert Systems with Applications, (2008), Vol. 34, pp. 2396-2405.