

# Improved Collaborative Filtering using Evolutionary Algorithm based Feature Extraction

Deepa Anand  
Assistant Professor  
Christ University  
Bangalore, India

## ABSTRACT

The ubiquity of Collaborative Filtering systems is evident in the wide variety of domains to which they have been applied successfully. However a major challenge to such systems is the high dimensionality and sparsity of the expressed preferences. Dealing effectively with large user profiles would improve the scalability of the system whereas reducing sparsity would increase the quality of recommendations. Several approaches in this direction have focused on feature selection and feature extraction in order to reduce the data dimension and thus make the recommendation process more scalable. Some of the features extraction techniques are based on extracting content based features. However many such solutions have been handcrafted and thus not guaranteed to work optimally under all data environments. This work explores Evolutionary algorithms based feature extraction techniques where the extracted features may describe user or item profiles. The features constructed/extracted thus are compact, dense and are discriminative. Moreover they have the advantage of requiring no extra information (such as content description) and are adaptive, delivering the optimal feature extraction scheme for the particular dataset. We have performed experiments with the popular MovieLens dataset and have compared the user-based and item-based evolutionary feature extraction schemes with respect to their accuracy. The experiments establish that the evolutionary feature extraction schemes score over traditional algorithms as well as content-based feature extraction schemes. Moreover we find that the item-based evolutionary feature extraction schemes outperform their user-based counterparts under varying parameter values.

## General Terms

Web Intelligence, Recommender Systems

## Keywords

Evolutionary Algorithms, Feature Extraction, Recommender Systems, User-based Collaborative Filtering, Item-based Collaborative Filtering

## 1. INTRODUCTION

Collaborative filtering (CF)[1][3] recommender systems predict item preferences for various users based on the preferences of other users with similar tastes. Collaborative filtering perfectly implements the idea of word-of-mouth promotion, in which the opinions of friends and benchmarking reports are the predominant influences on a purchaser's buying decision[12]. As one of the most successful technologies for recommender systems, it has been widely developed and improved over the past decade[13]. CF systems score over other recommendation techniques due to their ability to offer out the box suggestions and thus are the

dominant recommendation strategy. Such systems are categorized into memory based and model based algorithms. Memory-Based CF, which operates over the entire database to make a prediction in a totally online process, and Model-Based ACF, which uses the database to compile a similarity model is an online process[14]. Furthermore, memory-based CF can be performed based on user-based or item-based similarity. User-based CF computes the estimate of user preference for an item by finding similarity of a user with other users. The top K similar users' ratings on various items are then aggregated to offer recommendations. Item based CF on the other hand compare items for similarity and the rating for an unknown item by an active user is estimated by aggregating the user's ratings on similar items. The performance of both these variants, however, has been hampered by the high dimensional and sparse data that the algorithms learn from.

Attempts to make CF systems scalable have typically involved reducing the dimension of data to be utilized. Dimensionality reduction techniques such as Singular Value Decomposition, Latent Semantic Indexing [1] etc, have been utilized to bring down the dimensions and consequently the computational cost. Approaches utilizing the content based features of items condense the user profile consisting of preferences for item into a feature preference vector [7]. In domains where the number of features is much smaller than the number of items/users, the user/item profile becomes compact thus speeding up the similarity computation. We explore the evolutionary approach to deriving a compact user model where the features are extracted or constructed from the existing features (item ratings). This paper extends a previous work[15] to extract compact user profiles from existing preference elicitation. It is to be noted that a set of compact user profiles can be utilized in two ways vis-à-vis user based CF and item based CF. Applying the user-based CF on the compact profiles will reduce the computational effort in estimating similarity between a pair of users but the number of users compared is equivalent to traditional approaches. Another way of viewing the compact user profile matrix is to imagine the items having been replaced by pseudo-items which are less in number but are representative of the original set of items. An item based CF on the pseudo-items would imply lesser number of item-pairs between whom the similarity needs to be computed. A similar logic can be applied to compact user-profile for items. In this paper we compare these four variants where the compact user/item profiles are evolved using EA. The variants are compared against each other under varying data conditions.

The paper is organized as follows: Section 2 provides a brief review of recommender systems, evolutionary algorithms and feature selection/extraction techniques proposed in the past.

The proposed approach is detailed in Section 3 while Section 4 presents the experimental evaluation. We finally conclude with Section 5 which also presents future research directions

## 2. LITERATURE REVIEW

### 2.1 Collaborative Filtering

A Collaborative Filtering[1][8] systems identify a neighborhood set for an active user, for whom recommendations are to be made, by means of similarity measures which capture the degree to which their preferences match with those of users in the neighborhood set. User similarity is generally estimated through Pearson Correlation Coefficient (PCC) [8] and Vector Similarity[1]. PCC defines similarity between users  $x$  and  $y$  as:

$$sim(x, y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)(r_{ys} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)^2 \sum_{s \in S_{xy}} (r_{ys} - \bar{r}_y)^2}} \quad (1)$$

where  $S_{xy}$  is the set of items which users  $x$  and  $y$  have co-rated and  $\bar{r}_x$  is the mean rating for user  $x$ . Whereas Vector Space Similarity defines similarity as:

$$sim(x, y) = \frac{\sum_{s \in S_{xy}} r_{xs}r_{ys}}{\sqrt{\sum_{s \in S_{xy}} r_{xs}^2 \sum_{s \in S_{xy}} r_{ys}^2}} \quad (2)$$

To estimate item similarity however VS between a pair of item profiles is used. Once similar users are identified, the ratings of these users for the desired item are aggregated to predict the preference of the active user for the desired item. The various items not experienced by the active user, may then be presented to the user in descending order of their predicted scores. Resnick's formula[8] is typically used to aggregate the scores to get the final predicted value. It is defined as

$$pr_{i,k} = \bar{r}_i + \frac{\sum_{j \in N(i)} sim_{i,j} * (r_{j,k} - \bar{r}_j)}{\sum_{j \in N(i)} |sim_{i,j}|} \quad (3)$$

Effective estimate of user similarity is the key to the success of CF systems. However the numbers of rate able items far surpass the user's ability to consume them and with the result the ratings/preference matrix is very sparse. This is a major bottleneck in similarity computation which bases its computation on the set of common items that pair of users has rated. A sparse user profile implies a smaller or no overlap in their expressed preferences. The estimate of similarity between a pair of users is thus of low quality hampering the quality of recommendations. Scalability is another issue which is of major concern in making CF systems deliver timely suggestions. Web RS have typically millions of items

which can be acquired by a user and hence the user profiles are sizeable enough to be of concern while computing the user similarity.

Several solutions, to the above mentioned problem of sparsity and scalability, have been proposed in the past. Transitivity of similarity [5], trust[4] and tags has been harnessed in various forms to estimate the closeness between users who have very few or no common ratings. Many of these methods, however, have exclusively addressed the sparsity issue. One direction in making the systems more scalable has been to reduce the feature space. Feature selection techniques are a step in this direction. Feature selection techniques [6] work by selecting a subset of features with high discriminative capacity and utilize these features in comparing user profiles for similarity computation. The smaller user profile size implies lower computational cost while estimating user similarities and thus more timely recommendations. However, due to the sparseness of the original data the user profiles through the feature subsets are also sparse hence affecting the quality and coverage of recommendations. Feature extraction techniques [7] aided by content information of items construct user profiles by computing their interests on content based features. Hence the system gauges a higher user interest in a particular feature if several items, preferred by the user, contain the feature. Profiles built in this manner tend to be denser and more compact but rely on the existence of content based information. Several domains such as images may be difficult to describe and may not have the required information. Moreover there may be latent features, not captured in the content descriptions, which may characterize user preferences better. Dimensionality reduction techniques like Singular Value Decomposition have been explored in the past to derive a lower rank approximation of the ratings matrix and consequently reduce processing time. Sarwar et al [10] propose that the SVD-based approach to recommendations produced results that were better than a traditional collaborative filtering algorithm in case of a reasonably dense dataset. In [11] the authors investigate the utility of techniques such as SVD, Random Indexing(RI), Reflective Random Indexing (RRI) and Randomized Singular Value Decomposition (RSVD) for aiding collaborative filtering. They conclude that a combination of RRI and SVD delivers a better recommendation diversity, though SVD results in lesser prediction errors. Our previous work [15] proposed an evolutionary method of constructing a compact set of representative features which describe the user preferences effectively. It was found that the extraction of such new features not only reduced the dimensionality thus facilitating faster recommendations, but also were able to extract latent features not explicitly specified. The current work focuses on use of the compact user profile in two ways: for similarity computation or to apply item-based CF on the pseudo items generated. A similar approach is followed for each item, i.e. reduce the user space by replacing the large number of users by a smaller representative set of pseudo-users so that its utilization is also two-fold.

### 2.2 Evolutionary algorithms

Evolutionary algorithms (EAs) are stochastic optimization techniques based on the principles of natural evolution [2]. They operate on the Darwinian notion of survival of the fittest. These algorithms typically maintain a collection of potential solutions which constitute a population. The population is evolved over several generations until a predetermined number of iterations or until a stopping criterion is met. Potential candidates in the new generation are produced by applying some genetic operators on some of the

member of the current population. All the individuals in the current population as well as the newly produced offsprings are then evaluated for their quality as a solution to the problem at hand. The quality of an individual is measured by a fitness function and a fitness value is assigned to each individual. EAs typically employ crossover and mutation operators for the generation of new individuals. The crossover operator is based on the assumption that different parts of the optimal solution can be independently discovered, and be later combined to create better solutions [2]. Whereas crossover allows creation of two new individuals by allowing two parent chromosomes to exchange meaningful information, mutation is used to maintain the genetic diversity of the population by introducing a completely new member into the population.

EA and the related Genetic Algorithms have in several instances, been used for feature extraction. Peil et al., [9] propose a hybrid GA and kNN approach to classification. GA is used to select and extract representative features. These features are in turn are used for classification and the performance of the classifier is then fed back to the GA.

### 3. FEATURE EXTRACTION USING EVOLUTIONARY ALGORITHMS

In our previous work [15], we proposed an approach to evolve an optimal map to construct a compact set of features from the high dimensional user ratings matrix. The map is evolved so that the extracted features are highly discriminative. We reproduce a part of the work from [15] to simplify the discussion that follows.

#### 3.1 Proposed System

Consider a system with a set of  $n$  users  $U = u_1, u_2, \dots, u_n$  and a set of  $m$  items,  $I = i_1, i_2, \dots, i_m$ .

The ratings of various users for different items are given by a ratings matrix  $M$  such that rating for an item  $i_y$  by a user  $u_x$ ,

is given by  $M_{xy}$ . [15] explored a linear transformation of the rating matrix such that the rating matrix  $M_{xy}$  is mapped to a transformed matrix  $M'_{xk}$ , such that  $k \ll m$ . This is achieved through a transformation function

$$T: R^{n \times m} \rightarrow R^{n \times k}$$

where  $T$  can be represented by a matrix of size  $m \times k$ . Applying the transformation the user-item ratings matrix on function reduces the user-item preference matrix to a user-feature preference matrix of size  $n \times k$ . We can imagine the user-item rating matrix being reduced to a transformed preference matrix where the original set of items are replaced by a set of “pseudo-items” which though small in number, are able to capture all information related to user preferences as the larger dimensional item set. The previous work [15] exploited the reduced user-item matrix to find similarity between user pairs. However the reduced matrix may be harnessed in another way too, to receive recommendations. Fig. 1 and 2, illustrate both methods of generating recommendations employing the reduced user-item matrix. Following the procedure prescribed in [15] a user based CF algorithm may be applied and the transformed preference matrix is utilized for finding user similarity. Thus, for every user pair, the similarity estimation is less computation

intensive. However the aggregation of ratings from similar users was performed using the original rating matrix. We refer to this method of feature extraction and usage as **User-based CF using Evolutionary Item-Extraction(UBEI)**. The other option is to apply item-based CF on  $M'$ . Thus the “user profile” of the item for which the rating needs to be predicted, is compared to the ‘user profiles’ consisting of their preferences for the pseudo-items in order to estimate similarity. The computed similarities can be then used for computation of the predicted rating. Here the computational effort is reduced due to the fact that an item needs to be compared to less number of items for a prediction to be obtained. Henceforth we refer to this method as **Item-based CF using Evolutionary Item-Extraction(IBEI)**

A similar reasoning can be applied the other way, i.e. devising transformation function which reduce the user-item preference space to a “pseudo-user-item preference space, where the large number of users is replaced by a small number of pseudo-users who carry the discriminatory information from the original set of users by eliminating any redundancies therein. Thus the transformation function in this case  $S$  is

$$S: R^{n \times m} \rightarrow R^{l \times m}$$

defined as  $S$  hence can be represented by a matrix of dimensions  $l \times m$  so that  $M'' = S \times M$ . Analogous to the case of pseudo-items, the preference matrix based on “pseudo-users” can be used for User-based or Item-Based CF. We refer to these techniques as **User-based CF using Evolutionary User-Extraction(UBEU)** and **Item-based CF using Evolutionary User-Extraction(IBEU)** respectively.

The transformation matrix  $T$  or  $S$ (as the case may be) can be learnt using Evolutionary algorithms. Since both  $T$  and  $S$  are matrices whose dimensions are  $m \times k$  or  $l \times n$  respectively, where  $k$  and  $l$  can vary according to the number of features (items/users) needed to be extracted, we apply specialized crossover and mutation operators. To evolve matrix  $T$  we follow a two point column crossover where the crossover of two chromosomes are performed by interchanging a set of columns between the two matrices. So for two chromosome

$$C_1 = [a_1 \dots a_k] \text{ and } C_2 = [b_1 \dots b_k]$$

two points for each chromosome is randomly generated,  $a_1$  and  $a_2$  for  $C_1$  and  $b_1$  and  $b_2$  for  $C_2$  such that  $1 < a_1, a_2 < k$  and  $1 < b_1, b_2 < k$ .

The offsprings  $C'_1$  and  $C'_2$  are generated by interchanging the columns  $a_1$  to  $a_2$  in  $C_1$  and columns  $b_1$  to  $b_2$  in  $C_2$ . A similar methodology may be followed for crossover of matrix to obtain the optimum user transformation function  $S$ . Here we follow a two-point row crossover where for two chromosomes

$$C_1 = [x_1 \dots x_n] \text{ and } C_2 = [y_1 \dots y_n]$$

two points are randomly chosen along the rows of both chromosomes so that the portion of the chromosome between the two points, treated as row numbers, are exchanged. Chromosomes are mutated by randomly selecting a particular row/column and changing the degree of membership of a few chosen items for that particular feature represented by the row/column.

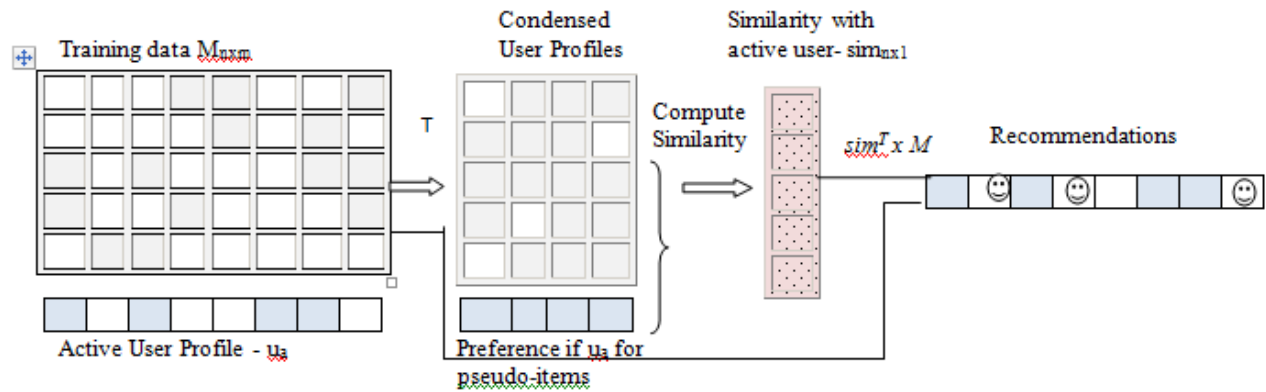


Fig.1 : Steps in UBEI – Computation of user similarity is more efficient due to smaller user profiles

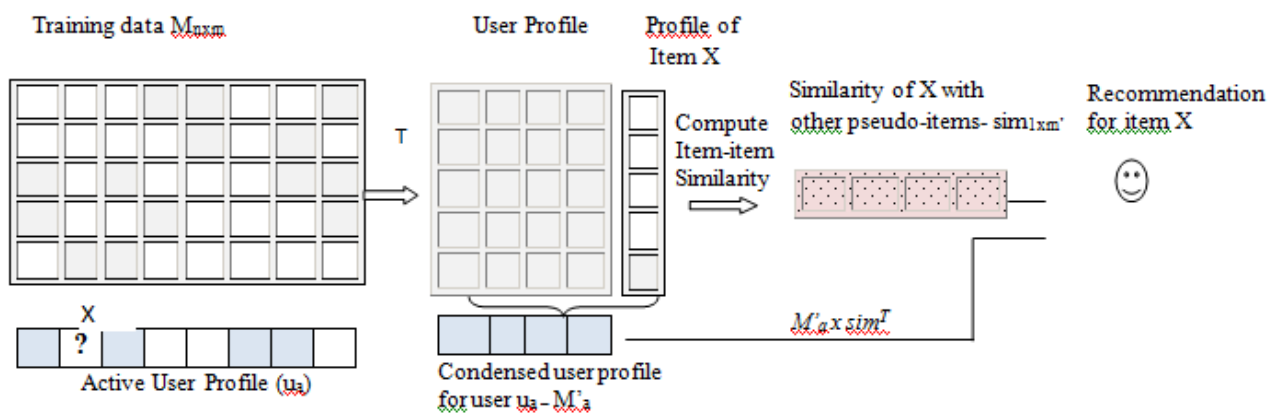


Fig 2: Steps in IBEI – Computation of item similarity is more efficient since the number of pseudo-items is much smaller than the original set of items.

Chromosomes in the initial population are constructed by generating a matrix of random real numbers in the range [0,1].

The matrix dimensions are themselves randomly generated as a first step, i.e. if we follow the item extraction schemes (UBEI or IBEI), then the number of users in the transformed matrix is the same as the original matrix but the number of items(features) would reduce, hence the requirement to randomly determine the number of items(features). A similar approach is followed for the user-extraction cases (UBEU and IBEU) where the number of pseudo-users is randomly generated. The number of extracted features is restricted to be in the range [3,50]. The generated chromosomes are filtered by setting entries in the chromosome below a certain threshold T to be zero.

### 3.2 Fitness Function

A fitness function quantifies the optimality of a chromosome and guides the process towards achieving its optimization goal by allowing fitter individuals to breed and thus hopefully improve the quality of individuals over the generations [3]. The fitness of the linear transformation function is measured by the quality of features constructed. The extracted features' quality can be gauged by the magnitude of prediction errors obtained via estimates of similarity derived from them. To compute the fitness value for a chromosome the available data is split into three sets the training set  $Tr$ , the validation set  $V$ ,

and the test set  $T$ . The training set is used for neighborhood construction and for prediction while the test set  $T$  is treated as unseen ratings which are used to evaluate and compare various recommendation approaches. Validation set is used to compute the fitness of chromosomes. The fitness of an individual is computed by considering the ratings in the validation set as unknown and using the training set to perform predictions. The fitness of an individual is then calculated as the mean absolute error of predictions using the chromosome on the training data.

## 4. EXPERIMENTAL EVALUATION

To establish the effectiveness of the proposed approaches we compare them against the traditional measures of similarity such as Pearson Correlation Coefficient (PCC)[8] and Vector Similarity (VS)[1] for both user-based and item-based CF. The various approaches to using evolutionary algorithm for feature-extraction and collaborative filtering are also compared with each other. To contrast the various approaches we use the Movielens dataset which is freely available. The Movielens dataset consists of 100,000 ratings provided by 943 users on 1682 movies. The ratings scale is in the range 1-5 with 1 - "bad" to 5 - "excellent".

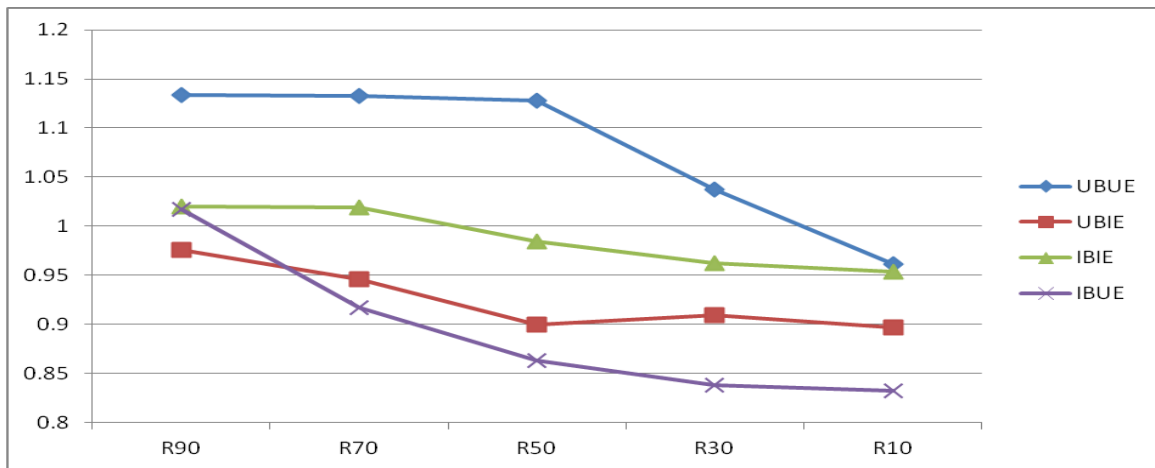


Fig 5: Comparison of different Evolutionary Feature Extraction mechanisms under varying sparsity conditions

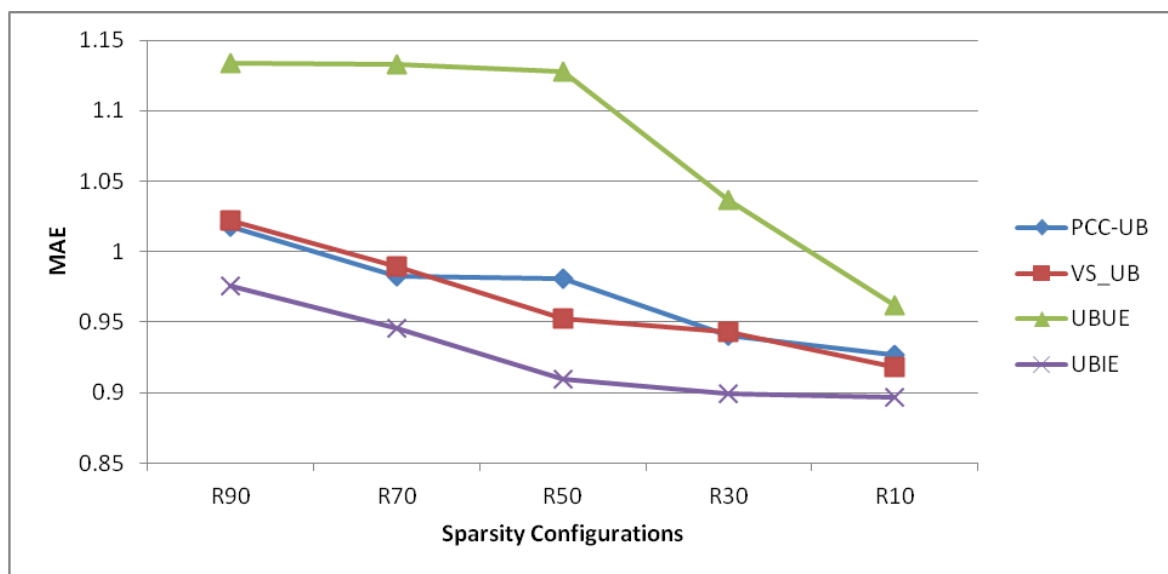


Fig. 3: Comparison of user-based CF techniques under varying sparsity configurations

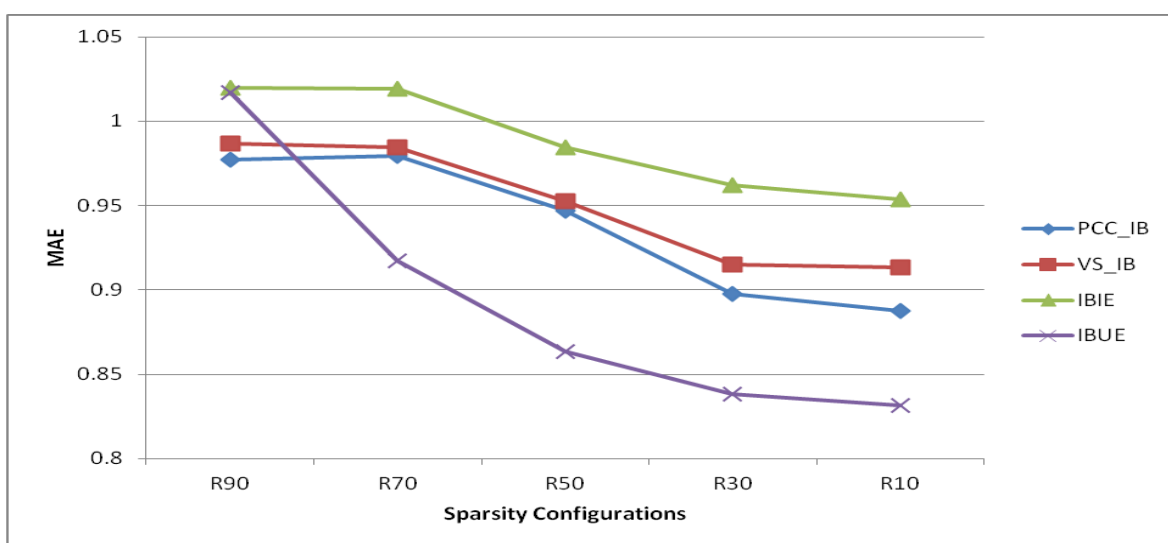


Fig. 4: Comparison of item-based CF techniques under varying sparsity configurations

The ratings are discrete. The various approaches compared are the PCC with both the user-based and item-based variations respectively called PCC-UB and PCC-IB and similarly the user and item based version of Vector similarity VS-UB and VS-IB. These methods are compared against UBEI, IBEI, IBEU and UBEU methods as detailed in the previous sections. The various approaches are compared by computing their Mean Absolute Error on the test set which is defined as;

$$MAE = \frac{1}{M_T} \sum_{k=1}^{M_T} |pr_k - r_k|, \quad (4)$$

where  $M_T$  is the number of ratings in the test dataset.  $pr_k$  is the predicted rating for the  $k$ th rating in the test set and  $r_k$  is the actual rating. For each user we randomly select 5 ratings for the test set and 5 ratings to be included in the validation set. The chosen ratings are removed from the training data set. The population size for each generation of the EA is set to 20. The maximum number of iterations is set to 10. The parameter T is set to 0.5. The size of the neighborhood set to 20.

Since sparsity is a challenge which RS face and real life datasets may have varying amount of sparsity, we evaluate the various techniques on varying levels of sparsity. The sparsity is varied by discarding 10%, 30%, 50%, 70%, 90% of the ratings from the training set for each user and retaining the rest of the ratings for training giving rise to configurations R10, R30, R50, R70, R90. Fig. 3 compares the approaches using user based CF i.e. PCC-UB, VS-UB, UBIE, UBUE. As is evident from the figure UBIE gives the least MAE under all sparsity configurations establishing the effectiveness of the features (pseudo-items) extracted. However UBUE gives the worst performance among all user-based methods, thus reducing the number of users and finding user similarity among pseudo-users so extracted is not of much help. This may be due to the low number of users(pseudo-users) contributing to the predictions. The PCC and VS techniques have almost comparative performance. Item based CF techniques are compared in Fig. 4. For all configurations except R90, IBUE performs the best. In fact it outperforms the other techniques by a wide margin. PCC has a slightly better performance than VS. However, IBIE has the worst performance. Fig. 5 shows the comparison of the various evolutionary feature extraction schemes and User/item based CF combination. Among all techniques IBUE performs the best under all sparsity conditions, UBIE is the next best. However both IBIE and UBUE give the largest MAEs with UBUE being the worst. This can be attributed to the fact that both IBIE and UBUE perform collaborative filtering(user or item based) where the number of users/items is reduced quite a bit. This may mean that fewer users/items contribute while aggregating the ratings and this might be the reason for the lowering of accuracy.

## 5. CONCLUSIONS

This paper explores evolutionary algorithms as a means to extract useful features (users or items) for collaborative filtering based recommendations. Features extracted are in the form pseudo-users or pseudo-items which effectively represent the large set of users/items respectively. For each extraction mechanism the user-based and item-based CF methods were explored giving rise to four techniques UBUE, IBIE, IBUE and UBIE. Among the various possibilities explored it was found that IBUE and UBIE outperform the

other techniques including traditional methods such as PCC and VS with respect to accuracy under most sparsity configurations. However the other two methods UBUE and IBIE give a poor performance. IBUE and UBIE are able to effectively extract pseudo-users and pseudo-items which capture the essence of the information contained in the ratings data and the improvement in performance is due to the reduction in sparsity and better representation of the ratings information.

In the future we plan to compare the proposed techniques with methods employing other machine learning techniques such as neural networks[16]. Incorporating additional information such as trust[4], tags[1] etc may also be explored. More extensive experiments on larger datasets with varied features might make the study more effective and useful.

## 6. REFERENCES

- [1] Adomavicius, G., Tuzhilin, A. 2005/ Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, IEEE Trans. on Knowl. and Data En., vol. 17(6), pp. 734–749, doi:10.1109/TKDE.2005.99.
- [2] Alba, E., Chicano, J. 2004. Training neural networks with GA hybrid algorithms in Proceedings of Conference on Genetic and Evolutionary Computation–GECCO, pp. 852-863
- [3] Anand, D. , Bharadwaj, K.K. 2010. Enhancing Accuracy of Recommender System through Adaptive Similarity Measures based on Hybrid Features,in Proceedings 2nd Asian Conference on Intelligent Information and Database Systems(ACIIDS 2010), LNAI 5991, pp.1-10,
- [4] Anand, D. , Bharadwaj, K.K. 2012. Pruning Trust-Distrust network via Reliability and Risk Estimates for Quality Recommendations, Social Network Analysis and Mining, Springer.
- [5] Anand, D., Bharadwaj, K.K.2012. Exploring Graph Based Global Similarity Estimates for Quality Recommendations, International Journal of Computational Science and Engineering, Inderscience(Accepted for publication)
- [6] Baltrunas, L., Ricci, F. 2007. Dynamic Item Weighting and Selection for Collaborative Filtering, In Berendt, B., Mladenic, D., Semeraro, G. , Spiliopoulou, M., and Stumme, G., editors, Web Mining 2.0, International Workshop Located at the ECML/PKDD 200, Warsaw, Poland.
- [7] Symeonidis, P. 2007. Content-based Dimensionality Reduction for Recommender Systems, in Proceedings of the 31st Annual Conference of the Gesellschaft für Klassifikation, pp.619-626
- [8] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl , J. 1994. GroupLens An open architecture for collaborative filtering of netnews. In Proceedings of ACM CSCW'94 conference on computer-supported cooperative work, pp. 175–186.
- [9] Pei, M. Goodman, E. D., Punch, W. F. 1997. Feature Extraction Using Genetic Algorithms, in Proceeding of International Symposium on Intelligent Data Engineering and Learning'98 (IDEAL'98), Hong Kong
- [10] Sarwar, B., Karypis, G., Konstan, J., Riedl, J. 2000. Application of Dimensionality Reduction in

- Recommender Systems—A Case Study, in Proceedings of ACM WebKDD Workshop.
- [11] Ciesielczyk, M., Szwabe, A. 2011. RSVD-based Dimensionality Reduction for Recommender Systems, *International Journal of Machine Learning and Computing*, Vol. 1, No. 2/
- [12] Felfernig, A. Friedrich, G., Schmidt-Thieme, L., 2007. Guest Editors' Introduction: Recommender Systems, *IEEE Intelligent Systems*, v.22 n.3, p.18-21, [doi: 10.1109/MIS.2007.52]
- [13] Ding, S., Zhao, S. Yuan, Q., Zhang,X., Fu, P., Bergman, L. 2008. Boosting collaborative filtering based on statistical prediction errors, in Proceedings of the ACM conference on Recommender systems, Lausanne, Switzerland [doi>10.1145/1454008.1454011]
- [14] O'Donovan, J., Smyth, B. 2005. Trust in recommender systems. in: Proceedings 10th international conference on Intelligent user interfaces(IUI '05), ACM, pp. 167–174.
- [15] Anand, D.: 2012. Evolutionary Algorithm based Feature Extraction for Enhanced Recommendations, in Proceedings of ARTCom 2012, LNEE pp. 410–413.
- [16] Lerner, B., Guterman, H., Aladjem, M., Dinstein, I.. 1996. Feature Extraction by Neural Network Nonlinear Mapping for Pattern Classification, in Proceedings of the 13th International Conference on Pattern Recognition, ICPR13, Vienna, vol. 4, 320-324.