Feature Extraction for Collaborative Filtering: A Genetic Programming Approach

Deepa Anand

Department of Computer Science, Christ University
Bangalore, Karnataka 560029, India

Abstract

Collaborative filtering systems offer customized recommendations to users by exploiting the interrelationships between users and items. Users are assessed for their similarity in tastes and items preferred by similar users are offered as recommendations. However scalability and scarcity of data are the two major bottlenecks to effective recommendations. With web based RS typically having users in order of millions, timely recommendations pose a major challenge. Sparsity of ratings data also affects the quality of suggestions. To alleviate these problems we propose a genetic programming approach to feature extraction by employing GP to convert from user-item space to user-feature preference space where the feature space is much smaller than the item space. The advantage of this approach lies in the reduction of sparse high dimensional preference information into a compact and dense low dimensional preference data. The features are constructed using GP and the individuals are evolved to generate the most discriminative set of features. We compare our approach to content based feature extraction approach and demonstrate the effectiveness of the GP approach in generating the optimal feature set.

Keywords: *Recommender Systems, Collaborative Filtering, Genetic Programming, Feature Extraction*

1. Introduction

Recommender systems (RS) [8][11] tackle the problem of discovering interesting and novel items from a high dimensional item space tailored to a user's tastes and preferences. They can be categorized according to the type of information employed to arrive at recommendations. For example, content based recommenders match potential items for their content similarity with the items preferred by the user in the past. Collaborative Filtering systems[1] on the other hand leverage user affinity based on historical rating data to predict the user preference for various items. The desirability of each item for a user is determined based on its appeal in the user's neighborhood which comprises of the set of users whose taste match closely with the current user. Collaborative filtering systems are further categorized into memory based and model based algorithms [7]. Model based algorithms use machine learning techniques to build user models from the available ratings data offline and use the built model online to offer recommendations. Memory based algorithms on the other hand follow the lazy evaluation strategy and perform all computations at the time that the items need to be recommended. The preprocessing step used by model based algorithms make them more scalable than their memory based counterparts but they suffer from their inability to incorporate the up to date ratings information. There have been attempts to fuse both memory based and model based methods. For example [2] and [15] propose to build a condensed user profile by leveraging on content based information of items which is used for assessing similarity between users. K-Nearest neighbor approach is then used to find the predicted score for active users.

A major issue with CF algorithms is the high dimensionality of feature space. The similarity between a pair of users is gauged by comparing their profiles. The user profile in a CF system consists of the ratings by the user for the different items. Since the number of items may be in order of millions the user profile is large and consequently the comparison of user profiles for similarity estimation might require time. The user profiles, in addition to being large, are also very sparse. The number of items rated by users is a small fraction of the number of items available in the system. This implies that the overlap in ratings while comparing user profiles is small and thus may hamper recommendation quality. Several solutions to the sparsity have been proposed in the past such as employing transitivity of similarity to increase similar user base [5] or using additional information in the form of trust [4], tags etc to estimate similarity between users not having a large item overlap. A solution to both the issues of sparsity and high dimensionality is to condense the user profile into a smaller dimensional dense feature space. Methods have been proposed to condense the ratings space into feature space by utilizing the content information associated with items [2][15].

We propose a genetic programming approach to constructing a compact user profile using the available rating data. Each of the constructed features is a function of ratings for a subset of items from the available items. The features are evolved till an optimal subset of features with a good discriminative capacity is achieved. The GP method scores over the content based feature extraction techniques since they are not hindered by unavailability of content information in certain domains. Moreover they may be able to capture latent features which may not be described by the content based features. Additionally the evolutionary approach evaluates the goodness of the new features as a group. This is important since it is possible that content-based or newly constructed features are effective in matching similar users individually but are not effective as a group. Moreover there may be interactions/interdependencies between these features which may determine the efficacy of the feature set. For e.g. in the movie domain, where movies may be described by the genre to which they belong, there may be strong correlation between preference for romance genre and comedy genre. A good feature extraction scheme would account for such interactions by, for instance, considering only one of the correlated features while discarding the other. The proposed approach evaluates a set of constructed features as a whole and thus is expected to take care of these interdependencies.

The rest of the paper is organized as follows: Section 2 discusses related work. The proposed GP approach is outlined in Section 3 while Section 4 details the experimental evaluation. Section 5 presents the conclusions and points some directions for future work.

2. Related Work

2.1 Genetic Programming

Genetic Programming is a relatively recent technology which has been demonstrated as a versatile tool for Automatic Program Generation in a variety of applications[[16]. GP belongs to a set of artificial intelligence problem-solving techniques based on the principles of biological inheritance and evolution. Each potential solution is called an individual (i.e., a chromosome) in a population. GP works iteratively applying genetic transformations, such as crossover and mutation, to a population of individuals to create more diverse and better performing individuals in subsequent generations [17]. Each member in a population is assessed for its quality by using a fitness function. The programs evolved by GP are generally represented as a tree. Moreover the size of the different chromosomes in the population may vary. Because of its intrinsic parallel search mechanism and powerful global exploration capability in a high-dimensional space, GP has been used to solve a wide range of hard optimization problems that oftentimes have no known optimum solutions [17]. An interesting application of GP is in the area of weather forecasting where the evaporation loss is predicted as a function of various climatic conditions [9].In the recent past GP based techniques have been harnessed in the area of information and image retrieval. For example [12] propose methods to construct optimal classifiers for information retrieval purposes. A GP based method to evolve image similarity measures from existing features has been proposed by Torres et al. [17]. In the area of recommender systems [3] proposes a method to find optimum similarity measures among users where the optimality of a similarity function is determined by the prediction error that it produces.

2.2 Collaborative Filtering

Collaborative filtering (CF) systems are inspired by real life decision making process wherein we gather opinions about inexperienced situations/items from our acquaintances and friends and make an informed decision. The prediction using CF is performed in various stages. The users are first assessed for their similarity with other users. The predicted score for an item by a user(active user) is then performed by aggregating the ratings of other similar users for the particular item. The items may then be arranged according to their predicted score to be presented to the user. CF systems traditionally use Pearson Correlation Coefficient to assess user similarity. The similarity computation is performed using the formula;

$$sim(x,y)=\frac{\sum_{i\in S_{xy}}(r_{x,i}-\bar{r}_x)(r_{y,i}-\bar{r}_y)}{\sqrt{\sum_{i\in S_{xy}}(r_{x,i}-\bar{r}_x)^2\sum_{i\in S_{xy}}(r_{y,i}-r_y)}} \qquad (1)$$

where Sxy is the set of items which users x and y have co-rated and $\bar{\boldsymbol{r}}_{\boldsymbol{x}}$ is the mean rating for user x. Vector similarity (VS) [7] on the other hand is defined as ;

$$sim(x,y)=\frac{\sum_{i\in S_{xy}}r_{x,i}r_{y,i}}{\sqrt{\sum_{i\in S_{xy}}{r_{x,i}}^2\sum_{i\in S_{xy}}{r_{y,i}}^2}}\ , \qquad (2)$$

Effective similarity measures play a key role in the recommendation process. However as discussed in the previous section the scarcity of ratings data implies that for several user pairs the similarity cannot be estimated at all, or even if they can be they are based on a very small subset of common ratings and thus their reliability is a suspect. A variety of solutions have been proposed to the sparsity problem. Additional information in the form content details of items [2], trust information [4] and tags assigned to various items by various users have been employed in the past for enhancing the quality of recommendations. Transitivity of similarity [5] has also been explored in the past to enhance the user to user links. Most of the proposed methods however exclusively address the sparsity issue.

The high dimensionality of item space hinders the ability to offer timely recommendations. Dimensionality reduction techniques such as SVD [6], has been employed in the past to reduce the user-item space and offer timely suggestions. Other methods employ content based information to reduce the item space.[17] proposes to use item content information to estimate the user's liking for each item feature, instead of his preference for various items. Since the number of features is a much smaller quantity than the number of items, the user preferences can be expressed in a much compact form. Moreover the preference for item features would be denser since if the user has expressed his preference for even a single item with the feature we would be able to estimate his degree of liking for the feature. Another such method[2] proposes estimating user preference for various movie genres by utilizing his preference for various movies. A user who has viewed several movies of the genre 'Comedy', for example, and rated them highly will have a high degree of preference for 'Comedy'. Since the number of genres is a small quantity the user-item preference space is reduced into user-genre preference space which greatly reduces the dimension, in addition to making the matrix dense. This matrix is then utilized to assess the similarity between users. For the final prediction however the original rating matrix is used. This method hence is able to combine the advantages of model based method (fast similarity computation) with the benefits of memory based algorithms (using up to date rating information for rating prediction). The disadvantage of the above mentioned methods is that they rely on the availability of item content information. Hence they are not applicable for domains in which content information is non-existent. We propose a method employing Genetic Programming to construct a compact set of discriminative dimensions which capture the latent features in the data. The proposed method in addition to not relying on available content information is also able to store the learnt features using a constant space.

3. Genetic Programming based Feature Extraction (GPFE)

We propose to build features as a function of item ratings. Consider a system with n users U={u1, u2, …, un} and a set of m items I = {i1, i2, i3 …im}. Let the number of features that an item can contain be k, F= {f1, f2, …, fk}. Let R be the rating matrix and hence has dimensions nxm and C be the mxk matrix such that Cij =1 if ith item contains jth feature. Using genetic programming we explore means to convert the nxm user-item rating matrix into a nxk user-feature preference matrix. This can be done in general by expressing the degree of liking of a feature f by any user as a function of preference of the same user for a subset of items $i_1^f, i_2^f, ..., i_l^f$ i.e.

$$C(u, f) = \eta(R(u, i_1^f), R(u, i_2^f), ..., R(u, i_l^f))$$

This is illustrated through an example below.

Example: Fig.1(a) shows a user-item ratings matrix. Assume that the system is aware of the content based features of the various items as in Fig. 1(b), which contains an entry 1 for a item , feature pair if the item contains the feature. Then the user preference(say for user 1) for each feature (say Feature 1) can be estimated by examining the rating conferred by user1 upon items 1, 3 and 4 since these items contain the feature. In this particular example the Fig 1(c) contains the feature preference for each user where the preference for a feature is computed by summing the ratings of the user for all items containing that feature, i.e.

$$C(u, Feature1) = R(u, i_1) + R(u, i_3) + R(u, i_4))$$
$$C(u, Feature2) = R(u, i_2) + R(u, i_5)$$

Domains having content based description of items can utilize this information to construct a compact user profile, but for domains not containing this information such an approach is not possible. Even in the absence of content information it is possible that there is a strong correlation in the pattern of ratings for a set of items since they may belong to a common category. Using genetic programming we, thus, endeavor to learn a set of functions which map ratings for subsets of items to individual features. The next few sections detail the chromosome representation, and the genetic operators used.

	I1	I2	I3	I4	I5
U1	5	3		5	1
U2	3		4		5
U3		4	2	3	
U4			3	3	4
U5	1	2	1		3

(a)

	I1	I2	I3	I4	I5
F1	1		1	1	
F2		1			1

(b)

	F1	F2
U1	10	4
U2	3	5
U3	5	4
U4	6	4
U5	2	5

(c)

Fig. 1(a) The user-item rating matrix. (b) The item-feature matrix. An entry of 1 means that the feature is present in the corresponding item (c) The constructed user-feature preference matrix.

3.1 Chromosome Representation

The function mapping the ratings for a subset of items into a feature preference degree can be represented as an expression tree. Eg. Feature 1 in Example 1 can be represented as shown in Fig. 2. Using a combination of Genetic Programming and Evolutionary algorithms a set of such features, each represented by an expression tree, is evolved. An individual I in the population thus is a set of Expression trees I={E1, E2 .., Ex} where x is the number of features. The initial population is chosen by randomly choosing the number of features and generating the expression trees corresponding to each feature. The leaf node in each expression tree is either the rating for an item or a constant whereas the internal nodes contain operators($+,-,\times,\div$).The leaf nodes contain an item rating. The limit on the number of features is fixed in the interval [3, 50].

3.2 Genetic Operators

New chromosomes in the population are created by employing the genetics-inspired oprerators of crossover and mutation. While crossover facilitates exchange of meaningful information among two parent trees and helps convergence to an optimal solution, mutation helps in maintaining genetic diversity in the population and avoids being trapped in local maxima/minima. The process iterates for several generations till a stopping criteria is met [3].

Crossover

The crossover operator are applied at two levels: Feature level and tree level. Consider two chromosomes $\boldsymbol{C1=\{E_1^1,E_2^1,\cdots,E_x^1\}}$ and $\boldsymbol{C2=\{E_1^2,E_2^2,\cdots,E_x^2\}}$. The crossover at the feature level involves interchanging a set of features between the chromosome. A one point crossover is followed wherein crossover points a1 and a2 are chosen from C1 and C2 respectively. The new chromosomes generated through this type of crossover will be $\boldsymbol{C1'=\{E_1^1,E_2^1,\cdots,E_{a1-1}^1,E_{a2}^2,E_{a2+1}^2,\cdots,E_y^2\}}$ and $\boldsymbol{C2'=\{E_1^2,E_2^2,\cdots,E_{a2-1}^2,E_{a1}^1,E_{a1+1}^1,\cdots,E_x^1\}}$. The other type of crossover involves choosing a feature (represented by the expression tree) each from the chromosomes and performs a crossover of the expression trees. Figure 3 shows one such crossover. Either of the two types of crossovers is performed with the probability of choosing the methods being equal.

Mutation

One feature from the chromosome chosen for mutation is altered in two possible ways. Either the feature is completely replaced by a new feature(new expression tree). The alternative is to alter the tree corresponding to the feature chosen by replacing a subtree by another. Either of the two forms of mutation is performed with equal probability.

Fitness Function

Each individual in the population is assessed for its quality using a score known as the fitness value. The quality of a chromosome is assigned based on its ability to solve the problem at hand adequately. In the case of GPFE the fitness of a chromosome is driven by the quality of predictions obtained through it. To estimate the fitness of individuals in the population, the rating data is divided into three parts, training set **Tr**, validation set **V** and the test set **T**. The ratings in **Tr** are treated as items already rated by the user and is utilized for neighborhood construction and ratings prediction. The validation set, on the other hand, is utilized for the learning the optimal set of features. The test set is used to evaluate the quality of the learnt features and for comparison with other methods.

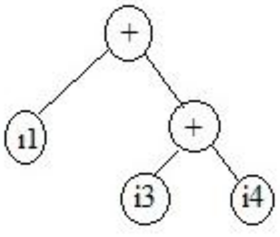

Fig 2: The feature construction tree for F1

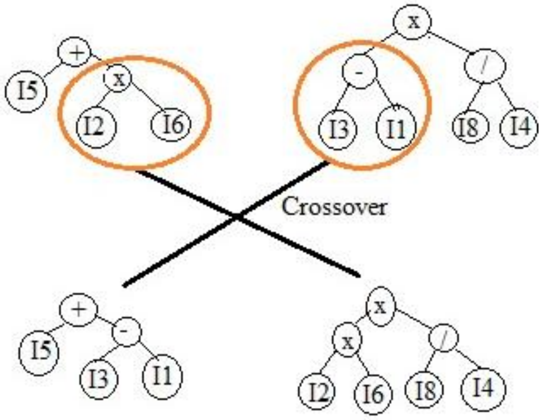

Fig 3: Tree level Crossover

The fitness for an individual is estimated by applying the feature transformations on the training matrix (**Tr**) to get user-feature preference matrix. This matrix in turn is used to compute user similarities. The computed similarities are used to predict the ratings using Eq. 2 for the set of ratings in **V**. The prediction error is then obtained as the mean absolute difference between the predicted and the actual ratings and is given by

$$fitness(I) = \frac{\sum_{r \in V} | r - r_I^{'} |}{|V|}$$

where r_i' is the predicted rating using individual *I* and *r* is the actual rating in the validation set.

GPFE based Recommendation Framework

Below we outline the proposed approach to recommendations based on GPFE

Step 1: Apply GP based on Tr and V to derive an optimal feature set C.

Step 2: Apply the learnt functions in C to the ratings in **Tr** to derive a user-feature preference matrix.

Step 3: Estimate the similarity matrix, sim(nxn) between all pairs of users. We use vector similarity Eq. 2 to compute the similarity.

Step 4: Compute predictions by using the training matrix Tr and the similarity , sim, computed in step 3 using Resnick's prediction formula[13]

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

It is to be noted that the running the genetic programming algorithm to determine the optimal feature set takes up a lot of computational resources and is time consuming, but it is to be noted the optimal feature construction process shall be performed as an offline process, while the constructed feature set will be used online for predictions.

4. Experimental Evaluation

We demonstrate the effectiveness of the GPFE approach by contrasting their recommendation accuracy with content based feature construction technique as outlined in [17] and traditional measures of similarity specifically, PCC [13] and VS[7]. MovieLens is a movie rating dataset which contains ratings given to 1682 movies by 943 users. MovieLens also contains content information items in the form of genres that a movie belongs to. There are 18 genres that a movie can belong to such as Romance, Comedy, Mystery etc. Since we are proposing a system for movie prediction, we follow the approach outlined in [17] for deriving the interest of a user in various features (genres). However we do not apply SVD since the method is used for ranking items rather than prediction. We henceforth refer to this method by Content based Feature Extraction (CBFE). The approaches are compared via their Mean Absolute Error on ***T*** which is defined as;

$$MAE = \frac{1}{|T|} \sum_{k=1}^{|T|} | pr_k - r_k |, \tag{3}$$

where **|*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

The ratings dataset is preprocessed to filter out users who have rated less than six items. Since sparsity is one of the major challenges facing CF algorithms, we study the effect of various levels of sparsity in the data, on the performance of the various algorithms. To do this we

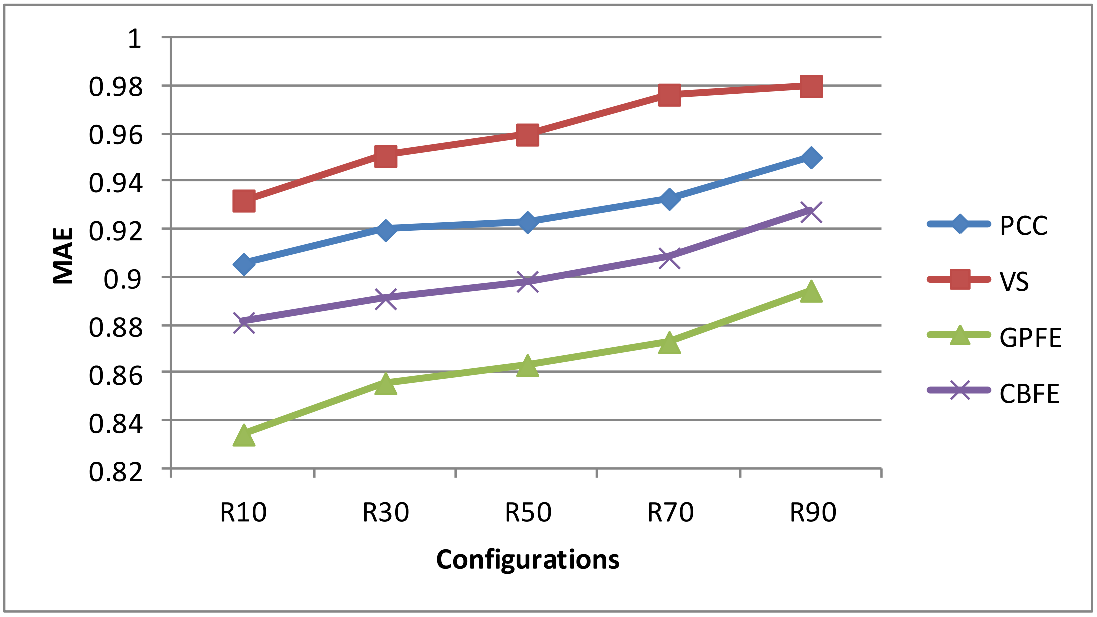

Fig 5. Comparison of PCC, VS, GPFE and CBFE with respect to MAE on varying sparsity levels

divide the dataset into training set, validation set and test set. For each user we set aside two random ratings each into the validation set and test set respectively. The rest of the ratings belong to the training set. To introduce sparsity we randomly remove 'R'% of ratings from the training set. We vary the R among the values in the set { 10, 30, 50, 70, 90} to get five configurations R10, R30, R50, R70, R90 respectively. The number of nearest neighbors K is set to 20. For the GP the population size is set to 20. The number of iterations is set to 15.

The accuracy of the various methods under varying sparsity levels is shown in Fig. 4. As is clear from the figure the proposed approach (GPFE) outperforms all the other methods under all sparsity levels. With increase in the sparsity levels the MAE of all methods increase, which is as expected. Among the different methods CBFE method outperforms the traditional methods of estimating similarity. VS based similarity measure performs the worst among all the methods. Fig. 4 shows the evolution of the fitness along the various iterations. As seen from the figure, initially the decrease in the MAE is rapid along the first few iterations (with the exception of iteration 10, when the error increases). Towards the last five iterations the decrease is very minor. Such experiments were performed with other configurations. Though in some configurations the number of optimal number of iterations is more than fifteen, we fix the number to fifteen since we wish to balance the quality of recommendations with the amount of time taken to determine the optimal feature map.

5. Conclusions and Future Work

The aim of the proposed work is to tackle the problem of high dimensionality and sparsity typical to RS data. To this end we propose a genetic programming based feature extraction technique to transform the user-item preference space into a condensed and dense user-feature preference space. The transformation functions so constructed can model any linear or non-linear function. The proposed approach is able to combine the advantages of both memory-based and model-based techniques since the condensed user profile is employed for user similarity computation whereas the original training matrix is used for the rating prediction. In addition to not relying on content information to guide the feature construction process the proposed technique also bases the evolution of the features as a set by thus enabling measurement of individual goodness of feature as well as accounting for interactions therein. Experimental comparisons demonstrate the enhanced recommendations produced by the proposed GP based approach as compared to content based feature construction techniques as well as traditional CF methods.

In the future we plan to study the effect of tweaking the fitness function by accounting for the coverage obtained as well as other measures such as diversity [1]. We also plan to evaluate the approach on more varied domains with larger datasets. Domains which require good suggestions for various users without the actual predicted ratings utilize measurement based on classification accuracy. The GP based method could be evaluated on its effectiveness in achieving high classification accuracy. We also plan to employ other approaches inspired by genetics such swarm optimization and ant colony optimization techniques[10] and propose to compare the various learning techniques in terms of their feature extraction abilities.

References

[1] A. A. Name, and B. B. Name, Book Title, Place: Press, Year.
[2] A. Name, and B. Name, "Journal Paper Title", Journal Name, Vol. X, No. X, Year, pp. xxx-xxx.
[3] A. Name, "Dissertation Title", M.S.(or Ph.D.) thesis, Department, University, City, Country, Year.
[4] A. A. Name, "Conference Paper Title", in Conference Name, Year, Vol. x, pp. xxx-xxx.

[1] 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. on Knowl. and Data En., Vol. 17, No. 6, 2005, pp. 734–749.
[2] M.Y.H. Al-Shamri and K.K. Bharadwaj, “Fuzzy-Genetic Approach to Recommender System Based on a Novel Hybrid User Model”, Expert Systems with Applications, Elsevier, Vol. 35, No. 3, 2008, pp. 1386-1399.
[3] D. Anand and K.K. Bharadwaj, “Adaptive user similarity measures for recommender systems: A genetic programming approach”, In: Proceedings 3rd IEEE international conference on Computer Science and Information Technology, IEEE, China ,2010, pp.121-125.
[4] D. Anand and K.K. Bharadwaj. “Pruning Trust-Distrust network via Reliability and Risk Estimates for Quality Recommendations”, Social Network Analysis and Mining, Springer, 2012
[5] D. Anand and K.K. Bharadwaj, “Exploring Graph Based Global Similarity Estimates for Quality Recommendations”, International Journal of Computational Science and Engineering, Inderscience(Accepted for publication)
[6] D. Billsus and M. Pazzani, “Learning collaborative information filters”, In International Conference on Machine Learning, Morgan Kaufmann Publishers, 1998.
[7] J.S. Breese, D. Heckerman, aned C. Kadie, “Empirical Analysis of Predictive Algorithms for Collaborative Filtering”. In Proceedings of the fourteenth annual conference on uncertainty in artificial intelligence. Morgan Kaufmann, 1998, pp. 3–52.
[8] R. Burke, “Hybrid Recommender Systems: Survey and Experiments” User Modeling and User-Adapted Interaction Vol. 12, 2002, pp. 331-370.
[9] K.S. Kasiviswanathan,R. Soundhara, R. Pandian, S. Saravanan, A. Agarwal, “Genetic programming approach on evaporation losses and its effect on climate change for Vaipar Basin”, International Journal of Computer Science Issues, Vol. 8, Issue 5, No 2, September 2011.
[10] S. Nadi, M.H. Saraee, M.D. Jazi, A. Bagheri, ”FARS: Fuzzy Ant based Recommender System for Web Users”, International Journal of Computer Science Issues, Vol. 8, Issue 1, January 2011
[11] U. Niranjan, R.B.V. Subramanyam, V. Khanaa, “An Efficient System Based On Closed Sequential Patterns for Web Recommendations”, International Journal of Computer Science Issues, Vol. 7, Issue 3, No 4, May 2010
[12] N. Oren, “Reexamining tf.idf based information retrieval with Genetic Programming”, In Proceedings of Annual conference of the South African Institute of Computer Scientists and Information Technologists, South African Institute for Computer Scientists and Information Technologists, 2002, pp.224-234
[13] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “Grouplens An open architecture for collaborative filtering of netnews”, In Proceedings of ACM CSCW’94 conference on computer-supported cooperative work, 1994, pp. 175–186.
[14] B.M. Sarwar, G. Karypis, J.A. Konstan, J.T. Riedl, ” Application of Dimensionality Reduction in Recommender System -- A Case Study” In Proceedings of the ACM WebKDD Workshop, 2000
[15] P. Symeonidis.”Content-based Dimensionality Reduction for Recommender Systems”, In Proceedings of the 31st Annual Conference of the Gesellschaft für Klassifikation, 2007, pp.619-626.
[16] W.A. Tackett, “Genetic Programming for Feature Discovery and Image Discrimination”, In International Conference on Genetic Algorithms, 1993.
[17] R.S. Torres, A.X. Falcão, B. Zhang, W. Fan, E.A. Fox, M.A. Gonçalves and P. Calado, “A new framework to combine descriptors for content-based image retrieval”, In Proceedings of 14th ACM international conference on Information and knowledge management, ACM, 2005, pp. 335-336.

Deepa Anand finished her M.Tech in Computer Science from Jawaharlal Nehru University(JNU) in Delhi in 2009 and her PhD in Computer Science from JNU in 2012. She is currently working as an Assistant Professor in the Department of Computer Science, Christ University. She has published papers in international journals and conferences. Her area of interest is in Machine Learning and Computational Web Intelligence.