Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 5, No. 3, August 2014), Pages: 101-111
www.jacr.iausari.ac.ir

An Optimal Similarity Measure for Collaborative Filtering Using Firefly Algorithm

Fatemeh Shomalnasab[1]✉, Mehdi Sadeghzadeh[2], Mansour Esmaeilpour[3]

(1) Department of Computer Engineering, College of Computer, Hamedan Science and Research Branch, Islamic Azad University, Hamedan, Iran
(2) Department of Computer Engineering, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran
(3) Department of Computer Engineering, Hamedan Branch, Islamic Azad University, Hamedan, Iran

starsh64@yahoo.com; sadegh_1999@yahoo.com; esmaeilpour@iauh.ac.ir

Received: 2013/12/23; Accepted: 2014/02/02

Abstract

Recommender Systems (RS) *provide personalized recommendation according to* the *user need by analyzing behavior of users and gathering their information.* One of the algorithms used in recommender systems is user-based Collaborative Filtering (CF) method. *The idea is that if users have similar preferences in the past, they will probably have similar preferences in the future. The important part of collaborative filtering algorithms is allocated to determine similarity between objects. Similarities between objects are classified to user-based similarity and item-based similarity. The most popular used similarity metrics in recommender systems are Pearson correlation coefficient, Spearman rank correlation, and Cosine similarity measure.*

Until now, little computation has been made for optimal similarity in collaborative filtering by researchers. For this reason, in this research, we propose an optimal similarity measure via a simple linear combination of values and ratio of ratings for user-based collaborative filtering by the use of Firefly algorithm; and we compare our experimental results with Pearson traditional similarity measure and optimal similarity measure based on genetic algorithm. Experimental results on real datasets show that proposed method not only improves recommendation accuracy significantly but also increases quality of prediction and recommendation performance.

Keywords: *Recommender System, Collaborative Filtering, Similarity Measure, Firefly Algorithm*

1. Introduction

In recent years, the Internet plays very important role in daily life. People use the Internet communicating with others, buying and selling products, searching information, and etc. Information management is almost impossible by the use of traditional tools due to the large amount of data and growing the Web documents because these documents are not organized logically. This issue is known as “information overload”. Therefore, some new tools and methods are needed to manage it. In order to customize the Web environment, Personalized Web has become a popular phenomenon to solve the problem of information overload.

The growing trend of information on the Internet has made a lot of difficulties in the process of decision making and selecting proper data and items for many Web users. A recommender system suggests the most appropriate item to the user who needs assistance in searching, sorting and filtering information by analyzing behavior of users and gathering their information. Therefore, it will save time and the required energy for searching large amount of information. Consequently, users can achieve their favorite and interesting items faster.

Since mid of 90s, many researches has been done on recommender systems and consequently significant progress has been achieved in this field. Recommender systems provide personalized recommendations according to user needs by eliminating irrelevant items and proposing the most interesting items according to user preferences. Therefore, RSs save the required time for searching. Some of the used methods in recommender systems are collaborative filtering, content-based filtering, and knowledge-based recommender system [1].

Collaborative filtering algorithms are classified in two types of memory-based algorithm and model-based algorithm. The model-based techniques are based on building a model of data and calculations on it; Models are built by use of data mining and machine learning algorithms. These algorithms can discover appropriate patterns. There are filtering algorithms based on various models such as Bayesian networks, clustering and etc. Memory-based algorithms directly calculate their recommendations based on user–item matrix that stored in memory [1]. To apply the collaborative Filtering methods, no information is required about content of items except ratings of users. Collaborative filtering approach takes a user-item matrix of ratings as an input, and its output is numerical prediction that will express amount of user interest to item or list of top-N [1]. The most popular algorithm used in collaborative filtering is the nearest neighbor algorithm based on the user or the nearest neighbor algorithm based on item. In other word, collaborative filtering method is an automated prediction way about user interests which works by gathering information from large number of users as collaborates. The idea is that if users have similar preferences in the past, they will probably have similar preferences in the future. In this view, each element of user-item matrix represents interest of user on an item. Ratings usually use Likert scale from 1 to 5 or 1 to 10, where 1 is "strongly dislike" and 5 or 10 is "strongly like".

In general, various techniques are used for inference in recommender systems; some methods to predict user needs use user similarity with previous users of system. Pearson correlation coefficient, Spearman rank correlation, and cosine similarity measure are metrics to compute similarities between objects. Pearson correlation coefficient is the best measure in user-based CF [2]. The important part of user-based collaborative filtering algorithms is determining similarity between any pair of users. The main purpose of current paper is to improve performance of collaborative filtering by determining measure that it increases accuracy of recommender system. We choose Firefly algorithm to determine optimal similarity measure. Proposed method causes increasing speed and efficiency of prediction process by use of swarm intelligence algorithm in collaborative filtering approach. This paper is organized as follows: In section 2 we describe user-based collaborative filtering. Section 3 discusses about design of optimal similarity measure. In section 4, firefly algorithm is provided. Section 5 presents the experimental results and section 6 is the conclusion and future work.

2. User-based Collaborative Filtering

The core of this method is to calculate similarity between pair of users. This method gets matrix of users ratings and active user's ID as inputs. CF identifies other users that they have had similar interests in past by user-based nearest-neighbor algorithm [11]. Then, it calculates a numerical prediction according to the nearest neighbor ratings for each item which active user was not seen yet. If rating prediction shows that a user is highly interested in a new item, it will put it in *Top_N* list or it will be recommend to the active user.

Above steps can be formulated formally as follows:

R is $n \times m$ matrix where $u \in 1 \dots n, i \in 1 \dots m$. Rows of the matrix represent set of users $U=\{u_1, \dots, u_n\}$ and the columns are set of items (product) $I=\{i_1, \dots, i_m\}$. Each element of the matrix is user rating on item *i*. If user *u* is not rated item *j*, corresponding element in the rating matrix is zero. Range of rating is defined on discrete numerical scale in set $\{m \dots M\}$ where m=1 indicates lack of interest and M=5 or M=10 indicates high interest of user on item *i*. The most popular similarity metric used in recommender systems is Pearson correlation coefficient which is calculated from the following equation:

$$sim(a,u) = \frac{\sum_{i=1}^{m}(r_{a,i}-\bar{r}_a)(r_{u,i}-\bar{r}_u)}{\sqrt{\sum_{i=1}^{m}(r_{a,i}-\bar{r}_a)^2 \sum_{i=1}^{m}(r_{u,i}-\bar{r}_u)^2}} \qquad (1)$$

where $sim(a,u)$ is similarity between active user a (the user which rating prediction is calculated for him) and other users u, $r_{a,i}$ is the rating of active user a on item i and $r_{u,i}$ the rating of user u on item i, $\bar{r}_a$ and $\bar{r}_u$ are respectively rating averages of active user and user u [2].

We consider ε neighborhood threshold to choose of K nearest neighbor to the active user, after determining the active user similarity with other users. It can be presented by the following formal definition:

$$s(a)=\{ u \mid sim(a,u) \geq \varepsilon, rank(sim(a,u)) \leq K\} \qquad (2)$$

Where $s(a)$ consist of top K users most similar to active user a [5].

3. Similarity Measure

Determining optimal similarity measure is defined as a combinatorial optimization problem where objective function has been considered as mean absolute error (MAE) to obtain prediction accuracy.

3.1 Definitions

In rating matrix R, each row indicates user ratings u on $I=\{i_1, \dots, i_m\}$ items. If a user doesn't rate an item, corresponding element in matrix R remains zero. Suppose that we have two vectors of rating matrix R as follow (ratings vector of user x and user y):

$$\begin{aligned} r_X &= (r_X^1, r_X^2, \dots, r_X^I) \\ r_y &= (r_y^1, r_y^2, \dots, r_y^I) \end{aligned} \qquad (3)$$

Where r_x^1 is the rating of user x on item 1. We consider a vector $V_{x,y}=(v_{x,y}^{(0)}, v_{x,y}^{(1)}, \dots, v_{x,y}^{(M-m)})$ in order to compare these two vectors. Dimension of $V_{x,y}$ is the difference between the maximum rating and minimum rating of Likert scale (m and M). To obtain vector $V_{x,y}$ between both users x and y, firstly we count number of items which both users have rated them (both are not zero) and consider as denominator b. The numerator a indicates number of items which have both users rated them and the rating differential is equal to i, where $i \in \{0 \dots M - m\}$. Therefore, each component of vector is ratio of rating as $V_{x,y}=\frac{a}{b}$. When i=0, it means that both user x and user y have rated same rating on item i, because difference between those two ratings is zero. Furthermore, when i=M-m it means that two users have maximum difference of opinion on item i, that is one of users given maximum rating and reversed another user given minimum rating [3].

3.2 Optimal Similarity Function

An optimal function is provided using weighted vector $V_{x,y}$ to determine similarity between two users where each weight is $w^{(i)} \in [-1,1]$ as follow:

$$sim_w(x,y)=\frac{1}{M-m+1}\sum_{i=0}^{M-m} w^{(i)} \cdot v_{x,y}^{(i)} \tag{4}$$

Where $v_{x,y}^{(i)}$ is ratio of rating on items that difference between those two ratings is i and both user x and user y have rated these items. $w^{(i)}$ represents the importance of the component $v_{x,y}^{(i)}$ for calculating $sim_w(x,y)$. M and m respectively are maximum rating and minimum rating in Likert scale.
The aim is to gain optimal vector of weights by swarm intelligence algorithm. We choose optimal similarity function which provided minimum mean absolute error among all generated similarity functions [3].

4. Firefly Algorithm (FA)

Firefly Algorithm is one of the swarm intelligence algorithms that an evolutionary model based on social behavior and inspired by nature. It is used to solve optimization problems. Fireflies produce short and rhythmic lights that their light patterns are different from each other. Population of algorithm is in fact fireflies, each of which has some lighting or fitness characteristics. In this algorithm, fireflies are compared with each other and the firefly which is less attractive moves toward the more attractive firefly. For simplicity, we can idealize these flashing characteristics as the following three rules:

- All fireflies are unisex so that one firefly is attracted to other fireflies regardless of their sex;
- Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less bright one will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If no one is brighter than a particular firefly, it moves randomly;
- The brightness or light intensity of a firefly is affected or determined by the landscape of the objective function to be optimized [9].

Attractiveness of fireflies is expressed by light intensity which is considered as the objective function. Finally, a firefly, which is selected as the most attractive, is optimal response to the problem.
Attractiveness is a relative parameter and from the view of other fireflies is measured and it depends on distance of fireflies from each other. Attractiveness changes by distance according to the following equation:

$$\beta = \beta_0 . e^{-\gamma . r^2} \tag{5}$$

Where β_0 is the maximum of attractiveness in interval [0,1], γ is absorption coefficient in $[0, \infty)$, and r is the distance between two fireflies. Distance $r_{i,j}$ is the Cartesian distance between any two fireflies i and j at x_i and x_j, respectively. It is obtained from the following equation [8]:

$$r_{i,j} = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k)}^{2}} \quad (6)$$

where $x_{i,k}$ is the *kth* component of spatial coordinate x_i of *ith* firefly. The movement of a firefly i that attracted to another more attractive (brighter) firefly j is calculated as follows:

$$x_i = x_i + \beta_0 . e^{-\gamma . r_{i,j}^2} (x_j - x_i) + \alpha . (rand - 0.5) \quad (7)$$

In above formula, the second term indicates attractiveness and the third term randomized parameter where $\alpha \in [0,1]$, *rand* is a random number in interval [0,1]. Standard Firefly algorithm pseudo code is given below [7]:

Objective function f(x), $x = (x_1, \dots, x_4)^T$
Generate initial population of fireflies x_i (i=1,2,…,n)
Light intensity I_i at x_i is determined by $f(x_i)$
Define light absorption coefficient γ
While (t > MaxGeneration)
  for i=1:n ; all n fireflies
    for j=1:i ; all n fireflies
      if ($I_j > I_i$), move firefly i towards j in d-dimension
      endif
  Attractiveness varies with distance r via exp $[-\gamma . r]$
  Evaluate new solution and update light intensity
    end for j
  end for i
Rank the fireflies and find the current best
end while
Post process results and visualization

4.1 Initial Population

In this paper, we used Firefly algorithm to determine optimal similarity function in collaborative filtering. Number of fireflies is selected from 15 to 40 in most problems. In the current paper, we choose 20 fireflies. As mentioned previously $w^{(i)} \in [-1,1]$ we calculate length of interval: $\frac{Ub-Lb}{M-m+1}$. Optimal values of vector $W=(w^{(0)},w^{(1)},w^{(2)},\dots,w^{(M-1)})$ must be in the following intervals when maximum rating is M=5 [3]:

$$
\begin{aligned}
&w^{(0)} \in [1, 0.6],\\
&w^{(1)} \in [0.6, 0.2],\\
&w^{(2)} \in [0.2, -0.2],\\
&w^{(3)} \in [-0.2, -0.6],\\
&?w\,?^{((}4))?[-0.6,-1].
\end{aligned}
\tag{8}
$$

We have $w^{(i)} . v_{x,y}^{(i)}$ in optimal similarity function. According to above, two users have the same opinion and given same rating to an item *i* when i=0. Unlike it, opinions of two users are opposite to each other when i=M-1. Thus, it is expected that similar opinions give the most positive weight and antithetic opinions give the most negative weight. We consider separate effects for ratings of users with similar opinions and conflicting opinions. In order to generate initial population, half of population randomly generated and the other half of population are randomly generated in the mentioned ranges [3].

4.2 Fitness Function

Several metrics were proposed to evaluate the effectiveness of recommendations in similarity optimization problem for collaborative filtering in recommender systems, such as: mean absolute error, accuracy, precision, and recall. We use mean absolute error as objective function to measure recommendation accuracy which is obtained by difference between predicted rating and real rating [14].
First, we search for *K* most similar neighbors to active user *a* by function defined in Eq. (3). Then, Rating prediction (P_u^i) is calculated for user *u* on item *i* according to the following equation [2]:

$$P_u^i = \bar{r}_u + \frac{\sum_{n\in k_u}[sim_w(u,n) * (r_n^i - \bar{r}_n)]}{\sum_{n\in k_u} sim_w(u,n)} \tag{9}$$

Where $\bar{r}_u$ is average of ratings made by the user *u*. The objective function is calculated from the following equation [3]:

$$fitness = MAE = \frac{1}{N_{User}} \sum_{u\in N_{User}} \frac{\sum_{i\in N_{Item}} |P_u^i - r_u^i|}{N_{Item}} \tag{10}$$

5. Experiments

In this section, we study the evaluation of our proposed method. Our experiment is based on modeling and testing on a real dataset. In this paper, we used MovieLens 1M dataset published on website[1]. This dataset contains 6040 users, 3,706 movies and 1,000,209 ratings. Minimum and maximum values are respectively, 1 and 5. Table 1 shows the above information in brief.

1. http://www.grouplens.org/node/73

Table 1. Information of MovieLens 1M

MovieLens Dataset	
#Users	6,040
#Movies	3,706
#Ratings	1,000,209
Min & Max values	1-5

5.1 MovieLens Dataset

During the past years, MovieLens dataset is as a popular reference in recommender systems researches. We also use this dataset to compare our results with results of previous methods.
This dataset have been collected by the GroupLens Research Project at the University of Minnesota. MovieLens is a web-based system that was for the first time suggested in fall 1997. Every week, thousands of users visit MovieLens site and receive film recommendations from this site. It has over 45,000 users who have expressed their opinions on 6600 films [12].

5.2 Parameters of Problem

In this paper, we consider initial population of fireflies 20, alpha, beta and gamma respectively equal to 0.5, 0.2 and 1 for implementing Firefly algorithm.
Furthermore, average of ratings for rating prediction is obtained from items which both users rated them and also *K* nearest neighbor to the active user should have rated that specific item *i* (where prediction is performing for this item).
We choose similarity threshold to determine *K* nearest neighbors to active user equal to 0.02. If U_i is set of all users which have rated item *i* and *s(a)* contains *K* nearest neighbor if $U_i \cap s(a) \neq \emptyset$, therefore, rating prediction can be calculated. On the other hand, all *K* nearest neighbor should be rated item *i*, if any of *K* nearest-neighbor to active user haven't rated specific item *i*, rating prediction for specific item *i* can't be calculated for the active user *a* [5]. MAE can be calculated if active user actually has rated item *i* and prediction has been calculated for item *i*.

5.3 Experimental Results

The proposed algorithm was implemented by MATLAB a2010 version 7.10.0 on Microsoft Windows 7 and 2.53GHz Intel Core i5 CPU. We implemented recommender system based on Pearson traditional metric and collaborative filtering recommender system based on genetic algorithm in order to evaluate and compare the results of our method.
First, we use only part of whole recommender system (training users and training items) to determine optimal similarity function. We evaluate optimal similarity function on test set. We selected part of MovieLens dataset and divided it into two subsets, 80% of users are considered as training users and 80% of items are considered as training items. Total number of function evaluations is 200 trial runs. Finally, vector *W* is obtained that minimizes objective function. We consider 20% of users which have not been used in Firefly algorithm as test users and 20% of items that have not been used in Firefly algorithm as test items for testing optimal similarity function. Proposed algorithm is run 12 times, in each time run it provides one mean absolute error proportional to *K* values in range {5, ... , 60}.

We use proposed method to determine the similarity between pair of users by choosing active user and calculating ratio of ratings. Then we select *K* nearest neighbor to active user. Finally, mean absolute error on test set is calculated according to Eq. (8). We implemented collaborative filtering method based on genetic algorithm on training set with the same cases [3]. We choose Pearson correlation to evaluate our method with traditional methods, because it has the best performance among other metrics [10].

5.4 Evaluation Metric

MAE is calculated by 12 times running Firefly Algorithm (FA), Genetic Algorithm (GA) and Pearson methods. Number of *K* nearest neighbor is defined in the range {5,…,60}. The different values help us to see trend of changes of MAE chart. The best results of mean absolute error are obtained by use of FA method, namely it results in fewer errors. We must choose median values of *K* in appropriate range {35,...,60} in order to obtain satisfactory results. Neighborhood selection is very important. When *K* is too high, too many neighbors can cause "additional noise" in predictions. In contrast, when *K* is too small, quality of predictions may be negatively affected or no prediction can be calculated. Fig. 2 shows MAE for three methods [1].

Figure 1. FA-method, GA-method and Pearson method comparative results using MovieLens for MAE

Precision and recall are the most popular metrics for evaluation of information retrieval systems. They have been used for recommender systems by many researchers. Precision is a measure of exactness and recall is a measure of completeness [6]. There are different methods to calculate precision and recall [13]. In this paper precision obtained based to the following equation:

$$Precision = \frac{1}{|U|} \sum_{u \in U} \frac{|Test \cap Top\_N|}{Top\_N} \tag{11}$$

And recall equation is:

$$Recall = \frac{1}{|U|} \sum_{u \in U} \frac{|Test \cap Top\_N|}{Number\ of\ items\ in\ Test\ set} \tag{12}$$

Where *U* is the number of users, *Test* is the real rating of test set and *Top_N* indicates number of top recommendations. We use precision and recall to evaluate quality of recommendations. Fig. 3 shows precision measure in number of recommendations in range {20,...,60}.

Figure 2. FA-method, GA-method and Pearson method comparative results using MovieLens for Precision

Recall measure in number of recommendations in range {20,...,60} is shown in Fig. 4.

Figure 3. FA-method, GA-method and Pearson method comparative results using MovieLens for Recall

The above figures show that by increasing number of recommendations, precision decreases and recall increases. This quality metrics improve the results for each value of *N*. Therefore, FA method not only improves recommendation accuracy but also provides better recommendations.

5.5 Performance Results

Table 2 shows the result of average of execution time for three methods.

Table 2. Execution time

Metric	Pearson correlation coefficient	GA method	FA method
Time (s)	14.62	5.22	5.14

As shown in Table 2, average of required time to provide predictions by using FA is 9.33s less than Pearson and 0.13s less than GA. Simple equation is necessary to calculate similarity function (used to determine *K* nearest neighbor) because searching the *K* most similar neighbors to active user requires the maximum time to find recommended items. The equation simplicity will improve recommendation process performance.

Similarity function equation based on Firefly algorithm ((linear combination of weights and ratings) see equation (3)) is simpler than the equation used in traditional metrics (see equation (1)) [4]; therefore, our proposed method provides recommendations faster than traditional metrics. Bobadilla et al. (2011) proposed an optimal similarity function using binary genetic algorithm. In their genetic algorithm, each vector of weights, represents a possible individual of the population. Each component $w^{(i)}$ in the vector W is represented by 10 bits. Consequently, the vector W is represented by string of 0s and 1s (with a length of $2^{10(M-m+1)}$ bits). To solve this problem, each binary chromosome must be converted to a continuous values. As usual, GA uses the common operators, selection, crossover and mutation. Therefore, continuous values must be converted to binary values again. Whereas in the Firefly algorithm, each vector of weights, W, represents a firefly which its components are real number in the range of [-1,1]. Thus, there is no need to decode binary values to continuous values and vice versa. Consequently, the rate of achieving optimal solution and convergence in firefly algorithm will increase significantly. Training step in our proposed method is faster than GA method. This issue has an important effect on the rate of generating optimal solution.

According to table 2, FA method runs 35% faster than Pearson method which shows a promising advantage in a recommender system.

6. Conclusion

Nowadays, we require systems which they have ability to direct users towards goods and services. A recommender system is an intelligent system that recognizes interests and preferences of users in the Internet environment, filters existing information and provides relevant recommendations to users.

The "similarity between pair of users" is the main feature of user-based collaborative filtering algorithms used in recommender systems. We have provided optimal similarity function in order to increase speed of finding nearest neighbors of active user and reduce its computation time. In this paper, we have studied Firefly algorithm to determine optimal similarity function. FA method provides better quality and faster results than traditional method and GA method. Proposal use of FA is a new approach in recommender system.

The proposed method will be tested on datasets with different features in future. In addition, we will work on most of offline calculation methods in background.

References

[1] D. Jannach, M. Zanker, A. Felfernig and G. Friedrich, "Recommender Systems An Introduction," Cambridge University Press, Printed in the United States of America, 2011.

[2] F. Ricci, L. Rokach, B. Shapira & P.B. Kantor, "Recommender Systems Handbook," Springer, New York Dordrecht Heidelberg London, 2011.

[3] J. Bobadilla, F. Ortega, A. Hernando & J. Alcal, "Improving Collaborative Filtering Recommender System Results and Performance Using Genetic Algorithms", *Journal of Knowledge-Based Systems*, Vol.24, No. 8, pp. 1310–1316, 2011.

[4] J. Bobadilla, F. Serradilla, J. Bernal, "A New Collaborative Filtering Metric that Improves the Behavior of Recommender Systems," *Journal of Knowledge-Based Systems*, Vol. 23, No. 6, pp. 520-528, 2010.

[5] A. Yamashita, H. Kawamura & K. Suzuki, "Similarity Computation Method for Collaborative Filtering Based on Optimization," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol.14, No.6, pp. 654-660, 2010.

[6] P. Bedi, and R. Sharma, "Trust Based Recommender System Using Ant Colony for Trust Computation," *Journal of Expert Systems with Applications*, Vol. 39, No. 1, pp. 1183–1190, 2012.

[7] X. S Yang, "Firefly Algorithms for Multimodal Optimization," *Proc. On Stochastic Algorithms: Foundations and Applications*, Vol. 5792, pp. 169-178, 2009.

[8] A. Yousif, A.H Abdullah, S.M Nor, A.A Abdelaziz, "Scheduling Jobs on Grid Computing Using Firefly Algorithm," *Journal of Theoretical and Applied Information Technology*, Vol. 33, No. 2, pp. 155-164, 2011.

[9] X. S Yang, "Firefly Algorithm, Stochastic Test Functions and Design Optimization," *Journal of Bio-Inspired Computation*, Vol. 2, No. 2, pp.78-84, 2010.

[10] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *Journal of ACM Transactions on Information Systems (TOIS)*, Vol. 22, No. 1, pp. 5–53, 2004.

[11] J. Herlocker, J.A. Konstan, J. Riedl, "An Empirical Analysis of Design Choices in Neighborhood-based Collaborative Filtering Algorithms," *Journal of Information Retrieval*, Vol. 5, No. 4, pp. 287 – 310, 2002.

[12] S. Gong, H. Ye, H. Tan, "Combining Memory-based and Model-based Collaborative Filtering in Recommender System," *Proc. On Circuits, Communications and Systems*, pp. 690 – 693, 2009.

[13] [13] M. Garden, "On The Use Of Semantic Feedback In Recommender Systems," M.S. thesis, School of Computer Science, McGill University, Montréal, 2004.

[14] S. Bhaidani, "Recommender System Algorithms," B.Sc thesis, Department of Mechanical and Industrial Engineering, University of Toronto, 2008.