A Comparative Study for Optimizing Retail Inventory Market Using Termite Colony Optimization

Radwa Ali, Neveen I. Ghali, Taghrid A. Imam

Al-Azhar University, Faculty of Science, Cairo, Egypt
radwa_ali1@yahoo.com, nev_ ghali@yahoo.com

Abstract

Recommender system has been changed from novelties used by a few E-commerce sites, to serious business tools that are re-shaping the world of E-commerce. Many of the largest commerce web sites are already using recommender systems to help their customers find products to purchase. A recommender system learns from customers and recommends products among the available products that they may be interested. An important contribution of this paper is a comparison between TCO (Termite Colony Optimization) approaches to recommender systems, these approaches provide decision making models which are used by termites to adjust their movement trajectories. This study has been tested by four datasets which are Indian Retail Market, MovieLens data, Chinese Grocery market, and Chinese Retail market datasets.

Keywords: *Termite colony optimization, Retail, Inventory, Recommender system*.

1. Introduction

Decisions have to be daily made in many activities such as shopping for clothes, download music clips in digital libraries, latest news on websites, etc.. But the information overload makes the decision making hard and time consuming. Therefore the recommender system has been introduced [3, 4], which is a new type of internet based software tools, designed to help users find their way through today's complex e-applications.In particular, e-commerce recommendation is a mature research field with theoretical support and practical experience, used for helping businesses and consumers buy and sell products and services through an electronic medium without using any paper documents. Several different aspects of the behavior of termite colonies have inspired different kinds of termite algorithms. For example foraging, brood storing, and division of labor, and cooperative transport. In all these examples, termites coordinate their activities via stigmergy, a form of indirect communication mediated by modifications of the environment. One of the most examples of successful termite algorithms is known as termite colony optimization (TCO).Many Artificial Intelligence (AI) techniques have been used in construction of the recommender system. One of the most studied and the most successful technique is TCO [2].

Hence a comparative study for two approaches using TCO to optimize the inventory of retail will be introduced.

The remainder of this paper is ordered as follows. A briefed overview on recommender systems and TCOare given in section 2. The two approaches are defined in section 3. Section 4 shows the experimental results and analysis for the two proposed approaches. Conclusion and future work are discussed in Section 5.

2. Background

2.1 Recommender Systems

Recommender systems were originally defined in 1997 by Resnick and Varian [4] as ones in which "people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients". But this definition places the emphasis on the recommender systems as supporting the collaboration between users. Later, a general definition is taking place, referring to recommender systems as those systems that "have the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options" [3]. The two basic entities which appear in any recommender system are the user and the item. A user is a person who utilizes the recommender system providing his opinion about various items and receives recommendations about new items from the system. Therefore, a recommender system can be viewed as a mapping of users and items to a set of utility values. The view of recommendation as a prediction task comes from the fact that this mapping is not, in general, defined on the whole domain of user-item pairs, and thus requires the system to estimate the interest values for some elements of the domain. More formally, let us assume the existence of a set of users, $U = \{u1, u2, ..., um\}$, and a set of items, $I = i_1, i_2, ..., i_n$. The profile for a user $u \epsilon U$ can be viewed as an n-dimensional vector of ordered pairs, u(n) = h(i_1, $s_u(i_1)$), (i_2, $s_u(i_2)$), ..., (i_n, $s_u(i_n)$), where $i_j's \epsilon I$ and s_u is a partial function for user u, assigning utility values to items in I. Thus, the function $s_u : I \rightarrow R$ represents the profile of a user u, mapping items to an ordered set of utility values, R [5, 6, 7]. Recommender systems can be classified into many categories depending on the information they use to recommend items. Three major recommendation techniques have been studied: Collaborative filtering recommendation, Content based recommendation, and Demographic recommendation. The initial conception of collaborative filtering systems refer to methods suggesting new items for a particular user based on users past liking and the opinions of other like-minded users. For example, suppose that there is a target user who likes items A and B. If there are many other users who like A, B and C, then C item will probably be recommended to the target user. Content-based systems recommend items based on content of items rather than other user's ratings of the system. Instead of deriving a user-to-item correlation, these systems use item-to-item correlation for generating recommendations. These systems gather content data about the items such as, title, author, etc. for the books or the director, genre, etc. for the movies. Gathering data can be achieved using various approaches, then,the user is asked to provide some ratings for the items randomly. Finally, the systems match un-rated items contents with the compiled user profile and assigns scores to the items depending on the match between user profile and item descriptions. The items are ranked according to their scores and presented to the user in order as output. Demographic recommender systems aim to categorize the user based on personal attributes and make recommendations based on demographic classes. Demographic systems depend on the assumption that all users belonging to a certain demographic group have similar taste or preference. Demographic techniques form "user-to-user" correlations [11, 12].

2.2 Termite Colony Optimization

Termite colony optimization TCO was introduced by Paul Grasse in 1959 [1] which has been inspired from the foraging behaviour of real termites. When searching for food, termites initially explore the area surrounding their nest in a random manner. As soon as a termite finds a food source, it evaluates the quantity and the quality of the food and carries some of it back to the nest. During the return trip, the termite deposits a pheromone trail on the ground. The pheromone is a chemical excreted by the insect which evaporates and disperses over time. The quantity of pheromone deposited, which may depend on the quantity and quality of the food, will guide other termites to the food source. Indirect communication between the termites via pheromone trails enables them to find shortest paths between their nest and food sources. If no pheromone exists, a termite moves randomly in any direction. A colony of termites has the ability to perform complex task by applying simple rules between its individuals. For example, consider a flat surface upon which termites and pebbles are distributed. The hill building example is a simple example which explains the behaviour of the termites. The termites want to build a hill from the pebbles. The termites try to collect all of the pebbles into one place. Termites act independently of all other termites, and move only on the basis of an observed local pheromone gradient. If a termite is not carrying a pebble and it encounters one, the termite will pick it up. If a termite is carrying a pebble and it encounters one, the termite will put the pebble down. The pebble will be infused with a certain amount of pheromone. With these rules, a group of termites can collect dispersed pebbles into one place [2].

The TCO algorithm is represented in algorithm (1). TCO employs a population of termites which move in a *D*-dimensional search space $S \subset R^D$ to find the optimal solution. Assume that we have a population of *N* termites. Each termite *i* in the population is represented with $Y_i = (Y_{i1}, Y_{i2}, ..., Y_{iD})$ which represents a feasible solution for an optimal problem in the *D*-dimensional search space S. In TCO, each position Y_i represents a hill with an associated quality which is represented as $fit(Y_i)$. The fitness value models amount of pheromones which are deposited on the hill. At each iteration of the TCO, the fitness value of each termite is formulated. The fitness value is used for computation of pheromone content at each position of the termites. The pheromone content of the termites at any position is computed based on the following equation

$$\psi_i(t) = 1 - \rho\psi_i(t-1) + \frac{1}{fit(Y_i)+1} \qquad (1)$$

Where, ρ the evaporation rate that belongs to [0,1], $\psi_i(t-1)$ and $\psi_i(t)$ respectively are the pheromone density level at the previous and current locations of i^{th} termite.The density level of pheromone provides the information to termites to adjust its movement towards one of these pheromone gradients. That is the termites evaluates the pheromone density level, and adjusts its movement towards a position with the highest level of pheromone density. A termite *i* considers the local best position, denoted as b_i, as its most suitable position and move towards that if its current position has smaller level of pheromone density compared to the best local position. The movement of the termite *i* is controlled using the following equation

$$Y_i(t) = Y_i(t-1) + \eta_b q_b (b_i - Y_i(t-1)) \qquad (2)$$

if $\psi_i(t-1)<\psi_{bi}(t-1)$, where $\$1 <\eta_b< 2$ and $0 <q_b< 1$ probabilistic values that controls the movement of the termite towards local position in the search space.

Algorithm 1 : TCO algorithm

Initialize: η_b= 1.5 and q_b = 0.055
Select a random item T_i where $i \leq n$
while $i \leq n$ **do**
  if $n \geq 2$ **then**
    calculate fitness function $FtnsM_i$
    calculate $\psi_i(t)$ using equation (1)
  end if
  if termite *i* has neighbors **then**
    if $\psi_i(t-1)<\psi_{bi}(t-1)$**then**
      calculate $Y_i(t)$ using equation (2)
    end if
  end if
  adjust step size s
end while

3. Problem Definition

Hence two approaches to optimize the inventory of retail to increase the frequency of selling a certain item using TCO will be defined. These approaches aim to recommend the item seems to be like by the user by maximizing fitness function. The first approach was proposed by *S*. Banerjee and et al. in [8]. The fitness function proposed in it was

$$\text{Ftns}_1 M_i = \frac{\log(\text{Mi}+1)}{n} \quad (3)$$

The second approach was proposed by R. Ali and et al. in [9]. The fitness function proposed in it was

$$\text{Ftns}_2 M_i = \frac{\sqrt{(\text{Mi}+1)}}{n} \quad (4)$$

for $1\leq i \leq n$ where, M_i= Repeat Purchase of the item of same or different brand, n =Total no. of product and $FtnsM_i$ is the fitness function at item i

The idea was to investigate a product consumption pattern of retail and thereby could devise a more efficient business logic of retail inventory system. This process of optimization has been achieved by using TCO.

Proposed algorithms solicited the basic design of retail market and market flow of purchase behavior has been monitored by termite's colonial behavior. Where $\eta_b \ and q_b$ are probabilistic values that controls the movement of the termites towards local best position in a Search Space. Optimized Value of the self organization part that is controlled automatically at t^{th}instance $Y_i(t) = S_i$. Define b_i to be locally available best position in a search space. The pheromone density level at the current Position of i^{th} termite is defined to be *i(t)* as in algorithm (2) where the fitness function is calculated using equation (3) and equation (4) in the first and the second approaches respectively.

Algorithm 2 : TCO algorithm

```
Initialize: η_b = 1.5 and q_b = 0.055
Select a random item T_i where i ≤ n
while i ≤ n do
    if n ≥ 2 then
        calculate fitness function FtnsM_i
    end if
    b_i ← FtnsM_i
    if termite i has neighbors then
        if ψ_i(t − 1) < ψ_bi(t − 1) then
            calculate Y_i(t) using equation (2)
        end if
    end if
    S_i ← Y_i(t)
end while
```

4. Experimental Results

In presented experiments data related to Indian retail market named as "Big Bazar (Large Market)" is used. The possible parameters in a retail market that are principally considered are Total number of product, Repeat purchase of the item of same or different brand, and the fitness function to be maximized. The proposed function also used the publicly available MovieLens 100K dataset from the MovieLens recommendation system [10]. Each user in this dataset has rated at least 20 movies. The dataset from this system contains over 100,000 ratings which were given by 943 users on 1682 movies using a 1(bad)-5(excellent) numerical scale and then it was converted into a user-movie matrix R that had 943 rows and 1682 columns. The movie is used as an item and the rates of users for movies as rates of users for items. Also the proposed function has been tested by a dataset contains information about purchases of a panel of thousands of consumers in both offline and online stores of a large grocery retailer, a supermarket chain in Spain. A set of consumers is observed in two different periods of time: 2003 (1 year) and 2007 (6 months). There is info about a broad range of categories (406), including fresh food and durables. Finally the proposed function used a Chinese transaction data which has been collected in February, 2001 from a supermarket.

4.1 Experimental results for datasets before applying TCO algorithm

The relation curve between the repeat purchase and frequency of selling is shown in Figure 1, Figure 2, Figure 3, and Figure 4 for Indian retail Market, MovieLens data, Chinese grocery market, and Chinese retail market data respectively, it has been observed that the frequency of selling a certain item has been increased the proposed objective function compared with the old function.

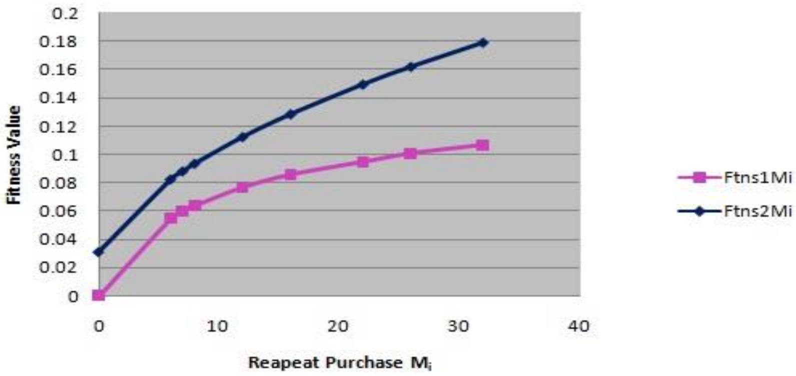

Figure 1. Rate of selling of a particular product for retail Indian market dataset before applying TCO algorithm

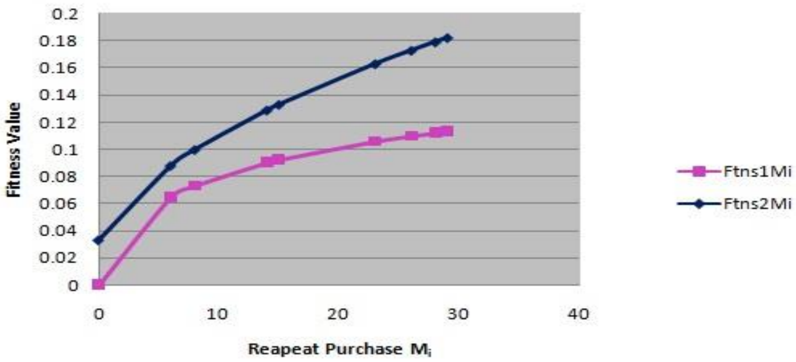

Figure 2. Rate of selling of a particular product for MovieLens dataset before applying TCO algorithm

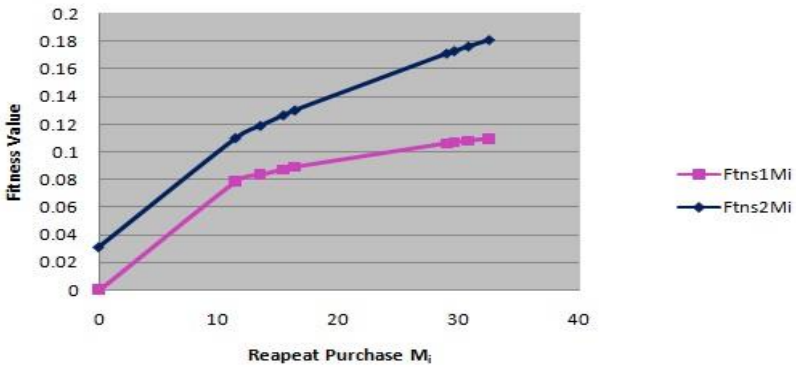

Figure 3. Rate of selling of a particular product for grocery Chinese market dataset before applying TCO algorithm

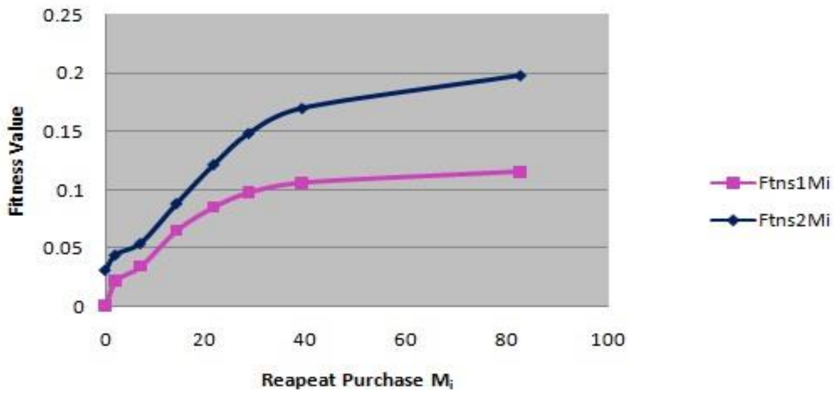

Figure 4. Rate of selling of a particular product for retail Chinese market dataset

4.2 Experimental results for datasets after applying TCO algorithm

Figures 5 to Figure 8 show the relation between the repeat purchase of items and its fitness function before and after applying TCO algorithm using retail market data set, MovieLens dataset, Grocery Chinese market dataset, and Retail Chinese market dataset respectively. It has been observed that the fitness function increases with the repeating purchase of each item.

Here, we have related the components of TCO in terms of retail market components to obtain equation (3) and equation (4) where $\psi_i(t)$ = T_{i+1}, $\psi_i(t-1)$ = T_i, ρ = 0.055,$fit(Y_i)$ = M_i, $Y_i(t\text{-}1)$ is the value of the self organization part that is controlled automatically at $(t\text{-}1)^{th}$ instant, η_b =1.5, $q_b = 0.055$, b_i=$FtnsM_i$, and $Y_i(t)$ = S_i is the optimized value of the self organization part that is controlled automatically at t^{th} instance.

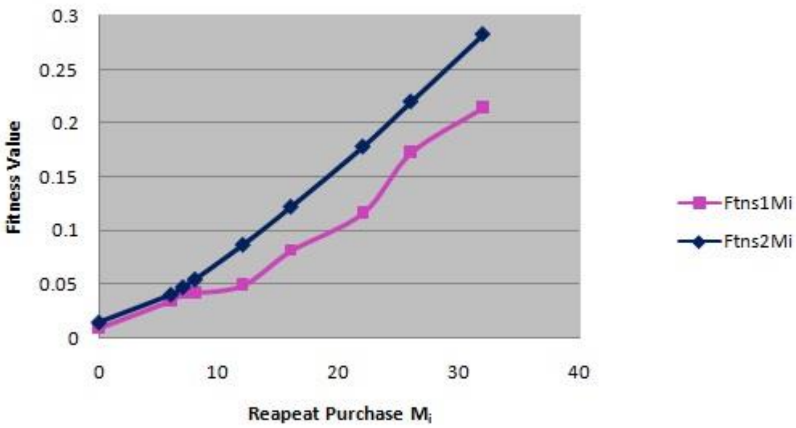

Figure 5. Rate of selling of a particular product for retail Indian market dataset after applying TCO algorithm

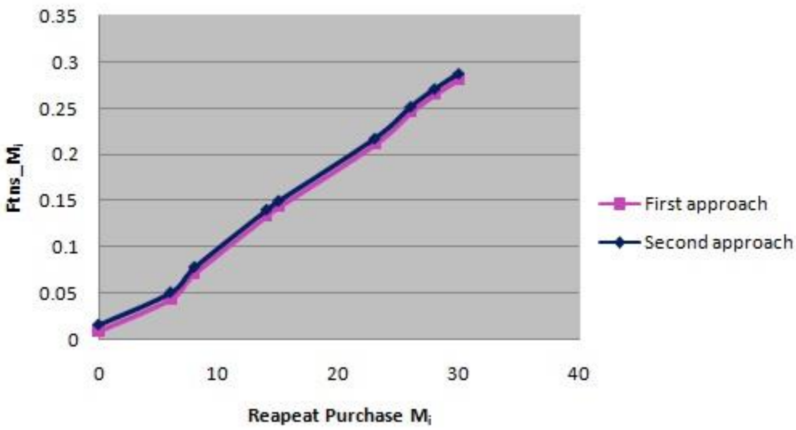

Figure 6. Rate of selling of a particular product for MovieLens dataset after applying TCO algorithm

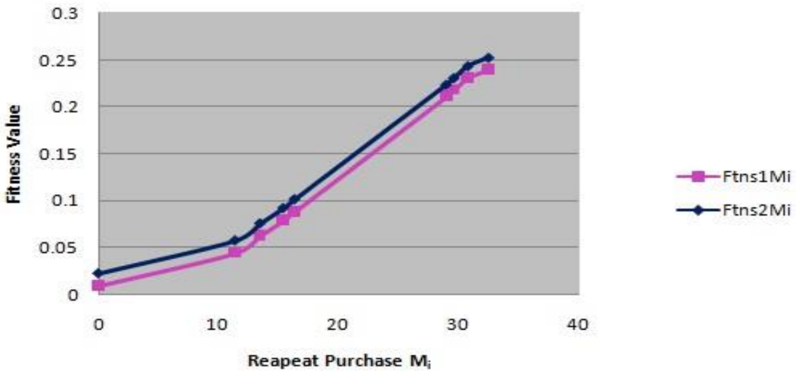

Figure 7. Rate of selling of a particular product for grocery Chinese market dataset after applying TCO algorithm

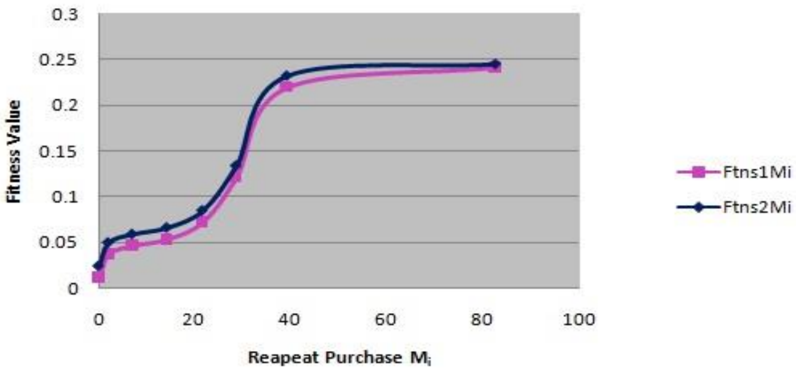

Figure 8. Rate of selling of a particular product for retail Chinese market dataset after applying TCO algorithm

Experimental resultsshow that the last item has been purchased for maximum number of times (32 times for Retail Market data and 30 times for MovieLens data and 32 for Grocery Chinese market data and 82 for Retail Chinese market data). Hence, the maximum rate of selling is 0.2146, 0.2816, 0.2397, and 0.242 for Retail Market, MovieLens data, Grocery Chinese market, and Retail Chinese market data respectively by using the fitness function in [8] and 0.2833, 0.2874, 0.2529, and 0.2451 for Retail Market, MovieLens data, Grocery Chinese market, and Retail Chinese market data respectively after using the proposed fitness function in [9]. Meanwhile, it is observed that whenever items are repeatedly purchased from market its total rate of selling also increased and that total rate has been optimized using the second approach.

5. Conclusion and Future work

This paper presents a comparative study for to approaches using TCO to optimize the inventory of retail. From experimental results it can concluded that the maximal fitness curves of the fitness function using the two approaches show that the fitness function value has been improved with the repeating purchase of each item by the second approach using retail market data, MovieLens data, Grocery Chinese market data, and Retail Chinese market data.

As a future direction, the collective behaviour of termites could be fine tuned with their parameters to avoid local minima.

References

[1] R. Hedayatzadeh, F. Salmassi, M. Keshtgari, R. Akbari and K. Ziarati, “Termite Colony Optimization: A Novel Approach for Optimizing Continuous Problems”, 18th Iranian Conference on Electrical Engineering (ICEE), pp.553-558, 2010.

[2] P.~Grassé, “Tentative interpretation of the behavior of termites manufacturers” nsectes Sociaux , Springer-Verlag, Vol. 6, No.1, pp.~41-81, 1959.

[3] R. Burkeh, “Hybrid Recommender Systems: Survey and Experiments”, User Modeling and User-Adapted Interaction Journal, Vol. 12, No. 4, pp. 331–370, 2002.

[4] P. Resnick, H. Varian , “Recommender systems”, Scientific Computing and Validated Numerics, Vol. 40, No. 3, pp. 56–58, 1997.

[5] E. Vozalis, K. G. Margaritis, “Analysis of recommender systems' algorithms”. In Proceedings of the sixth Hellenic European conference on computermathematics and its applications (HERCMA), pp. 1- 14, 2003.

[6] B. Mobasher,R. Burke, R. Bhaumik and C. Williams, “Towards Trustworthy Recommender Systems: An Analysis of Attack Models and Algorithm Robustness”, ACM Transactions on Internet Technology, Vol. 7, No. 2, pp. 1–41, 2007

[7] G. Adomavicius, A. Tuzhilin, “Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions”, Knowledge and Data Engineering, IEEE Transactions, Vol. 17, No. 6, pp. 734 - 749, 2005

[8] S. Banerjee, N. Ghali, A. Roy and A. Hassanein, “Investigating Optimization in Retail Inventory: A Bio-inspired Perspective towards Retail Recommender System”, 12th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 161 - 165, 2012.

[9] R. Ali, S. Banerjee, , N. Ghali and A. Hassanein, “Towards Retail Market Recommendations using Termite Colony Optimization”, Proceedings of the IEEE 22nd International Conference on Computer Theory and Application (ICCTA), pp. 13-15, 2012.

[10] MovieLens dataset, Group lens research, http://grouplens.org/node73

[11] K. Rao, and V. Talwar, “Application Domain and Functional Classification of Recommender Systems—A Survey”, Journal of Library and Information Technology, Vol. 28, No. 3, pp. 17--35, 2008

[12] R.~Burkeh, “Hybrid Recommender Systems: Survey and Experiments”, User Modeling and User-Adapted Interaction Journal, Vol. 12, No. 4, pp.331--370, 2002.