Chiang Mai J. Sci. 2014; 41(5.2) : 1429-1438
http://epg.science.cmu.ac.th/ejournal/
Contributed Paper

An Agent Model for Information Filtering using Revolutionary RSVD Technique

Dussadee Praserttitipong [a] * and Peraphon Sophatsathit [b]

[a] Department Computer Science, Faculty of Science, Chiang Mai University, Chiang Mai, 50202, Thailand.

[b] Advanced Virtual and Intelligent Computing (AVIC) Center, Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok, 10330, Thailand.

*Author for correspondence; e-mail: dussadee.p@cmu.ac.th

Received: 20 March 2012
Accepted: 11 July 2013

ABSTRACT

This paper proposes a collaborative software agent model. The agent works in a distributed environment making recommendation based on its up-to-date knowledge. This knowledge is partly acquired from other collaborative agents to combine with its own prior knowledge by means of a revolutionary regularized singular value decomposition (rRSVD) technique. The technique is used as an adaptation process for the agent to learn and update the knowledge periodically. This process employs one of the three agent adaptation models, namely, 2-phase, 1-phase, or non-adaptation that is suitable for the operating bandwidth, along with a fast incremental knowledge adaptation algorithm. As a consequence, the adapted agent will be able to work alone in a distributed environment at a satisfactory level of performance.

Keywords: recommender systems, collaborative agent, adaptation process, distributed environment

1. INTRODUCTION

Intelligent agents have emerged in the last decade as an alternative to assist the users in a distributed way for helping their users find relevant items such as service or goods based on their preference or opinion [1], Commonly, their tasks can be reduced to the job of estimating rating scores of the items that have not been seen by their users. The results of these systems can be both the predicted rating scores for some particular items or a set of *top-n* recommended items.

There are several intelligent agent systems have been presented in literature [2, 3, 4, 5, 6, 7]. Most of them advise their users with the pre-learning inferential model initiated from the characteristics of both user's profile and item's profile. However, the item's profile of some problem domains (for example, graphical images and audio streams) is difficult to analyze and perform automatic features extraction [4]. This must be supplemented by a manual profile specifying process that calls for additional workload on the user's part, thereby hinders the advantages of the agent

systems.

To model this kind of agent system, the collaborative filtering (CF) technique [3, 4, 5, 6, 7], which is a successful type in conventional recommender system techniques, is examined in this paper. Instead of creating an inferential model, the predicted rating scores are made according to the calculated similarity of historical data among collaborative agents. The agent is suggested with the predicted rating scores evaluated from other agents who have similar taste.

The CF supporting techniques which are rooted on singular value decomposition (SVD) technique [10, 11] are studied. The main idea of SVD-based technique is to reduce insignificant users or items by capturing latent relationships between users and items, thereby the noise in the resulting user-item matrix is decreased which leads to higher prediction accuracy. However, the conventional CF technique generally performs the evaluation on the central repository where user's opinions are stored. This centralized evaluation can induce a bottleneck to the system [2].

A simple solution is to distribute the workload of this SVD technique to the agents, who are the entire client's of the system. Nevertheless, the distribution is not suitable for the client platform with less availability resources, such as mobile device platforms. In addition, the fact that knowledge sharing sessions among all collaborative agents must be held at all time to enable knowledge sharing process makes it impractical to maintain full connection across all collaborative agents. Some meaningful information might be lost and the agents have to determine the rating score assessment based on incomplete knowledge information. Thus, we set out to investigate the conventional SVD-based technique to make it practical for embedding into the agent model working in a decentralized environment.

This paper proposes a collaborative agent model. The main know-how from other agents can be incorporated as prior knowledge of the agent. An incremental update algorithm for this prior knowledge is also presented to enabling an adaptability to this proposed agent model. The remaining of this paper is organized as follows. Section 2 explains the methods and data sets used in the experiment. Section 3 discusses the experimental results. Section 4 summarizes the proposed approach and some final thoughts.

2. PROPOSED AGENT MODEL

In order to establish the necessary principles and methods for the proposed approach, some preliminaries are elucidated to set up the basic theoretical building blocks.

2.1 Agent Modeling Methods

A few referenced vocabularies to be used in this literature are identified as follows:

Let r be an $m \times n$ matrix, where $m \geq 1$ be the number of agents and $n \geq 1$ be the number of items. The element $r_{a,i}$ is a rating score of a user a *over an item i. The value of $r_{a,i}$ is a discrete value, where user has rated on item. Otherwise, the value of $r_{a,i}$ is null.*

The collaboration is established based on evaluation of historical rating scores. The historical know-how from other agents is learned at the central repository site and stored as prior knowledge in the information server. The newly acquired knowledge, represented by a user-item rating matrix, can be decomposed in two aspects, namely, the knowledge with respect to the user's point of view ($K^{\{A\}}$) and the knowledge gathered from other users' point of view ($K^{\{I\}}$). Finally, all constructive knowledge is

encapsulated and ported to the client agent as prior knowledge.

The proposed model encompasses three processing steps as follows: (1) Evaluation process, (2) Training process, and (3) Incremental adaptation process.

2.1.1 Evaluation process

When an agent faces with a decision situation making on a particular item (such as service or goods), the rating scores are required. The agent's knowledge, combined with additional details acquired from other collaborative agents, is utilized to estimate the results. An estimation equation of rating score prediction is formulated as

$$\hat{r} = \text{round}(\text{average}(K^{\{A\}}, K^{\{I\}})) \quad (1)$$

where $\hat{r}$ represents a predicted rating score.

Definition 1:
Let K be the knowledge with respect to a particular point of view, the value of this knowledge can be determined as

$$K = \bar{r} + \Delta \quad (2)$$

where $\bar{r}$ is an average rating score and Δ is the prior knowledge of the concerning point of view.

Thus, the variables in Equation (1) can be expressed to accommodate the above user's point of view as

$$K^{\{A\}} = \bar{r}^{\{A\}} + \Delta^{\{A\}} \quad (3)$$
$$K^{\{I\}} = \bar{r}^{\{I\}} + \Delta^{\{I\}} \quad (4)$$

where matrix $\bar{r}^{\{A\}}$ collects average rating scores of all explicit rating values working out by the agents and matrix $\bar{r}^{\{I\}}$ gathers average rating score of all explicit rating values on the item. These two variables denote adaptable knowledge part of the agent. This knowledge information will be regularly updated according to the changing environments, i.e., new rating scores, new users, and new items. The $\Delta^{\{A\}}$ and $\Delta^{\{I\}}$ denote prior knowledge with respect to the user's point of view and prior knowledge gathering from other users' point of view, respectively.

Rather than carry out the decision making at the central repository site, the evaluation process takes place at the agent site in the distributed environment. Hence, full connection across all collaborative agents is no longer needed. The degree of dependence on others is unfastened because the know-how of other collaborative agents is already encapsulated in its prior knowledge. Thereby the agent can provide the prediction autonomously. The processing time of this prediction process is $O(1)$. This makes the proposed approach suitable for the agent to operate in real time environment.

2.1.2 Training process

This training process is executed at the central repository site, in which the relationships among the agents are created within a finite set of trusted registered agents. The collections of other agents' attitudes are assembled via an off-line process. Thus, the knowledge sharing session among other collaborations can be assured.

The main point for modeling this knowledge according to this training process is presented based on the RSVD technique which was informally introduced by [11, 12]. However, the shortcomings of this primitive RSVD technique make them inadequate to operating in real time under a distributed environment. The revolutionary RSVD function is thus proposed to support the training process that allows an agent to update its knowledge autonomously in a distributed environment. The overall training algorithm for modeling the agent's knowledge comprises

$K^{\{A\}}$, $K^{\{I\}}$ as described below

1. Calculate the average rating score for all explicit rated elements working out by each agent and collect them in matrix $\bar{r}^{\{A\}}$. The value of the average rating score made by agent a, denoted $\bar{r}_a^{\{A\}}$ or $\bar{r}_a$ for short, is determined from

$$\bar{r}_a = \frac{\sum_{j'=1}^{n'} r_{j'}}{n'} \quad (5)$$

where set R' collects all rating scores made by agent a according to Definition 2 (defined in Definition 2 of the incremental adaptation process). In other words, R' is the set of all elements in row of the user-item rating matrix $r_{a,i}$, where $r_{a,i}$ is not null and is less than or equal to the number of items n.

2. Calculate the average rating score of all explicit rating values made on each item to form matrix $\bar{r}^{\{I\}}$. The value of average rating score of an item i, denoted $\bar{r}_i^{\{I\}}$ or $\bar{r}_i$ for short, is determined from

$$\bar{r}_i = \frac{\sum_{j'=1}^{n'} r_{j'}}{n'} \quad (6)$$

where the set R' collects all rating scores made on item i according to Definition 2. In other words, R' is the set of all elements in column i of the user-item rating matrix $r_{a,i}$ which is not null and $n' <= m$ (n' is less than or equal to the number of users m).

3. Execute the revolutionary RSVD function to obtain matrix $\Delta^{\{A\}}$, where $\Delta^{\{A\}}$ is an $m \times n$ matrix defined earlier.

4. Execute the revolutionary RSVD function to obtain matrix $\Delta^{\{I\}}$, where $\Delta^{\{I\}}$ is an $m \times n$ matrix defined earlier.

5. The resulting knowledge from this process comprises $\{\bar{r}^{\{A\}}, \bar{r}^{\{I\}}, \Delta^{\{A\}}, \Delta^{\{I\}}\}$, such that
5.1 An average rating for item i, ($\bar{r}_i$), is sent back to the information server where the concerning item is posted.
5.2 The remaining information $\{\bar{r}_a, D^{\{A\}}_{a,*}, D^{\{I\}}_{*,i}\}$ is sent in response to the request of agent a for total renewing of prior knowledge.

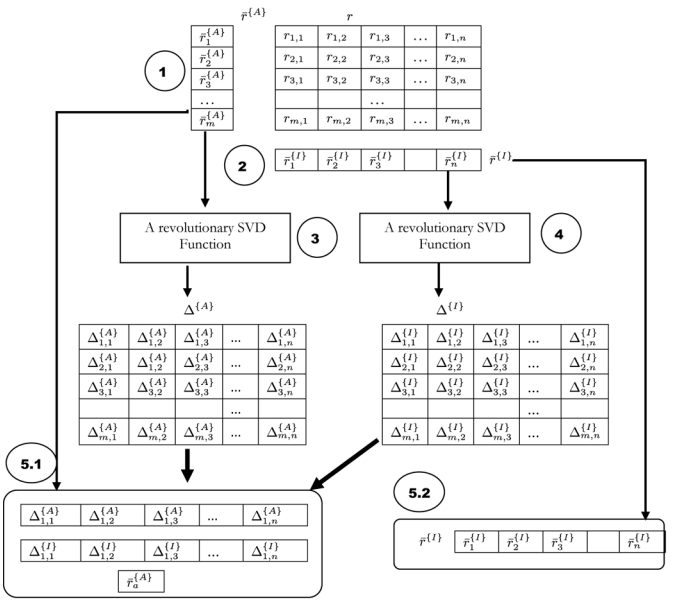

Figure 1. Evolutionary of knowledge composition according to the learning process.

The knowledge evolution obtained from each step in this algorithm is shown in Figure 1.

According to RSVD technique, the baseline estimation equation for predicting the unknown $r_{a,i}$ is given by

$$\hat{r} = \mu + bi + ba + q^T p \quad (7)$$

where μ is the average rating score of all rated elements in user-item rating matrix, p and q are primitive RSVD parameters [13]. Besides, the observed bias (deviations) of both user a and item i, denoted by ba and bi, respectively, are added to enhance the performance of the primitive RSVD.

This research adopted the improved RSVD technique to arrive at the higher accuracy results. The baseline estimation equation for predicting the unknown rating in Equation (7) can be rearranged to comply with the rRSVD as

$$\hat{r} = \bar{r} + b + q^T p \quad (8)$$

The training process estimates the values of the rRSVD model parameters which are b, p and q (the combination of these values were previously denoted as $\Delta^{\{A\}}$ and $\Delta^{\{I\}}$). It is executed repeatedly to get $\hat{r} \approx r$.

This revolutionary RSVD function is invoked two times along the training process to construct $\Delta^{\{A\}}$ and $\Delta^{\{I\}}$.

The inputs of this function are

o user-item rating matrix (r), and

o an $m \times 1$ matrix $\bar{r}^{\{A\}}$, for $\Delta^{\{A\}}$, or an $n \times 1$ matrix $\bar{r}^{\{I\}}$ for $\Delta^{\{I\}}$.

o γ and λ are the stochastic gradient descent method constants. The values of γ and λ are set to 0.005 and 0.02, respectively [11, 12].

Function: Revolutionary RSVD

1. Create the revolutionary RSVD model parameters for the matrices b, p and q. The size of matrices p and q are $m \times k$ and $n \times k$, respectively, while the size of matrix b is equal to the size of matrix $\bar{r}$.

2. Randomize all elements of matrices b, p and q with the small values.

3. Set *Iter* = 1

4. LOOP UNTIL *Iter* = *LimitIter* OR *MAE* < = *LimitMAE*

4.1 Compute the value of the estimated rating score of all rated elements,

$$\hat{r} = \text{round}(\bar{r} + b + q^T p)$$

4.2 Calculate the prediction error by comparing the value of the estimated rating score in matrix $\hat{r}$ and all rated elements in matrix r,

$$e_{a,i} = \begin{cases} r_{a,i} - \hat{r}_{a,i} & \text{if } r_{a,i} \text{ has been rated} \\ 0 & \text{otherwise} \end{cases}$$

4.3 Evaluate the model parameters by means of a stochastic gradient descent method to minimize the regularized squared error according to

- $b = b + \gamma(e - \lambda \cdot b)$
- $p = c + \gamma(e \cdot d - \lambda \cdot c)$
- $q = d + \gamma(e \cdot c - \lambda \cdot d)$

4.4 Increment the value of *Iter* by 1

4.5 Calculate the value of *MAE*, where

$$\text{MAE} = \frac{\sum_{a=1}^{|m_i|} \frac{(\sum_{i=1}^{n_a} f_{a,i})}{|n_a|}}{|m_i|}$$

5. END LOOP
6. return $\Delta = b + q^T p$

According to Koren [11, 12], the learning process repeats for setting the model parameters until the terminal conditions such as LimitIter and LimitMAE are reached. There is no explicit termination condition, but the algorithm normally loops until the error rate (*error*) is close to zero. In contrast, our algorithm states the termination condition based on the value of Mean Average Error (MAE), which is the average absolute errors corresponding to actual ratings-prediction pairs, along with their average [9, 17]. Lower MAE [6] value corresponds to more accurate user ratings prediction. The appropriate value for *LimitMAE* established in this research is 0.50, in which the learning process will not lead to over-fitting.

2.1.3 Incremental adaptation processes

Further investigations are established in order to arrive at some forms of an adaptation agent which are practical enough to embed into the agent model working in the decentralized environment. When the user or another agent has provided a new explicit rating value on an item, the knowledge captured from the training phase must be refreshed in some way so as to integrate the new knowledge into the agent's repertoire. This process is called the adaptation process.

Two incremental adaptation processes are proposed, namely, incremental adaptation at client site and incremental adaptation at server site.

(a) Incremental Adaptation Process at Client Site.

After agent *a* receives a new explicit rating value, it adapts itself according to the change by updating its adaptable part $\bar{r}^{\{A\}}$.

(b) Incremental Adaptation Process at Information Server Site.

After the agent receives a new explicit rating value on an item, the same value is also sent to the information server (i.e. the server that owns the information about all items). The average rating score of all explicit rating values made on that item in matrix $\bar{r}^{\{I\}}$ is updated accordingly.

In order to model the incremental update equation, the definition for evaluating an average rating score is established for subsequent use as follows:

Definition 2:

Let R' be a set of all known rating scores with respect to a particular point of view $R' = \{r_1, r_2, ..., r_{n'}\}$, and $|R'| = n'$, r, be average rating score of all known rating scores, hence r can be calculated as

$$r = \frac{\sum_{j'=1}^{n'} r_j}{n'} \tag{9}$$

A general form of the incremental update equation for both $\bar{r}^{\{A\}}$ and $\bar{r}^{\{I\}}$ is by the following Theorem.

Theorem 1:

Let R be a set of new decided rating scores, $|R| = n$. If the knowledge about the rating scores is changed from R' to $R' \cup R$, then an average rating score of all known rating scores is changed to $\bar{r}^{[n]}$ where the value of $\bar{r}^{[n]}$ is given by

$$\bar{r}^{[n]} = \alpha \bar{r}^{[0]} + \Omega \tag{10}$$

$\bar{r}^{[0]} = \bar{r}$, $\alpha = \frac{n'}{n' + n}$ *and*

$$\Omega = \frac{\sum_{j=1}^{n} r_j}{n' + n}$$

These processes are performed in real time which imposes a critical time constraint

on the update process. Instead of performing total maintenance of the agent's knowledge which takes approximately $O(m^3)$, an incremental update process is proposed. The execution time for an agent to handle a one-item adhoc change while keeping the average value up-to-date takes $O(n)$ and the accompanying incremental update cost becomes $O(1)$.

The theorem can be proved by mathematical induction.

Proof. Let $P(n)$ be the statement

$$\bar{r}^{[n]} = \alpha\bar{r}^{[0]} + \Delta$$
$$= \frac{n'}{n'+n}\bar{r}^{[0]} + \frac{\sum_{j=1}^{n} r_j}{n'+n}$$

We will show that $P(n)$ is true for every $n \geq 1$.

Basis step: for $n = 1$, $P(1)$ denoted by $\bar{r}^{[1]}$ can be written as

$$\bar{r}^{[1]} = \frac{[r_1 + r_2 + \ldots + r_{n'}] + r_{n'+1}}{n'+1}$$

$$= \frac{\sum_{j=1}^{n} r_j + r_{n'+1}}{n'+1}$$

$$= \frac{(n' \times \bar{r}^{[0]}) + r_{n'+1}}{n'+1};$$

this is because $\bar{r}^{[0]} = \frac{\sum_{j=1}^{n} r_j}{n'}$

$$= \left(\frac{n' \times r^{[0]}}{n'+1}\right) + \left(\frac{r_{n'+1}}{n'+1}\right)$$
$$= \frac{n' r^{[0]}}{n'+n} + \frac{\sum_{j=1}^{n} r_j}{n'+n}; \text{ replace 1 with } n$$
$$= \frac{n' r^{[0]}}{n'+n} + \frac{\sum_{j=1}^{n} r_j}{n'+n}$$

Thus, $P(1)$ holds.

Inductive hypothesis: $P(k)$: $\bar{r}^{[k]}$

$$\bar{r}^{[k]} = \frac{[r_1 + r_2 + \ldots + r_{n'}] + [r_{n'+1} + r_{n'+2} + \ldots + r_{n'+k}]}{n'+k}$$

$$= \frac{n'}{n'+k}\bar{r}^{[0]} + \frac{\sum_{j=1}^{n} r_j}{n'+k}$$

Inductive step: $P(k+1)$: $\bar{r}^{[k+1]}$

$$\bar{r}^{[k+1]} = \frac{[r_1 + r_2 + \ldots + r_{n'}] + [r_{n'+1} + r_{n'+2} + \ldots + r_{n'+k}] + r_{n'+(k+1)}}{n' + (k+1)}$$
$$= \frac{[r^{[k]} \times (n'+k)] + r_{n'+(k+1)}}{n' + (k+1)}$$

According to the induction hypothesis,

$$= \frac{[[\frac{n'}{n'+k} \times r^{[0]} + \sum_{j=1}^{n'+k} r_j] \times (n'+k)] + r_{n'+(k+1)}}{n' + (k+1)}$$

$$= \frac{n' \times r^{[0]} + \sum_{j=1}^{k} r_j + r_{n'+(k+1)}}{n' + (k+1)}$$

$$= \frac{n' \times \bar{r}^{[0]} + \sum_{j=1}^{k+1} r_j}{n' + (k+1)}$$

$$= \frac{n'}{n' + (k+1)} \times r^{[0]} + \frac{\sum_{j=1}^{k+1} r_j}{n' + (k+1)}$$

$$= \frac{n'}{n' + (k+1)} r^{[0]} + \frac{\sum_{j=1}^{k+1} r_j}{n' + (k+1)}$$

Thus, $P(k+1)$ holds. By induction, $P(n)$ holds for all n.

Bearing the above adaptation process in mind, we propose three agent adaptation models that designate how the agent operates as follows:

(a) A 2-phase agent model using revolutionary RSVD technique (rRSVD2)

An adaptive agent performs some adaptation processes at both client site and information server site. This agent model is suitable for the agent that operates over high available bandwidth.

(b) A 1-phase agent model using revolutionary RSVD technique (rRSVD1)

An adaptive agent performs an adaptation processes at client site. This agent model is suitable for the agent that operates over low available bandwidth.

(c) A non-adaptation agent model using revolutionary RSVD technique (rRSVD0)

A non-adaptive agent makes its decision based on the prior knowledge obtained from the training process. This agent model is suitable for the agent that operates over poor available bandwidth or off-line.

2.2 Experimental Materials.

We acquired our dataset from MovieLens (ML), a research recommender site maintained by the GroupLens project (www.movielens.mun.edu). This data set contains 943 users, 1682 movies, and 100,000 ratings (discrete values from 1 to 5). The sparsity of this data set is 93.7%. The accuracy was measured by MAE.

3. RESULTS AND DISCUSSION

We compared our algorithm with the SVD-based technique as the baseline experiment since it was one of the successful CF technique [2]. In addition, we also compared our approach with other SVD-based techniques, namely, LSI/SVD [15], RSVD [13], and improved RSVD model [11, 12], and SVD + + [11, 12]. The results of these comparisons are shown in Figure 2.

It is apparent from Figure 2, that the MAE obtained from the proposed 2-phased agent model using revolutionary RSVD technique (rRSVD2) is the lowest. This implies that the performance increases when high bandwidth is available. The 1-phased agent model using revolutionary RSVD technique (rRSVD1) also provides good MAE results. This is because the knowledge refreshment of rRSVD1 model only adapts some parts of agent's knowledge.

In case of distributed environment and off-line situations in which none of the adaptation process could be practically performed, the MAE obtained from the proposed non-adaptation agent model using revolutionary RSVD technique (rRSVD0) is lower than other proposed SVD-based techniques as shown in Table 1. This implies that the proposed technique gives rise to high accurate results as it can perform independently in distributed environment and off-line situation as well.

It can be concluded that for high bandwidth where the agents can send its feedback to the information server, and some incremental adaptation processes are taken into account at both agent and information server sites, the best accurate results will be obtained. As the available bandwidth becomes lower where the agents are hard to send their feedback to the information server, the incremental adaptation can only perform at the agent site to refresh some part of its knowledge. The prediction accuracy results are still better than other SVD-based techniques.

4. CONCLUSION

This paper proposes a collaborative agent model, operated by means of three processing steps. This constitutes a collaborative processing model which is suitable for agents to operate in distributed environment and off-line manner. The expertise from other agents is learned at the central repository site and adapted as prior knowledge by the originating agent. The underlying principle lies in the revolutionary RSVD technique to determine the rating scores, given the above knowledge and other derived details from the information server. Evaluations are carried

out at the agent site. A fast incremental update processes of prior knowledge is devised to enhance the adaptability and performance of this proposed agent model.

The scope of this proposed model is restricted to the limitations of discrete rating scores that preclude a wider range of applicability. Further investigation on continuous values is a challenging research endeavor remained to be explored.

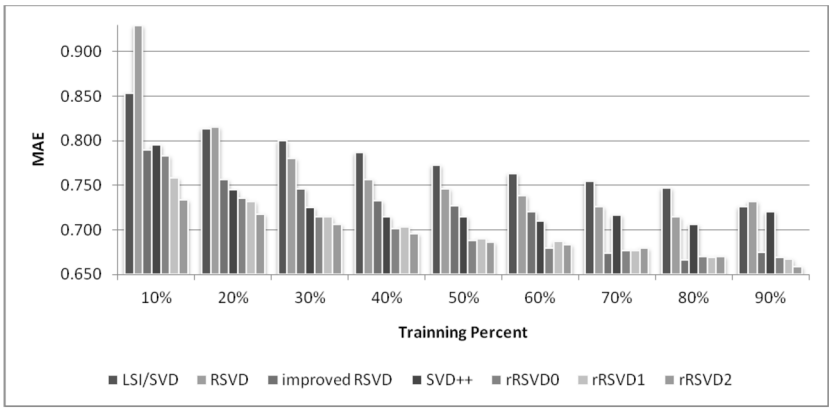

Figure 2. Evaluation of MAE according to different size of training/testing sets.

Table 1. Comparative statistics of the proposed approach and exiting approaches

	LSI/SVD	RSVD	improved RSVD	rRSVD0
10%	0.853	0.929	0.790	**0.783**
20%	0.814	0.815	0.756	**0.736**
30%	0.800	0.780	0.746	**0.715**
40%	0.787	0.757	0.733	**0.702**
50%	0.772	0.746	0.727	**0.688**
60%	0.763	0.738	0.721	**0.680**
70%	0.754	0.726	**0.674**	0.677
80%	0.747	0.715	**0.666**	0.671
90%	0.726	0.731	0.675	**0.670**

REFERENCES

[1] Adomavicius G. and Tuzhilin A., Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Transactions on Knowledge and Data Engineering*, 2005; **17(6)**: 734-749.

[2] Cacheda F., Carneiro V., Fern′andez D. and Formoso V., Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems, *ACM Transactions on the Web*, 2011; **5(1)**: 2:1-2:33.

[3] Carbo J. and Molina J.M., Agent based collaborative filtering based on fuzzy recommendations, *Int. J. Web Eng. Technol.*, 2004; **1**: 414-426.

[4] Funk S., Netfiix update: Try this at home, Available at: *http://sifter.org/simon/journal/20061211.html.*, 2006.

[5] George T. and Merugu S., A Scalable Collaborative Filtering Framework

Based on Co-clustering, *Proceedings of the 5th IEEE International Conference on Data Mining*, 2005; 625-628.

[6] Godoy D. and Amandi A., An Agent-based Recommender System to Support Collaborative Web Search Based on Shared User Interests, *Proceedings of the 13th International Conference on Groupware: Design, Implementation, and Use*, 2007; 303-318.

[7] Herlocker J.L., Konstan J.A., Borchers A. and Riedl J., An Algorithmic Framework for Performing Collaborative Filtering, *Proceedings of the 1999 Conference on Research and Development in Information Retrieval*, 1999; 230-237.

[8] Herlocker J.L., Konstan J.A., Terveen L.G. and Riedl J.T., Evaluating collaborative filtering recommender systems, *ACM Transactions on Information Systems*, 2004; **22(1)**: 5-53.

[9] Hern′andez del Olmo F. and Gaudioso E., Evaluation of recommender systems: A new approach, *IEEE Transactions on Knowledge and Data Engineering*, 2008; **35(3)**: 790-804.

[10] Khoshneshin M. and Street W.N., Incremental Collaborative Filtering via Evolutionary Co-clustering, *Proceedings of the 4th ACM Conference on Recommender Systems*, 2010; 325-328.

[11] Koren Y., Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model, *Proceedings of the* 5th *International Conference on Computer and Information Technology*, 2008; 426-434.

[12] Koren Y., Collaborative Filtering with Temporal Dynamics, *Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining*, 2009; 447-455.

[13] Paterek A., Improving Regularized Singular Value Decomposition for Collaborative Filtering, *Proceedings of the KDD Cup Workshop at the 13th ACM International Conference on Knowledge Discovery and Data Mining*, 2007; 39-42.

[14] Sarwar B., Karypis G., Konstan J. and Riedl J., Item-based Collaborative Filtering Recommendation Algorithms, *Proceedings of the 10th International Conference on World Wide Web*, 2001; 285-295.

[15] Sarwar B.M., Karypis G., Konstan J.A. and Riedl J.T., Application of Dimensionality Reduction in Recommender System a Case Study, *Proceedings of ACM WebKDD 2000 Web Mining for E-Commerce Workshop*, 2000; 82-90.

[16] Su X. and Khoshgoftaar T.M., A survey of collaborative filtering techniques, *Adv. Artif. Intell.*, 2009; 1-20.

[17] Vozalis E. and Margaritis K.G., Analysis of Recommender Systems' Algorithms, *Proceedings of the 6th Hellenic European Conference on Computer Mathematics and Its Applications*, 2003; 1-14.