International Journal of Artificial Intelligence,
ISSN 0974-0635; Int. J. Artif. Intell.
Autumn (October) 2013, Volume 11, Number A13

A Loosely Collaborative Dependency Framework for a Fast Adaptive Agent Model using Extended RSVD

Dussadee Praserttitipong [1] **and Peraphon Sophatsathit**[2]

Advanced Virtual and Intelligent Computing (AVIC) Center,
Department of Mathematics and Computer Science,
Faculty of Science, Chulalongkorn University, Bangkok, 10330, Thailand
Email:[1]D.Praserttitipong@gmail.com, [2]Peraphon.S@Chula.ac.th

ABSTRACT

This paper proposes a loosely dependent agent model, where the underlying training is based on a collaborative filtering, i.e., an extended Regularized Singular Value Decomposition (RSVD) technique. Such software agents normally act as a cooperative assistant to carry out some tasks on behalf of their user, thereby operate alone smoothly in any virtual environment. The extended RSVD technique runs periodically at the server site to rapidly update the agent's experiences. The derived knowledge in conjunction with those from other collaborative agents are encapsulated as prior knowledge and sent back to the client agent. Thus, the agent can execute its tasks having timely repertoire of experience to make suitable decisions. The adaptation is said to complete.

Keywords: recommender systems, collaborative agent, incremental update, extended RSVD technique.

Computing Classification System: H.3.3.

1 INTRODUCTION

The advent of Internet connectivity creates an information-centric society where millions of people access large amount of information stored in the networks on a daily basis. This is accomplished by means of information retrieval systems that serve as essential tools to guide users to the information they are seeking. The emergence of intelligent agents in the last decade provides an alternative to assist the users acquiring needed information in a distributed way. Built upon traditional finding tools such as search engines which report information items to the users according to the specified keywords, these agents help their users find relevant items (such as services or goods) based on users' profiles, i.e., users' preference or opinions (Daniela Godoy, 2007). This task is usually reduced to the job of estimating item rating score under investigation that have not been seen by the users. The results of these systems can be both the predicted rating scores for some particular items and a set of top-n recommended items.

Several intelligent agent systems have been studied in literatures (Mukherjee, Sajja and Sen, 2003; Carbo and Molina, 2004; Daniela Godoy, 2007; Garcia, Tarin and Giret, 2011). Some of them suggested items similar to the ones that the user preferred in the past. An inference model based on the relationship between historical user's decisions and item contents are established via a back-end learning process (Mukherjee et al., 2003; Carbo and Molina, 2004). This inference model is used for evaluating the rating scores of the unrated items at runtime environment. However, this inference learning model is limited to the domain in which the contents or the characteristics of the items are difficult to analyze. Regardless of the item contents, some intelligent agent systems construct their decisions by taking advantage of the knowledge acquired from other collaborative agents. The decision making process is made according to the calculation of user profiles or preferences similarity. Thus, this kind of agent systems can be implemented for any kind of decision items. However, building comprehensive knowledge from user profiles and preferences require some explicit inquiry processes for fulfilling the complete information. These additional workloads might hinder the advantages of user cooperation (Daniela Godoy, 2007). Moreover, the fully knowledge sharing sessions across all collaborative agents cannot be guaranteed to succeed because the on-line partnership with other agents might increase the overall network congestion, while some sessions might not be able to establish. Thus, some useful information might be lost and the agents have to make recommendations with incomplete information. Therefore, further investigations are needed in order to arrive at some form of agent's knowledge acquisition technique that will enable distributed decision making process with the least requirements for on-line cooperation from other agents.

The collaborative filtering (CF) technique, which is the best known type in conventional centralized recommender system technique, is investigated for designing the agent's knowledge construction process. Rather than performing profiles matching, the agent's knowledge is created via an off-line centralized learning process performed on historical rating scores of all collaborative parties (Han and Qiao, 2011; Haghanikhameneh, Payam Hassany Shariat Panahy and Mousavi, 2012). The agents who have similar tastes are selected for acquiring extra knowledge on unfamiliar items. Hence, users are free from the burden of providing explicit data that can be obtained elsewhere and the decision making can be performed on any kind of items. However, high quality recommendation techniques, such as RSVD technique, require expensive time for knowledge acquisition and maintenance process. They are constrained by data sparsely problem, where the number of historical ratings obtained is usually very small compared with the number of ratings that need to be predicted. Therefore, a knowledge refreshment process is required to combine new knowledge with prior knowledge in order to increase the prediction accuracy in some way.

This paper proposes a modified centralized CF technique called extended RSVD technique to set up an adaptive knowledge model for one agent. The knowledge for this agent is constructed in a centralized server learning historical rating data from other agents and from the agent itself through the proposed extended RSVD learning technique. Then, this knowledge is sent to the designated remote agent for use as its prior knowledge. An incremental knowledge update process is also presented to enabling adaptability. This proposed agent's adaptability scheme plays a part in solving the high cost for knowledge acquisition and maintenance process. In addition, this adaptability enables the agent to make its own decisions in a loosely collaborative dependency manner where a fully connected network is not necessary, thereby solving a typical network congestion problem caused by knowledge sharing sessions across all collaborative agents. A reference architecture for the proposed adaptive agent model and evaluation of dependency degree are also furnished to validate the merit of the proposed framework. Details on how the proposed technique is developed and implemented are described as follows. Section 2 explains a few relevant related works. Section 3 summarizes the configuration of a few pertinent models and infers a measurement scheme for dependency evaluation. Section 4 clarifies the proposed model and reference architecture. Section 5 exhibits the experimental results along with discussions. Section 6 concludes with and some final thoughts.

2 RELATED WORKS

Some terminologies and relevant techniques are established based on a few related works for subsequent references as follows (Adomavicius and Tuzhilin, 2005; Hernandez del Olmo and Gaudioso, 2008; Cacheda, Carneiro, Fernandez and Formoso, 2011):

> Let r be an $m \times n$ user-item rating matrix. The element $r_{a,i}$ is a discrete rating score of user a over item i, and the row r_a can be represented as $r_a = (r_{a,i}, r_{a,2}, \ldots, r_{a,n})$. If item i has been rated by user a, $r_{a,i} \geq 1$.

The SVD-based technique which is one of the successful conventional centralized technique (Cacheda et al., 2011), has been developed and improved in the last decade. The main idea of this technique is to decompose user-item rating matrix to other matrices using different approaches that are summarized below:

1. **Latent Semantic Indexing/SVD:** The latent semantic indexing/singular value decomposition (LSI/SVD) method was introduced by Sarwar *et al* (Sarwar, Karypis, Konstan and Riedl, 2000; Vozalis, Markos and Margaritis, 2009). The learning process performs over a user-item rating matrix (r) in deriving the factorized matrices. The main idea of this approach is to reduce insignificant users or items by capturing latent relationships between users and items, thereby the dimension of the matrix decreases.

2. **Regularized SVD:** Regularized SVD (RSVD) is a technique inspired by effective matrix factorization method from the domain of natural language process (Paterek, 2007). This technique was originally proposed in the context of CF environment by Simon Funk (Funk, 2006). In this model, each item is represented as a set of aspects and each user is represented as a set of values indicating their preference for the various aspects of the items. The learning process arrives at the factorized matrices by utilizing a minimized squared error function. The learning process repeats setting the model parameters until the terminal conditions are reached. There is no explicit value for the terminal conditions (Paterek, 2007), but normally the algorithm loops until the error rate is close to zero.

3. **Improved RSVD:** This technique was informally introduced in (Koren, 2008; Koren, 2010) by adding some bias parameters to RSVD model, indicating the observed bias (deviations) of the users over the item.

4. **SVD++:** This method was formally introduced by Koren (Koren, 2008; Koren, 2010). In this method, the adaptation of the recommendation model is the main concern and is realized with the help of implicit feedbacks.

3 AN AGENT FRAMEWORK FOR SUPPORTING CF TECHNIQUES

A dependency degree is a measurement introduced in this paper to evaluate the extent to which the agent or user a relying on other parties executes a prediction process, as well as adapts its knowledge according to a new rating score being fed back from its users. This value is modelled in terms of the occurrence of collaboration that an agent a must involve with other parties for making appropriate decisions and refreshing its knowledge with respect to an item i. Thus, if the dependency degree increases, the connection cost also increases. The reference model of the current agent models are summarized below.

3.1 A distributed adaptive agent model

A simple solution for implementing an adaptive agent model is to create knowledge sharing sessions among collaborative agents and retain these connections at all time. A general model for this solution including interaction flows between the cooperating components is presented in Figure 1.

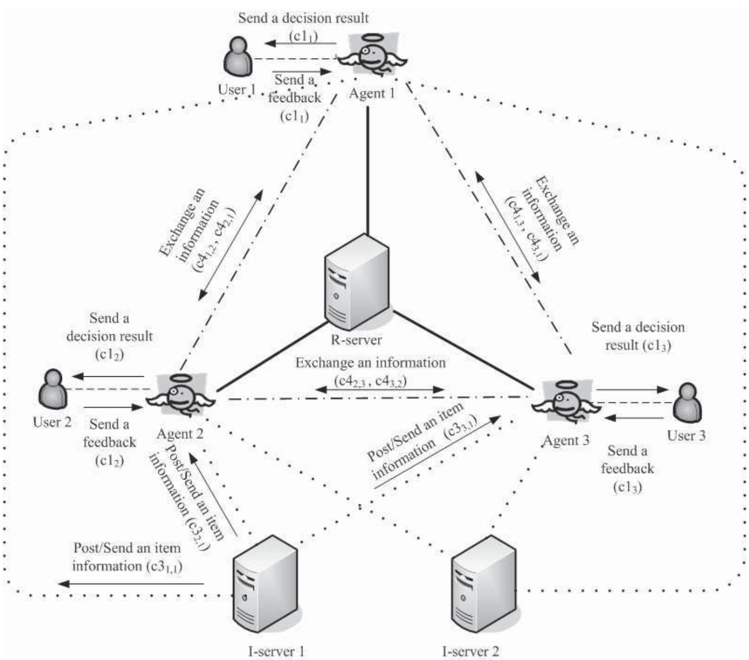

Figure 1: A general schema for the distributed adaptive agent model

As shown in Figure 1, the predicting process takes place at the client's site. The knowledge required is periodically updated by an adaptation process, which is performed according to the user feedback in conjunction with additional information acquired from other agents. I-server is an information server that stores the information about the items and R-server is accounted for the central repository site where prior knowledge of the agents is archived, processed, and generated according to the learning process.

The dependency value for refreshing the prior knowledge and making appropriate decisions of an agent a with respect to an item i can be assessed as follows:

- Primary dependency evaluation:
  - An agent a must retain a connection with its user in 2-way connections for sending an appropriated recommendation results to its user and getting its user feedback. This dependency costs $2(c1_a)$.
  - An agent a must hold its connections with other agents in 2-way connections for getting others' opinion and sharing its opinion to others. This dependency costs $2\sum_{a'=1}^{m-1} c4_{a,a'}$, where $a' \neq a$.
  - An I-server a presents its current item i to an agent a. This dependency costs $c3_{a,i}$.
- Secondary dependency evaluation: determines the costs of connecting other $m-1$ collaborative agents (a') to other parties in the system
  - There are $m-1$ collaborative agents that must hold their communication with their own users in 2-way connections. This dependency costs $2\sum_{a'=1}^{m-1} c1_{a'}$.
  - There are connection costs from an I-server a posting its current item i to other collaborative agents. This dependency costs $\sum_{a'=1}^{m-1} c3_{a,i}$.

Thus, the overall dependency value for this distributed adaptive agent model becomes

$$
\begin{aligned}
cp &= 2(c1_a) + 2\sum_{a'=1}^{m-1} c4_{a,a'} + c3_{a,i} + 2\sum_{a'=1}^{m-1} c1_{a'} + \sum_{a'=1}^{m-1} c3_{a',i} \\
&= 2(c1_a) + 2\sum_{a'=1}^{m-1} c1_{a'} + c3_{a,i} + \sum_{a'=1}^{m-1} c3_{a',i} + 2\sum_{a'=}^{m-1} c4_{a,a'} \\
&= 2\sum_{a=1}^{m} c1_a + \sum_{a=1}^{m} c3_a + 2\sum_{a'=}^{m} c4_{a,a'}
\end{aligned}
$$

The drawback of this approach is the assumption that all the collaborating agents must be fully connected among them. Actually, this requirement cannot be guaranteed in general. This is due to the fact that the on-line partnership with other agents might increase the overall network congestion, or the connection session might not be successfully established. As a result, some useful information from the collaboration might be lost and the agents have to determine the rating score based on incomplete information.

3.2 An agent model using non-adaptive conventional CF technique

In order to model such a distributed agent system, the CF technique is employed in this paper. This model is depicted in Figure 2 which illustrates an agent system whose decision making process relies on a conventional centralized R-server. The knowledge is formulated from either memory-based CF or model-based CF recommender system. The predicting process is conducted at the central repository. There is no maintenance process to be performed.

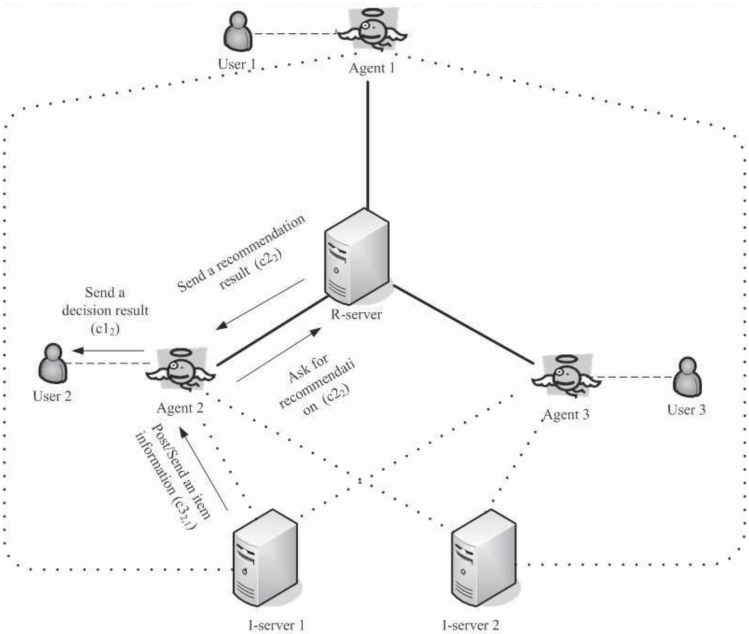

Figure 2: A general schema for the agent model using a non-adaptive conventional CF technique

The dependency value for the agent model using a non-adaptive conventional CF technique of an agent a can be assessed as follows:

- Primary dependency evaluation:
  - In contrast to the distributed adaptive agent model, the user feedback is not taken into account for this approach because the agent performs its own decision making process purely on historical knowledge collected at the centralized R-server. An agent a must retain its connection to its user in 1-way connection for sending an appropriated recommendation results. This dependency costs $c1_a$.
  - An agent a must hold its connection to the centralized R-server in 2-way connections for sending its request seeking recommendation and receiving the recommended result. This dependency costs $2(c2_a)$.

- An I-server a presents its current item i to an agent a. This dependency costs $c3_{a,i}$.

The dependency value of the predicting process with respect to an item i for the non-adaptive conventional CF techniques can be computed from

$$cp = c1_a + 2c2_a + c3_{a,i}$$

Examples of this approach are user-based CF (Herlocker, Konstan, Borchers and Riedl, 1999), item-based CF (Sarwar, Karypis and Joseph Konstan, 2001), Naive Bayes classification approach, SVM classification approach, collaborative filtering via co-clustering (COCL) (George and Merugu, 2005), Latent Semantic Indexing Singular Value Decomposition (LSI/SVD) (Sarwar et al., 2000; Vozalis et al., 2009; Paterek, 2007).

3.3 An agent model using adaptive conventional CF technique

This model attempts to rectify the above non-adaptive conventional CF centralized approach being constrained by the high cost for knowledge acquisition and maintenance process. Some adaptive CF techniques, such as an evolutionary co-clustering (ECOCL) (Mukherjee et al., 2003) and SVD++(Koren, 2008; Koren, 2010) have been studied. In this model, an agent makes its decision based on the information recommended by the R-server. The knowledge is formulated from the model-based CF recommender system. The prediction process is conducted at the R-server.

Figure 3 depicts a general schema for this approach. All user feedbacks are sent to update historical data at the R-server. This attempt is known as adaptation process. Thus, the interactions among all agents to predict an item i and to refresh prior knowledge about that item are taken into account. The Dependency value of an agent a with respect to an item i can be assessed as follows:

- Primary dependency evaluation:
  - An agent a must retain a connection with its user in 2-way connections for sending an appropriated recommendation result to the user and getting the user feedback. This dependency costs $2(c1_a)$.
  - An agent a must connect to the centralized R-server 3 times sending its request for seeking recommendation on item i, receiving the recommended result, and sending its user feedback to the R-sever for knowledge maintenance/update. This dependency costs $3(c2_a)$.
  - An I-server a presents its current item i to an agent a. This dependency costs $c3_{a,i}$.
- Secondary dependency evaluation determines the costs of connecting other $m-1$ collaborative agents (a') to other parties in the system
  - There are $m-1$ collaborative agents that must hold their communication with their own users in 2-way connections. This dependency costs $2\sum_{a'=1}^{m-1} c1_{a'}$.

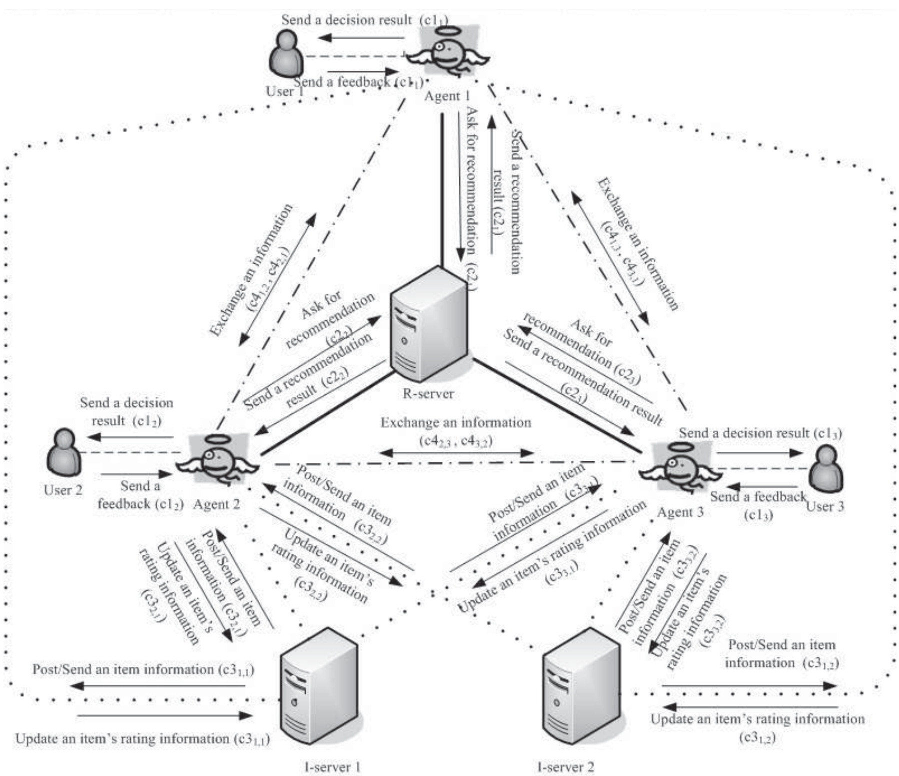

Figure 3: A general schema for the agent model using adaptive conventional CF technique

– The $m-1$ collaborative agents must also connects to the centralized R-server 3-times for sending their request for seeking recommendation for an item i, receiving the recommended result, and sending their user feedback to the R-sever. This user feedback is used for on-line maintenance of the knowledge by R-sever. This dependency costs $3\sum_{a'=1}^{m-1}(c2_a)$.

– There are connection costs from an I-server a posting its current item i to other collaborative agents. This dependency costs $\sum_{a'=1}^{m-1} c3_{a,i}$.

Thus, the overall dependency value for this distributed adaptive agent model becomes

$$\begin{aligned} cp &= 2(c1_a) + 3c2_a + c3_{a,i} + 2\sum_{a'=1}^{m-1} c1_{a'} + 3\sum_{a'=1}^{m-1} c2_a + \sum_{a'=1}^{m-1} c3_{a',i} \\ &= 2(c1_a) + 2\sum_{a'=1}^{m-1} c1_{a'} + 3c2_a + 3\sum_{a'=1}^{m-1} c2_a + c3_{a,i} + \sum_{a'=1}^{m-1} c3_{a',i} \\ &= 2\sum_{a=1}^{m} c1_a + 3\sum_{a=1}^{m} c2_a + \sum_{a=1}^{m} c3_{a,i} \end{aligned}$$

It is apparent that these conventional centralized recommender systems generally perform the evaluation at the central repository where user opinions are stored. This centralized evaluation can induce a bottleneck in the system (Carbo and Molina, 2004). Further investigations are needed in order to arrive at some forms of modified CF technique which are practical enough to embed it into the agent model working in the decentralized environment.

4 THE PROPOSED MODEL

We propose a loosely collaborative dependency framework extended with a fast incrementally update algorithm. The main idea is to untie the on-line user-to-user knowledge sharing connections. Besides, know-how from other agents can be incorporated incrementally as prior knowledge of the agents. Details of this proposed model are described in the following subsections.

4.1 Reference architecture

Even though the knowledge derived from conventional CF can be distributed for off-line applications, both RSVD and improved RSVD techniques are constrained by sparse data problems, where the number of known ratings available for learning is usually very small compared to the number of ratings to be predicted. In order to build a comprehensive repertoire of agent's knowledge, acquiring knowledge from other collaborative agents is a viable alternative for knowledge enhancement scheme.

Building on top of the general model as depicts in Figure 1, know-how from other agents are learned from the central repository site and used as the prior knowledge for the agent to make its local decision. When a rating score is required, this prior knowledge and the adaptable knowledge part derived from the information server are integrated to estimate a new rating at the agent site.

The key point of the estimation procedure lies in an efficient adaptation process that yields more accurate results. At the agent site, the agent performs incremental adaptation knowledge update by means of knowledge gathering know-how from the I-server, user feedback, and other agent knowledge. Figure 4 depicts a general reference architecture of the proposed adaptive agent model.

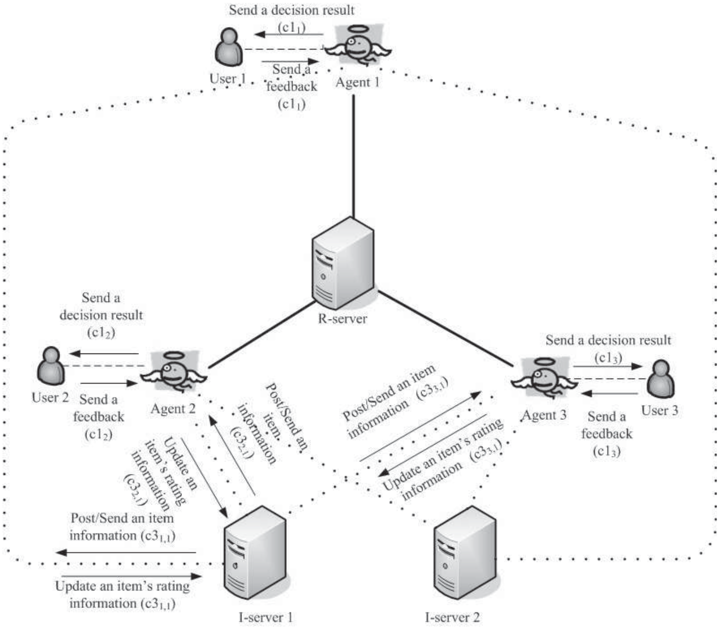

Figure 4: A reference architecture of the proposed adaptive agent model

This consolidated knowledge is used to predict new scores, where dependency with respect to an item i of the adaptive can be assessed as follows:

- Primary dependency evaluation:
  - An agent a must retain a connection with its user in 2-way connections for sending an appropriated recommendation results to the user and getting the user feedback. This dependency costs $2(c1_a)$.
  - An I-server a presents its current item i to an agent a. This dependency costs $c3_{a,i}$.
- Secondary dependency evaluation determines the costs of connecting other $m-1$ collaborative agents (a') to other parties in the system
  - There are $m-1$ collaborative agents that must hold their communication with their own users in 2-way connections. This costs $2\sum_{a'=1}^{m-1} c1_{a'}$.

– There are connection costs from an I-server a posting its current item i to other collaborative agents. This dependency costs $\sum_{a'=1}^{m-1} c3_{a,i}$.

Thus, the overall dependency value for this distributed adaptive agent model becomes

$$\begin{aligned} cp &= 2(c1_a) + c3_{a,i} + 2\sum_{a'=1}^{m-1} c1_{a'} + \sum_{a'=1}^{m-1} c3_{a',i} \\ &= 2(c1_a) + 2\sum_{a'=1}^{m-1} c1_{a'} + c3_{a,i} + \sum_{a'=1}^{m-1} c3_{a',i} \\ &= 2\sum_{a=1}^{m} c1_a + \sum_{a=1}^{m} c3_a \end{aligned}$$

This process is performed without any need for on-line connection from other agents or servers because all required knowledge is encapsulated in its prior knowledge. Thus, the agent can update with loosely dependency from other parties. The process consists of three steps, (a) prediction process, (b) learning process, and (c) incremental adaptation process.

4.2 Prediction process

This technique exploits the improved RSVD technique (Koren, 2008; Koren, 2010) and extends the prediction accuracy of RSVD technique by adding some bias parameters to RSVD model to indicate the observed bias (deviations) of both user a, denoted by b_a, and the observed bias of item i, denoted by b_i. The baseline estimation becomes

$$\hat{r}_{a,i} = \mu + ba + bi + \sum_{k=1}^{K} (p_{a,k} \times q_{i,k}) \tag{4.1}$$

where μ is the average of all known rating values and ba, bi,p_* and q_* are model parameters whose value is specified according to the learning process. This set up of the proposed extended RSVD technique permits changing the construction of decision making knowledge model from the overall rating model (μ) to mainly stating a user perspective ($\bar{r}_a$) and other perspective ($\bar{r}_i$). Thus, the agent's adaptation mechanism is loosely based on other agent's collaboration which enables a more practical knowledge refreshment process.

When an agent a faces with a decision situation making on an item i, the prior knowledge combining with derived information from the I-server are utilized. The estimated rating value for $\hat{r}_{a,i}$ is given as

$$\begin{aligned} \hat{r}_{a,i} &\approx \text{average}(K^{\{A\}}, K^{\{I\}}) \\ &\approx \text{average}([\bar{r}_a + \alpha_{a,i}^{\{A\}}], [\bar{r}_i + \alpha_{a,i}^{\{I\}}]) \end{aligned} \tag{4.2}$$

where the estimated $\hat{r}_{a,i}$ is comprised of the knowledge with respect to the user viewpoint ($\{A\}$) and the knowledge gathered from other user viewpoint ($\{I\}$), $\bar{r}_a$ is the average rating score of all explicit rating values working out by an agent a, and $\bar{r}_i$ is the average rating score of all explicit rating values applying on item i. These two variables represent the adaptable knowledge part of an agent. This information will change according to the underlying environment, i.e., new rating score, new user, and new item. $\alpha_{a,i}^{\{A\}}$ and $\alpha_{a,i}^{\{I\}}$ denote prior knowledge with respect to the user viewpoint and prior knowledge gathering from other user viewpoint, respectively.

4.3 Learning process

The learning process enhances the RSVD technique of the conventional centralized recommender system. A primer RSVD function is presented below.

FUNCTION: A Primer RSVD
Input: (a) user rating matrix r, (b) average rating matrix $\bar{r}$.
Output: prior knowledge matrix (α)

---

1. Create the model parameters matrices, ba, bi, p and q where ba, bi, p and q are $m \times 1$, $n \times 1$, $m \times k$, and $n \times k$ matrices, respectively.
2. Randomize the value of all elements in the matrices ba, bi, p and q with small values.
3. LOOP UNTIL $Iter = LimitIter$ OR $MAE \leq LimitError$
   (a) Set $MAE = 0$
   (b) FOR any given rating $r_{q,i}$, where $\hat{r}_{a,i}$ has been rated,
      i. a prediction $\hat{r}_{a,i}$ is made according to
      $$\hat{r}_{a,i} = \text{round}(\mu + ba + bi + q_i^T p_a) \tag{4.3}$$
      ii. a predicting error is calculated from $e_{a,i} = r_{a,i} - \hat{r}_{a,i}$
      iii. the RSVD model parameters are assigned using a stochastic gradient descent method for minimizing the regularized squared error as follows:
         - $bi_i = bi_i + \gamma(e_{a,i} - \lambda \cdot bi_i)$
         - $ba_a = ba_a + \gamma(e_{a,i} - \lambda \cdot ba_a)$
         - $q_i = q_i + \gamma(e_{a,i} \cdot p_a - \lambda_2 \cdot q_i)$
         - $p_a = p_a + \gamma(e_{a,i} \cdot q_i - \lambda \cdot p_a$
      iv. calculate the value of MAE, where $MAE = MAE + |r_{a,i} - \hat{r}_{a,i}|$
   (c) END FOR
   (d) Recalculate the value of MAE using $MAE = MAE/m$
   (e) Increment the value of $Iter$ by 1
4. END LOOP
5. Return prior knowledge (α), where $\alpha_{a,i} = ba + bi + q_i^T p_a$

According to Koren (Koren, 2008; Koren, 2010), the values of γ and λ are set to 0.005 and 0.02, respectively. The model parameters are learned by the stochastic gradient descent method for minimizing the regularized squared error. The learning process repeats for setting the model parameters until the terminal conditions (i.e., LimitIter and LimitError) are reached. There is no explicit terminal condition in conventional algorithms but loop until the error rate is close to zero. In contrast, our algorithm states the terminal condition based on the value of MAE, where it is an average absolute error corresponding to actual ratings-prediction pairs, along with their average (Adomavicius and Tuzhilin, 2005; Hernandez del Olmo and Gaudioso, 2008; Cacheda et al., 2011). Lower MAE value corresponds to more accurate user rating prediction. The appropriate value for $LimitError$ suggested in this paper is 0.5, where the learning process will not lead to overfitting. The variable k is the largest singular values, where $k \leq n$ yields the lowest MAE value for decomposing matrix r into the model parameters of the learning process. Thus, the size of the vital model parameter matrix p and q depends on the characteristics of the data set.

In addition, the matrix r is replaced with an average rating matrix ($\bar{r}^{\{A\}}$) of all users. The variable $\bar{r}$ in Equation 4.2 is replaced by $\bar{r}^{\{A\}}$ in case of modeling the knowledge with respect to the user viewpoint ($\hat{r}^{\{A\}}$). In this case, the matrix in Equation 4.2 is replaced by the average rating matrix ($\bar{r}^{\{I\}}$) for all items.

The overall learning process for modeling both $\hat{r}^{\{A\}}$ and $\hat{r}^{\{I\}}$ proceeds as follows. The collaboration is established based on evaluation of historical rating scores. The essential know-how collected from other agents is encapsulated as the prior knowledge for porting to the clients by means of the proposed extended RSVD technique. Finally, all constructive knowledge is consolidated and encapsulated in the agent for subsequent remote execution. The algorithm for the learning process can be described below.

FUNCTION: Extended RSVD
Input: user rating matrix (r),
Output: resulting knowledge $\bar{r}^{\{A\}}$, $\bar{r}^{\{I\}}$, $\alpha^{\{A\}}$, $\alpha^{\{I\}}$

---

1. Calculate an average rating score of all explicit rating values working out by an agent a, denoted by $\bar{r}^{\{A\}}$. The value of $\bar{r}^{\{A\}}$ is determined from

$$\bar{r}_a^{\{A\}} = \frac{\sum_{i=1}^{|n_a|} r_{a,i}}{|n_a|} \tag{4.4}$$

   where n_a denotes a set of items that have been rated by user a and $n_a \leq n$.

2. Calculate an average rating score of all explicit rating values on item i, denoted by $\bar{r}^{\{I\}}$. The value of $\bar{r}^{\{I\}}$ is determined from

$$\bar{r}_i^{\{I\}} = \frac{\sum_{a=1}^{|n_a|} r_{a,i}}{|m_i|} \tag{4.5}$$

   where m_i denotes a set of users that rate item i and $m_i \leq m$.

3. Execute a primer SVD function with $\bar{r}_a^{\{A\}}$ to acquire matrix $\alpha^{\{A\}}$

4. Execute a primer SVD function with $\bar{r}_i^{\{I\}}$ to acquire matrix $\alpha^{\{I\}}$

5. The resulting knowledge from this algorithm comprises of
   (a) The average rating for item i, ($\bar{r}_i^{\{I\}}$), is sent back to the information server where the concerning item i is posted.
   (b) The remaining information $\bar{r}_a^{\{A\}}$, $\alpha_{a,*}^{\{A\}}$, $\alpha_{a,*}^{\{I\}}$ is sent in response to the agent a, that requests for total renewing its prior knowledge.

This process is executed at the R-server, 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 complete knowledge sharing session across all others collaboration can be assured.

4.4 Fast incremental adaptation process

When an agent or user has provided a new explicit rating score on an item, the knowledge captured from the training phase must be refreshed in some way. Two incremental adaptation processes are proposed, namely, incremental adaptation at client site and at I-server site. The baseline equation for updating the average rating score is given by

$$\bar{r} = \frac{(\bar{r} \times z) + newrating}{z + 1} \tag{4.6}$$

- In case of incremental adaptation process taking place at client site, the value of $\bar{r}$ and z in Equation 4.6 are set to $\bar{r}_a^{\{A\}}$ and $|n_a|$, respectively.
- In case of incremental adaptation process taking place at I-server, the value of $\bar{r}$ and z in Equation 4.6 are set to $\bar{r}_i^{\{I\}}$ and $|m_i|$, respectively.

These processes are performed in real-time which impose a critical time constraint on the update process. Instead of carrying out total agent's knowledge adaptation which takes approximately $O\left(m^3\right)$, the incremental update process adds the one item that costs only $O\left(1\right)$.

5 RESULTS EVALUATIONS AND DISCUSSIONS

There are two types of result to be evaluated and discussed in this section. They are (1) empirical results, which are gained from the simulation process over the well-known data sets in machine learning research, and (2) dependency evaluation results, which are acquired from the proposed dependency degree measurement.

5.1 Empirical results

We employed the 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, an 100,000 ratings (discrete values from 1 to 5). The value of variable which was suitable for this data set was 100. Two experiments were conducted. Experiment I was carried out to assess prediction accuracy and the cost incurred, while experiment II evaluated the performance in comparison with other existing techniques.

5.1.1 Experiment I

In this experiment, we used a standard 5-fold cross-validation based on 80-20 off-line training and test sets, respectively. Two metrics were employed to evaluate the recommendation results. They are

1. Prediction accuracy: uses Mean Absolute Error (MAE) (Adomavicius and Tuzhilin, 2005; Hernandez del Olmo and Gaudioso, 2008) for evaluating the accuracy.

2. Execution cost: computes the execution cost of each algorithm by applying time complexity measure.

We compared our approach with a number of existing techniques, namely, user-based CF (Herlocker et al., 1999), item-based CF (Sarwar et al., 2001), Naive-Bayes classification approach, SVM classification approach, COCL (George and Merugu, 2005), ECOCL (Mukherjee et al., 2003), LSI/SVD (Sarwar et al., 2000; Vozalis et al., 2009), RSVD (Paterek, 2007), improved RSVD model (Koren, 2008; Koren, 2010), and SVD++ (Koren, 2008; Koren, 2010). The first three algorithms were classified as memory-based CF techniques and the remaining are model-based CF techniques. The results of these comparisons are shown in Table 1.

Table 1: Comparative statistics of the proposed approach and the existing approaches

Technique	**MAE**	Training	Predicting		Adaption
		Cost	**Cost**	**Place**	**Technique**
User-based CF	0.7750	–	$O(mn)$	R-Server	None
Item-based CF	0.7800	–	$O(mn)$	R-Server	None
NB	0.8150	$O(n^2)$	$O(m^2)$	R-Server	None
SVM	0.7800	$O(n^2)$	$O(m^2)$	R-Server	None
COCL	0.7340	$O(m^3)$	$O(1)$	R-Server	None
ECOCLE	0.7160	$O(m^3)$	$O(1)$	R-Server	Incremental
LSI/SVD	0.7438	$O((m+n)^3)$	$O(1)$	R-Server	None
RSVD	0.7145	$O(mnk)$	$O(1)$	R-Server	None
Improved RSVD	0.7023	$O(mnk)$	$O(1)$	R-Server	None
SVD++	0.7150	$O(mn^2k)$	$O(1)$	R-Server	None
Extended RSVD	0.6706	$O(mnk)$	$O(1)$	Client	Incremental

As can be seen from Table 1, the execution cost incurred for the prediction process of most model-based CF techniques is lower than the prediction cost of memory-based CF techniques due to its fast incremental adaptation. Furthermore, the MAE values obtained from the memory-based CF techniques are higher than those from model-based CF techniques. These results confirm that the model-based CF techniques are better than the memory-based CF techniques in terms of prediction accuracy and cost.

Additionally, the prediction process for most model-based CF techniques takes $O(1)$ which makes it fast and practical to perform in a real-time environment. The techniques classified in SVD-based family, especially RSVD, improved RSVD, SVD++, Extended RSVD yield good MAE results. It is apparent that the MAE obtained from the proposed extended RSVD technique is the lowest one and yields the highest prediction accuracy.

5.1.2 Experiment II

To determine the accuracy of the proposed approach, the experiment was conducted based on different number of training data sets. The size of the training sets varied from 10%-90%. We employed the SVD-based technique to be the baseline experiment. Then, we compared our approach with other four techniques in SVD family, namely, LSI/SVD (Sarwar et al., 2000; Vozalis et al., 2009), RSVD (Paterek, 2007), improved RSVD model (Koren, 2008; Koren, 2010), and SVD++ (Koren, 2008; Koren, 2010). The accuracy performed over the different size of training/testing sets is shown in Figure 5.

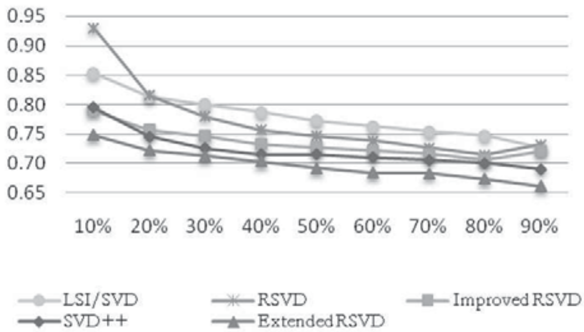

Figure 5: An evaluation of MAE according to different size of training/testing sets

We see that the proposed extended RSVD technique presents the lowest MAE values in every test set. This implies that the accuracy obtained from our proposed model is higher than other techniques.

5.2 Dependency evaluation results

The dependency degree of each approach is evaluated to summarize how flexible each model performed under the agent environment. The dependency degree of each approach can be formulated as shown in Table 2.

Table 2: Comparative statistics of the proposed approach and existing approaches

Approach	Dependency
A distributed adaptive agent model	$cp = 2\sum_{a=1}^{m} c1_a + \sum_{a=1}^{m} c3_a + 2\sum_{a'=}^{m} c4_{a,a'}$
An agent model using non-adaptive conventional CF technique	$cp = c1_a + 2c2_a + c3_{a,i}$
An agent model using adaptive conventional CF technique	$cp = 2\sum_{a=1}^{m} c1_a + 3\sum_{a=1}^{m} c2_a + \sum_{a=1}^{m} c3_{a,i}$
A proposed adaptive agent model	$cp = 2\sum_{a=1}^{m} c1_a + \sum_{a=1}^{m} c3_a$

Notice from Table 2 that dependency degree of the agent model using non-adaptive convention CF technique is the lowest one. However, the model possesses two main drawbacks that can offset the worth of its loose dependence on others.

- First, the accuracy results obtained from the agent model using non-adaptive convention CF techniques (such as the one that uses LSI/SVD or RSVD) are significantly lower than the accuracy obtained from adaptive approach (such as the one that uses ECOCLE technique and Extended RSVD technique).
- Second, some of the agent models using non-adaptive convention CF techniques, for instance, user-based CF (Herlocker et al., 1999), item-based CF (Sarwar et al., 2001), Naive-Bayes classification approach, and SVM classification approach, consume higher prediction processing time that make them impractical for real-time environment.

Thus, it can be concluded that the adaptive agent approach is better than the non-adaptive approach. Dependency degree of the proposed adaptive agent model is lower than the result obtained from other adaptive agent models. This result indicates that our proposed agent is more loosely dependence than other collaborative agent schemes.

6 CONCLUSION

Conventional collaborative agent systems are obstructed from the agent's knowledge achievement process as they rely on the completeness of on-line connections to other agents. To overcome this problem, this paper proposes an extended RSVD technique, which is a modified version of a conventional CF technique, to be used as the knowledge acquisition process. The expertise from other agents is learned at the central repository site and encompassed as prior knowledge for the originating agent via the extended RSVD technique. Evaluations are carried out at the client site having only a few connections with others. This indicates that the problem of complete connection required by the conventional collaborative agent system is resolved by this loose dependency agent framework. In addition, the fast incremental update processes of prior knowledge are devised to enhance the adaptability and performance of agent deployment in real-time environment. One caveat of the proposed model lies in its scope which is restricted to applications based on discrete rating score. It would thus be of interest to further investigate on continuous rating scale which will in turn enable a more general predicting process for wider range of applications.

References

Adomavicius, G. and Tuzhilin, A. 2005. 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* **17**: 734–749.

Cacheda, F., Carneiro, V., Fernandez, D. and Formoso, V. 2011. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems, *ACM Transactions on the Web* **5**: 2:1–2:33.

Carbo, J. and Molina, J. M. 2004. Agent-based collaborative filtering based on fuzzy recommendations, *International Journal of Web Engineering and Technology* **1**(4): 414–426.

Daniela Godoy, A. A. 2007. An agent-based recommender system to support collaborative web search based on shared user interests, *Proceedings of the* 13^{th} *International Conference on Groupware: Design Implementation, and Use*, CRIWG'07, Springer-Verlag, Berlin, Heidelberg, pp. 303–318.

Funk, S. 2006. Netflix update: Try this at home, *http://sifter.org/simon/journal/20061211.html.*

Garcia, I. G., Tarin, L. S. and Giret, A. 2011. A multi agent architecture for single user and group recommendation in the tourism domain, *International Journal of Artificial Intelligence.* **6**(S11): 161–182.

George, T. and Merugu, S. 2005. A scalable collaborative filtering framework based on co-clustering, *Proceedings of the* 5^{th} *IEEE International Conference on Data Mining*, ICDM '08, Houston, Texas, USA, pp. 625–628.

Haghanikhameneh, F., Payam Hassany Shariat Panahy, n. N. K. and Mousavi, S. A. 2012. A comparison study between data mining algorithms over classification techniques in squid dataset, *International Journal of Artificial Intelligence.* **9**(A12): 37–58.

Han, H. and Qiao, J. 2011. An efficient algorithm for feedforward neural network reconstructing and its application, *International Journal of Artificial Intelligence.* **7**(A11): 125–141.

Herlocker, J. L., Konstan, J. A., Borchers, A. and Riedl, J. 1999. An algorithmic framework for performing collaborative filtering, *Proceedings of the* 22^{nd} *annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, Berkeley, California, USA, pp. 230–237.

Hernandez del Olmo, F. and Gaudioso, E. 2008. Evaluation of recommender systems: A new approach, *Expert Systems with Applications* **35**(3): 790–804.

Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model, *Proceedings of the* 14^{th} *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, Las Vegas, Nevada, USA, pp. 426–434.

Koren, Y. 2010. Collaborative filtering with temporal dynamics, *Communications of the ACM* **53**: 89–97.

Mukherjee, R., Sajja, N. and Sen, S. 2003. A movie recommendation system-an application of voting theory in user modeling, *User Modeling and User-Adapted Interaction* **13**(1-2): 5–33.

Paterek, A. 2007. Improving regularized singular value decomposition for collaborative filtering, *Proceedings of the KDD Cup Workshop at the* 13^{th} *ACM International Conference on Knowledge Discovery and Data Mining*, KDDCup '07, San Jose, California, USA, pp. 39–42.

Sarwar, B., Karypis, G. and Joseph Konstan, a. J. R. 2001. Item-based collaborative filtering recommendation algorithms, *Proceedings of the* 10^{th} *International Conference on World Wide Web*, WWW '01, Hong Kong, Hong Kong, pp. 285–295.

Sarwar, B. M., Karypis, G., Konstan, J. A. and Riedl, J. T. 2000. Application of dimensionality reduction in recommender system – a case study, *Proceedings of ACM WebKDD 2000 Web Mining for E-Commerce Workshop*, Boston, MA, USA, pp. 82–90.

Vozalis, M., Markos, A. and Margaritis, K. 2009. On the performance of SVD-based algorithms for collaborative filtering, *Proceedings of the* 4^{th} *Balkan Conference in Informatics*, Thessaloniki, Greece, pp. 245–250.