Journal of Theoretical and Applied Information Technology
20th October 2015. Vol. 80 No.2
© 2005 - 2015 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A HYBRID OF SINGULAR VALUE DECOMPOSITION AND REGRESSION ANALYSIS COLLABORATIVE FILTERING WITH LINEAR INCREMENTAL UPDATE METHOD

[1] WIJAK SRISUJJALERTWAJA, [2] DUSSADEE PRASERTTITIPONG

[1, 2] Department of Computer Science, Faculty of Science, Chiang Mai University, Chiang Mai, THAILAND

E-mail: [1] wijak.s@cmu.ac.th, [2] dussadee.p@cmu.ac.th

ABSTRACT

Collaborative filtering (CF) approach comprises of several well-known techniques which successful in creating personalized recommendations. Singular Value Decomposition (SVD) based technique is the dominant class of CF techniques. The techniques rooted from SVD concept mostly return the high accuracy recommendation results than others. These SVD-based techniques are come up with the concept of model-based CF techniques, in which the relationship of historical ratings between users and items are learned. The learning processes usually perform in off-line mode. The model parameters are assessed according to this off-line process for constructing the knowledge models which are further use in on-line recommendation environments. However, the accuracy of the SVD-based technique is reduced according to the increasing number of solutions that they have to evaluate compared with the steady number of knowledge that they have learned, namely users-items sparse problem. On the other hand, the memory-based CF techniques are also suggested in literatures. These techniques are relied on collecting knowledge about ratings between users and items in computational area and re-computing the entire knowledge every time when the recommendations are called for. Thus, the users-items sparse problem is not an obstructer for memory-based CF techniques, because the new knowledge is always encompassed to the fundamental knowledge. Even though, memory-based CF techniques do not cause of difficulty in the users-items sparse problem, they impractical for implementing in on-line environment. Because of memory-based CF techniques take a lot of time for estimating just only one recommendation result. Hence, this paper proposed a hybrid of SVD-based technique and memory-based technique for CF. A regression analysis algorithm is proposed as a memory-based CF technique. The incremental update method with linear time refreshment also presented in this paper for making the practical on-line knowledge maintenance. The empirical experiment was established. The accuracy results acquired from the hybrid between the revolutionary RSVD technique and the proposed linear regression analysis with incremental update method depicted the highest accuracy, especially in users-items sparse situations.

Keywords: *Recommender Systems, Collaborative Filtering, Singular Value Decomposition, Incremental Update, Users-Items Sparse Problem*

1. INTRODUCTION

In recent years, large amount of people access information stored in an internet every day for searching information which they interested in. Normally, the results returned from the conventional information retrieval systems are a ton of irrelevant information. It is, therefore, this information requires an evaluation with the help of expert knowledge to make its more meaningful. Recommender systems are known as computer-based systems for dealing with the problem of information overload by providing personalized recommendations of items (such as services or goods) to their users [1, 2, 3]. The recommended results of these systems are relied on the predicted rating scores of the items that have not been seen by their users.

Collaborative filtering (CF) is the most successful approach for building such a personalized recommendation [1, 2, 3, 5]. It uses the known historical rated scores from of other users for evaluated the estimated rating scores for unknown items to its current users. Various techniques have been developed to address their recommended solutions based on CF concept.

Journal of Theoretical and Applied Information Technology
20th October 2015. Vol. 80 No.2
© 2005 - 2015 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

These techniques can be categorized into two categories [1, 2]: memory-based CF techniques and model-based CF techniques.

Memory-based CF techniques normally make their prediction relied on similarities measurements between users and/or items. Many techniques have been used to compute the similarity such as correlation-based approach and cosine-based approach [2, 6, 7]. The historical database represents relationships between users and items with the corresponding rating scores are corrected in form of users-items rating matrix. This entire rating matrix is fitted into the calculation area (or memory). The similarity calculation algorithm is called for every time their users facing with their new situations. The main advantage of memory-based CF approach is an easy of implementation. In addition, a new knowledge about users' rating scores is combined into the users-items matrix and it is available for using in the next evaluation. Hence, the systems' knowledge is refreshing all the time. However, this approach is not reasonable for online system. Because the rating prediction process for producing a single recommendation solution must be performed over all elements collected in users-items rating matrix [2], this computation consumes a significant processing time. Besides, the whole users-items rating matrix must be preserved in the memory all the time.

Model-based CF techniques relied on the concept of pre-constructed rating prediction model, which represents the behavior of the users assembled from users-items rating matrix, according to off-line learning algorithms. Several learning algorithms have been studied for establishing the model, e.g., clustering [4, 7], graph theory, Bayesian classifiers, artificial neural networks, SVD-base techniques [8, 9, 11]. The model parameters are figured out from the primitive users-items rating matrix at the learning time. The overall size of these model parameters is smaller than the entire users-items rating matrix. When the recommendation is asked for, these model parameters are combined in order to estimate the rating score for new face of item of the user. This rating prediction process is faster than memory-based CF techniques, because it performs over the less volume elements. In addition, the prediction results acquired from model-based CF techniques produce high quality recommendations than the memory-based CF algorithms. However, it is constrained by the time consuming for knowledge refreshing process. Generally, the new rated scores made after the learning time are not combined until the next off-line learning process. Because of the learning process for construction of model requires a considerable amount of time. This problem may cause the less accuracy recommendation results, when the number of ratings that need to be predicted are very large compared with the number of rating scores encountered at the learning process. This problem is named as users-items sparse problem. [2].

To alleviate with the above problems, SVD-based techniques, which are outstanding techniques in model-based CF techniques [3], have been studied. In addition, the linear regression technique with the proposed linear time incremental update algorithm is introduced in this paper. This knowledge refreshing process is relied on the memory-based CF techniques which simple enough to incorporated into conventional SVD-based techniques for practical using in online recommendation environments. Thus, both of memory-based CF techniques and model-based CF techniques are combined for taking some advantages of both approaches. Moreover, the accuracy for the users-items sparse problem is improved because of the complement of recommendation systems knowledge.

The remaining of this paper is organized as follows. Section 2 explains about related works and Section 3 presents some proposed ideas. Section 4 exhibits the experimental results along with discussions. In addition, Section 5 accounts for conclusions and some final thoughts.

2. RELATED WORKS

The main concept of CF is to collect the database representing historical relationships between users and items with the corresponding rating scores in form of users-items rating matrix, hereafter referred to as matrix r. A size of matrix r is an $m \times n$ matrix, where m be the number of users and n be the number of items. An element $r_{u,i}$ is a rating score of user u over item i. The value of $r_{u,i}$ is said to be a rated element, if a user u has been rated on an item i.

One of the successful CF techniques [3] are SVD-based techniques which are inspired by an effective matrix factorization technique rooted from natural language processing research domain [3, 8, 9, 10, 11, 12]. Several SVD-based techniques such as the RSVD [9], the improved RSVD model [10], and a revolution SVD [12] are investigated.

According to a learning process, matrix r is factorized to smaller matrices. The re-composition of these factorized matrices acquires

Journal of Theoretical and Applied Information Technology
20th October 2015. Vol. 80 No.2
© 2005 - 2015 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

an estimated users-items rating, hereafter referred to as matrix $\hat{r}$. The learning process iterates until the values of the estimated rated elements are almost equal to rated elements, or $\hat{r} \approx r$.

3.1 Regular SVD technique (RSVD)

In the RSVD technique [9], each item was represented by a set of aspects and each user was represented by a set of values indicating their preference for various aspects of the items. A baseline estimation for an unknown rating ($\hat{r}_{u,i}$) was assigned as

$$\hat{r}_{u,i} = p_u^T q_i \quad (1)$$

where p_u and q_i were model parameters that factorized from matrix r. The size of p and q were $m \times k$ and $n \times k$, respectively. A variable k was an appropriate size of factorizing matrices acquired from learning process. The values of p and q were specified according to the learning process by minimizing the least square errors function $e_{u_i}^2 = (r_{u,i} - \hat{r}_{u,i})^2 = (r_{u,i} - p_u^T q_i)^2$. In addition, the regular term $\lambda(\|p_u\|^2 + \|q_i\|^2)$ was added to avoid over-fitting by penalizing the magnitudes of the parameters. The optimization function for this learning process was given as

$$\min_{p_*,q_*} \sum_{(u,j)\in r} (r_{u,j} - p_u^T q_i)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2)$$

An iterative gradient descent method on the value of training loss ($l_{u,i}$) was taken into account in RSVD learning process for estimating the model parameters. The value of $l_{u,i}$ could be computed as $l_{u,i} = (e_{u_i}^2 - \lambda \cdot p_u^T p_u + \lambda \cdot q_i^T q_i)/2$. The gradient of $l_{u,i}$ was given as

$$\frac{\partial}{\partial p_u} l_{u,i} = -e_{u,i} q_i + \lambda \cdot p_u$$
$$\frac{\partial}{\partial q_i} l_{u,i} = -e_{u,i} p_u + \lambda \cdot q_i$$

The learning process iterated for modifies the parameters by a magnitude proportional to a constant γ in the opposite direction of the gradient, yielding

- $q_i += \gamma(e_{u,i} \cdot p_u - \lambda \cdot q_i)$
- $p_u += \gamma(e_{u,i} \cdot q_i - \lambda \cdot p_u)$

The values of γ and λ were set to 0.005 and 0.02, respectively [8, 9, 11]. The learning process repeated for setting the model parameters until the terminal conditions were reached. There was no explicit termination condition, but the algorithm normally loops until the error rate ($e_{u,i}$) was close to zero. The time complexity of this learning algorithm was $O(mnk)$ or approximately $O(n^3)$.

3.2 The improved RSVD technique

The improved RSVD technique [11] extends the prediction accuracy of RSVD technique by adding some bias parameters to RSVD model. The parameters to indicate the observed bias of both a user u, denoted by b_u, and an observed bias over a item i, denoted by c_i. The baseline estimation was established as

$$\hat{r}_{u,i} = \mu + b_u + c_i + p_u^T q_i \quad (2)$$

where μ was the average of all known rating values and b_u, c_i, p_u and q_i were model parameters whose value was specified according to the learning process.

$$\min_{b_*,c_*,p_*,q_*} \sum_{(u,i)\in r} (r_{u,i} - \mu - b_u - c_i - p_u^T q_i)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2 + b_u^2 + c_i^2)$$

A training loss of this learning process was $l_{u,i} = (e_{u_i}^2 - \lambda p_u^T p_u + \lambda q_i^T q_i + \lambda b_u^2 + \lambda c_i^2)/2$. Thus, the gradient of this $l_{u,i}$ could be computed as

$$\frac{\partial}{\partial p_u} l_{u,i} = -e_{u,i} q_i + \lambda \cdot p_u$$
$$\frac{\partial}{\partial q_i} l_{u,i} = -e_{u,i} p_u + \lambda \cdot q_i$$
$$\frac{\partial}{\partial b_u} l_{u,i} = -e_{u,i} + \lambda \cdot b_u$$
$$\frac{\partial}{\partial c_i} l_{u,i} = -e_{u,i} + \lambda \cdot c_i$$

The learning process also performs as RSVD learning technique for modifies the parameters by a magnitude proportional to a constant γ in the opposite direction of the gradient, yielding

- $q_i += \gamma(e_{u,i} \cdot p_u - \lambda \cdot q_i)$
- $p_u += \gamma(e_{u,i} \cdot q_i - \lambda \cdot p_u)$
- $b_u += \gamma(e_{u,i} \cdot -\lambda \cdot b_u)$
- $c_i += \gamma(e_{u,i} - \lambda \cdot c_i)$

The learning process also iterated until the error rate was closed to zero. The time complexity of this learning algorithm was approximately $O(n^3)$.

Journal of Theoretical and Applied Information Technology
20th October 2015. Vol. 80 No.2
© 2005 - 2015 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In addition, the results returned from the improved RSVD technique were higher accuracy than the RSVD technique.

3.3 The revolutionary RSVD technique

A user-item rating matrix was decomposed [12] in two aspects, namely, the knowledge with respect to the user point of view ($K^{\{U\}}$) and the knowledge gathered from other users' point of view ($K^{\{I\}}$). Finally, all constructive knowledge was encapsulated and ported to the client agent as prior knowledge. An estimation equation of rating score prediction was formulated as

$$\hat{r}_{u,i} = \text{average}(K^{\{U\}}, K^{\{I\}}) \quad (3)$$

Thus, the variables in Equation (1) could be expressed to accommodate as

$$K^{\{U\}} = \bar{r}_u + \Delta^{\{U\}} \quad (4)$$
$$K^{\{I\}} = \bar{r}_i + \Delta^{\{I\}} \quad (5)$$

where $\bar{r}_u$ and $\bar{r}_i$ were a average rating score of made by a user u and a average rating score of made on an item i acquired from a learning process, respectively. The $\Delta^{\{A\}}$ and $\Delta^{\{I\}}$ denote knowledge with respect to the user's point of view and knowledge gathering from other users' point of view, respectively.

$$\Delta^{\{A\}} = \bar{r}_u + b_u + p_u^T q_i$$
$$\Delta^{\{I\}} = \bar{r}_i + b_i + s_u^T t_i$$

The model parameters with respect to the user's point of view ($\Delta^{\{A\}}$), i.e. b_u, p_u and q_i, were specified according to the learning process by utilizing a minimized squared error function as

$$\min_{b_u, p_*, q_*} \sum_{(u,i)\in r} (r_{u,i} - \bar{r}_u - p_u^T q_i)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2 + b_u^2)$$

A training lose of this learning process was $l_{u,i} = (e_{u_i}^2 - \lambda p_u^T p_u + \lambda q_i^T q_i + \lambda b_u^2)/2$. Thus, the gradient of this $l_{u,i}$ could be computed as

$$\frac{\partial}{\partial p_u} l_{u,i} = -e_{u,i} q_i + \lambda \cdot p_u$$
$$\frac{\partial}{\partial q_i} l_{u,i} = -e_{u,i} p_u + \lambda \cdot q_i$$
$$\frac{\partial}{\partial b_u} l_{u,i} = -e_{u,i} + \lambda \cdot b_u$$

In addition, the other model parameters with respect to knowledge gathering from other users' point of view ($\Delta^{\{I\}}$), i.e. b_i, s_u and t_i were also specified according to the learning process by utilizing a minimized squared error function as

$$\min_{b_i, s_*, t_*} \sum_{(u,i)\in r} (r_{u,i} - \bar{r}_i - s_u^T t_i)^2 + \lambda(\|s_u\|^2 + \|t_i\|^2 + b_i^2)$$

A training loss of this learning process was $l_{u,i} = (e_{u_i}^2 - \lambda s_u^T s_u + \lambda t i^T t i + \lambda b_i^2)/2$. Thus, the gradient of this $l_{u,i}$ could be computed as

$$\frac{\partial}{\partial s_u} l_{u,i} = -e_{u,i} t_i + \lambda \cdot s_u$$
$$\frac{\partial}{\partial t_i} l_{u,i} = -e_{u,i} s_u + \lambda \cdot t_i$$
$$\frac{\partial}{\partial b_i} l_{u,i} = -e_{u,i} + \lambda \cdot b_i$$

In contrast with the RSVD and the improved RSVD, the revolution RSVD learning algorithm states the termination condition based on the value of Mean Average Error (MAE), which was the average absolute errors corresponding to actual ratings-prediction pairs, along with their average [2].

$$MAE = \frac{\sum_{u=1}^{m} \sum_{i=1}^{n} (r_{u,i} - \hat{r}_{u,i})}{m \times n} \quad (6)$$

The lower MAE value indicated the more accuracy of the prediction results. The appropriate value of the termination condition was established in [12] as the iteration must be performed until MAE<= 0.50. This algorithm returned the highest accuracy results comparing with the RSVD and the improved RSVD. The time complexity for its eager learning process was also $O(n^3)$. In addition, the incremental process was also proposed. However, the accuracy in case of the sparse situations needed some improvements for achieving the higher accuracy values.

3.4 Regression-based technique

Regression analysis was a technique for finding the relationship between a dependent variable and one or more independent variables. In [13], the relationship between two rated items was studied as

$$f_{i,j}(r_{u,i}) = r_{u,i} \times \alpha_{i,j} + \beta_{i,j} \quad (7)$$

where $r_{u,i}$ was stated as an independent variable. There were two regression analysis constants values, i.e., $\alpha_{i,j}$ and $\beta_{i,j}$, which were assigned their values by using the minimization of the mean squared error. Besides, an estimation equation for rating score prediction was formulated as

Journal of Theoretical and Applied Information Technology
20th October 2015. Vol. 80 No.2
© 2005 - 2015 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\hat{r}_{u,i} = \sum_{j=1}^{n} w_{i,j} \times f_{i,j}(r_{u,i}) \quad (8)$$

where $w_{i,j}$ was the weight of an item i depended on an j.

Even though, the simple linear regression technique ordinarily was classified into model-based CF approach in the literatures [2], it was simple enough for implementing it as if a memory-based CF approaches. Because the time complexity of this learning algorithm was approximately $O(n^2)$ which was less than the time complexity for implementing SVD-based techniques.

3. PROPOSED APPROACH

In this paper, we propose the hybrid technique which is the combination of SVD-based techniques and the linear regression analysis technique. An estimation equation of rating score prediction proposed in this paper is formulated as

$$\hat{r}_{u,i} = \text{average}(\Gamma_{u,i}, \Omega_{u,i}) \quad (9)$$

where

- $\Gamma_{u,i}$ be an estimated rating score evaluated from SVD-based technique described in the literature reviews
- $\Omega_{u,i}$ be an estimated rating score evaluated from the proposed linear regression technique

3.1 A generalized of SVD-based techniques for rating scores evaluation

In this paper several SVD-based techniques have been studied, the generalized algorithm for the learning process to calculate all model parameters is summarized as following.

- *Initial model parameters (such as p_u and q_i, etc.)*
- *DO*
  - $FOR\ u = 1\ to\ m$
    - i. $e_{u,i} = r_{u,i} - \hat{r}_{u,i}$
    - ii. $FOR\ k = 1\ to\ z$
      - *Evaluate model parameters*
    - iii. $END\ FOR$
  - $END\ FOR$
  - $MAE = \sum_{i=1}^{m} r_{u,i} - \hat{r}_{u,i}$
  - *Calculate* $\hat{r}_{u,i}$
- *LOOP UNTIL* $MAE <= LimitError$

$\approx O(n^3)$

As mentioned in the literatures [12], the SVD-based techniques are categorized into model-based CF techniques. All model parameters have been specified their values according to the off-line learning process. The learning algorithm normally evaluates its model parameters by applying a magnitude proportional to a constant γ in the opposite direction of the gradient. Accroding to [18], the values of γ, λ and *LimitError* are set to 0.005, 0.02 and 0.5, respectively. The time complexity of this algorithm costs $O(n^3)$.

3.2 A linear regression analysis algorithm for rating scores evaluation

In order to present the proposed linear regression analysis technique for dealing with CF recommended rating scores, some terminologies are defined for subsequently use as following.

For any user u, historical rated values collected in matrix r are transformed into matrix x. x_j be a rated value of an user u made on an item i at decision making round j^{th}. In addition, matrix y is created for collecting average rating scores of items which have been rated by a user u, where y_j be an average rating score of an item i that users u sort out the solution at rounds j^{th}

First of all, the pre-processing is taken into account for creating a matrix x and a matrix y. The values of decision making rounds are sorted corresponding to the values of item index i. For illustrating the meaning of these terminologies, an example of user-item rating matrix is shown as Figure 1.

User/item	i=1	...	i=24	i=25	i=26	...	i=n
u=1							
u=2							
u=1							
...							
U	2		5	3	4		
...							
u=m							
$\bar{r}_i$	2.4		3.5	4.0	3.2		

Figure 1: An example of user-item rating matrix

Journal of Theoretical and Applied Information Technology
20th October 2015. Vol. 80 No.2
© 2005 - 2015 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Suppose user u has been rated on items 1, 24, 25 and 26 with the rated values 2, 5, 3 and 4, respectively. The historical rated values in matrix r are converted to a matrix x. Besides, the corresponding average rating scores of those rated elements are assigned to a matrix y. The transformation of this example is demonstrated in Figure 2.

j	an index of the corresponding rated item i	x_j	y_j
$j=1$	1	2	2.4
$j=2$	24	5	3.5
$j=3$	25	3	4.0
$j=4$	26	4	3.2

Figure 2: An example of matrix x and a matrix y of a user U

The linear regression analysis for estimating rating score is proposed as

$$\Omega_{u,i} = \alpha + \beta \bar{r}_i \qquad (10)$$

where $\bar{r}_i$ is the average rating score of an item i. β and α are linear regression variables which can be calculated as follow:

$$\beta = \frac{\sum_{j=1}^{z}(x_j - \bar{x})(y_j - \bar{y})}{\sum_{j=1}^{z}(x_j - \bar{x_j})^2} \qquad (11)$$

$$\alpha = \bar{y} - \beta \bar{x} \qquad (12)$$

A simple linear regression analysis algorithm for a user U can be stated as follow:

FOR $u = 1$ *to* m
/*Average values calculation */
i. *Initial* $sumx = 0$ *and* $sumy = 0$
ii. *FOR* $j = 1$ *to* z
- $sumx+ = x_j$
- $sumy+ = y_j$

iii. *END FOR*
iv. $\bar{x} = sumx/z$
v. $\bar{y} = sumy/z$

/* Regression parameters calculation*/
i. *Initial* $sum1 = 0$ *and* $sum2 = 0$
ii. *FOR* $j = 1$ *to* z
- $sum1+ = (x_j - \bar{x}) \times (y_j - \bar{y})$
- $sum2+ = (x_j - \bar{x})^2$

iii. *END FOR*
iv. $\beta = \frac{sum1}{sum2}$
v. $\alpha = \bar{y} - \beta \bar{x}$

END FOR

$\approx O(n^2)$

The time complexity of this simple linear regression analysis for one user takes $O(n)$. Since the user of recommender system are m users, this overall calculation process approximately costs $O(n^2)$. Thus, this algorithm can be classified into polynomial time algorithm.

As mentioned in the earlier section, the memory-based CF techniques refreshing their knowledge by re-calculating their parameters every time for making their evaluation. However, re-calculating regression parameters via simple linear regression analysis by an Equation (11) and an Equation (12) for one user takes linear time computation. The main iterations for refreshment mainly cause from average values evaluation part and an β evaluation. In this paper, we try to reduce the processing time of these refreshment processes by re-formulate an average evaluation equation and an Equation (11) .

3.2.1 A constant time refreshment for an average value calculation

In this paper, there are many equations dealing with their variables in form of average values. An average can be generally computed as

$$\bar{a} = \frac{\sum_{i=1}^{d} a_i}{d} \qquad (13)$$

where $a = \{a_1, a_2, \ldots, a_d\}$ and $|a| = d$. An average value of this general set is denoted as $\bar{a}$.

When the new set of elements $(a' = a_{d+1}, a_{d+2}, \ldots, a_{d+d'},\ |a'| = d')$ is added into set a, the conventional refreshing process takes $O(n)$ computation time.

However, the complexity of an average calculation process can be reduced as

$$\bar{a}^{new} = \frac{d\bar{a}^{old} + (a_{d+1}, a_{d+2}, \ldots, a_{d+d'})}{d + d'} \qquad (14)$$

where $\bar{a}^{old}$ and $\bar{a}^{new}$ be an average value of set u with size d and an average value of set a with size $d + d'$. This refreshment process costs $O(c)$, which is a constant time computation.

3.2.2 A constant time refreshment for regression parameters calculation

The main iterations for refreshment regression parameters causes from an β evaluation. Thus, an Equation (8) is reformulated as followings.

Journal of Theoretical and Applied Information Technology
20th October 2015. Vol. 80 No.2
© 2005 - 2015 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$
\begin{aligned}
\beta &= \frac{\sum_{j=1}^{z}(x_j-\bar{x})(y_j-\bar{y})}{\sum_{j=1}^{z}(x_j-\bar{x_j})^2} \\
&= \frac{\sum_{j=1}^{z}(x_jy_j - x_j\bar{y} - y_j\bar{x} + \bar{x}\bar{y})}{\sum_{j=1}^{z}(x_jx_j - x_j\bar{x} - x_j\bar{x} + \bar{x}\bar{x})} \\
&= \frac{\sum_{j=1}^{z}(x_jy_j - x_j\bar{y} - y_j\bar{x}) + z\bar{x}\bar{y}}{\sum_{j=1}^{z}(x_jx_j - x_j\bar{x} - x_j\bar{x}) + z\bar{x}\bar{x}} \\
&= \frac{\sum_{j=1}^{z}(x_jy_j - x_j\bar{y} - y_j\bar{x}) + z\frac{\sum_{j=1}^{z}x_j}{z}\bar{y}}{\sum_{j=1}^{z}(x_jx_j - x_j\bar{x} - x_j\bar{x}) + z\frac{\sum_{j=1}^{z}x_j}{z}\bar{x}} \\
&= \frac{\sum_{j=1}^{z}(x_jy_j - x_j\bar{y} - y_j\bar{x}) + \sum_{j=1}^{z}x_j\bar{y}}{\sum_{j=1}^{z}(x_jx_j - x_j\bar{x} - x_j\bar{x}) + \sum_{j=1}^{z}x_j\bar{x}} \\
&= \frac{\sum_{j=1}^{z}(x_jy_j - x_j\bar{y} - y_j\bar{x} + x_j\bar{y})}{\sum_{j=1}^{z}(x_jx_j - x_j\bar{x} - x_j\bar{x} + x_j\bar{x}} \\
&= \frac{\sum_{j=1}^{z}(x_jy_j - \bar{x}y_j)}{\sum_{j=1}^{z}(x_jx_j - \bar{x}x_j)} \\
&= \frac{\sum_{j=1}^{z}x_jy_j - \bar{x}\sum_{j=1}^{z}y_j}{\sum_{j=1}^{z}x_jx_j - \bar{x}\sum_{j=1}^{z}x_j}
\end{aligned}
$$

Thus, an Equation (11) is changed to

$$\beta = \frac{\sum_{j=1}^{z}x_jy_j - \bar{x}\sum_{j=1}^{z}y_j}{\sum_{j=1}^{z}x_jx_j - \bar{x}\sum_{j=1}^{z}x_j} \qquad (15)$$

Then, let assign cached information collected in memory which are four $m \times 1$ matrices. These matrices are $sumy$, $sumxx$ and $sumxy$.

Let $sumx$ *be* $\sum_{j=1}^{z}x_j$, $sumy$ *be* $\sum_{j=1}^{z}y_j$ $sumxy$ *be* $\sum_{j=1}^{z}x_jy_j$ *and* $sumxx$ *be* $\sum_{j=1}^{z}x_jx_j$.

It is, therefore, the refreshment process for regression parameters can be performed by means of incremental update. The time complexity of this process is reduced to $O(c)$, which is also a constant time computation.

3.2.3 A proposed linear time refreshment for regression parameters calculation

A linear regression analysis refreshment algorithm, which relied on incremental update and collecting some cached information, for a user u can be described as follows:

/* Average values refreshment process*/

1) $\bar{x} = \frac{z\bar{x} + x_{z+1}}{z+1}$
2) $\bar{y} = \frac{z\bar{y} + y_{z+1}}{z+1}$

} $O(c_1)$

/* Regression parameters refreshment process*/

1) $sumx+ = x_j$
2) $sumy+ = y_j$
3) $sumxy+ = x_j \times y_j$
4) $sumxx+ = x_j \times x_j$
5) $\beta = \frac{sumxy - \bar{x} \times sumy}{sumxs - \bar{x} \times sumx}$
6) $\alpha = \bar{y} - \beta\bar{x}$

} $O(c_2)$

The time complexity of the refreshment for one user takes $O(c)$. Since the user of recommender system are m users, this overall calculation process costs $O(m)$. Thus, this algorithm can be classified into linear time algorithm.

4. RESULTS AND DISCUSSION

In this study, we used MovieLens [14] datasets which contained data corresponding to movie ratings collected from the MovieLens recommender system (http://movielens.umn.edu). We randomly divided the dataset into the training subsets and testing subsets. The percentage of the training sets were varied from 10%-90%. Thus, there were nine situations which were taken into account in this empirical experiment. The highlighted of this experiment was on the low percentage of training subsets which were classified in the users-items sparse problem.

We applied a proposed linear regression analysis with an incremental update method to several SVD-based techniques, which was stated in [2] as the successful class of CF. The different techniques found in the literatures, i.e., the RSVD, the improved RSVD, and the revolutionary RSVD, were selected as a benchmark techniques. We used a MAE value which was presented in an Equation (6) as an accuracy evaluation measurement. The lower MAE value implied the higher accuracy acquired from the recommender system. The results of the experiments were shown as following. The range of MAE values that focused on were 0.80-0.65.

Journal of Theoretical and Applied Information Technology
20th October 2015. Vol. 80 No.2
© 2005 - 2015 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

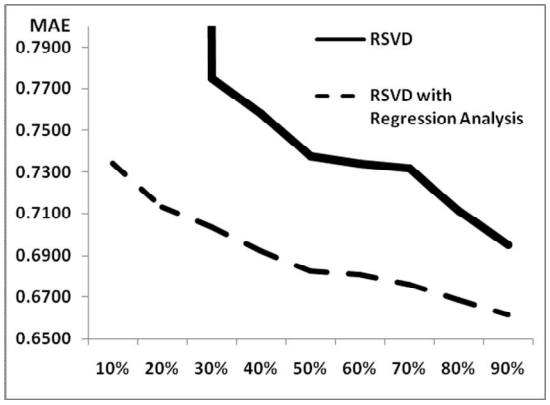

Figure 3. A comparison of accuracy between the RSVD and the RSVD with regression analysis

As shown in Figure 3, the accuracy results return from RSVD technique were higher than the focused range in case of 10% and 20% of training data sets. The MAE values acquired from both 10% and 20% of training data sets were 4.99 (as shown in Table 1), which were significantly higher that the range of MAE values that we focused on. Thus, these could be said that the primitive RSVD technique did not good enough for working under the users-items sparse problem. On the other hand, a hybrid of a linear regression analysis technique proposed in this paper and the RSVD technique reported with the higher accuracy than the primitive RSVD in every aspects. Especially, for the low percentage of training subsets, this hybrid technique produced the notably greater accuracy outcomes than the results of primitive RSVD technique.

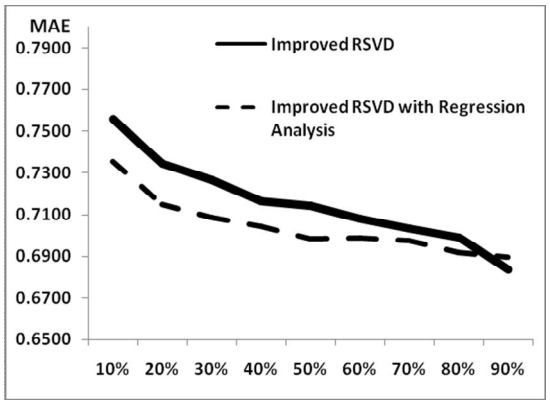

Figure 4. A comparison of accuracy between the improved RSVD and the improved RSVD with regression analysis

In addition, another experiment was conducted for testing the accuracy of rating prediction results acquired from applying a linear regression analysis technique with an incremental update method to the improved RSVD technique. As can be seen in Figure 4, the results returned from this hybrid technique were also higher quality than the results achieved from a primitive improved RSVD technique.

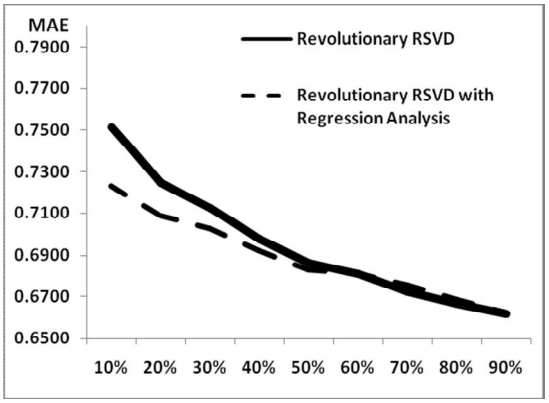

Figure 5. A comparison of accuracy between the revolutionary RSVD and a revolutionary RSVD with regression analysis

Moreover, the results in Figure 5 depicted that the rating prediction results returned from a hybrid of the revolutionary RSVD and a proposed linear regression analysis with an incremental update method presented the lowest MAE values in the users-items sparse problem (in case of 10%-40% of training data sets).

Journal of Theoretical and Applied Information Technology
20th October 2015. Vol. 80 No.2
© 2005 - 2015 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1. An evaluation of MAE according to different size of training/testing sets

Technique \ Percent of training set	10%	20%	30%	40%	50%	60%	70%	80%	90%
RSVD	4.9991	4.9998	0.7748	0.7580	0.7376	0.7338	0.7317	0.7111	0.6949
Improved RSVD	0.7556	0.7342	0.7263	0.7165	0.7141	0.7081	0.7034	0.6988	0.6836
Revolutionary RSVD	0.7514	0.7246	0.7126	0.6974	0.6863	0.6807	**0.6720**	**0.6664**	**0.6615**
RSVD with Regression Analysis	0.7340	0.7130	0.7034	0.6922	**0.6824**	**0.6805**	0.6757	0.6682	0.6616
Improved RSVD with Regression Analysis	0.7355	0.7150	0.7089	0.7041	0.6984	0.6987	0.6972	0.6917	0.6895
Revolutionary RSVD with Regression Analysis	**0.7233**	**0.7087**	**0.7027**	**0.6919**	0.6828	0.6812	0.6754	0.6686	0.6624

Table 1 was shown the numerical values of MAE according to different size of training/testing sets of every experiment performed in this section. The accuracy of the results acquired from the hybrid of the revolutionary RSVD technique and a regression analysis technique with an incremental update method were prominent accuracy than others in case of the low percentage of training subsets. However, for the normal situation in which the size of training sets were almost equal or bigger than the test sets, this hybrid method also produced the comparable accuracy results to others. Thus, it could be concluded that the hybrid of the revolutionary RSVD technique and a regression analysis technique with an incremental update method was better enough to deal with both normal situation and the users-items sparse problem.

5. CONCLUSIONS

In order to address the users-items sparse problem which are the problem in CF approach, this paper proposes a technique which combines techniques from both memory-based CF and model-based CF approach to take advantages of both approaches. SVD-based techniques, which are successful techniques in memory-based CF approach, are investigated to enhance their capability by adding knowledge refreshing function. The simple linear regression technique with the proposed knowledge refreshing technique is incorporated into SVD-based techniques. From the empirical experiment, it can be concluded that the hybrid of the revolutionary RSVD technique and a regression analysis technique with an incremental update method returns the accuracy results in every aspect of training subsets. Besides, this proposed technique can reduce the time complexity for a refreshment process of overall recommendations into linear time complexity by taking an advantage of incremental update process.

Furthermore, we would like to evaluate our proposed technique on dataset of domains other than movies, such as BookCrossing dataset. In order to confirm that it can be applied in other domains.

Acknowledgements

This research has been supported by Faculty of Science Research Fund, Chiang Mai University.

REFERENCES

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

[2] G. Adomavicius and A. Tuzhilin. 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(6):734-749,2005.

[3] F. Cacheda, V. Carneiro, D. Fern´andez, and V. Formoso. Comparison of collaborative filtering algorithms: Limitations

© 2005 - 2015 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web*, 5(1):2:1-2:33, 2011.

[4] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:1-20, 2009.T. George and S. Merugu. A scalable collaborative filtering framework based on co-clustering. In *Proceedings of the 5th IEEE International Conference on Data Mining*, pages 625-628, 2005J. Carbo and J. M. Molina. Agent based collaborative filtering based on fuzzy recommendations. *International Journal of Web Engineering and Technology*, 1:414-426, 2004.

[5] F. Hern´andez del Olmo and E. Gaudioso. Evaluation of recommender systems: A new approach. *IEEE Transactions on Knowledge and Data Engineering*, 35(3):790.804, 2008.

[6] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 1999 Conference on Research and Development in Information Retrieval*, pages 230-237, 1999.

[7] E. Vozalis and K. G. Margaritis. Analysis of recommender systems' algorithms. In *Proceedings of the 6th Hellenic European Conference on Computer Mathematics and Its Applications*, pages 1-14, 2003.

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

[9] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the* 5th *International Conference on Computer and Information Technology*, pages 426-434, 2008.

[10] Y. Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining*, pages 447-455, 2009.

[11] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of the KDD Cup Workshop at the 13th ACM International Conference on Knowledge Discovery and Data Mining*, pages 39-42, 2007.

[12] D. Praserttitipong and P. Sophatsathit. An Agent Model for Information Filtering using Revolutionary RSVD Technique. *Chiang Mai Journal of Science*, Vol.41 No.5/2 (OCTOBER 2014), page 1429-1438, 2014.

[13] S. Vucetic and Z. Obradovic . Collaborative Filtering Using a Regression-Based Approach. *Knoledge and Information Systems*, Vol.41 No.5/2 (OCTOBER 2014), page 1429-1438, 2014.

[14] GroupLens Research. MovieLens data sets. http://grouplens.org/datasets/movielens/, [23 April 2015].