International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **3(B)**, March **2012**

 ISSN 1349-4198
pp. 2157–2172

IMPROVING MISSING-VALUE ESTIMATION IN MICROARRAY DATA WITH COLLABORATIVE FILTERING BASED ON ROUGH-SET THEORY

Bo-Wen Wang and Vincent S. Tseng*

Department of Computer Science and Information Engineering
National Cheng Kung University
No. 1, University Road, Tainan City 701, Taiwan
bwwang@mail.stut.edu.tw; *Corresponding author: tsengsm@mail.ncku.edu.tw

Received February 2011; revised June 2011

Abstract. *Data mining techniques have been used to extract useful knowledge from DNA microarray gene expression data for discovering the relations between novel diseases and their related genes. However, DNA microarray gene expression data often contain missing values that must be dealt with to prevent them from significantly affecting analysis results. Hence, a number of missing-value imputation approaches have been proposed. In this paper, an intelligent imputation approach named the CFBRST (Collaborative Filtering Based on Rough-Set Theory) method is proposed to impute missing values more accurately than currently done by existing approaches. Experimental results on real microarray gene expression datasets reveal that the proposed approach can effectively improve missing-value estimation. The collaborative filtering (CF) approach is often used in recommender systems due to its excellent performance. The proposed CFRBS method is based on the CF method and rough-set theory. The CFBRST method is compared with the k-nearest neighbor (k-NN) imputation algorithm. Experimental results show that the CFBRST method has better accuracy than that of a k-NN approach for yeast cDNA microarray datasets, especially when the percentage of missing values is high.*

Keywords: Collaborative filtering, Rough-set theory, Gene expression data, Missing values, Imputation

1. **Introduction.** In recent years, DNA microarray gene expression data have been widely used in numerous studies to determine the relation between novel diseases and their related genes. These studies proposed some effective techniques for extracting useful knowledge from thousands of gene expression levels simultaneously under various conditions [11,12,24].

In the field of bio-informatics, DNA microarray gene expression data analysis is used for applications such as drug discovery, protein sequencing, cancer classification [13], and the identification of genes relevant to a certain diagnosis or therapy. However, DNA microarray gene expression data often contain missing values for various reasons, including image corruption [29], hybridization error, dust, and insufficient resolution. Unfortunately, the missing values significantly affect gene expression data analysis results. A lot of information is lost when genes with missing values are ignored or directly deleted. For example, it has been shown that missing values may seriously disturb or even prevent subsequent data analysis [1]. Furthermore, high-quality of DNA microarray gene expression data analysis that heavily relies on the quantity of missing values can actually provide researchers with valuable information.

To deal with such problems, many imputation algorithms have been developed to recover the missing values before the actual data analysis is conducted. Imputation algorithms include the SVDimpute method [36], the k-nearest neighbor (k-NN) method [36], the local least-squares (LLS) approach [18], the Bayesian approach [29], the collateral missing value imputation approach [33], and the Gene Ontology k-nearest neighbor (GOKNN) method [8,10,25,37]. Although these imputation algorithms deal with missing values well when the required condition is satisfied, they also have several limitations. kNNimpute performs best on non-time series data or noisy time series data, whereas SVDimpute works well on time series data with low noise levels and with a strong global correlation structure. LLSimpute and GOKNN have the best performance when strong local correlation exists in the data.

The collaborative filtering (*CF*) approach is often used in recommender systems. This approach provides recommendations based on the similarity of preferences between users. The advantages of the *CF* approach are that the recommendation relies on other users' experiences and that it has better accuracy than content-based [2,3,26] recommender systems. Two types of basic *CF* algorithm have been proposed. The first type is memory-based (user-based) *CF* algorithms, which provide recommendations according to the preferences between an active user and his or her top-k nearest neighbors. The GroupLen[31] system is a user-based *CF* algorithm. The second type is model-based *CF* [5,7,27,28,40] algorithms, which first train a model based on a training dataset and then provide recommendations according to this model. In addition to user-based *CF* and model-based *CF* methods, hybrid *CF* algorithms provide recommendations by combining user-based *CF* and model-based *CF* [9,30,34] or by combining the *CF*-based recommendation approach and the content-based recommendation approach [3,17,26].

Since the data format of DNA microarray datasets is similar to that of datasets used in *CF* recommender systems, the present study develops an imputation method named the *CFBRST* (Collaborative Filtering Based on Rough-Set Theory) method for imputing missing values in DNA microarray datasets. Rough sets have been shown to be very useful in various applications [15,21,23,35]. The main advantage of the proposed method is that rough-set prediction considers not only the similarities between genes but also those between conditions. By integrating the similarities between genes and those between conditions, the nearest genes and conditions can be derived simultaneously to infer more accurate missing values. With this, the proposed method further achieves higher precision by performing rough-set-based prediction. The missing values imputed in the preprocessing phase are used to infer the missing values in the prediction phase. Without imputed values, it is difficult to derive accurate missing values. The experimental results show that the *CFBRST* method has better accuracy than that of the kNNimpute algorithm in yeast cDNA microarray datasets. The remainder of this paper is organized as follows. A review of related work is given in Section 2. In Section 3, the proposed approach for imputing missing values in a DNA microarray dataset is described. Experimental evaluations of the proposed approach are presented in Section 4. Finally, conclusions and future work are stated in Section 5.

2. **Related Work.** The estimation of missing values in DNA microarray gene expression data has been studied in recent years. Several imputation algorithms have been proposed for imputing the missing values in DNA microarray gene expression data. In the k-NN method [36], for gene A with a missing value, the top k neighboring genes similar to gene A are first selected according to their expression values. The missing value of gene A can then be imputed by the weighted average of the values from the top k similar genes. The weighted average is the contribution of each gene based on the similarity between

gene A and its neighbors. The similarity between two genes is computed as the Euclidean distance based on their expression values:

$$d(g_1, g_2) = \sqrt{\sum_{k=1}^{p} (e_k - f_k)^2}$$

where $g_1 = \langle e_1, e_1, \ldots, e_p \rangle$ and $g_2 = \langle f_1, f_2, \ldots, f_p \rangle$ are two genes, $d(g_1, g_2)$ is the Euclidean distance (dissimilarity) between g_1 and g_2, and the predicated value α can be computed using:

$$\alpha = \frac{\sum_{i=1}^{k} b_i \Big/ d(A, i)}{\sum_{i=1}^{k} 1 \Big/ d(A, i)}$$

where b_i is the expression value of the j^{th} condition in the i^{th} neighbor.

Tuikkala et al. [37] proposed an imputation method that considers gene ontology (GO) information to improve missing-value estimation. The method (GOKNN) calculates the semantic similarity between two genes from their GO annotations. The semantic similarity is used as additional information of the two genes, whose similarity is computed by combining the semantic similarity with the similarity computed from expression values in the pure k-NN imputation method. The experimental results show that this method outperforms the pure k-NN imputation method especially when the percentage of missing values is high.

One of the most effective algorithms for missing-value imputation is local least-squares (LLS) imputation [18]. The LLS algorithm selects the top k nearest neighboring genes and then predicts the missing values using the least-squares method.

A number of collaborative filtering algorithms have been proposed for recommender systems. The GroupLen system is a well-known user-based *CF* algorithm. The basic idea of the system is to find the active user's nearest neighbors from the massive number of users' ratings, aggregate the ratings of the active user's nearest neighbors to predict the values of unrated items, and then recommend the top k items with the highest values to the active user. However, the user-based *CF* algorithm has a high computation time because it has to compute the similarities between the active user and all users in the dataset. To alleviate this problem, cluster-based *CF* algorithms [7,38,41] have been proposed. Cluster-based *CF* algorithms work by identifying groups of users who have similar preferences first, and then computing the similarities between the active user and all groups. The active user belongs to the group that has the highest similarity with it, then applies user-based *CF* algorithm in the group to provide recommendation. This approach is more efficient than a user-based *CF* algorithm, but its accuracy is lower.

An item-based *CF* algorithm was proposed in 2001 [32]. In contrast to a user-based *CF* algorithm, it measures items' similarities from the massive number of users' ratings, finds the nearest neighbors of items, predicts the values of the unrated items for the active user, and then recommends the top k items with highest values. In general, item-based *CF* algorithms outperform user-based *CF* algorithms and cluster-based *CF* algorithms.

Some model-based *CF* algorithms regard recommendation as behavior classification. Xu et al. [40] proposed a learning algorithm that constructs a personalized recommender system based on support vector machines (SVMs). Based on frequent itemset lattices, Nikovski et al. [28] presented an induction of compact decision trees as the optimal recommendation policy. Using a Bayesian network, Breese et al. [7] attempted to solve the problem of personalized recommendation using probabilistic formulations. Lin et

al. [22] found the overlaps of several users' tastes that match the active user's taste by utilizing the discovered user associations and article associations.

Many kinds of hybrid *CF* algorithm have been recently proposed. Chuan et al. [9] solved the recommendation problem by combining user-based *CF* regression and item-based *CF* filtering. Based on the similarity fusion of user-based and item-based *CF*, Wang et al. [39] used a probabilistic framework to exploit available data in the user-item matrix. Other hybrid *CF* algorithms [4,6,14,17,26,30] that combine a content-based recommender system and user-based/item-based *CF* outperform individual methods.

3. **Proposed Approach.** The data format of DNA microarray datasets is similar to that for datasets used in recommender systems, where genes can be regarded as users and conditions can be regarded as items. The collaborative filtering (*CF*) algorithm can then be applied to the DNA microarray datasets to accurately estimate missing values. The critical problem for an excellent imputing missing-value system is how to predict the missing values correctly. The proposed method approximates an optimal solution for predicting the unknown values by combining user-based *CF* and rough-set theory.

3.1. **Overview of the proposed approach.** The proposed *CFBRST* approach can be characterized by considering the correlation of genes in the DNA microarray data and a rough-set-based prediction framework to predict the missing values. This method can predict good missing values to facilitate subsequent DNA microarray gene expression data analysis. The framework of the *CFBRST* approach, shown in Figure 1, can be divided into the following two stages.

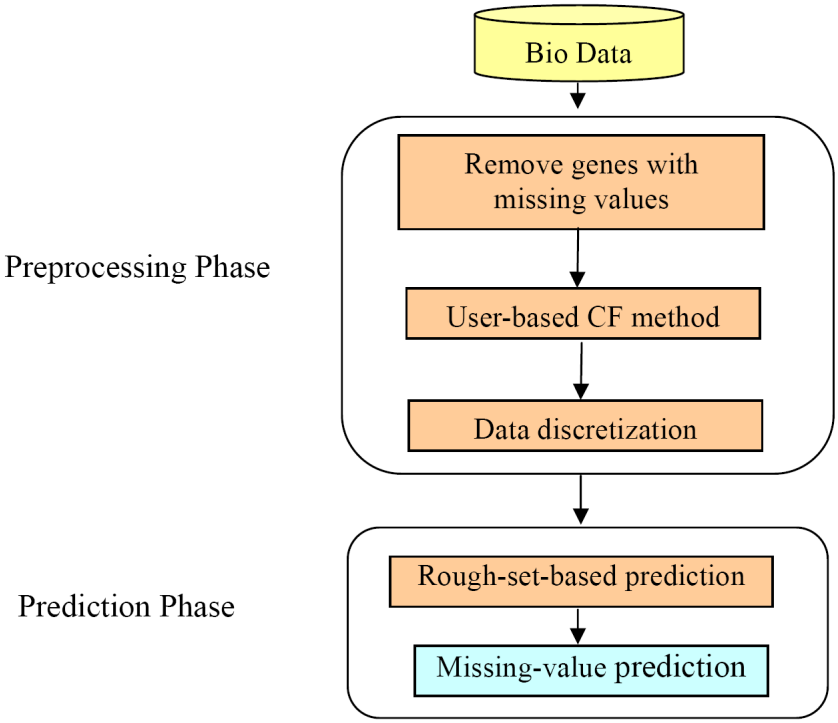


FIGURE 1. Framework of the proposed approach

Preprocessing Stage: To predict the missing values using the rough-set-based method, all the data in the dataset must be known except the predicted missing values and the data type in a dataset must be categorical. The user-based *CF* method is first used to fill the missing values. Then, the numerical data is transformed into categorical data.

Prediction Stage: After the preprocessing stage, the unknown values for the dataset can be imputed using the user-based *CF* method and the numerical data are transformed into categorical data. The rough-set-based prediction method is used to impute the unknown values for the items of the matrix.

3.2. **Preprocessing stage.** A lot of genes have missing values in a gene expression dataset. Thus, rows (genes) with missing values are first removed to make the dataset a complete matrix. Then, some of the data are deleted at random to produce a testing dataset with missing values. The user-based *CF* method is applied to the testing dataset and to estimate the missing values that were deleted to produce a complete matrix.

There are two main steps in user-based *CF* algorithms, namely similarity computation and prediction computation.

Similarity Computation. In this paper, the Pearson correlation coefficient is used to compute the similarity between two genes i and j.

$$sim(i,j) = \frac{\sum_{c\in C}(y_{i,c} - \overline{y_i})(y_{j,c} - \overline{y_j})}{\sqrt{\sum_{c\in C}(y_{i,c} - \overline{y_i})^2}\sqrt{\sum_{c\in C}(y_{j,c} - \overline{y_j})^2}} \tag{1}$$

here, $y_{i,c}$ denotes the value of gene i on condition c, $\overline{y_i}$ is the average value of gene i, and C is the set with known values of both genes i and j. Suppose that gene i is the gene with missing values to be imputed. The similarities between gene i and the other genes in the dataset are computed, and then the top k genes that have highest similarities with gene i are selected. After the top k nearest neighbors of gene i have been chosen, the missing values of the gene i are predicted according to the expression value of the selected top k nearest neighbors of gene i.

Prediction Computation. The missing values of gene i can be predicted according to the expression value of the selected top k nearest neighbors of the gene i. The missing values of gene i can be predicted using:

$$p_{i,c} = \overline{y_i} + \frac{\sum_{j\in J} sim(i,j) \times (y_{j,c} - \overline{y_j})}{\sum_{j\in J} |sim(i,j)|} \tag{2}$$

where $p_{i,c}$ denotes the predicted value of gene i on condition c, $y_{j,c}$ denotes the value of gene j on condition c, $\overline{y_i}$ is the average value of gene i, and J is the set of genes similar to gene i.

Data Discretization. To use rough-set theory to increase the prediction accuracy of the missing values, the numerical dataset must be transformed into a categorical dataset. The data of the dataset must thus be discretized.

First, the number of categories the dataset should have must be decided. Then, for each gene i , the following two steps are used to discretize the data of the dataset.

(1) $\text{step}(i) = (\text{value}_{\max(i)} - \text{value}_{\min(i)})/N$

(2) $\text{value}_{\text{transformed}} = 1 + \text{Round}((\text{value}_{\text{original}} - \text{value}_{\min(i)})/\text{step}(i))$

here, $\text{value}_{\max(i)}$ and $\text{value}_{\min(i)}$ are the maximum value and the minimum value in gene i, respectively, and $\text{value}_{\text{original}}$ and $\text{value}_{\text{transformed}}$ are the original value and transformed value, respectively. N is the number of discrete levels used to discretize the original value, and $\text{step}(i)$ is the step size for gene i. After discretizing the dataset, the transformed value of the data is an integer between 1 and $N+1$. An example is shown below to demonstrate data discretization.

Example 3.1. *Suppose that the data in Table 1 is the original gene expression data. The value of N is set to 4, $step(1) = \dfrac{1.2 - (-0.4)}{4} = 0.4$, and the transformed value of $cond_1$*

in $gene_1$ *is* $1 + round\left(\frac{(0.9-(-0.4))}{0.4}\right) = 4$. *Similarly,* $step(2) = 0.3$ *and* $step(3) = 0.3$. *Table 2 shows the transformed gene expression data.*

TABLE 1. Original gene expression data

Gene ID	$cond_1$	$cond_2$	$cond_3$	$cond_4$	$cond_5$
1	0.9	T	0.5	1.2	−0.4
2	0.9	0.85	0.6	1.1	−0.1
3	0.95	1.0	−0.2	−0.2	−0.15

TABLE 2. Transformed gene expression data

Gene ID	$cond_1$	$cond_2$	$cond_3$	$cond_4$	$cond_5$
1	4	T	3	5	1
2	4	4	3	5	1
3	5	5	1	1	1

3.3. **Prediction stage.** After the preprocessing tasks, the proposed prediction approach is used. The goal of this stage is to impute the missing values accurately to improve subsequent analysis.

Rough-Set-Based Prediction. Generally speaking, a dataset in real applications can be decomposed into two subsets, namely the complete dataset and the incomplete dataset. An incomplete dataset has missing attribute values. A missing value in a dataset can significantly affect data analysis. Unfortunately, most DNA microarray gene expression datasets are incomplete. Several studies have attempted to infer the missing values by learning from objects with known values [16,19,20]. The method proposed here estimates the unknown values in a transformed gene-condition matrix with the help of rough-set theory.

As illustrated in Figure 2, the first task of rough-set-based prediction is to determine a class attribute (or condition). The class attribute is derived from the Pearson correlation coefficient between *targetcond* and the other filtered conditions (lines 2-7). Therefore, the class attribute most relevant to *targetcond* can be used to infer the unknown value $v_{i,targetcond}$. Based on rough-set theory, after the elementary set of the class attribute is generated, the available elementary subset *group* which contains the active gene (gene with missing value to be predicted) is selected from the elementary set of the class attribute (lines 8-9). Then, *targetcond* is combined with the most relevant condition into a new condition set $Ccond_k$ (line 11). Then, the algorithm partitions the genes into an elementary set of $Ccond_k$ according to the condition values (line 14). Next, the matching equivalence class set must be found. That is, if the subsets of the elementary set of *targetcond* are all contained in *group* and the number of genes exceeds the constraint *cg*, the subsets are collected as a potential equivalence class set (line 16). From the potential equivalence class set, the subsets whose combined condition values are the same as the condition values of *targetcond* are selected as the equivalence class set (lines 17-18). If no equivalence class set is found, the procedure iteratively partitions the genes into another elementary set of the combination of $Ccond_k$ and the other conditions. Finally, the unknown value $v_{i,targetcond}$ is derived (line 25). Examples are shown below to demonstrate the prediction process.

Example 3.2. *Suppose that there are 10 genes and 5 conditions in the database. Table 3 shows that an un-imputed matrix containing ten genes* $\{g_1, g_2, \ldots, g_{10}\}$*, four selected conditions* $\{cond_1, cond_3, cond_4, cond_5\}$*, and target condition* $targetcond$*, which represents a missing value in the 1*st *gene that is to be predicted. Except* $targetcond$*, each condition for the 1*st *gene (the 1*st *tuple) has a non-zero value. In contrast, the condition values to the other genes may be zero in Table 3 since some genes have missing values. In this matrix, zero represents a missing value. Table 4 shows the resulting matrix after the user-based imputation operation. Except the* $targetcond$ *value, each condition value should be a non-zero value after the user-based imputation operation.*

TABLE 3. Example of an un-imputed matrix

Gene ID	$cond_1$	$targetcond$	$cond_3$	$cond_4$	$cond_5$
1	4	T	3	5	1
2	4	4	3	5	1
3	5	5	0	1	1
4	2	1	3	0	2
5	4	4	2	2	2
6	0	3	1	0	1
7	3	4	2	2	2
8	1	1	0	1	1
9	4	0	3	1	1
10	0	5	3	1	0

TABLE 4. Example of an imputed matrix

Gene ID	$cond_1$	$targetcond$	$cond_3$	$cond_4$	$cond_5$
1	4	T	3	5	1
2	4	4	3	5	1
3	5	5	1	1	1
4	2	1	3	2	2
5	4	4	2	2	2
6	3	3	1	1	1
7	3	4	2	2	2
8	1	1	1	1	1
9	4	4	3	1	1
10	4	5	3	1	1

Example 3.3. *This example is based on Example 3.2. By calculating the Pearson correlation coefficient between* $targetcond$ *and the other conditions, the derived set* $\{sim(1, target\ cond), sim(3, targetcond), sim(4, targetcond), sim(5, targetcond)\}$ *is* $\{0.92, 0.14, -0.06, -0.22\}$*. Therefore, the class attribute is* $cond_1$*. Next, as shown in Table 5, the group is generated from the elementary set of* $cond_1$ *with respect to* $\{g_1, g_2, g_5, g_9, g_{10}\}$*. Assume that the condition constraint cc is 2 and the gene constraint gc is 2. Without considering class attribute* $cond_1$*,* $sim(3, targetcond)$ *is the largest. Hence, the condition most relevant to* $targetcond$ *is condition 3. The system combines* $targetcond$ *and* $cond_3$ *as* $Ccond_2 = \{targetcond, cond_3\}$ *to to meet the condition constraint. Thereby the elementary sets referred to Citm1 are generated. As shown in Table 6, seven partitions of the elementary set* $(ELSet)$ *of* $Ccond_1$ *were found; the matching subsets are thus* $\{g_2, g_9\}$ *and*

Input: The transformed gene-condition matrix $MX_{m\times n}[v_{i,j}]$, the target condition *targetcond* for gene g_i, the condition constraint *cc*, and the gene constraint *cg*;
Output: The *targetcond* value $v_{i,\,targetcond}$;
Algorithm RoughSet_Prediction

```
1.  find the top s similar conditions to targetcond;
2.  for k=1 to s-1 do
3.    if sim(k, targetcond)>maxsim then
4.      maxsim= sim(k, targetcond);
5.      class=k;
6.    end if
7.  end for
8.  for j=1 to |G| do
9.    if v_{j, class} = v_{i, class} then group= group ∪ g_j;
10.  k=cc;
11.  Ccond_k= targetcond ∪ cond_mr, where cond_mr is the most relevant
       condition to targetcond and cond_mr ∈ CandidateCond;
12   CandidateCond= CandidateCond \ cond_mr;
13.    repeat
14.      ELSet = the elementary set of Ccond_k, where ELSet=∪ tg_h and
           tg_h={g_x…., g_y, …., g_z| where v_{x, targetcond} = v_{y, targetcond}, v_{y, targetcond} = v_{z,
           targetcond}, x≠y, y≠z};
15.      for each elementary set tg_h ⊆ ELSet do
16.        if tg_h ⊆ group, | tg_h |≧cg then
17.          for j=x to | tg_h | do
18.            if v_{j, cond_h} = v_{i, cond_h} then
19.              pvalue= pvalue + v_{x, targetcond};
20.              cnt++;
21.    .       end if
22.          end for
23.        end if
24.      end for
25..      v_{i, targetcond}= pvalue / cnt;
26.      if v_{i, targetcond}≠0 then
27.        k++;
28.        Ccond_k= Ccond_k ∪ cond_mr, where cond_mr is the next relevant item to
             targetcond and cond_mr ∈ CandidateCond;
29.        CandidateCond= CandidateCond \ cond_mr;
30.      end if
31.    until v_{i, targetcond}≠0
32.  return v_{i, targetcond};
```

FIGURE 2. Algorithm for rough-set-based prediction

$\{g_{10}\}$. *In this step,* $\{g_{10}\}$ *has to be dropped since the number of users in* $\{g_{10}\}$ *cannot exceed the gene constraint. Therefore, the potential equivalence class set is* $\{g_2, g_9\}$. *Because the value of* $cond_3$ *for* $\{g_2, g_9\}$ *is the same as that of* $targetcond$ *for* g_1, *the equivalence class set* $\{g_2, g_9\}$ *is judged. Consequently, the iterative operation stops finding combinations of* $Ccond_2$ *and the other conditions to look for the equivalence class set. Finally, the prediction value* $v_{1,targetcond}$ *is 4. Note that the purpose of the gene constraint and the condition constraint is to enhance the prediction precision. That is, the larger the gene and condition constraints, the fewer the number of matching elementary sets, the higher the*

equivalence granularity, and the fewer the number of matching equivalence class sets. For example, if the gene constraint is set to 1, the matching elementary sets include $\{g_2, g_9\}$ *and* $\{g_{10}\}$*. Therefore, the prediction value* $v_{1,targetcond}$ *is (4+5)/2=4.5, which is different from the value derived when only* $\{g_{10}\}$ *was used.*

TABLE 5. Example of the elementary set of the class attribute $cond_1$

Elementary set	Gene ID	$cond_1$
1 (group)	1 (active)	4
	2	4
	5	4
	9	4
	10	4
2	8	1
3	4	2
4	6	3
	7	3
5	3	5

TABLE 6. Example of the elementary set of {*targetcond*, $cond_3$}

Elementary set	Gene ID	*targetcond*	$cond_3$
1	8	1	1
2	4	1	3
3	6	3	1
4	5	4	2
	7	4	2
5	2	4	3
	9	4	3
6	3	5	1
7	10	5	3

Missing-Value Prediction

In order to impute original missing values, the values predicted in rough-set-based prediction ($\text{value}_{\text{transformed}}$) must be transformed into original values ($\text{value}_{\text{original}}$):

(1) $\text{step}(i) = (\text{value}_{\max(i)} - \text{value}_{\min(i)})/N$ and

(2) $\text{value}_{\text{transformed}} = 1 + \text{Round}((\text{value}_{\text{original}} - \text{value}_{\min(i)})/\text{step}(i))$

where:

$$\text{value}_{\text{original}} = (\text{value}_{\text{transformed}} - 1) \times \text{step}(i) + \text{value}_{\min(i)}$$

An example is shown below to demonstrate the process.

Example 3.4. *Suppose that the data in Table 7 is the transformed gene expression data, the value of N is 4,* $step(1) = 0.4$, $value_{\min(1)} = -0.4$, $step(2) = 0.3$, $value_{\min(2)} = -0.1$, $step(3) = 0.3$, $value_{\min(3)} = -0.2$. *The original gene expression data is shown in Table 8.*

Table 7. Transformed gene expression data

Gene ID	$cond_1$	$cond_2$	$cond_3$	$cond_4$	$cond_5$
1	4	4	3	5	1
2	4	4	3	5	1
3	5	5	1	1	1

Table 8. Original gene expression data

Gene ID	$cond_1$	$cond_2$	$cond_3$	$cond_4$	$cond_5$
1	0.8	0.8	0.4	1.2	−0.4
2	0.8	0.8	0.5	1.1	−0.1
3	1.0	1.0	−0.2	−0.2	−0.2

4. **Experimental Evaluations.** This section presents the empirical evaluations. The main goal of the experiments was to measure the performance of the proposed *CFBRST* approach. Four methods were compared: the *CFBRST* approach, the user-based (gene-based) CF imputation approach, the k-NN imputation approach, and the row (gene) average (filling missing values with row average) imputation approach. The experimental results show that the proposed *CFBRST* approach outperforms the other imputation approaches.

4.1. **Datasets.** Experiments with three public yeast cDNA microarray datasets were performed. The first dataset (diauxic) is from a study of temporal gene expression (Derisi et al., 1997). It consists of 6068 genes and 7 conditions. After genes with missing values were removed, 5875 genes remained. The second dataset (elutriation) is the elutriation part of a study on yeast cell-cycle gene expression (Spellman et al., 1998). There are 6075 genes and 14 conditions in this dataset. After genes with missing values were removed, 5766 genes remained. The last dataset (phosphate) is from a study of phosphate accumulation (Ogawa et al., 2000). There are 6013 genes and 8 conditions in this dataset. After genes with missing values were removed, 5783 genes remained. A summary of the three datasets is shown in Table 9.

Table 9. Summary of the three datasets used for experiments

Name	***M***	***M'***	***C***
Diauxic	6068	5875	7
Elutriation	6075	5766	14
Phosphate	6013	5783	8

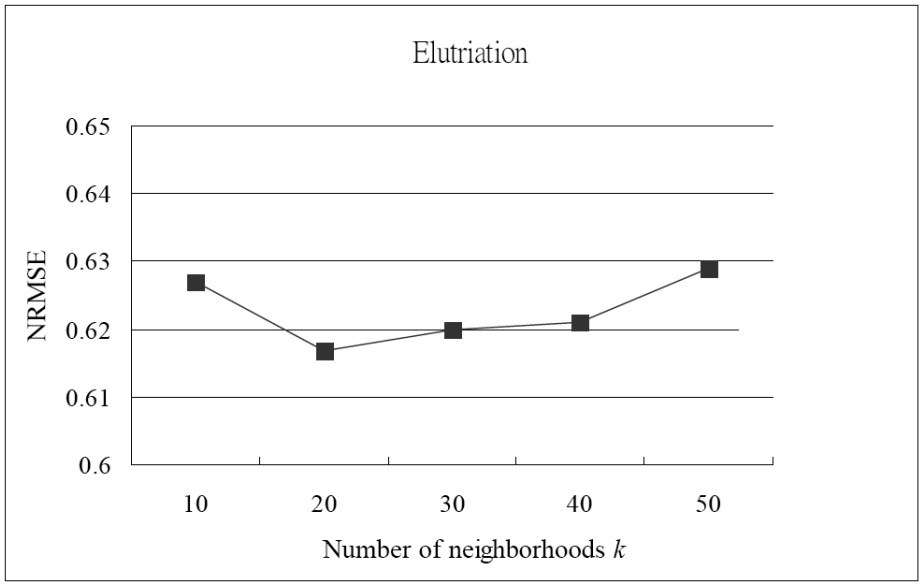


FIGURE 3. Effect of the number of nearest neighbors on *CFBRST* estimation for the elutriation dataset

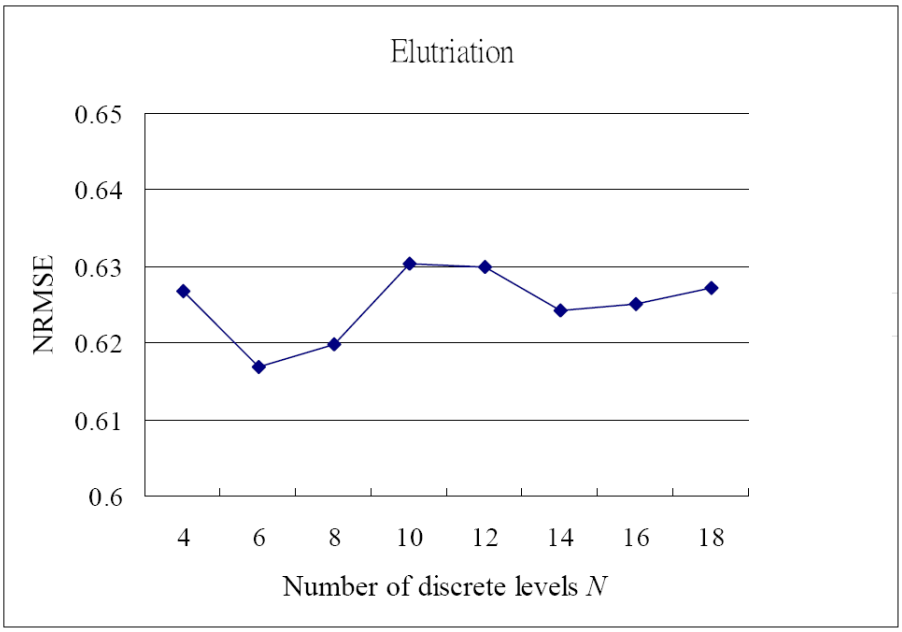


FIGURE 4. Effect of the number of discrete levels N on *CFBRST* estimation for elutriation dataset

4.2. **Evaluation metrics.** Mean absolute error (MAE) is often used to measure the effectiveness in CF recommender systems. It represents the difference between the actual values and predicted values. It is defined as:

$$MAE = \frac{\sum_{i=1}^{N} |y_{pre_i} - y_{act_i}|}{S}$$

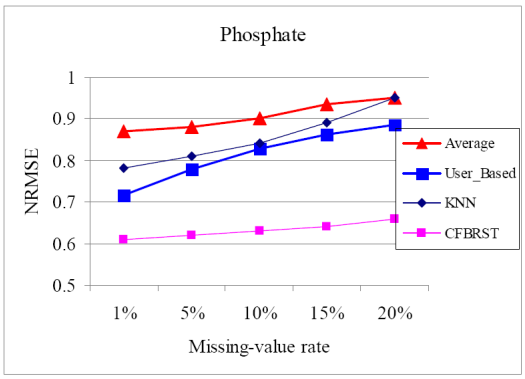
where y_{pre_i} and y_{act_i} are the i^{th} predicted value and the i^{th} actual value, respectively, and S is the number of the predicted entries. Generally, the smaller the MAE is, the better the approach performs.

Normalized root mean squared error (*NRMSE*) is often used as an evaluation criterion in missing-value imputation algorithms. It is defined as:

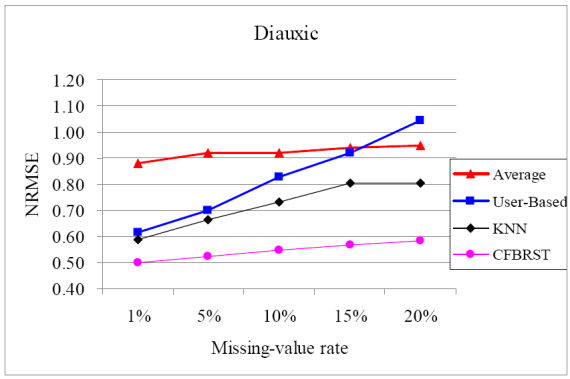
$$NRMSE = \sqrt{\frac{mean[(y_{pre} - y_{act})^2]}{std[y_{act}]}}$$

where y_{pre} and y_{act} are vectors containing the predicted values and the actual values for all estimated entries, respectively. Similar to *MAE*, the smaller the *NRMSE* is, the better the approach performs.

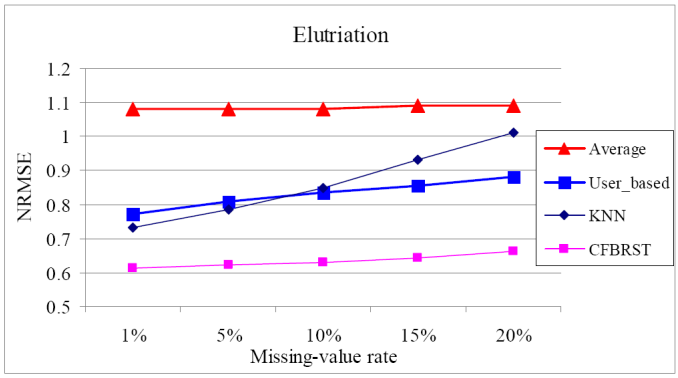
In this paper, *NRMSE* is used as the evaluation metric to compare the proposed algorithm with other imputation algorithms because it is widely used to evaluate the effectiveness of algorithms for microarray gene expression data.



(a)



(b)



(c)

FIGURE 5. Comparison of $NRMSE$ values of the tested imputation methods for various missing-value rates

4.3. **Evaluations for the parameter settings.** Before evaluating the performance of the *CFBRST* approach, suitable parameter settings were determined for the experiments. Two parameters play important roles in the experiments. The first is the neighborhoods k, which determines the size of the nearest neighbors for the user-based *CF* algorithm. The other parameter is the number of discrete levels N, which controls the number of categories when the numerical dataset is transformed into the categorical dataset.

Experiment for determining neighborhood size *k*. The size of the neighborhood can affect the prediction quality. When the number of similar neighbors is insufficient, the performance decreases due to an overemphasis on similar genes. On the other hand, too many similar neighbors deteriorates accuracy because too many irrelevant neighbors will be used to predict the missing values. To determine this parameter, Troyanska et al. 2001 observed that the best results of the k-NN imputation algorithm can be obtained when the neighborhood size is between 10 and 20. Tuikkala et al. 2006 found that a neighborhood size of 20 is suitable for the GOKNN imputation algorithm. An experiment (Figure 3) was performed in this study by varying the neighborhood size k. *NRMSE* was computed for the elutriation dataset. The results show that a neighborhood size of 20 is optimal. Hence, the neighborhood size was set to 20 in subsequent experiments.

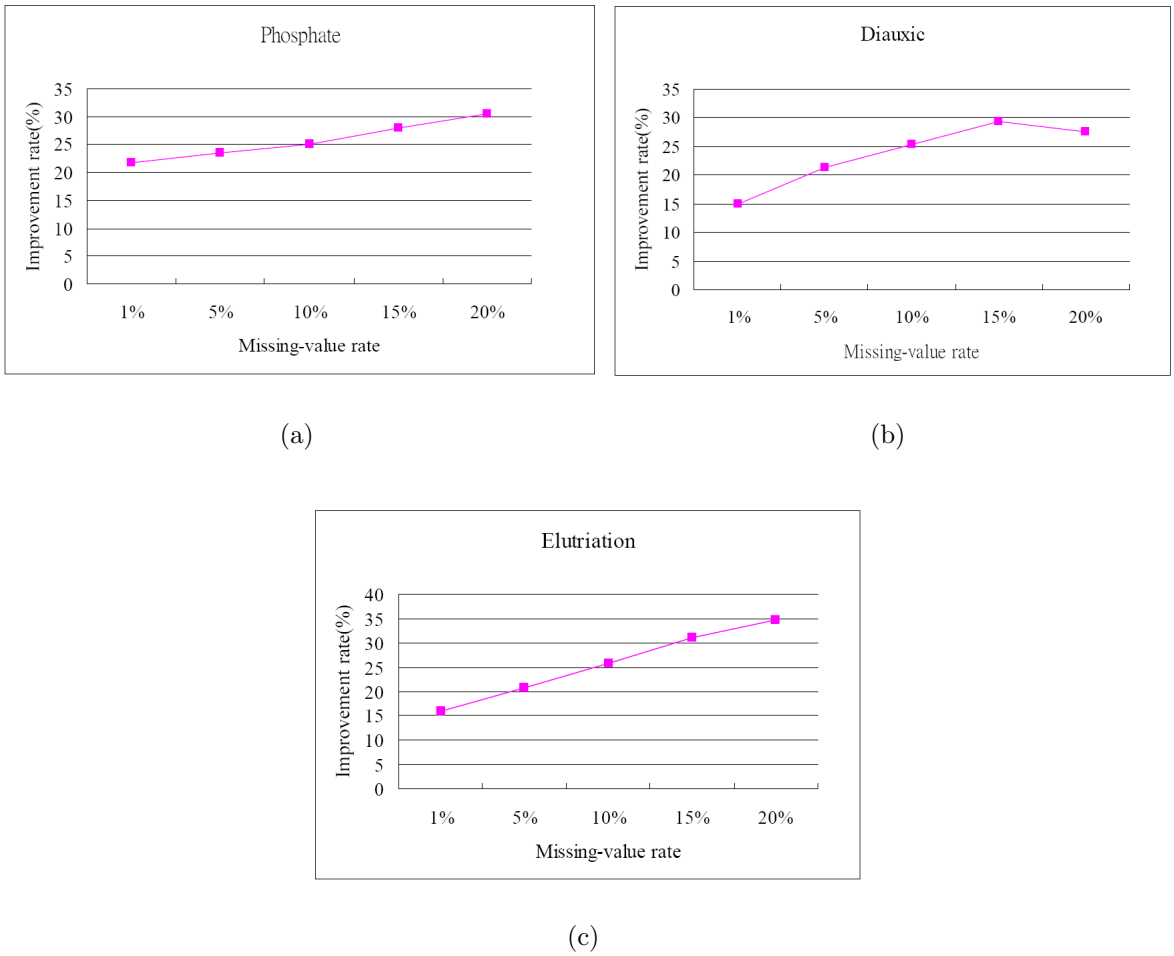


FIGURE 6. Improvement rate of $NRMSE$ values for various missing-value rates

Experiment for determining number of discrete levels *N*. After user-based *CF* prediction, rough-set theory is applied to increase the prediction accuracy of the missing values. The data of the dataset must be discretized to transform the numerical dataset into a categorical dataset. The number of the discrete levels N affects the accuracy of the *CFBRST* approach. An experiment was conducted to determine the number of discrete levels N for subsequent experiments. Figure 4 reveals that the impact of the number of discrete levels N is very small and that the best result was obtained for $N = 6$. Therefore, this parameter was set to 6 for subsequent experiments.

4.4. **Comparisons between *CFBRST* and other approaches.** To assess the quality of the proposed algorithm, several experiments were conducted using the *CFBRST* algorithm, the user-based (gene-based) *CF* imputation approach, the k-NN imputation approach, and the row (gene) average imputation approach. Figure 5 shows the results for various missing-value rates (percentage of missing values) for the three datasets. For all missing-value rates, the $NRMSE$ value for the proposed *CFBRST* approach is smaller than those of the other imputation algorithms for the three datasets.

Moreover, the improvement rate (IR) of *CFBRST* compared with the k-NN imputation algorithm was determined as:

$$IR = \frac{NRMSE_{KNN} - NRMSE_{CFBRST}}{NRMSE_{KNN}}$$

where $NRMSE_{CFBRST}$ is the $NRMSE$ value of the proposed *CFBRST* algorithm.

Figure 6 illustrates the IR of $NRMSE$ values for various missing-value rates. The improvement rate rises with increasing missing-value rate.

5. **Conclusions.** The Collaborative Filtering Based on Rough-Set Theory (*CFBRST*) method, which combines collaborative information and rough-set theory, was proposed for imputing the missing values of microarray gene expression data. Three real yeast microarray datasets were used to evaluate the performance of the proposed *CFBRST* imputation approach. The experimental results show that the *CFBRST* algorithm consistently outperforms other imputation algorithms in terms of estimation accuracy. The improvement rate of the *CFBRST* method compared to the k-NN imputation algorithm increased with increasing missing-value rate.

In future work, biological information such as Gene Ontology (GO) Annotation or protein information will be integrated into the *CFBRST* approach to impute missing values in microarray datasets.

Acknowledgment. This research was supported by National Science Council, Taiwan, under grant No. NSC 100-2627-B-006-020.

REFERENCES

[1] E. Acuna and C. Rodriguez, The treatment of missing values and its effect in the classifier accuracy, *Classification, Clustering and Data Mining Applications*, pp.639-648, 2004.

[2] M. Balabanović and Y. Shoham, Fab: Content-based collaborative recommendation, *Communications of the ACM*, vol.40, pp.66-72, 1997.

[3] J. Basilico and T. Hofmann, Unifying collaborative and content-based filtering, *Proc. of ACM International Conference on Machine Learning*, pp.65-72, 2004.

[4] C. Basu, H. Hirsh and W. Cohen, Recommendation as classification: Using social and content-based information in recommendation, *Proc. of the 15th National Conference on Artificial Intelligence*, pp.714-720, 1998.

[5] R. M. Bell, Y. Koren and C. Volinsky, Modeling relationships at multiple scales to improve accuracy of large recommender systems, *Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007.

[6] Y. Blanco-Fernandez, J. J. Pazos-Arias, M. Lopez-Nores, A. Gil-Solla and M. Ramos-Cabrer, VATAAR: An improved solution for personalized TV based on semantic inference, *IEEE Transaction on Consumer Electronics*, vol.52, pp.421-429, 2006.

[7] J. S. Breese, D. Heckerman and C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, *Proc. of the 14th Conference on Uncertainty in Artificial Intelligence*, pp.43-52, 1998.

[8] E. Camon, M. Magrane, D. Barrell, V. Lee, E. Dimmer, J. Maslen, D. Binns, N. Harte, R. Lopez and R. Apweiler, The gene ontology annotations (GOA) database: Sharing knowledge in uniprot with gene ontology, *Nucleic Acids Research*, vol.32, pp.262-266, 2004.

[9] C. Yu, J. Xu and X. Du, Recommendation algorithm combining the user-based classified regression and the item-based filtering, *Proc. of the 8th International Conference on Electronic Commerce*, pp.574-578, 2006.
[10] F. M. Couto, M. Silva and P. Coutinho, Measuring semantic similarity between Gene Ontology terms, *Data & Knowledge Engineering*, vol.61, pp.137-152, 2007.
[11] J. L. DeRisi, V. R. Iyer and P. O. Brown, Exploring the metabolic and genetic control of gene expression on a genomic scale, *Science*, vol.278, pp.680-686, 1997.
[12] S. Dudoit, Y. H. Yang, M. J. Callow and T. P. Speed, Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments, *Statistica Sinica*, vol.12, pp.111-139, 2002.
[13] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. BloomField and E. S. Lander, Molecular classification for cancer: Class discovery and class prediction by gene expression monitoring, *Science*, vol.286, pp.531-537, 1999.
[14] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker and J. Riedl, Combining collaborative filtering with personal agents for better recommendations, *Proc. of the 16th National Conference on Artificial Intelligence*, pp.439-446, 1999.
[15] S. W. Han and J. Y. Kim, A new decision tree algorithm based on rough set theory, *International Journal of Innovative Computing, Information and Control*, vol.4, no.10, pp.2749-2757, 2008.
[16] T. P. Hong, L. H. Tseng and S. L. Wang, Learning rules from incomplete training examples by rough sets, *Expert Systems with Applications*, vol.22, pp.285-293, 2002.
[17] X. Jin, Y. Zhou and B. Mobasher, A maximum entropy web recommendation system: Combining collaborative and content features, *Proc. of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2005.
[18] H. Kim, G. H. Golub and H. Park, Missing value estimation for DNA microarray gene expression data: Local least squares imputation, *Bioinformatics*, vol.21, pp.187-198, 2005.
[19] M. Kryszkiewiccz, Rough set approach to incomplete information system, *Information Sciences*, vol.112, pp.39-49, 1998.
[20] J. Liang and Z. Xu, Uncertainty measures of roughness of knowledge and rough sets in incomplete information systems, *The 3rd World Congress on Intelligent Control and Automation*, vol.4, pp.2526-2529, 2000.
[21] L. C. Lin, Z. Jing, J. Watada, T. Kashima and H. Ishii, A rough set approach to classification and its application for the creative city development, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(B), pp.4859-4866, 2009.
[22] W. Lin, S. A. Alvarez and C. Ruiz, Collaborative recommendation via adaptive association rule mining, *Proc. of the International Workshop on Web Mining for E-Commerce*, 2000.
[23] X. Liu, W. Wu and J. Hu, A method of fuzzy multiple attribute decision making based on rough sets, *International Journal of Innovative Computing, Information and Control*, vol.4, no.8, pp.2005-2010, 2008.
[24] D. J. Lockhart and E. A. Winzeler, Genome, gene expression and DNA arrays, *Nature*, vol.405, pp.827-836, 2000.
[25] P. W. Lord, R. D. Stevens, A. Brass and C. A. Goble, Investigating semantic similarity measures across the gene ontology: The relationship between sequence and annotation, *Bioinformatics*, vol.19, pp.1275-1283, 2003.
[26] P. Melville, R. J. Mooney and R. Nagarajan, Content-boosted collaborative filtering for improved recommendations, *Proc. of the 18th National Conference on Artificial Intelligence*, pp.187-192, 2002.
[27] B. Mobasher, R. Burke, R. Bhaumik and C. Williams, Effective attack models for shilling item-based collaborative filtering systems, *Proc. of the 2005 WebKDD Workshop*, 2005.
[28] D. Nikovski and V. Kulev, Induction of compact decision trees for personalized recommendation, *Proc. of the ACM Symposium on Applied Computing*, pp.575-581, 2006.
[29] S. Oba, M. A. Sato and I. Takemasa, A Bayesian missing value estimation method for gene expression profile data, *Bioinformatics*, vol.19, pp.2088-2096, 2003.
[30] A. Popescul, L. Ungar, D. Pennock and S. Lawrence, Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments, *Proc. of the 17th Conference in Uncertainty in Artificial Intelligence*, pp.437-444, 2001.
[31] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl, GroupLens: An open architecture for collaborative filtering of netnews, *Proc. of the ACM Conference on Computer Supported Cooperative Work*, pp.175-186, 1994.

[32] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, Item-based collaborative filtering recommendation algorithms, *Proc. of the 10th International World Wide Web Conference*, pp.285-295, 2001.
[33] M. S. Sehgal, I. Gondal and L. S. Dooley, Collateral missing value imputation: A new robust missing value estimation algorithm for microarray data, *Bioinformatics*, vol.21, pp.2417-2423, 2005.
[34] U. Shardanand and P. Maes, Social information filtering: Algorithms for automating "word of mouth", *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pp.210-217, 1995.
[35] S. Tan, X. Cheng and H. Xu, An efficient global optimization approach for rough set based dimensionality reduction, *International Journal of Innovative Computing, Information and Control*, vol.3, no.3, pp.725-736, 2007.
[36] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein and R. B. Altman, Missing value estimation methods for DNA microarray, *Bioinformatics*, vol.17, pp.520-525, 2001.
[37] J. Tuikkala, L. Elo, O. S. Nevalainen and T. Aittolallio, Improving missing value estimation in microarray data with gene ontology, *Bioinformatics*, vol.21, no.5, pp.566-572, 2006.
[38] L. H. Ungar and D. P. Foster, Clustering methods for collaborative filtering, *Proc. of the Workshop on Recommender Systems at the 15th National Conference on Artificial Intelligence*, pp.114-129, 1998.
[39] J. Wang, A. P. de Vries and M. J. T. Reinders, Unifying user-based and item-based collaborative filtering approaches by similarity fusion, *Proc. of the 29th ACM SIGIR Conference on Information Retrieval*, pp.501-508, 2006.
[40] J. A. Xu and K. Araki, A SVM-based personal recommendation system for TV programs, *Proc. of the 12th Multi-Media Modeling Conference*, 2006.
[41] G. R. Xue, C. Lin, Q. Yang, W. Xi, H. J. Zeng, Y. Yu and Z. Chen, Scalable collaborative filtering using cluster-based smoothing, *Proc. of the 28th ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.114-121, 2005.