Collaborative Filtering with Dimension Reduction Technique and Clustering for E-Commerce Product

Daffa Barin Tizard Riyadi, Z K A Baizal∗
School of Computing, Informatics, Telkom University, Bandung, Indonesia
Email: 1daffabarin@student.telkomuniversity.ac.id, 2∗baizal@telkomuniversity.ac.id
Correspondence Author Email: baizal@telkomuniversity.ac.id

Abstract—The rapid development of internet users over the last decade has led to an increase in the use of electronic commerce (e-commerce). The existence of a recommender system influences the success of e-commerce. Collaborative Filtering (CF) is one of the most frequently used recommender system methods. However, in real cases, sparsity problems generally occur. This is generally caused because only a small number of users give ratings to items. In this study, we propose the combination of clustering and dimension reduction methods on the Amazon Review Data to overcome the sparsity problem. The clustering method with K-Means is used to group users based on item preferences. Meanwhile, we used Singular Value Decomposition (SVD) for dimension reduction to improve the performance of the recommender system in sparse data. The results show that the combination of SVD and K-Means is successful in predicting ratings with an RMSE value of less than 2, significant performance increase compared to previous study. The use of SVD is proven to be able to overcome sparsity, with a decrease in RMSE of 9.372%.

Keywords: Recommender System; Collaborative Filtering; Amazon Review Dataset; K-Means; Singular Value Decomposition

1. INTRODUCTION

Electronic commerce (e-commerce) has experienced rapid growth. Online shopping has become one of the habits of internet users worldwide. This is supported by the development of internet users over the last decade. As of January 2023, internet users reached more than 5 billion [1]. The recommender system has a vital role in the development and success of selling items in e-commerce which has been proven to help increase user satisfaction and form user loyalty [2] and other essential aspects such as income and growth [3].

Collaborative Filtering (CF) is one of the most frequently used methods in building recommender systems [4]. In its development, one of the CF methods is memory-based CF. Memory-based CF is divided into two methods, which are user-based and item-based. The main idea of the CF user-based method is to assume similar users have the same preference for an item [5]. Meanwhile, the main idea of the item-based method is to assume the user will buy an item that has the same characteristics as the item purchased before [5]. Another method is model-based CF by utilizing user ratings to train a machine-learning model to obtain user rating prediction results for an item [6]. Traditional machine learning models commonly used in CF-based models include decision trees, support vector machines, rule-based methods, regression models, bayes classifiers, and neural networks [7].

The main problem that often happens when using the CF method is the sparsity in rating items which causes poor recommendation results. Thus, large user and item data dimensions must be used [5]. However, high dimensional data can reduce CF based recommender system efficiency. The method that can be used to deal with the problem of high-dimensional data is to use clustering [8]. However, this method cannot deal with sparsity problems caused by a rating matrix, which is an incomplete relationship between users and items [5]. In real cases, it is common for users only give ratings to some items. One solution to the problem of sparsity in clustering results is to use the dimension reduction method by removing data characteristics that are not dominant to improve the performance of the recommender system.

Research on recommender systems has been carried out in the past few years. Several experiments have been done using variety of algorithms to create a recommender system in different domains. In [9] Fakhr, A. A. et al. used user-based CF by using other users’ preferences for a certain item. Pearson similarity was used to calculate the similarity between users and predicting users’ rating. This study produces high MAE with value as high as $MAE = 2.531$. The cause of this is the sparsity problem in the dataset. In [10] Ayundhita, M. S. et al.
proposed a conversational recommender system based on functional requirements in e-commerce domain. This study shows recommender system perform better by prioritizing functional requirements compared to common recommender system used in e-commerce. In [11] Ardimansyah, M. I. et al. used matrix factorization at the preprocessing stage to deal with sparsity problems in the memory-based CF method. Ardimansyah, M. I. et al. found that the sparsity level in the MovieLens dataset was 93.7%. In this study, sparse matrix that represents item ratings given by users in the case where user didn’t give item ratings will be represented as empty value. After the preprocessing, all the empty value in the sparse matrix will be filled with result of matrix factorization itself, thus turning the previous matrix into dense. The results obtained show a significant increase in recommender system performance after conducting matrix factorization in both user-based CF and item-based CF. In [6] Zarzour H. et al. creates a new approach to developing a CF recommender system by combining clustering and dimension reduction methods in MovieLens dataset. This is intended to improve the performance of the recommended system and overcome sparsity and cold-start problems, as well as make it easier to increase system scalability. K-Means were used to cluster user based on item preferences and creating a cluster-center-item rating matrix that represents average rating of user in a cluster for each item. SVD were used to obtain decomposition matrices and later will be carried to calculate the similarity for each matrix. Furthermore, the similarity was used to predict the ratings. This study shows the success of dimension reduction in improving the performance of the recommender system compared with KNN based and K-Means based recommender system. In [12] Stratigi, M. et al. compared the ratings datasets and review datasets in Amazon Review Data [13] by implementing Valence Aware Dictionary for Sentiment Reasoning (VADER) approach. The results of this study indicate low performance with RMSE > 2 in the recommender system that is made. This is due to the potential for inconsistency between user ratings and reviews.

From related study, it is proven by combining clustering and dimension reduction can overcome the sparsity problem. We propose a recommender system that combine clustering and dimension reduction methods to overcome the sparsity problem in the Amazon Review Data. K-Means and SVD are the algorithms that we use for user clustering and dimension reduction respectively. We used Amazon Review Data [13] in the five core Musical Instrument category as dataset, which contains user ratings for musical instrument items. It is expected that the system to be built can produce an accurate CF recommender system and solve problems that commonly occur in CF recommender systems in general.

2. RESEARCH METHODOLOGY

2.1 Research Stages

Figure 1 shows the process of building our proposed recommender system.

![Recommender System Creation Flow](image)

In this study, we built an e-commerce recommender system. To create a prediction for items, cluster-similarity matrix as shown below,

$$
\begin{bmatrix}
C_{sm_{1,1}} & C_{sm_{1,2}} & C_{sm_{1,3}} & C_{sm_{1,4}} & \cdots & C_{sm_{1,n}} \\
C_{sm_{2,1}} & C_{sm_{2,2}} & C_{sm_{2,3}} & C_{sm_{2,4}} & \cdots & C_{sm_{2,n}} \\
C_{sm_{3,1}} & C_{sm_{3,2}} & C_{sm_{3,3}} & C_{sm_{3,4}} & \cdots & C_{sm_{3,n}} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
C_{sm_{n,1}} & C_{sm_{n,2}} & C_{sm_{n,3}} & C_{sm_{n,4}} & \cdots & C_{sm_{n,n}}
\end{bmatrix}
$$

needs to be created. Both rows and columns represent user cluster. Notation $C_{sm(a,b)}$ represents value of the similarity between a-th cluster and b-th cluster. Building process of recommender system until item prediction is given is shown in Figure 1. First, dataset is preprocessed until user-item rating matrix is created. Based on user-item rating matrix, user clustering using K-Means is used to cluster users and transforming user-item rating matrix
into centroid-item rating matrix. In the dimension reduction stage, SVD is used to get reduced U matrix. Based on reduced U matrix, system will create cluster-similarity matrix in similarity calculation stage. Prediction for items will be provided by using equation (equation similarity) using cluster-similarity matrix.

2.2 Dataset

In this study we used 5-core Amazon Review Data [13] in the Musical Instruments category as the dataset. The dataset has 231392 ratings from 27530 users and 10620 items. Thus, the sparsity level of the data is 99.9% as shown in Figure 2, or 6% more sparse than the MovieLens dataset [11]. Before the preprocessing, the dataset consisted of several columns, according to Table 1.

![Figure 2. User-Item Rating Matrix Sparsity Level Visualization](image)

<table>
<thead>
<tr>
<th>Table 1. Dataset Column Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column</td>
</tr>
<tr>
<td>reviewerID</td>
</tr>
<tr>
<td>asin</td>
</tr>
<tr>
<td>reviewerName</td>
</tr>
<tr>
<td>vote</td>
</tr>
<tr>
<td>style</td>
</tr>
<tr>
<td>reviewText</td>
</tr>
<tr>
<td>overall</td>
</tr>
<tr>
<td>summary</td>
</tr>
<tr>
<td>unixReviewTime</td>
</tr>
<tr>
<td>reviewTime</td>
</tr>
<tr>
<td>image</td>
</tr>
</tbody>
</table>

2.3 Preprocessing

In the context of our proposed recommender system, only a few columns of data are needed, which are 'reviewerID', 'asin', and 'overall'. The researcher also only chose data users who had at least rated items 25 times [14]. After the preprocessing is complete, the dataset has 133797 ratings from 27014 users and 1769 items, with a sparsity level of 99.7%. After clean data is obtained, a user-item rating matrix is made according to Figure 3. In the user-item rating matrix, columns represent items, rows represent items, and \( r(m,n) \) represents a user's rating on item \( n \).

![Figure 3. User-Item Rating Matrix](image)

2.4 User Clustering

User clustering is used to cluster user based on item’s preferences with K-Means. K-Means is among the most used clustering method in recommender system industry's [6]. In making a recommender system, K-Means algorithm is used to group users in K size. Each containing users with similar interests in items in terms of ratings. The main idea of the K-Means algorithm is to minimize a squared error function in the following equation

\[
\text{Argmin} \sum_{i=1}^{k} \sum_{x_j \in S_i} ||x_j - \mu_i||^2
\]
where \( \{x_0, ..., x_n\} \) are sets of objects, \( S = \{S_0, ..., S_k\} \) are groups of objects to minimize the sum of squares, and centroid \( c_i \) is the average of \( S_j \). After users are grouped based on preferences for items, the next step is to transform the user-item rating matrix into a centroid-item rating matrix according to Figure 4.

<table>
<thead>
<tr>
<th>( i_1 )</th>
<th>( i_2 )</th>
<th>( i_3 )</th>
<th>...</th>
<th>( i_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_1 )</td>
<td>( r_{1,1} )</td>
<td>( r_{1,2} )</td>
<td>( r_{1,3} )</td>
<td>...</td>
</tr>
<tr>
<td>( u_2 )</td>
<td>( r_{2,1} )</td>
<td>( r_{2,2} )</td>
<td>( r_{2,3} )</td>
<td>...</td>
</tr>
<tr>
<td>( u_3 )</td>
<td>( r_{3,1} )</td>
<td>( r_{3,2} )</td>
<td>( r_{3,3} )</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>( u_m )</td>
<td>( r_{m,1} )</td>
<td>( r_{m,2} )</td>
<td>( r_{m,3} )</td>
<td>...</td>
</tr>
</tbody>
</table>

![Figure 4. Transformation of User-Item Matrix to Centroid-Item Rating Matrix](image)

Likewise with the user-item rating matrix, in centroid-item rating matrix, columns represent items, rows represent clusters, and \( a_{k,n} \) represents the average user rating of items in cluster \( K \).

### 2.5 Dimension Reduction

One way to solve the sparsity problem is to use dimension reduction [14]. SVD is one of the most successful and frequently used dimensional reduction techniques in recommender systems. In the dimension reduction paradigm, SVD is classified as a numerical method, in contrast to PCA, which uses an analytic approach. The main idea of SVD is to find feature spaces with lower dimensions. The SVD of a matrix \( A \) of size \( m \times n \) can be expressed with following equation,

\[
A_{m,n} = U_{m,r} S_{r,r} V_{r,n}^T (2)
\]

The \( U \) matrix, \( m \times r \) in size, with the notation \( r \) representing the \( r \)-th eigenvector of the matrix \( A^T \times A \) [15]. \( S \) is a diagonal matrix of size \( r \times r \) with a singular value of matrix \( A \). Meanwhile, \( V^T \) is an orthogonal matrix of size \( r \times n \), which is a right singular vector with columns representing items [16]. Dimension reduction with SVD was made by using only \((100 - t)\% \) of the first column in the \( U \) matrix, \((100 - t)\% \) of the first rank in \( S \), and \((100 - t)\% \) of the first row in \( V^T \) matrix. In our proposed recommender system to be built, only the matrix was used for dimension reduction. The matrix below is \( U \) matrix before dimension reduction.

\[
\begin{bmatrix}
    u_{1,1} & u_{1,2} & u_{1,3} & \cdots & u_{1,r-1} & u_{1,r} \\
    u_{2,1} & u_{2,2} & u_{2,3} & \cdots & u_{2,r-1} & u_{2,r} \\
    u_{3,1} & u_{3,2} & u_{3,3} & \cdots & u_{3,r-1} & u_{3,r} \\
    \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
    u_{m,1} & u_{m,2} & u_{m,3} & \cdots & u_{m,r-1} & u_{m,r}
\end{bmatrix}
\]

After dimension reduction was carried out, only \((100 - t)\% \) of the \( U \) column remained. The matrix below is reduced \( U \) matrix after dimension reduction is done.

\[
\begin{bmatrix}
    u_{1,1} & u_{1,2} & u_{1,3} & \cdots & u_{1,r(100-t)\%} \\
    u_{2,1} & u_{2,2} & u_{2,3} & \cdots & u_{2,r(100-t)\%} \\
    u_{3,1} & u_{3,2} & u_{3,3} & \cdots & u_{3,r(100-t)\%} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    u_{m,1} & u_{m,2} & u_{m,3} & \cdots & u_{m,r(100-t)\%}
\end{bmatrix}
\]

### 2.6 Similarity Calculation

After the dimension reduction stage, the matrix similarity value was measured using cosine similarity (COS). COS is commonly used to measure the similarity between two vectors. COS is one of the most frequently used similarity calculation methods in recommender systems with the CF approach [6]. COS was used to measure the similarity between user clusters and create a cluster-similarity matrix. In accordance with equation **Error! Reference source not found.** [17], the notations \( x \) and \( y \) represent the \( a \)-th cluster and the \( b \)-th cluster which are rows of the \( U \) matrix represented in vectors. The \( \|a\| \) and \( \|b\| \) notations represent the lengths of the \( a \) and \( b \) vectors, respectively.
2.7 Rating Prediction

After obtaining the cluster-similarity matrix, the stages of predicting item ratings in a cluster were performed using the following equation [18].

\[
\hat{r}_{ci} = \bar{r}_c + \frac{\sum_{n \in \text{neighbors}} \text{sim}_{cn} \cdot (r_{n,i} - \bar{r}_n)}{\sum_{n \in \text{neighbors}} \text{sim}_{cn}}
\]

with \( \hat{r}_{ci} \) denotes predicted rating for item \( i \) in the cluster \( c \), \( \bar{r}_c \) denotes the average rating of cluster \( c \), \( N \) denotes the neighbors' cluster vector of cluster \( c \), \( \text{sim}_{cn} \) denotes the similarity of cluster \( c \), \( r_{n,i} \) denotes item rating \( i \) from cluster \( n \), and \( \bar{r}_n \) represents the average rating of neighbor cluster vector \( n \).

3. RESULT AND DISCUSSION

In this section we will explain the stages of making a recommender system by providing sample data for each stage. The recommender system that we built has been tested 5 times with different \( K \) values where \( K = \{10,20,30,40,50\} \). The dimension reduction process with SVD was tested 3 times with different \( t \) values where \( t = \{10,20,30\} \), which are represented by K-Means-10-reduction, K-Means-20-reduction, and K-Means-30-reduction.

3.1 Dataset

The dataset sample before preprocessing is shown in Table 2.

<table>
<thead>
<tr>
<th>overall</th>
<th>verified</th>
<th>reviewTime</th>
<th>reviewerID</th>
<th>asin</th>
<th>reviewerName</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>True</td>
<td>11/16/2016</td>
<td>AOZVLEOST0J3G</td>
<td>B009ZX8ZJG</td>
<td>JD</td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>True</td>
<td>01/12/2016</td>
<td>AJ94N31A434D</td>
<td>B0063678K</td>
<td>W. D.</td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>True</td>
<td>04/28/2016</td>
<td>A156CKCKXPH8S5</td>
<td>B002E1G5C</td>
<td>Cenerentola</td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>True</td>
<td>08/08/2013</td>
<td>A1E5FQZTM8OC1</td>
<td>B002GXYVO</td>
<td>Kenwood</td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>False</td>
<td>10/04/2014</td>
<td>A1OR8R0BADATAI</td>
<td>B001YODDC</td>
<td>Guitar Dog</td>
<td>...</td>
</tr>
</tbody>
</table>

In this study we split the dataset into training and testing data with a percentage of 80% and 20% respectively after the preprocessing. We make sure that for every user and item that present in testing data must also present in training data. Sample of both testing and training data are shown in Table 3 and Table 4 respectively.

<table>
<thead>
<tr>
<th>overall</th>
<th>reviewerID</th>
<th>asin</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>A1RIVJ07HKE1F9</td>
<td>B006MYECVE</td>
</tr>
<tr>
<td>5</td>
<td>A2PX3NR11VTXPG</td>
<td>B008FDSWJ0</td>
</tr>
<tr>
<td>5</td>
<td>A27HOMP2IFOOYC</td>
<td>B009MDMQJ4</td>
</tr>
<tr>
<td>5</td>
<td>A3EZTFLYUHCHBF</td>
<td>B00XQCRR5U</td>
</tr>
<tr>
<td>5</td>
<td>A3EMSV4RL1SV4N</td>
<td>B002GMYGA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>overall</th>
<th>reviewerID</th>
<th>asin</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>A1TDR4NT5G28NI</td>
<td>B001TMCDVW</td>
</tr>
<tr>
<td>3</td>
<td>A26RNTHH6P38YJ</td>
<td>B002GWFEOQ</td>
</tr>
<tr>
<td>5</td>
<td>A29A4ZAX7XDLJC</td>
<td>B002E1NNC</td>
</tr>
<tr>
<td>4</td>
<td>A20PWG5RGSGH5</td>
<td>B002OP0WC</td>
</tr>
<tr>
<td>2</td>
<td>AGI10EQ9HIZN2</td>
<td>B00GROJWAW</td>
</tr>
</tbody>
</table>

3.2 User-Item Rating Matrix

The following matrix is a sample for user-item rating matrix,

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]
where rows represent users and column represent items. The matrix has high sparsity level, this is due to the size of the matrix is 26485 × 1769 only with the limit of a user rating at least 25 times. The sparsity level of this matrix is 99.7%.

### 3.3 Centroid-Item Rating Matrix

The result of user clustering with K-Means can be transformed into centroid-item rating matrix. Matrix below is the sample data of centroid-item rating matrix,

\[
\begin{pmatrix}
0 & 0 & 4 & 0 & 0 \\
5 & 0 & 5 & 0 & 0 \\
5 & 1 & 5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

where rows represent clusters and column represents items. Value 0 is possible due to the high sparsity of the user-item rating matrix, as there will be a possible case of clusters without a single user rating for a certain item.

### 3.4 Reduced U Matrix

In this study, we will only be considering U matrix after the SVD process. in the case of K = 10 and t = 10, the U matrix has size 10 × 10 with rows representing clusters and columns representing eigenvectors of a matrix A^T × A according to the following 10 × 10 matrix,

\[
\begin{pmatrix}
-0.721 & 0.423 & \cdots & 0.007 & 0.011 \\
-0.291 & -0.514 & \cdots & 0.001 & -0.017 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
-0.161 & -0.147 & \cdots & 0.102 & -0.016 \\
-0.409 & 0.360 & \cdots & 0.012 & -0.002 \\
\end{pmatrix}
\]

After dimension reduction was carried out, only 90% of the U column remained according to the 10 × 9 reduced U matrix below with the system removing the last column from the U matrix.

\[
\begin{pmatrix}
-0.721 & 0.423 & \cdots & 0.007 \\
-0.291 & -0.514 & \cdots & 0.001 \\
\vdots & \vdots & \ddots & \vdots \\
-0.161 & -0.147 & \cdots & 0.102 \\
-0.409 & 0.360 & \cdots & 0.012 \\
\end{pmatrix}
\]

### 3.5 Cluster-Similarity Matrix

After the similarity calculation stage had been carried out, the U matrix turned into a cluster-similarity matrix,

\[
\begin{pmatrix}
1 & 0.000202 & \cdots & 0.000192 & 0.000026 \\
0.000202 & 1 & \cdots & -0.0000279 & -0.000038 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0.000192 & -0.0000279 & \cdots & 1 & -0.000036 \\
0.000026 & -0.000038 & \cdots & -0.000036 & 1 \\
\end{pmatrix}
\]

measuring 10 × 10 in size, with each row and column representing a cluster. The diagonal matrix has a value of 1 because it is the similarity between the two clusters.

### 3.6 Application Implementation

In this stage, a performance measurement algorithm is needed to evaluate the recommender systems, such as RMSE. The main idea of RMSE is to calculate the square root of the average value of all the differences between predicted and actual value. Therefore, a small RMSE value implies better performing recommender system [19]. RMSE can be expressed in equation [20] where \( y \) denotes the actual value, \( \hat{y} \) is the system’s given predicted value, and \( n \) is number of ratings predicted in a trial.

\[
RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}
\]

(5)

Research generally shows a high RMSE value with RMSE > 1. This is common in research with high dimensional data [12] because the sparsity level in the dataset reaches a value of 99.7% even after preprocessing with a user at least giving a rating 25 times at items. However, the success of SVD is proven to increase the performance of the recommender system in the dimension reduction process. Figure 5 shows the RMSE comparison between all the tested approaches. At K = 10, the greater the value of t, the lower the performance.
due to the loss of information from the U matrix, which is only $10 \times 10$ in size. For all values $K > 10$, K-Means-30-reduction works better than all methods when using $K > 10$. Meanwhile, for all values $10 < K < 50$ K-Means-20-reduction also performs better than K-Means. K-Means-10-reduction is not suitable for application with $K > 20$.

![Figure 5. Comparison of RMSE](image)


<table>
<thead>
<tr>
<th>K</th>
<th>K-Means</th>
<th>K-Means-30-reduction</th>
<th>RMSE Difference (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>2.055066</td>
<td>1.885936</td>
<td>8.583%</td>
</tr>
<tr>
<td>30</td>
<td>2.263957</td>
<td>2.153186</td>
<td>5.016%</td>
</tr>
<tr>
<td>40</td>
<td>2.398194</td>
<td>2.255984</td>
<td>6.111%</td>
</tr>
<tr>
<td>50</td>
<td>2.539080</td>
<td>2.311766</td>
<td>9.372%</td>
</tr>
<tr>
<td>average</td>
<td>2.31407425</td>
<td>2.151718</td>
<td>7.270%</td>
</tr>
</tbody>
</table>

4. CONCLUSION

CF-based recommender system has the main problem of sparsity which can produce a poor recommender system. This research focuses on developing a recommender system in the e-commerce domain and improving the recommender system’s performance by increasing accuracy. The results show a success in generating item recommendation with our proposed method. However, the system generally produces high RMSE values, yet significantly lower than previous research using the same dataset. This is caused by the high sparsity level of the dataset, which reaches 99.7% even after preprocessing of limiting to only user that has done at least 25 ratings. Therefore, the neighbor similarity level in the cluster-similarity matrix tends to become very low.

REFERENCES


