Chapter 4

Results and Discussion

This chapter describes the experimental results of the proposed method on synthetic and real-world data sets. The results of the proposed methods are compared with common methods, i.e., the VCR-KM, frequency-based, user-based, and item-based methods.

The experimental results are divided into five sections. Section 4.1 describes the proposed recommendation systems and the common methods for comparing the experiment results. Section 4.2 describes the details of the parameters and data sets in this research. Section 4.3 shows the results of clustering on the three synthetic data sets by using the VCM-GAs and VCM-MAs. Section 4.4 shows the results of clustering on the five real-world data sets by using the VCM-GAs, VCM-MAs, and VCM-KM. Section 4.5 shows the performance of the top-*N* recommendation systems on real-world data sets.

4.1 Comparison of Top-*N* Recommendation Systems

There are sixteen top-*N* recommendation systems used in the comparison. The user-based, item-based, and frequency-based recommendation systems are denoted by the UB, IB, and FB, respectively. The detailed information of the UB, IB, and FB methods were described in section 2.2. The recommendation system based on the VCM-KM is denoted by the VCR-KM. The detailed information of the VCM-KM was described in section 3.3. In the proposed methods, the top-*N* recommendation method based on the VCM-GA1 is denoted by the VCR-GA1. The hybrid methods between the VCR-GAs and UB methods are denoted the VCR-GA1-UB and VCR-GA2-UB. The top-*N* recommendation method based on the VCM-GA2 is denoted the VCR-GA2. The hybrid methods between the VCR-GAs and IB methods are denoted the VCR-BA1-IB and VCR-GA2-IB. The top-*N* recommendation method based on the VCM-MA1 is denoted by the VCR-MA1. The hybrid methods between the VCR-MAs and UB methods are denoted by the VCR-MA1-UB and VCR-MA2-UB. The hybrid methods between the VCR-MAs and IB methods are denoted by the VCR-MA1-IB and VCR-MA2-IB. The top-*N* recommendation method based on the VCM-MA2 is denoted by the VCR-MA2. The hybrid methods between the VCR-MAs and UB methods are denoted by the VCR-MA1-UB and VCR-MA2-UB. The hybrid methods between the VCR-MAs and IB methods are denoted by the VCR-MA1-IB and VCR-MA2-IB. The details of the hybrid methods were described in section 3.4.

4.2 Synthetic and Real-World Data Sets and Parameter Setting

In this section, the synthetic data sets, real-world data sets, and the parameter setting are described. There are three synthetic data sets and five real-world data sets.

4.2.1 Synthetic Data Sets

In the synthetic data sets, we create three data sets containing three, five, and seven clusters, respectively. In the first data set, there are 20 rows and 20 columns containing three clusters. In the second data set, there are 34 rows and 34 columns containing five clusters. In the third data set, there are 48 rows and 48 columns containing seven clusters. Each cluster in the images contains 24 pixels. For each image, the rows are randomly interchanged. The process is repeated with the columns. We use these data sets to evaluate the clustering performance in the next section.

4.2.2 Real-World Data Sets

The first real-world data set is the transaction of purchasing from Gazelle.com, leg wear and leg care e-tailer collected by Blue Martini Software on KDD-CUP2000 (KDD). In this data set, there are 3,465 purchases in total by 1,831 customers. However, there are missing data, for example, the customer ID, the item ID, the number of purchases, in some transactions. After removing those incomplete transactions, there are 1,697 customers and 247 items. However, there are only 271, 110, and only 14 customers who purchase at least two, three, and four items, respectively. Moreover, there are only 102 items that are purchased at least twice. In this data set, we select 110 customers who purchase at least three items. Hence, the size of the KDD data set is 110 customers and 247 items. The ten-fold cross validation is performed to divide the data set into the training and test sets.

The second data set is the transaction of purchasing at Thaiherbs-Thaimassage shop (TTS) (http://www.thaiherbs-thaimassage.com). There are 707 transaction records in this data set. The data set consists of 371 customers and 175 items purchased. However, there are only 112 and 55 customers who purchase at least two and three items, respectively. In addition, there are only 95 items that are purchased at least twice. Hence, the 112 customers who purchase at least two items are selected. The size of the TTS data set contains the 112 customers and 175 items. The training and test sets are divided by using the ten-fold cross validation.

The third data set is the transaction of visiting the entree Chicago restaurant (ECR), collected by the University of California on the UCI machine learning repository. This data set was recorded interactions in the 4^{th} quarter of 1996. Each user is presented by a session of user interaction with the system. There are 1,786 users (i.e. 1,786 sessions) and 674 restaurants. We selected 611 users who visited at least five times. The visited restaurants were mapped into the binary values (0 = unvisited, 1 = visited). We randomly selected 122 users, i.e. 20% of 611 users, as the test set. The remaining users are of the training set.

The fourth data set is the restaurant and consumer data set (RCM) collected by the Department of Computer Science, National Center for Research and Technological Development in Mexico. This data set contains 1,161 rating for 130 restaurants rated by 138 users. The rated restaurants are mapped into the binary values (0 = unrated restaurant, 1 = rated restaurant). We selected 115 users who rated at least four and 130 restaurants. The ten-fold cross validation was performed to divide the data set into the training and test sets.

The fifth data set is the MovieLens collected by the Group Lens Research Project at the University of Minnesota [6, 12]. This data set contains 100,000 ratings for 1,682 movies rated by 943 users. Each user has rated at least 20 movies. The rated movies are mapped into the binary values (0 = unrated movie, 1 = rated movie). We randomly selected 189 uses, i.e. 20% of 943 users, as the test set. The remaining users are the training set.

For each data set, the nonzero entries and total entries are used to consider the sparsity level [12]. The sparsity level of a data set for data matrix *R* is defined as

$$SL = 1 - \frac{\text{nonzero entries}}{\text{total entries}}, \tag{4.1}$$

where *SL* is the sparsity level. Table 4.1 shows the sparsity level of the five real-world data sets.

Table 4.1 Sparsity level of the real-world data sets.

Data set	Nonzero entries	Total entries	Sparsity level
KDD	393	27,170	0.9855
TTS	608	19,712	0.9692
RCM	907	18,209	0.9502
ECR	3,958	235,846	0.9832
MovieLens	100,000	1,570,988	0.9369

The sparsity problem occurs when the frequency of the purchased items is too small. We compared the number of items purchased or visited or rated in each data set. Figure 4.1 shows the number of purchased items. The purchased item means the same value of the visited and rated item (i.e., visited = rated = purchased = 1, otherwise = 0). I5 denotes the number of the purchased items between 1 and 5. I10 denotes the number of the purchased items between 6 and 10. I15 denotes the number of the purchased items between 11 and 15. I20 denotes the number of the purchased items between 16 and 20. I50 denotes the number of the purchased items between 21 and 50. I100 is the number of the purchased items between 51 and 100. I200 denotes the number of the purchased items between 101 and 200. I600 denotes the number of the purchased items more than 201.

The results show that the frequency of the purchased items in the KDD, TTS, RCM, and ECR data sets is much smaller than the frequency of the purchased items in the MovieLens. In the KDD, TTS, RCM, and ECR data sets, the number of I5 is high while the number of I10, I15, I20, and I50 is low. In the MovieLens data set, the number of the I10, I15, I20, I50, I100, I200, and I600 is higher than other data sets. Hence, the KDD, TTS, RCM, and ECR data sets should have higher sparsity level than the MovieLens data sets. The results in Table 4.1 clearly show that the KDD, TTS, RCM, and ECR data sets have higher the sparsity level than the MovieLens data set.

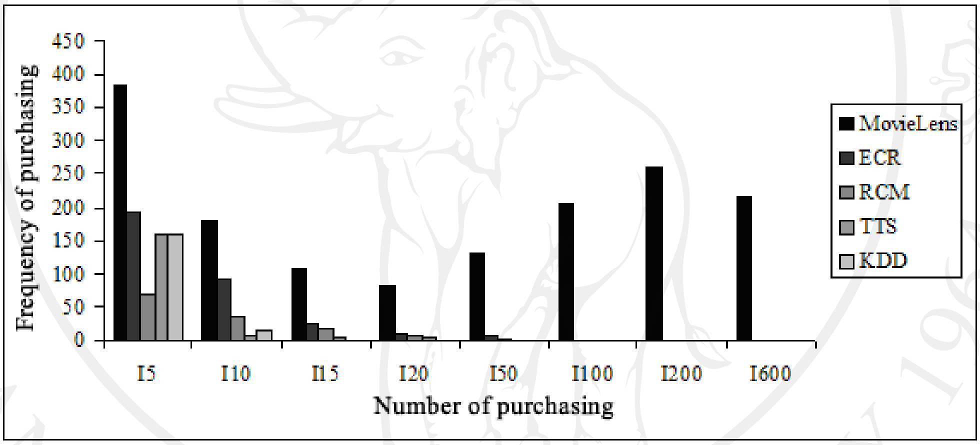

Figure 4.1 Number of items purchased.

In this research, we set up the parameters of two fitness functions in the GA and MA as follows. In the KDD, TTS, and ECR data sets, we set up the parameters as follows: size of population is 80, mutation is 0.01, and crossover rate is 0.6. The weights of fitness function in eq.(3.1) are set to 0.5 for both α and β. The weights of fitness function in eq.(3.4) are set to 0.2, 0.2, 0.3, and 0.3, respectively. In the recommendation systems for the KDD and TTS data sets, the neighborhood size of the user-based and item-based set is limited to ten. In the recommendation systems for the ECR data set, the neighborhood size of the user-based is 50 and the neighborhood size of the item-based is 10. In the recommendation systems for the RCM data set, the neighborhood size of the user-based is 30 and the neighborhood size of the item-based is 10. In the recommendation systems on the MovieLens data set, the neighborhood size of the user-based is 50 and the neighborhood size of the item-based is 50. The cosine similarity measure is used for the user-based and item-based top-*N* recommendation systems. The number *N* of the top-*N* recommendation systems is set to five (i.e., top-5 recommendation).

We divide the MovieLans and ECR data sets by randomly selecting customers into 20% for test set and 80% for training set. The details were described in section 4.2.2. However, In the KDD, TTS, and RCM data sets, these data sets are not officially divided into the training and test sets, however, we need to have training and test sets to evaluate the generalization properties of the recommendation methods. The cross validation method is a standard solution for the aforementioned limitation. In our experiments, the ten-fold cross validation is performed. We briefly describe the cross validation here. In the ten-fold cross validation, the entire data set is divided into ten groups of approximately the same size. In the first validation, the first set is kept as the test set or validation set while the nine remaining group are used as the training set. In our case, the data in training set is used to create the clustered image. Then, the derived clustered image is used to provide recommended items to the customers in the test set. The process is repeated on the remaining groups ten times. Hence, each data set will be used as the test data whose information has never been used in the training process. In this research, there will be ten values of each valuation measure from ten validations. For each evaluation measure, we report the results in terms of the average of those ten values.

4.3 Results of Clustering on Synthetic Data Sets Using VCM-GAs, VCM-MAs, and VCM-KM

In the real-world data sets, it is extremely difficult or impossible to evaluate whether a clustering method is able to properly cluster the users and items. Hence, three synthetic data sets were created to represent the data sets with known ground truths. It should be noted that all synthetic data sets are actually binary. The gray level versions are shown below so that we can visualize the clustering performance. To indicate the elements in the same or different cluster, we label elements in the same cluster using the same gray level. For the element in different clusters, the gray levels are different. Figure 4.2(a) shows the original three clusters as a binary image. Each of the clusters is represented by the gray color values, 100, 150, and 200 as shown in Figure 4.2. This data set consists of 20 rows and 20 columns. In this scenario, customers are represented by rows and items are represented by columns. Figure 4.2(b) shows the corresponding image after randomly interchanging rows and columns. It is the input data for the visual clustering process. Figure 4.2(c)-(g) show the result of clustering in Figure 4.2(b) by using the VCM-GA1, VCM-GA2, VCM-MA1, VCM-MA2, and VCM-KM, respectively. It can be clearly seen that all of the proposed methods achieve three clusters. Although the shape and location of each cluster is different from the original, the members in each cluster are the same as that in the original image.

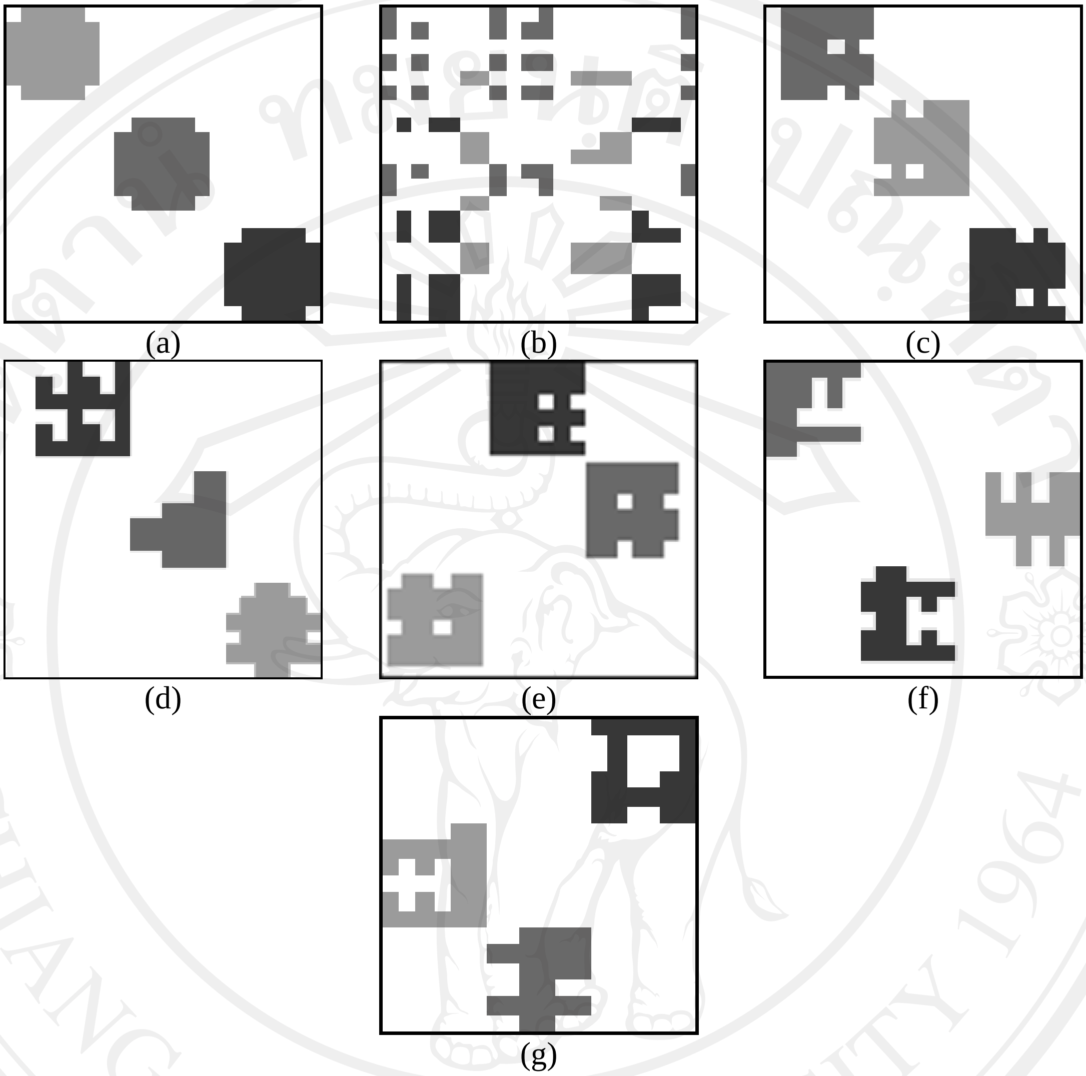

Figure 4.2(a) Original 3-cluster binary image, (b) Row-column-interchanged binary image, (c)-(g) Result of Fig 4.2(b) using the VCM-GA1, VCM-GA2, VCM-MA1, VCM-MA2, and VCM-KM.

Figure 4.3 shows the three clusters from the view point of the gray color values. There are three clusters in this image. Each element in the first cluster is represented by the gray color and is given a value, 100, each element in the second cluster is represented by the gray color with the value 150, and each element in the third cluster is represented by the gray color value 200.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	27	18	19	20
1	0	100	100	100	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	100	100	100	100	100	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	100	100	100	100	100	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	100	100	100	100	100	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	100	100	100	100	100	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	100	100	100	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	150	150	150	150	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	150	150	150	150	150	150	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	150	150	150	150	150	150	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	150	150	150	150	150	150	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	150	150	150	150	150	150	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	150	150	150	150	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	200	200	200	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	200	200	200	200	200
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	200	200	200	200	200
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	200	200	200	200	200
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	200	200	200	200	200
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	200	200	200	0

Figure 4.3 Gray color values of three clusters.

Figure 4.4 shows the clustering result of Figure 4.3 which randomly interchanges rows-columns positions by using the VCM-GA1. It shows that the gray color values have different positions. However, the value in each cluster is in the same group. These clustering results confirm that the VCM-GA1 is able to cluster the information in a binary image.

0	150	150	150	150	150	150	0	0	0	0	0	0	0	0	0	0	0	0	0
0	150	150	150	150	150	150	0	0	0	0	0	0	0	0	0	0	0	0	0
0	150	150	150	0	150	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	150	150	150	150	150	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	150	150	150	150	150	150	0	0	0	0	0	0	0	0	0	0	0	0	0
0	150	150	150	0	150	150	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	100	0	100	100	100	0	0	0	0	0	0	0
0	0	0	0	0	0	0	100	100	100	100	100	100	0	0	0	0	0	0	0
0	0	0	0	0	0	0	100	100	100	100	100	100	0	0	0	0	0	0	0
0	0	0	0	0	0	0	100	100	100	100	100	100	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	100	0	100	100	100	0	0	0	0	0	0	0
0	0	0	0	0	0	0	100	100	100	100	100	100	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	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	0	0	0	0	0	0	0	0	0	0	200	200	200	0	200	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	200	200	200	200	200	200	0
0	0	0	0	0	0	0	0	0	0	0	0	0	200	200	200	200	200	200	0
0	0	0	0	0	0	0	0	0	0	0	0	0	200	200	200	0	200	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	200	200	200	200	200	200	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	200	200	200	200	0

Figure 4.4 Result of the clustering on three clusters in the view point of the gray color values.

Figure 4.5 shows another result of clustering by using the VCM-GA1 in the view point of row and column positions. Each element in Figure 4.5 is represented by values. The first value is the position of row and the second value is the position of column as shown in Figure 4.3. For example element of the row position is 1 and the column position is 2, it contains the values 11 and 11. The first value means the row position 11 and the column position 11 as shown in Figure 4.3. The gray color value in this element in Figure 4.4 is 150 as shown in Figure 4.3. Although the results of clustering show that the shapes and row-column positions are changed, the information in the clusters are the same as in the original data set. So, the results of clustering clearly show that the VCM-GA1 is able to properly retrieve the information in binary images.

0	11,11	11,12	11,10	11,7	11,9	11,13	0	0	0	0	0	0	0	0	0	0	0	0	0
0	10,11	10,12	10,10	10,7	10,9	10,13	0	0	0	0	0	0	0	0	0	0	0	0	0
0	8,11	8,12	8,10	0	8,9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	9,11	9,12	9,10	9,7	9,9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	12,11	12,12	12,10	12,7	12,9	9,13	0	0	0	0	0	0	0	0	0	0	0	0	0
0	13,11	13,12	13,10	0	13,9	12,13	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	6,4	0	6,3	6,2	6,5	0	0	0	0	0	0	0
0	0	0	0	0	0	0	3,1	3,4	3,6	3,3	3,2	3,5	0	0	0	0	0	0	0
0	0	0	0	0	0	0	5,1	5,4	5,6	5,3	5,2	5,5	0	0	0	0	0	0	0
0	0	0	0	0	0	0	4,1	4,4	4,6	4,3	4,2	4,5	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1,4	0	1,3	1,2	1,5	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2,1	2,4	2,6	2,3	2,2	2,5	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	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	0	0	0	0	0	0	0	0	0	0	15,17	15,19	15,16	0	15,18	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	18,17	18,19	18,16	18,15	18,18	18,20	0
0	0	0	0	0	0	0	0	0	0	0	0	0	16,17	16,19	16,16	16,15	16,18	16,20	0
0	0	0	0	0	0	0	0	0	0	0	0	0	19,17	19,19	19,16	0	19,18	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	20,17	20,19	20,16	20,15	20,18	19,20	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	17,19	17,16	17,15	17,18	17,20	0

Figure 4.5 Results of the clustering in the view point of the row and column positions.

To make sure that the VCM-GAs, VCM-MAs, and VCM-KM are able to cluster the information in a binary image, we also create the other two synthetic data sets containing five and seven clusters to test them. The concept is the same as in the three-cluster image. Figure 4.6(a) and Figure 4.7(a) show the original images of the five and seven clusters, respectively. The image sizes are 34×34 and 48×48, respectively.

Figure 4.6(b) and Figure 4.7(b) show the input binary images which are randomly interchanged row and column positions in Figure 4.6(a) and Figure 4.7(a). Figure 4.6(c)-(g) demonstrate the clustering results of images in Figure 4.6(b) by using the VCM-GA1, VCM-GA2, VCM-MA1, VCM-MA2, and VCM-KM, respectively. Figure 4.7(c)-(g) show the clustering results of images in Figure 4.7(b) by using the VCM-GA1, VCM-GA2, VCM-MA1, VCM-MA2, and VCM-KM, respectively. All of results also confirm that the VCM-GAs, VCM-MAs, and VCM-KM are able to properly cluster the users and items in a binary image.

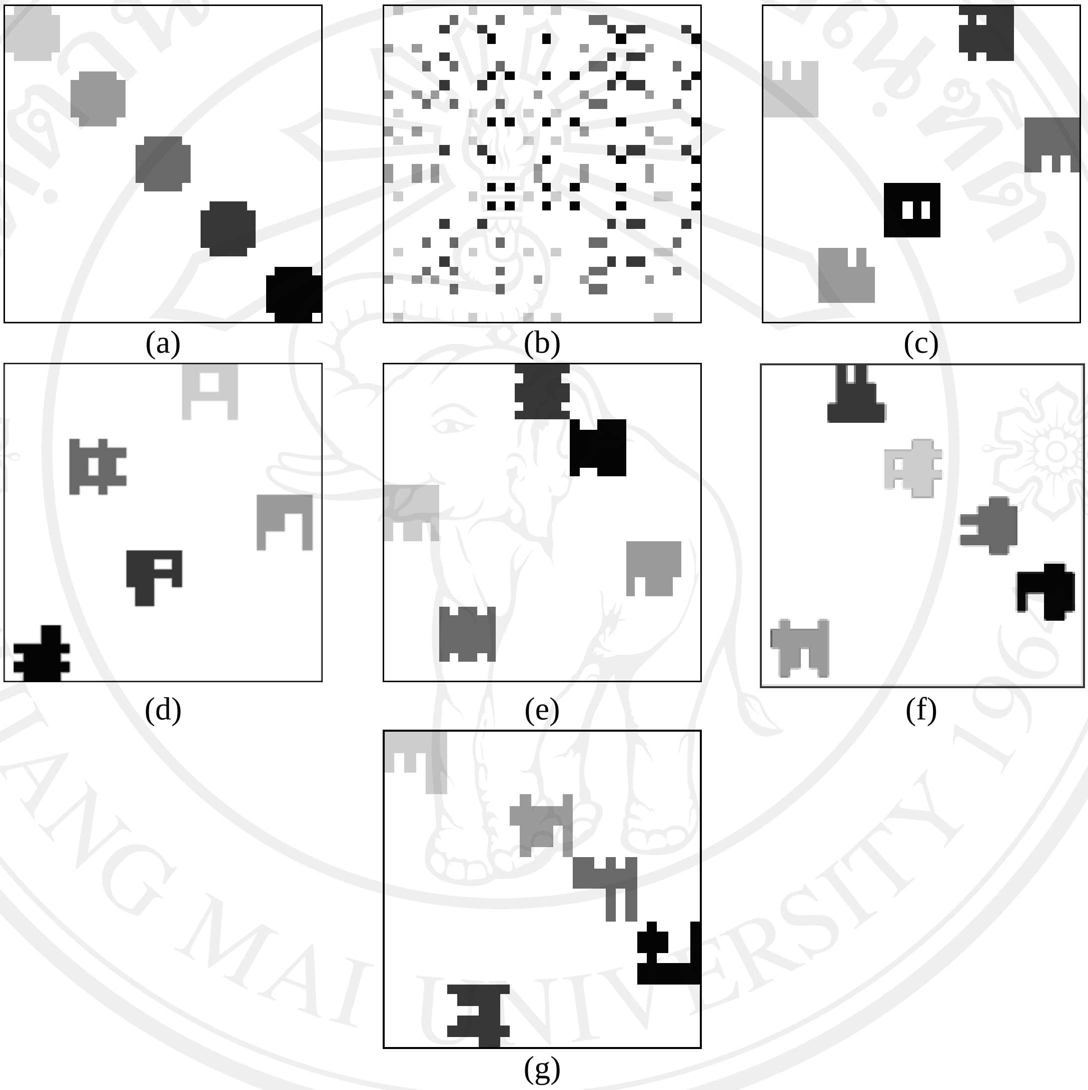

Figure 4.6(a) Original 5-cluster binary image, (b) Row-column-interchanged binary image, (c)-(g) Result of Fig 4.6(b) using the VCM-GA1, VCM-GA2, VCM-MA1, VCM-MA2, and VCM-KM.

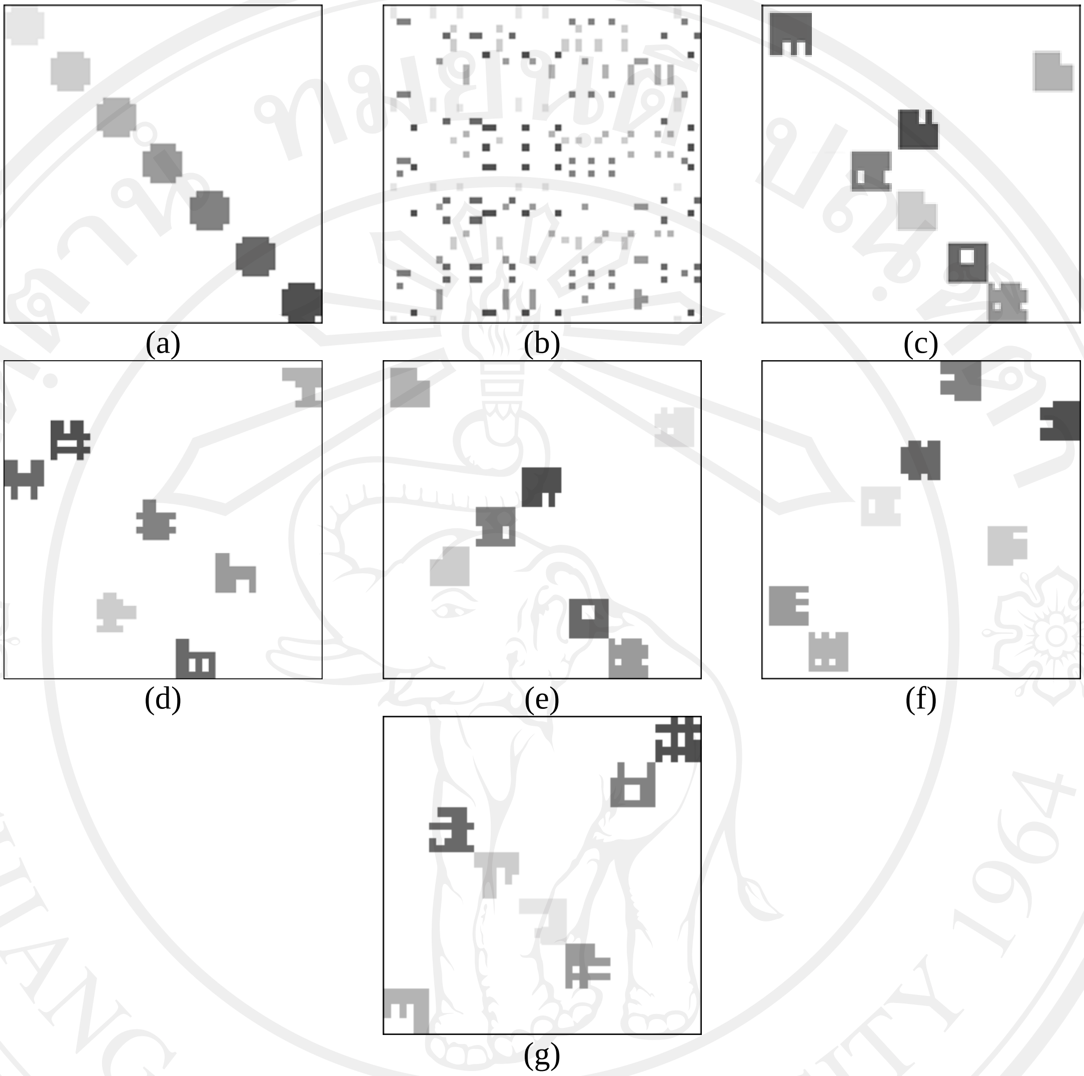

Figure 4.7(a) Original 7-cluster binary image, (b) Row-column-interchanged binary image, (c)-(g) Result of Fig 4.7(b) using the VCM-GA1, VCM-GA2, VCM-MA1, VCM-MA2, and VCM-KM.

All of results clearly show that the proposed methods are able to cluster the information in a binary image because these methods achieve three, five, and seven clusters, respectively. Although the shape and location of each cluster are different from the original one, the elements in each cluster are the same as that in the original image. Hence, it is possible to have different sizes and shapes.

4.4 Result of Clustering on Real-World Data Sets Using VCM-GAs, VCM-MAs, and VCM-KM

The clustering results using the VCM-GAs, VCM-MAs, and VCM-KM on the three synthetic data sets confirm that all of them are able to cluster the information in binary images. In this section, we apply the VCM-GAs, VCM-MAs, and VCM-KM to cluster the users and items in the real-world data sets. This process occurs after creating binary images from the purchased records. The detail for creating a binary image was described in section 4.2. In the next section, we use the derived clusters to generate the top-*N* items in the recommendation systems.

In this section, the binary images of the real-world data sets are shown. Each image is used as the input data for the process of clustering. Figure 4.8-4.12 show the binary images for the KDD, TTS, RCM, ECR, and MovieLens data sets, respectively.

Figure 4.8 KDD binary image.

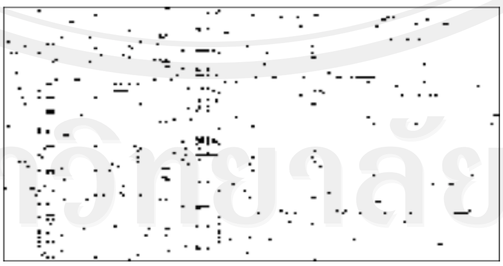

Figure 4.9 TTS binary image.

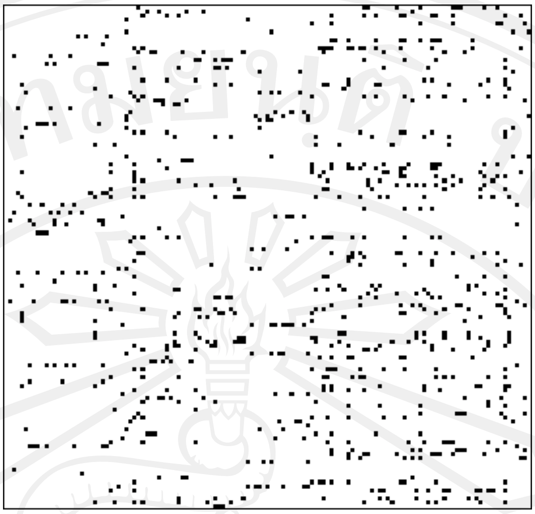

Figure 4.10 RCM binary image.

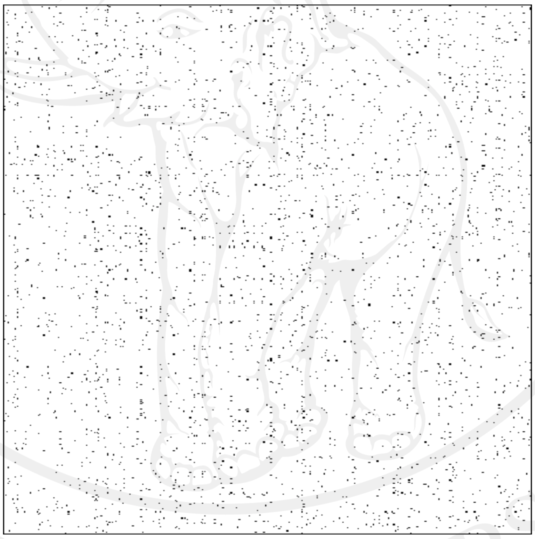

Figure 4.11 ECR binary image.

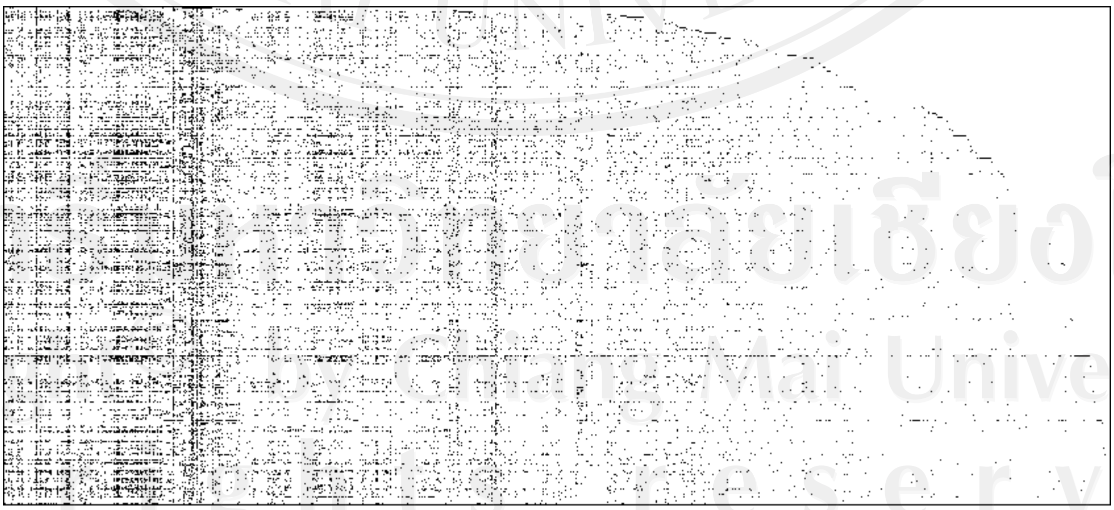

Figure 4.12 MovieLens binary image.

In this section, we apply the VCM-KM to cluster the information in each image. There are three processes to cluster the information. The first process is to cluster the users (rows) with items (columns) as the features. The second process is to cluster the items (columns) with users (rows) as the features. The third process is to group the elements in the same cluster, i.e., group users (rows) and group items (columns). The numbers of clusters in the VCM-KM is extremely difficult or impossible to determine. Hence, we fix the following numbers of clusters. In the KDD, TTS, RCM, and ECR data sets, the number of clusters is fixed to 20. In the MovieLens data set, the numbers of clusters is fixed to 100. The clustering result of Figure 4.8 is shown in Figure 4.13. Figure 4.14 illustrates the clustering result of Figure 4.9. The clustering result of Figure 4.10 is shown in Figure 4.15. Figure 4.16 demonstrates the clustering result of Figure 4.11. The clustering result of Figure 4.12 is shown in Figure 4.17.

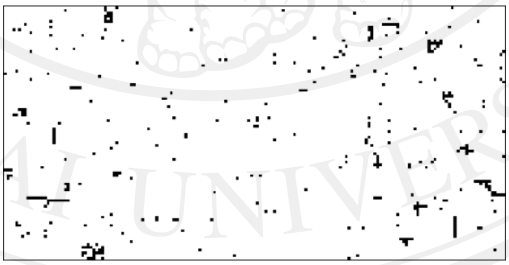

Figure 4.13 Result of Fig 4.8 on KDD image using the VCM-KM.

Figure 4.14 Result of Fig 4.9 TTS image using the VCM-KM.

Figure 4.15 Result of Fig 4.10 on RCM binary using the VCM-KM.

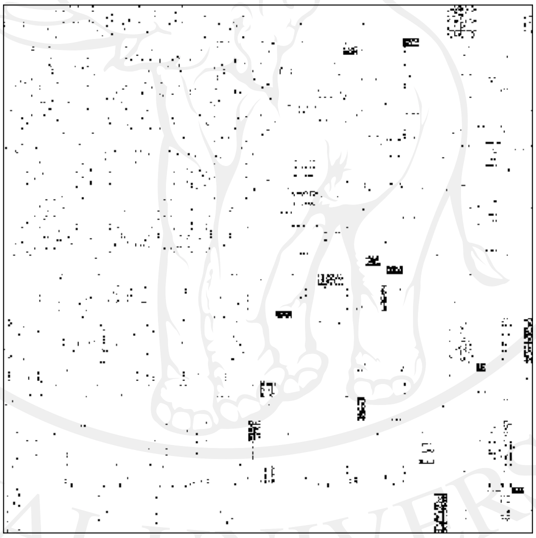

Figure 4.16 Result of Fig 4.11 on ECR image using the VCM-KM.

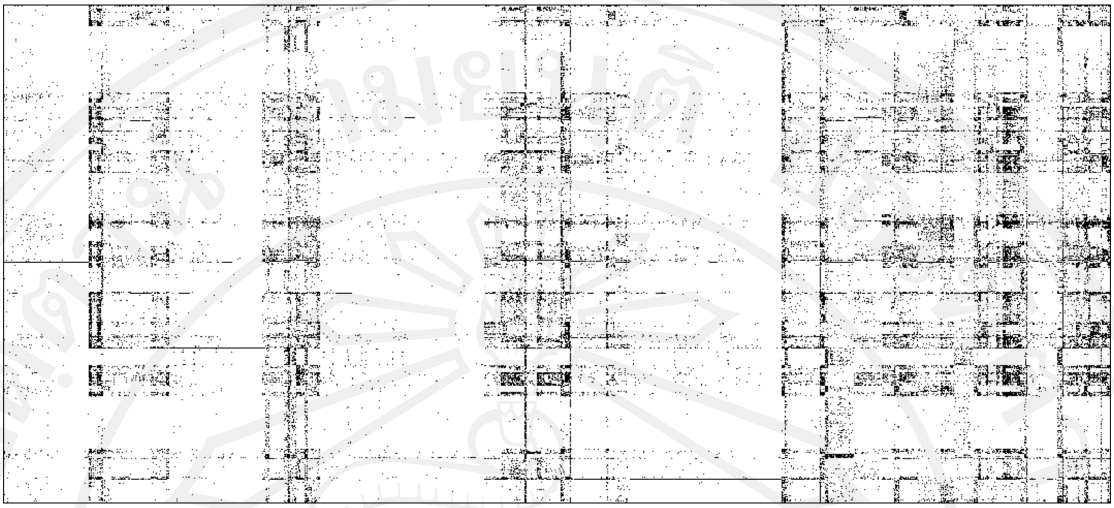

Figure 4.17 Result of Fig 4.12 on MovieLens image using the VCM-KM.

In this section, we apply the VCM-GA1 to cluster the information in each image. The clustering results of Figure 4.8, 4.9, 4.10, 4.11 and 4.12 by using the VCM-GA1 are shown in Figure 4.18, 4.19, 4.20, 4.21, and 4.22, respectively.

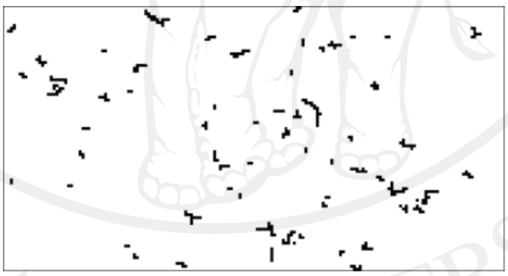

Figure 4.18 Result of Fig 4.8 on KDD image using the VCM-GA1.

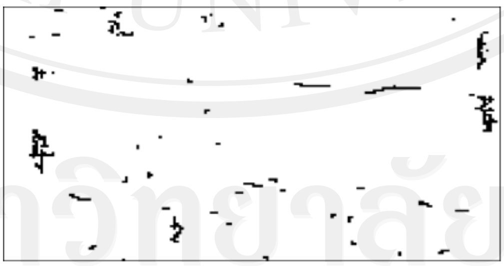

Figure 4.19 Result of Fig 4.9 on TTS image using the VCM-GA1.

Figure 4.20 Result of Fig 4.10 on RCM image using the VCM-GA1.

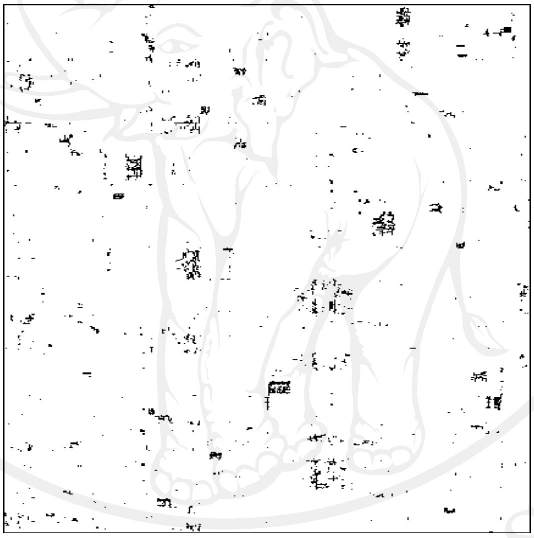

Figure 4.21 Result of Fig 4.11 on ECR image using the VCM-GA1.

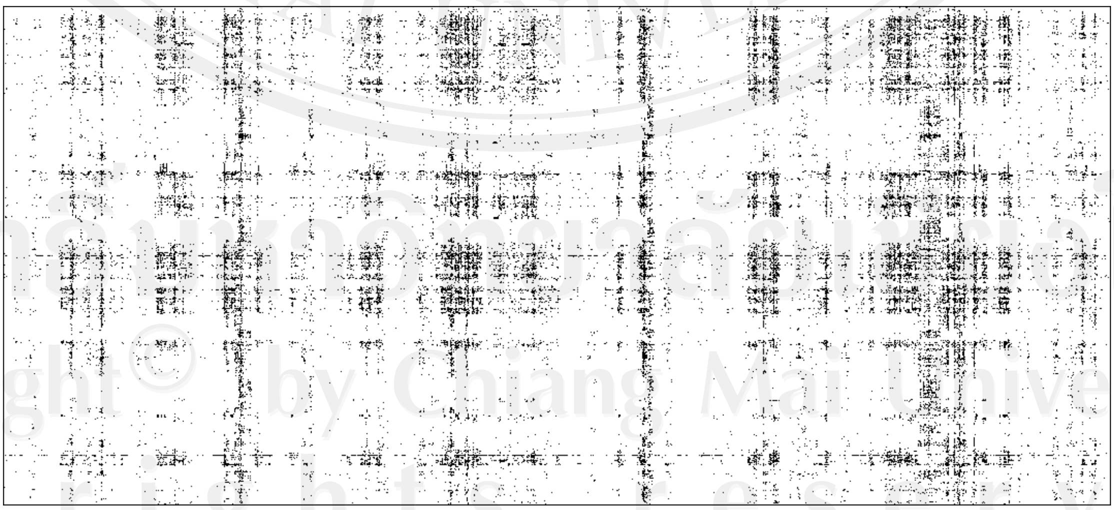

Figure 4.22 Result of Fig 4.12 on MovieLens image using the VCM-GA1.

In this section, The VCM-GA2 is applied to cluster the information in each image. Figure 4.23, 4.24, 4.25, 4.26, and 4.27 show the clustering results in Figure 4.8, 4.9, 4.10, 4.11, and 4.12 by using the VCM-GA2.

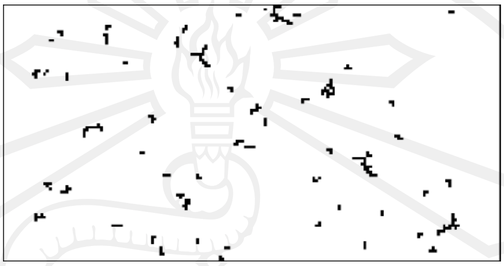

Figure 4.23 Result of Fig 4.8 on KDD image using the VCM-GA2.

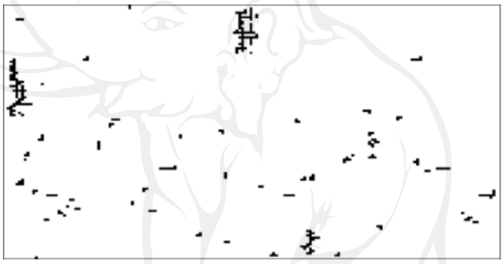

Figure 4.24 Result of Fig 4.9 on TTS image using the VCM-GA2.

Figure 4.25 Result of Fig 4.10 on RCM image using the VCM-GA2.

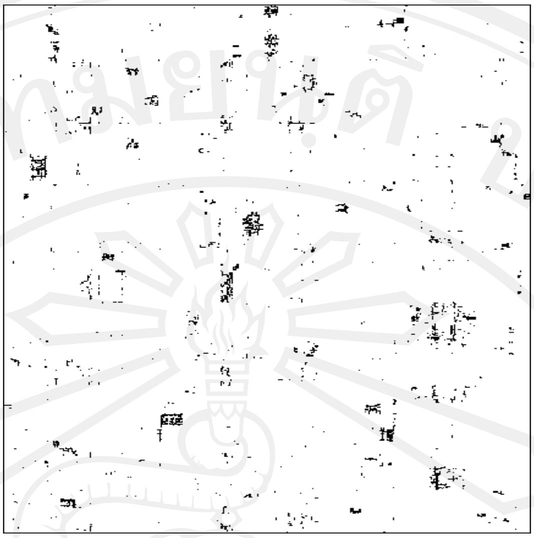

Figure 4.26 Result of Fig 4.11 on ECR image using the VCM-GA2.

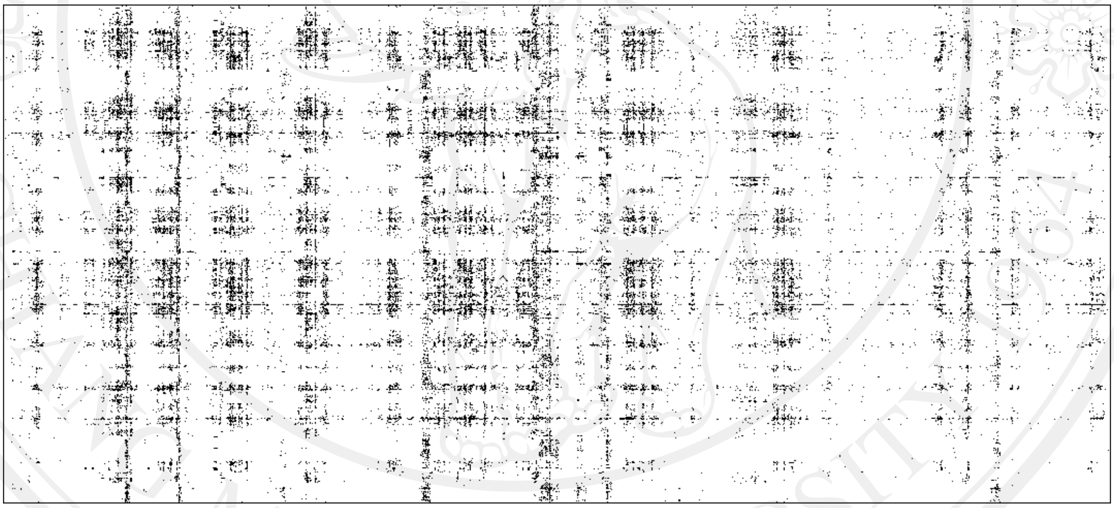

Figure 4.27 Result of Fig 4.12 on MovieLens image using the VCM-GA2.

In this section, the VCM-MA1 is applied to cluster the information in each image. Figure 4.28, 4.29, 4.30, 4.31, and 4.32 demonstrate the clustering results of Figure 4.8, 4.9, 4.10, 4.11, and 4.12 by using the VCM-MA1.

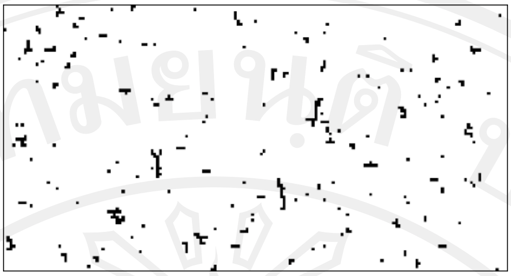

Figure 4.28 Result of Fig 4.8 on KDD image using the VCM-MA1.

Figure 4.29 Result of Fig 4.9 on TTS image using the VCM-MA1.

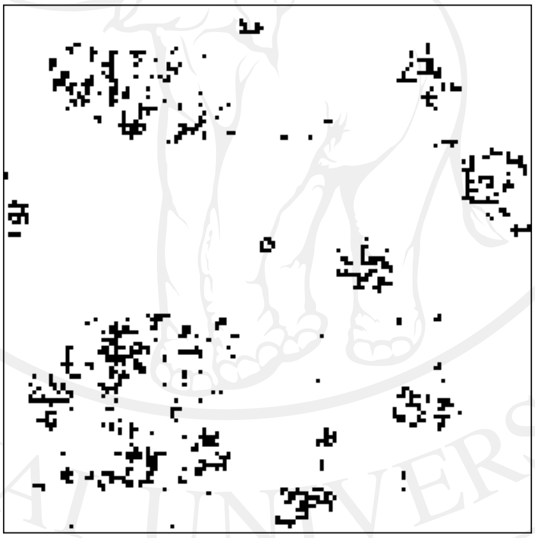

Figure 4.30 Result of Fig 4.10 on RCM image using the VCM-MA1

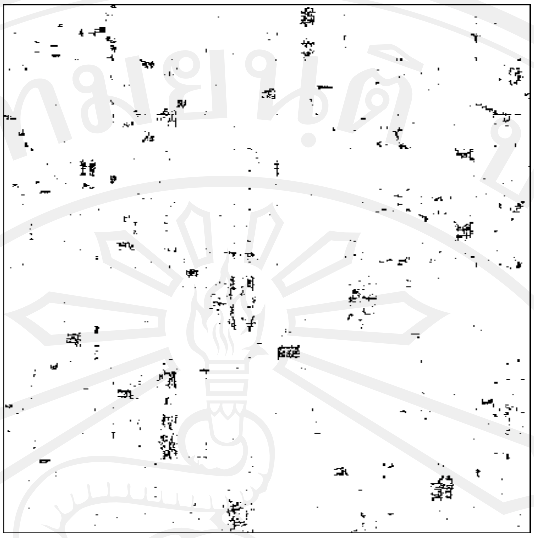

Figure 4.31 Result of Fig 4.11 on ECR image using the VCM-MA1.

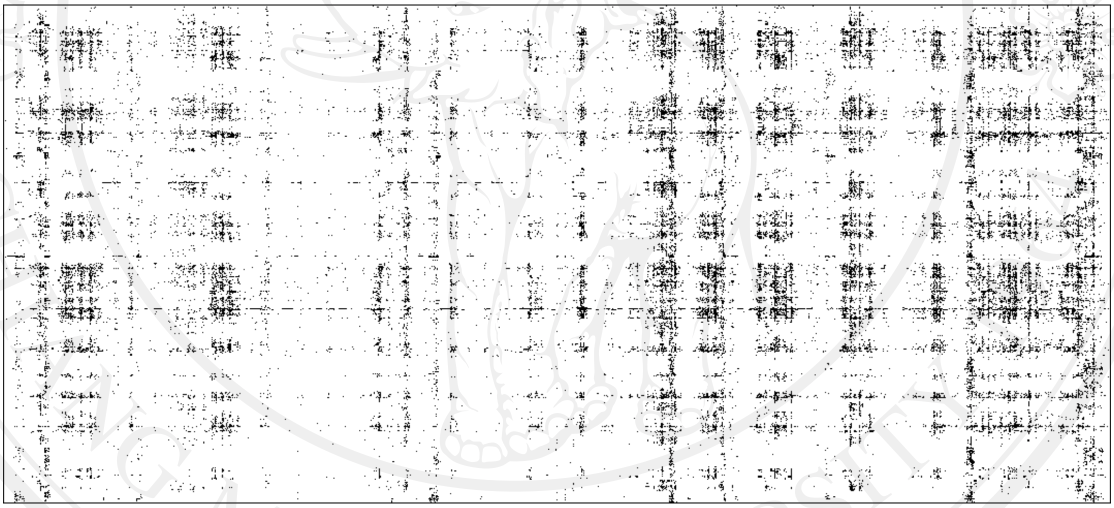

Figure 4.32 Result of Fig 4.12 on MovieLens image using the VCM-MA1.

To cluster the information in each image, the VCM-MA2 is applied. Figure 4.33 shows the clustering result of Figure 4.8. The clustering result of Figure 4.9 is shown in Figure 4.34. Figure 4.35 demonstrates the clustering result of Figure 4.10. The clustering result of Figure 4.11 is demonstrated in Figure 4.36. Figure 4.37 illustrates the clustering result of Figure 4.12.

Figure 4.33 Result of Fig 4.8 on KDD image using the VCM-MA2.

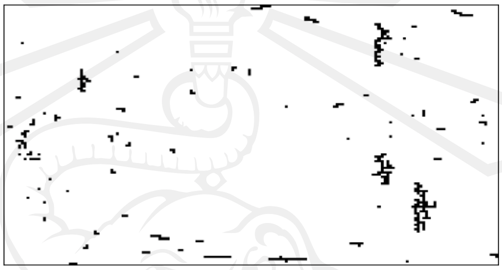

Figure 4.34 Result of Fig 4.9 on TTS image using the VCM-MA2.

Figure 4.35 Result of Fig 4.10 on RCM image using the VCM-MA2.

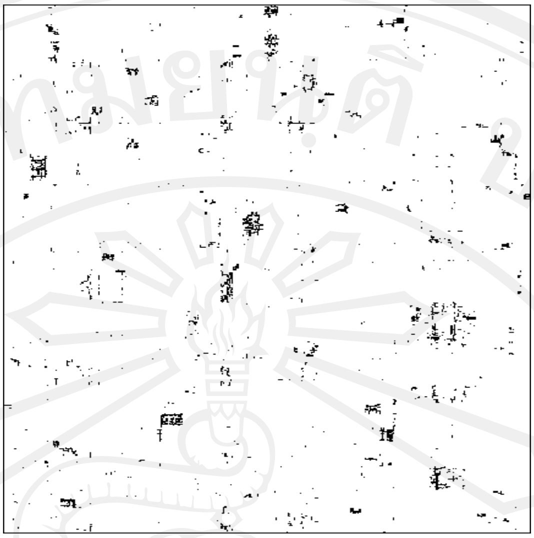

Figure 4.36 Result of Fig 4.11 on ECR image using the VCM-MA2.

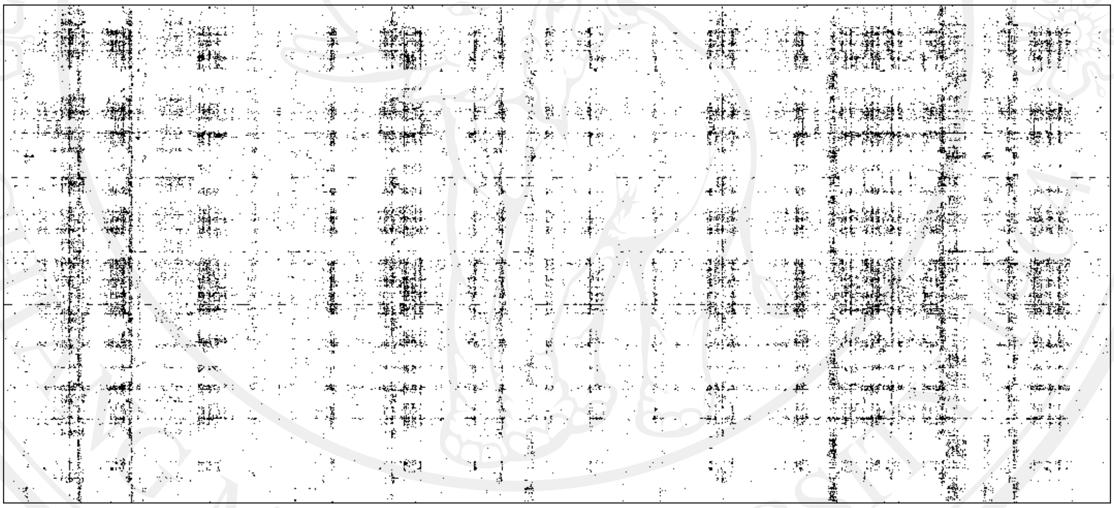

Figure 4.37 Result of Fig 4.12 MovieLens image using the VCM-MA2.

The results of clustering show that the VCM-GAs, VCM-MAs, and VCM-KM are able to cluster the users and items in five real-world data sets because the resulting images have fewer numbers of clusters than in the corresponding original images. Moreover, it is difficult to determine the actual numbers of clusters from a binary image. In the next section, the size of clusters in each image is counted.

In this section, we investigate the size of clusters in each data set before and after clustering. The size of clusters is the numbers of elements in a cluster. The size between 1 and 3, 4 and 10, 11 and 99, and more than 100 are labeled as “S”, “M”, “L”, “VL”, respectively. In this research, a cluster is a group of elements using the 4-connected neighborhood. Hence, if an image is well-clustered, then the numbers of “VL”, “L”, and “M” should be high and the number of “S” should be low.

We compare the size of clusters on the real-world data sets before and after clustering. Table 4.2 shows the sizes of clusters before clustering. Table 4.3-4.7 show the size of clusters after clustering using the VCM-KM, VCM-GA1, VCM-GA2, VCM-MA1, and VCM-MA2, respectively. After clustering using the VCR-KM, VCM-GA1, VCM-GA2, VCM-MA1, and VCM-MA2, the results show that the numbers of “VL”, “L”, and “M” increase and the numbers of “S” decrease. The results also show that the proposed methods, i.e., VCM-GAs and VCM-MAs, yield larger numbers of “VL”, “L”, and “M” than that of the VCM-KM and yield lower number of “S” than that of the VCM-KM. In the VCM-KM, the system tries to group the elements in the rows and columns. In the VCM-GAs and VCM-MAs, the systems try to decrease the numbers of the small-size clusters and try to increase the numbers of the large-size clusters. Hence, the results show that the VCM-GAs and VCM-MAs yield lower numbers of “S” than the VCM-KM. Decreasing the numbers of “S” increases the numbers of “VL”, “L”, “M”.

Table 4.2 Real-world data sets before clustering.

Data set	Size of cluster				
	VL	L	M	S	Total
KDD	0	1	9	298	308
TTS	0	3	13	277	293
RCM	0	0	0	39	681
ECR	0	0	24	2,965	2,989
MovieLens	7	1,635	5,116	36,179	42,937

Table 4.3 Real-world data sets using VCM-KM after clustering.

Data set	Size of cluster				
	VL	L	M	S	Total
KDD	0	17	11	179	207
TTS	0	10	8	168	186
RCM	0	30	39	339	408
ECR	0	78	109	1,299	1,486
MovieLens	71	1,751	3,209	22,560	27,591

Table 4.4 Real-world data sets using VCM-GA1 after clustering.

Data set	Size of cluster				
	VL	L	M	S	Total
KDD	0	10	41	111	162
TTS	0	10	15	113	138
RCM	0	44	44	151	239
ECR	4	105	157	521	787
MovieLens	77	2,237	3,463	19,635	25,412

Table 4.5 Real-world data sets using VCM-GA2 after clustering.

Data set	Size of cluster				
	VL	L	M	S	Total
KDD	0	11	36	113	160
TTS	0	8	71	63	142
RCM	0	40	35	99	174
ECR	5	112	125	411	653
MovieLens	92	2,178	3,282	17,577	23,129

Table 4.6 Real-world data sets using VCM-MA1 after clustering.

Data set	Size of cluster				
	VL	L	M	S	Total
KDD	0	10	39	116	165
TTS	0	8	46	84	138
RCM	0	40	36	102	178
ECR	5	113	125	430	673
MovieLens	90	2,190	3,315	17,706	23,301

Table 4.7 Real-world data sets using VCM-MA2 after clustering.

Data set	Size of cluster				
	VL	L	M	S	Total
KDD	0	11	43	99	153
TTS	1	11	26	35	73
RCM	1	36	33	101	171
ECR	7	108	121	442	678
MovieLens	96	2,171	3,239	17,032	22,538

The results show that the total number of clusters before clustering is larger than that the number of clusters after clustering by using the VCM-KM, VCM-GAs, and VCM-MAs. Hence, the VCM-KM, VCM-GAs, and VCM-MAs are able to cluster the users and items in the five real-world data sets.

In this section, we used the Dunn's index [46] to validate the proposed clustering methods. The Dunn's index attempts to indentify compact and well-separated clusters. It is defined as

$$DN = \min_{n=1,\dots,c}\left\{\min_{m=n+1,\dots,c}\left(\frac{d(c_i,c_j)}{\max_{k=1,\dots,c} diam(c_k)}\right)\right\}, \tag{4.2}$$

$$d(c_i,c_j) = \min_{x\in c_i, y\in c_j}(d(x,y)), \tag{4.3}$$

$$diam(C) = \max_{x,y\in C} d(x,y), \tag{4.4}$$

$d(c_i,c_j)$ is the dissimilarity function between two clusters c_i and c_j. $diam(C)$ is the diameter of cluster C. It is used to measure the dispersion of the clusters. The large values of the Dunn's index indicate the compact and well-separated clusters. The Euclidean distance is used to calculate the dissimilarity between clusters and the diameter of clusters. Table 4.8 shows the Dunn's index on five real-world data sets.

Table 4.8 Dunn's index on five real-world data sets.

Method	Data set				
	KDD	TTS	RCM	ECR	MovieLens
Before clustering	0.8127	0.3453	0.9428	1.0525	0.1753
VCM-KM	0.3156	0.3063	0.0673	0.1907	0.0333
VCM-GA01	0.1085	0.2220	0.0534	0.1827	0.0158
VCM-GA02	0.1387	0.2543	0.0562	0.1315	0.0101
VCM-MA01	0.1209	0.2811	0.0562	0.1623	0.0141
VCM-MA02	0.1240	0.1732	0.0448	0.1443	0.0087

In the Dunn's index, the distance between the clusters is expected to be large and the diameter of the clusters is expected to be small. However, the results show that the Dunn's index is not able to indicate the compact and well-separated clusters in our proposed clustering methods. The first reason is that the distance between the clusters cannot represent the well-separated clusters although the distance between clusters is large. The second reason is that the Dunn's index uses the largest diameter of clusters to normalize the distance between clusters but the largest diameter of clusters before clustering is smaller than that after clustering in the proposed clustering methods. Hence, the Dunn's index is large. Table 4.8 shows the results of the Dunn's index. It is clear that the Dunn's index is not suitable for validating our proposed clustering methods.

When the Dunn's index on the five clustering methods are compared, the results show that the indices before clustering are larger than that after clustering on all five real-world data sets. It is not surprising to get these results. The Dunn's index will likely be large if the largest cluster is small, i.e., small $diam(C)$. It cannot tell how well the clusters are formed. For each data set, it is clearly seen that the data before clustering are scattered. That results in several small clusters. The diameter of the largest cluster is therefore small. After the clustering is performed, the clusters are well-grouped. That results in the bigger clusters. The Dunn's index is therefore smaller. The results also show that the values of Dunn's indices after clustering by the VCR-KM are larger than that after clustering by the VCM-GAs and VCM-MAs. The clustering results using the VCM-KM on the five real-world data sets (see Figures 4.13-4.16) clearly show that the largest diameter is smaller than that after clustering using the VCM-GAs and VCM-MAs (see Figures 4.18-4.37). We also found that the values of Dunn's indies after clustering based on the GA and MA with the fitness function in eq.(3.4) are smaller than that with the fitness function in eq.(3.1). This is because the fitness function in eq.(3.4) tries to cluster without any shape consideration, whereas the fitness function in eq.(3.1) tries to do that the compactness the compactness based on of each clusters. Moreover, the values of Dunn's index after clustering by the MA are smaller than that after clustering by the GA. This is because the diameter of cluster based on the GA is smaller than that of the MA.

4.5 Experimental Results of Top-5 Recommendation Systems on Real-World Data Sets

In the experiment, five real-world data sets and F-measure were used to evaluate the top-*N* recommendation systems. The number of the top-*N* recommendation systems is set to 5 (i.e., top-5 recommendation). We compared the performance of the 12 proposed methods (i.e., VCR-GA1, VCR-GA1-UB, VCR-GA1-IB, VCR-GA2, VCR-GA2-UB, VCR-GA2-IB, VCR-MA1, VCR-MA1-UB, VCR-MA1-IB, VCR-MA2, VCR-MA2-UB, and VCR-MA2-IB) with the VCR-KM, UB, IB, and FB.

4.5.1 Top-5 Recommendation Systems on KDD Data Set

Table 4.9 is the example of the top-*N* recommendation results on the KDD data set. In this data set, the training and test sets are divided by using the ten-fold cross validation method. Each of the training set is the input of VCM-GAs to find the optimized clusters. In this table, the column labeled as "*Rec*" is the number of the recommended items. The column labeled as "*B*" is the number of the hidden items. The column labeled as "*Hit*" is the number of hits. There are ten customers in the test set. For example, considering customer #3 and one item in the basket, the system recommends five items. The numbers of hidden items in the active customer's basket are two. The number of hits sets is one. Hence, the precision, recall, and F1 are 0.20, 0.50, and 0.29, respectively. In the same customer with two items in the basket, the numbers of recommended items are five. The number of hidden items is one. The number of hits is one. So, the precision, recall, and F1 are 0.20, 1.00, and 0.33, respectively.

Table 4.9 Example of top-*N* recommendation on the KDD data set.

Test Set	Number of items in the basket											
	One item						Two items					
	Rec	*B*	*Hit*	*Pre*	*Re*	*F1*	*Rec*	*B*	*Hit*	*Pre*	*Re*	*F1*
1	5	2	0	0.00	0.00	0.00	5	1	0	0.00	0.00	0.00
2	2	2	0	0.00	0.00	0.00	3	1	0	0.00	0.00	0.00
3	5	2	1	0.20	0.50	0.29	5	1	1	0.20	1.00	0.33
4	5	3	0	0.00	0.00	0.00	5	2	0	0.00	0.00	0.00
5	5	2	0	0.00	0.00	0.00	5	1	0	0.00	0.00	0.00
6	5	3	0	0.00	0.00	0.00	5	2	0	0.00	0.00	0.00
7	3	2	0	0.00	0.00	0.00	5	1	0	0.00	0.00	0.00
8	2	2	0	0.00	0.00	0.00	5	1	0	0.00	0.00	0.00
9	5	2	0	0.00	0.00	0.00	5	1	0	0.00	0.00	0.00
10	4	2	1	0.25	0.50	0.33	5	1	1	0.20	1.00	0.33

Parameter setting of the GA and MA algorithms

The parameters of the fitness function in the GA and MA are ones of the factors in the process. In the global search of the GA and MA, We tested the parameters of global search and selected the optimal parameters are selected using our methods.

A synthetic data set was created for testing parameters in the global search. This data set consists of 20 clusters with 137 rows and 137 columns. It was mapped into a binary image. The input binary image was then created by randomly interchange the row and column positions. Then, we used this image to test the global search method, i.e., the genetic algorithm.

In this research, the parameters are: population size = 80, mutation rate = 0.1. For the crossover rate, we tested it at 1, 0.7, 0.6, and 0.5. The result of testing shows that the crossover rate of 0.6 is the best as shown in Table 4.10.

Table 4.10 Effectiveness of the parameter setting of the global search.

Population size	Mutation rate	Crossover rate	Converted iterations
80	0.1	1	840
80	0.1	0.7	640
80	0.1	0.6	620
80	0.1	0.5	1,000

In our methods, we have two objective functions of the global search. The first objective function focuses on the compactness and the number of clusters in a binary image. The second objective function has four parameters. The third parameter is *k*-cluster level that defined by user. For example, *k*-cluster level is 5. After the sorted size of clusters in descending order, the 5^{th} cluster was determined. The image with the largest 5^{th} cluster has the highest priority. However, in the real-world data sets, the size of clusters is not known. So, we tested the *k*-clusters sensitivity for both the VCM-GA2 and VCM-MA2.

To test *k*-clusters level, we created a synthetic data set with five clusters in a binary image. We tested *k* at 2, 5, and 7. Figure 4.38-4.39 shows the sensitivity of the *k*-cluster level on the synthetic data set using the VCM-GA2 and VCM-MA2. The results of both figures show that the *k*-cluster level has no effect with the VCM-GA2 and VCM-MA2. However, the *k*-cluster level has an effect with the time of clustering process. If the *k*-cluster level is set close to the actual number of clusters, the time of the clustering process is the best.

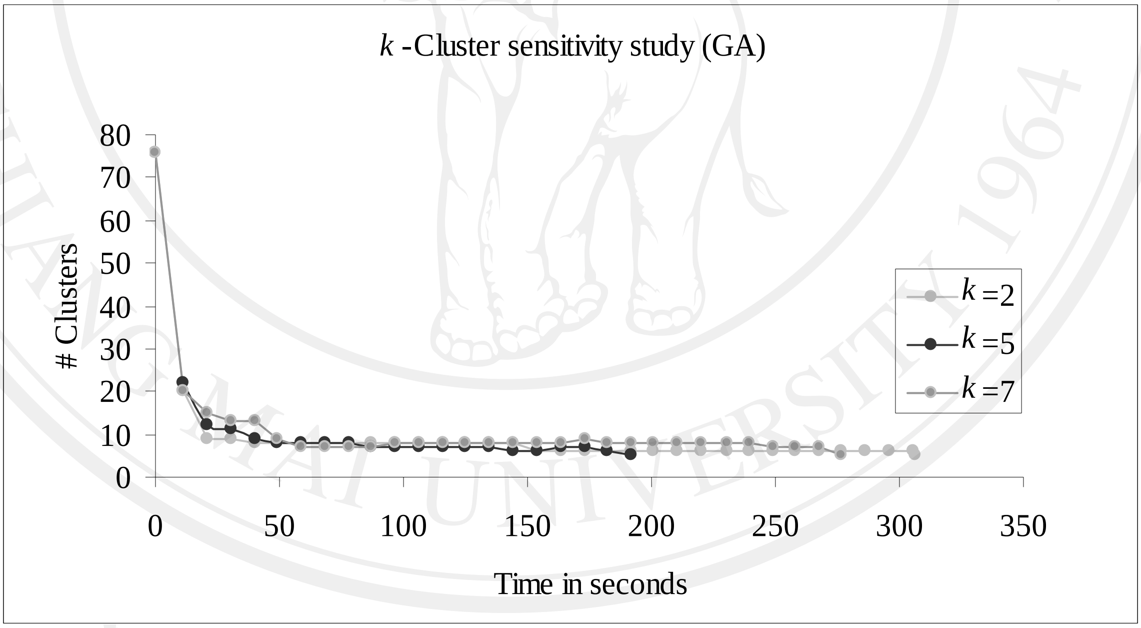

Figure 4.38 The *k*-cluster level sensitivity study in the VCM-GA2.

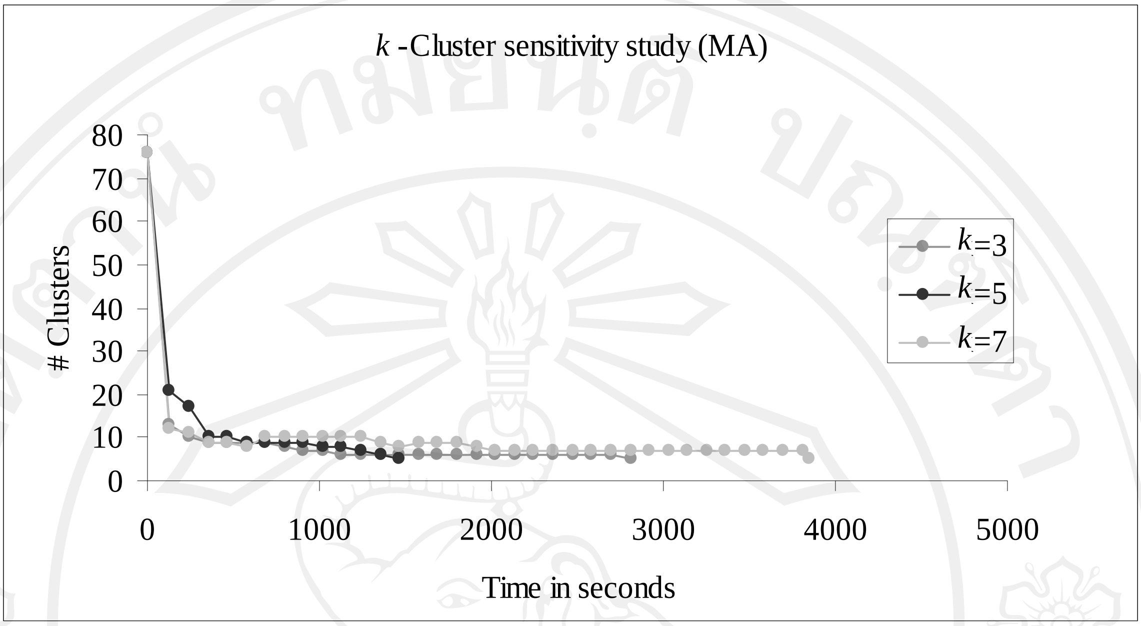

Figure 4.39 The *k*-cluster level sensitivity study in the VCM-MA2.

Table 4.11 shows the performance of the recommendation systems. There are sixteen methods that were compared on the KDD data set. The F-measure was used to evaluate the recommendation performance. The details of F-measure were described in section 3.4. The common methods including the VCR-KM, UB, IB, and FB were used as the baseline. The details of the common methods were described in section 2.2. Our methods are VCR-GA1, VCR-GA1-UB, VCR-GA1-IB, VCR-GA2-UB, VCR-GA2-IB, VCR-MA1, VCR-MA1-UB, VCR-MA1-IB, VCR-MA2, VCR-MA2-UB, and VCR-MA2-IB. The details of our methods were described in section 3.3. To evaluate the performance of the recommendation systems, the ten-fold cross validation was used. In the test set, one and two items were put into the basket.

Table 4.11 Recommendation performance comparison on KDD data set when one and two items are selected into the basket (evaluated on test sets of 10-fold cross validation).

Method	Number of items in the basket					
	One item			Two items		
	Precision	Recall	F1	Precision	Recall	F1
UB	0.03	0.07	0.04	0.05	0.11	0.07
IB	0.04	0.07	0.05	0.04	0.11	0.06
FB	0.02	0.03	0.02	0.01	0.05	0.02
VCR-KM	0.05	0.06	0.05	0.09	0.08	0.08
VCR-GA1	0.05	0.06	0.05	0.07	0.15	0.10
VCR-GA1-UB	0.04	0.07	0.05	0.06	0.15	0.09
VCR-GA1-IB	0.05	0.08	0.06	0.07	0.19	0.10
VCR-GA2	0.07	0.09	0.08	0.09	0.13	0.11
VCR-GA2-UB	0.05	0.09	0.06	0.05	0.19	0.08
VCR-GA2-IB	0.05	0.09	0.06	0.05	0.12	0.07
VCR-MA1	0.05	0.06	0.05	0.08	0.17	0.11
VCR-MA1-UB	0.04	0.07	0.05	0.07	0.16	0.10
VCR-MA1-IB	0.05	0.08	0.06	0.08	0.20	0.11
VCR-MA2	0.07	0.10	0.08	0.09	0.16	0.12
VCR-MA2-UB	0.06	0.09	0.07	0.06	0.12	0.08
VCR-MA2-IB	0.05	0.10	0.07	0.05	0.14	0.07

The precision results with the one item in the basket demonstrate that our methods have better performance than the UB, IB, and FB methods. In among the common methods, the results show that the VCR-KM method performs the better. The VCR-GA2, VCR-MA2, and VCR-MA2-UB methods have better performance than the common methods. The VCR-GA1, VCR-GA1-IB, VCR-GA2-UB, VCR-GA2-IB, VCR-MA1, VCR-MA1-IB, and VCR-MA2-IB have the same performance as the VCR-KM method. In the recommendation systems based on the GA with the fitness function in eq.(3.1), i.e., the VCR-GA1, VCR-GA1-UB, and VCR-GA1-IB, the VCR-GA1-IB is the best performance. In the recommendation systems based on the GA with the fitness function in eq.(3.4), i.e., the VCR-GA2, VCR-GA2-UB, and VCR-GA2-IB, the performances of these recommendation systems are the same.

However, the performances of the recommendation based on the GA with the fitness function in eq.(3.4) have better performance than the recommendation systems based on the GA with the fitness function in eq.(3.1). In the recommendation systems based on the MA where the fitness functions are as in eq.(3.1) and (3.4), the results show that the VCR-MA2 has the best performance.

The recall results with one item in the basket show that the VCR-GA1-IB, VCR-GA2, VCR-GA2-UB, VCR-GA2-IB, VCR-MA1-IB, and VCR-MA2 methods have better performance level than the common methods. Among the recommendation systems based on the GA with the fitness function in eq.(3.1), i.e., VCR-GA1, VCR-GA1-UB, and VCR-GA1-IB, the VCR-GA1-IB performs the best. Among recommendation systems based on the GA with the fitness function in eq.(3.4), i.e., VCR-GA2, VCR-GA2-UB, and VCR-GA2-IB, the VCR-GA2-IB performs the best. In the recommendation systems based on the MA with the fitness functions in eq.(3.1) and (3.4), i.e., VCR-MA1, VCR-MA1-UB, VCR-MA1-IB, VCR-MA2, VCR-MA2-UB, and VCR-MA2-IB, the results show that the VCR-MA2 performs the best.

The F1 results with one item in the basket show that the proposed methods perform better than the common methods. In the recommendation systems based on the GA with the fitness function in eq.(3.1), i.e., VCR-GA1, VCR-GA1-UB, and VCR-GA1-IB, the VCR-GA1-IB performs the better. Among the recommendation systems based on the GA with the fitness function in eq.(3.4), i.e., VCR-GA2, VCR-GA2-UB and VCR-GA2-IB, the VCR-GA2-IB performs the best . Among recommendation systems based on the MA with the fitness functions in eq.(3.1) and

(3.4), i.e., VCR-MA1, VCR-MA1-UB, VCR-MA1-IB, VCR-MA2, VCR-MA2-UB, VCR-MA2-IB, the VCR-MA2 has the best performance.

The precision results with two items in the basket show that the proposed methods perform better than the common methods. Among the recommendation systems based on the GA with the fitness function in eq.(3.1), i.e., VCR-GA1, VCR-GA1-UB, and VCR-GA1-IB, the VCR-GA1-UB has the lowest. Among the recommendation systems based on the GA with the fitness function in eq.(3.4), i.e., VCR-GA2, VCR-GA2-UB, and VCR-GA2-IB, the VCR-GA2-IB has the best. In the recommendation based on the MA with the fitness function in eq.(3.1) and (3.4), i.e., VCR-MA1, VCR-MA1-UB, VCR-MA1-IB, VCR-MA2, VCR-MA2-UB, VCR-MA2-IB, the results show that the VCR-MA2 has the best.

The recall results with items in the basket show that the proposed methods perform better than the common methods. Among the recommendation systems based on the GA with the fitness function in eq.(3.1), i.e., VCR-GA1, VCR-GA1-UB, and VCR-GA-IB, the results show that the VCR-GA-IB has the best. Among the recommendation systems based on the GA with the fitness function in eq.(3.4)., i.e., VCR-GA2, VCR-GA2-UB, and VCR-GA2-IB, the results also show that the VCR-GA2-IB has the best. Among the recommendation systems based on the MA with the fitness function in eq.(3.1) and (3.4), i.e., VCR-MA1, VCR-MA1-UB, VCR-MA1-IB, VCR-MA2, VCR-MA2-UB, and VCR-MA2-IB, the results also show that the VCR-MA1-IB and VCR-MA2 have the best performance.

The F1 results with two items in the basket show that the VCR-GA1-IB, VCR-GA2, VCR-GA2-UB, VCR-GA2-IB, VCR-MA1-IB, VCR-MA2, and VCR-MA2-UB methods perform better than the common methods. In the common methods, the VCR-KM method performs better than the UB, IB, and FB. Among the recommendation systems based on the GA with the fitness function in eq.(3.1), i.e., VCR-GA1, VCR-GA1-UB, and VCR-GA1-IB, the VCR-GA1-IB has the best performance. Among the recommendation systems based on the GA with the fitness function in eq.(3.4), i.e., VCR-GA2, VCR-GA2-UB, and VCR-GA2-IB, the VCR-GA2-IB has the best performance. Among the recommendation systems based on the MA with the fitness function in eq.(3.1) and (3.4), i.e., VCR-MA1, VCR-MA1-UB, VCR-MA1-IB, VCR-MA2, VCR-MA2-UB, VCR-MA2-IB, the results show that the VCR-MA2 has the best performance.

The results clearly show that among the recommendation systems based on the GA with the fitness functions in eq.(3.1) and (3.4), the recommendation systems based on the GA with the fitness function in eq.(3.4) has the best performance level. Moreover, the recommendation systems based on the MA with the fitness function in eq.(3.4) yield better performance levels compared with the recommendation systems based on the GA and the common methods. The results clearly show that the fitness function in eq.(3.4) yields a higher performance level than the fitness function in eq.(3.1). In addition, the results clearly show that the recommendation systems based on the MA (that is the extension of the GA) yield higher performance levels than the recommendation systems based on the GA.

1. Top-5 Recommendation Systems on TTS Data Set

Table 4.12 shows the performance of the recommendation systems for comparing the proposed methods and the common methods on the TTS data set. There are four common methods and twelve proposed methods as before. Only one item is put into the basket because the frequencies of purchasing are too low. The detail of this data set is as described in section 4.2.

Table 4.12 Recommendation performance comparison on TTS data set when one item is selected into the basket (evaluated on test sets of 10-fold cross validation).

Method	Precision	Recall	F1
UB	0.13	0.24	0.17
IB	0.09	0.19	0.12
FB	0.07	0.16	0.10
VCR-KM	0.17	0.20	0.18
VCR-GA1	0.14	0.34	0.18
VCR-GA1-UB	0.15	0.37	0.19
VCR-GA1-IB	0.15	0.39	0.20
VCR-GA2	0.28	0.48	0.30
VCR-GA2-UB	0.19	0.50	0.25
VCR-GA2-IB	0.18	0.47	0.24
VCR-MA1	0.14	0.36	0.20
VCR-MA1-UB	0.14	0.37	0.20
VCR-MA1-IB	0.15	0.38	0.21
VCR-MA2	0.29	0.49	0.36
VCR-MA2-UB	0.20	0.52	0.29
VCR-MA2-IB	0.19	0.48	0.27

In the common methods, i.e., VCR-KM, FB, UB, and IB, the precision results show that VCR-KM has the better performance. In the recommendation systems based on the GA with the fitness function in eq.(3.1), i.e., VCR-GA1, VCR-GA1-UB, and VCR-GA1-IB, the VCR-GA1 has the lowest performance. The recommendation systems based on the GA with the fitness function in eq.(3.4), i.e., VCR-GA2, VCR-GA2-UB, and VCR-GA2-IB, the VCR-GA2 has the highest performance levels. Comparison among the recommendation systems based on the GA shows that the systems based on the GA with the fitness function in eq.(3.4) have better performance level than that of the systems based on the GA with the fitness function in eq.(3.1). It is clear that the fitness function in eq.(3.4) works better. In the recommendation systems based on the MA with the fitness functions in eq.(3.1) and (3.4), i.e., VCR-MA1, VCR-MA1-UB, VCR-MA1-IB, VCR-MA2, VCR-MA2-UB, and VCR-MA2-IB, the VCR-MA2 has the best performance. The precision results also show that the systems based on the fitness function in eq.(3.4) yield better performance levels than the systems based on the fitness function in eq.(3.1). Moreover, the systems based on the MA yield the better performance levels than the systems based on the GA.

The recall results show that the proposed methods perform better than the common methods. In the common methods, the results show that the UB has the highest performance level. Among the recommendation systems based on the GA with the fitness function in eq.(3.1), i.e., VCR-GA1, VCR-GA1-UB, and VCR-GA1-IB, the VCR-GA1-IB has the highest performance level. Among the recommendation systems based on the GA with the fitness functions in eq.(3.4), i.e., VCR-GA2, VCR-GA2-UB, and VCR-GA2-IB, the VCR-GA2-IB has the highest performance level. Comparing the results based on the GA with the fitness functions in eq.(3.1) and (3.4), shows that of the systems based on the GA with the fitness function in eq.(3.4) yield the higher performance level than that of the systems based on the GA with the fitness function in eq.(3.1). Comparing the systems results based on the GA and the MA with the fitness function in eq.(3.4), the results show that the systems based on the MA yield the better performance levels than the performance of the systems based on the GA.

The F1 results show that the VCR-KM yields the highest performance level among the common methods. In the recommendation systems based on the GA with the fitness function in eq.(3.1), the VCR-GA1-IB yields the highest performance level. Among the recommendation systems based on the GA with the fitness function in eq.(3.4), the VCR-GA2 yields the highest performance level. Comparing the systems based on the GA with the fitness functions in eq.(3.1) and (3.4), the systems based on the MA yield the better performance level than the performance of the systems based on the GA.

4.5.2 Top-5 Recommendation Systems on ECR Data Set

For the ECR data set, we tested the neighborhood size on the user-based and item-based recommendation systems. To determine the effect of neighborhood size, we performed the experiment by using the user and item-neighborhoods of sizes 10, 20, 30, and 50 (i.e., k = 10, 20, 30, 50). We also tested the effect of neighborhood size on the ECR data set for top-5 recommendation systems. In the test set, the items 1 to 5 were randomly selected into the basket. From the results of the neighborhood sensitivity as shown in Figure 4.40-4.45, we found that the user-neighborhood size = 50 and the item-neighborhood size = 10 are suitable.

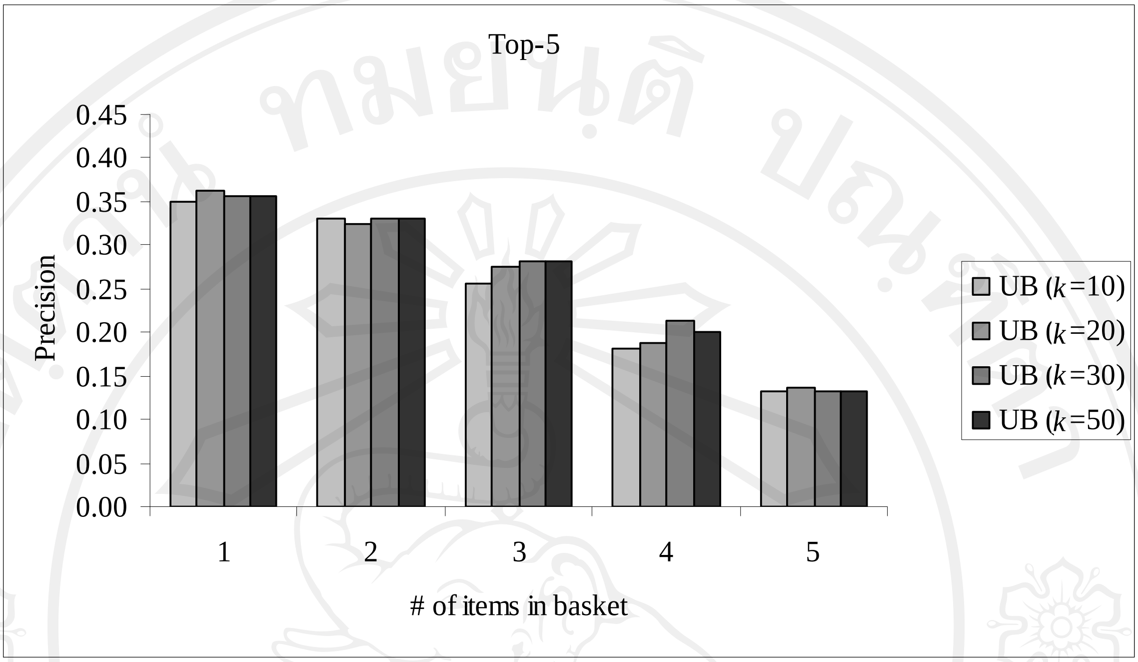

Figure 4.40 Average precision sensitivity of neighborhood size using user-based recommendation system on ECR data set.

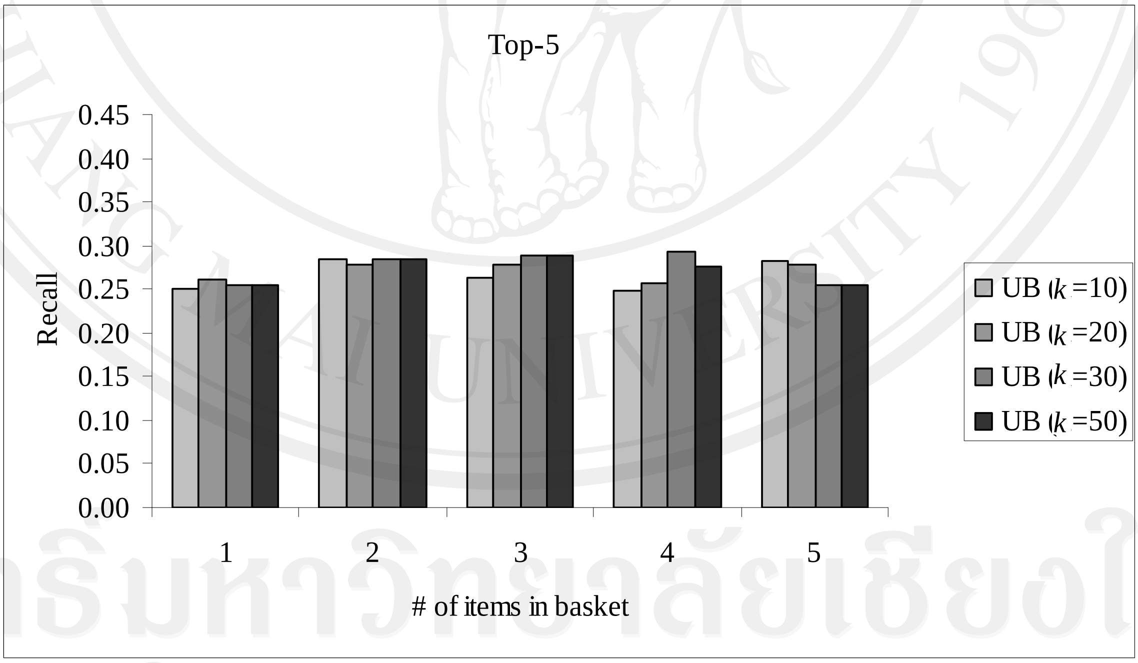

Figure 4.41 Average recall sensitivity of neighborhood size using user-based recommendation system on ECR data set.

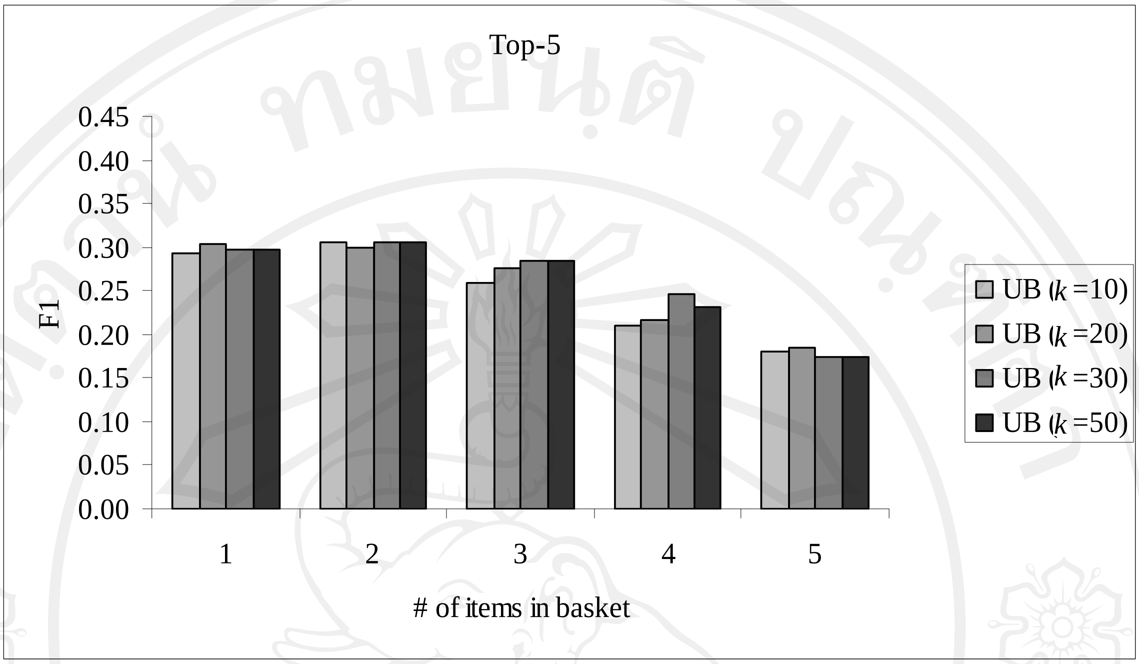

Figure 4.42 Average F1 sensitivity of neighborhood size using user-based recommendation system on ECR data set.

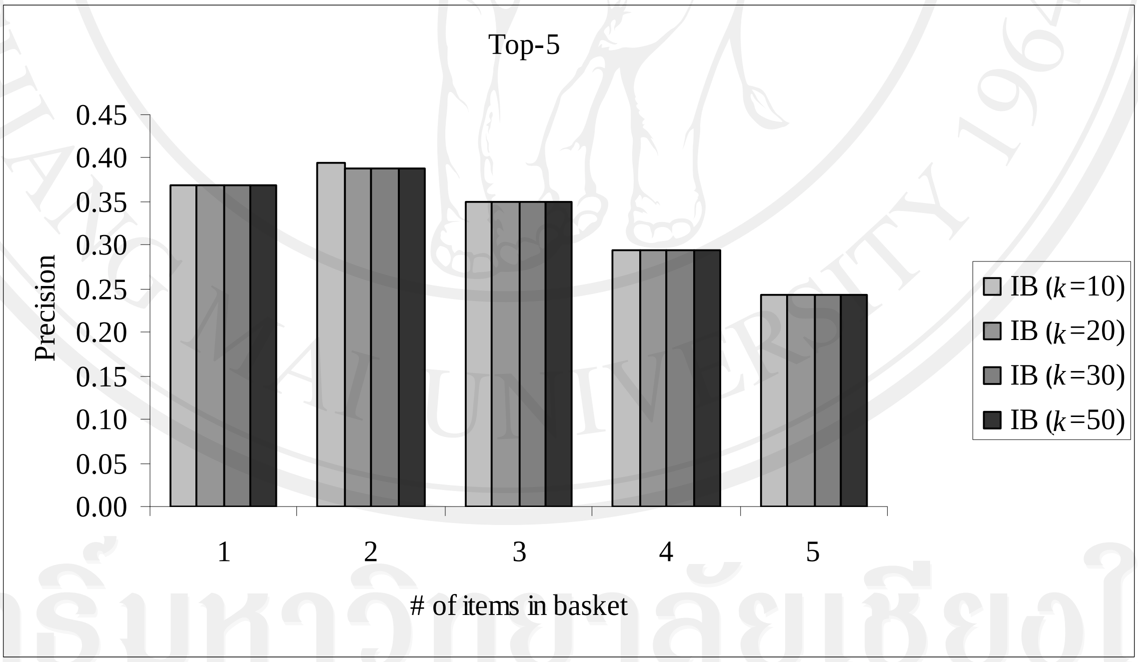

Figure 4.43 Average precision sensitivity of neighborhood size using item-based recommendation system on ECR data set.

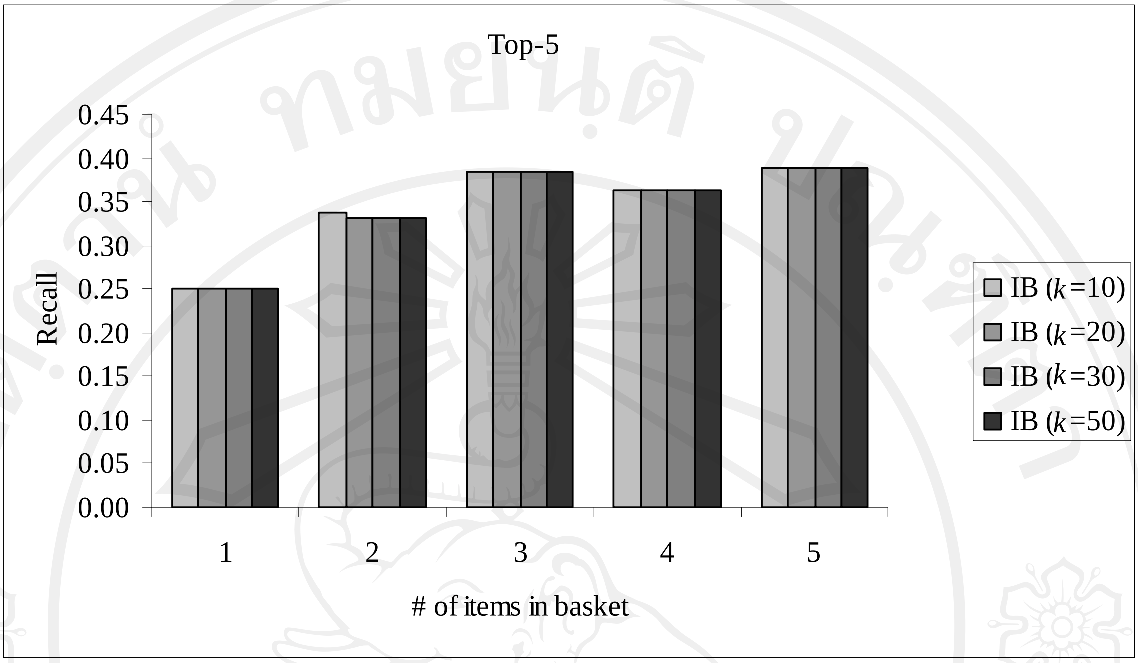

Figure 4.44 Average recall sensitivity of neighborhood size using item-based recommendation system on ECR data set.

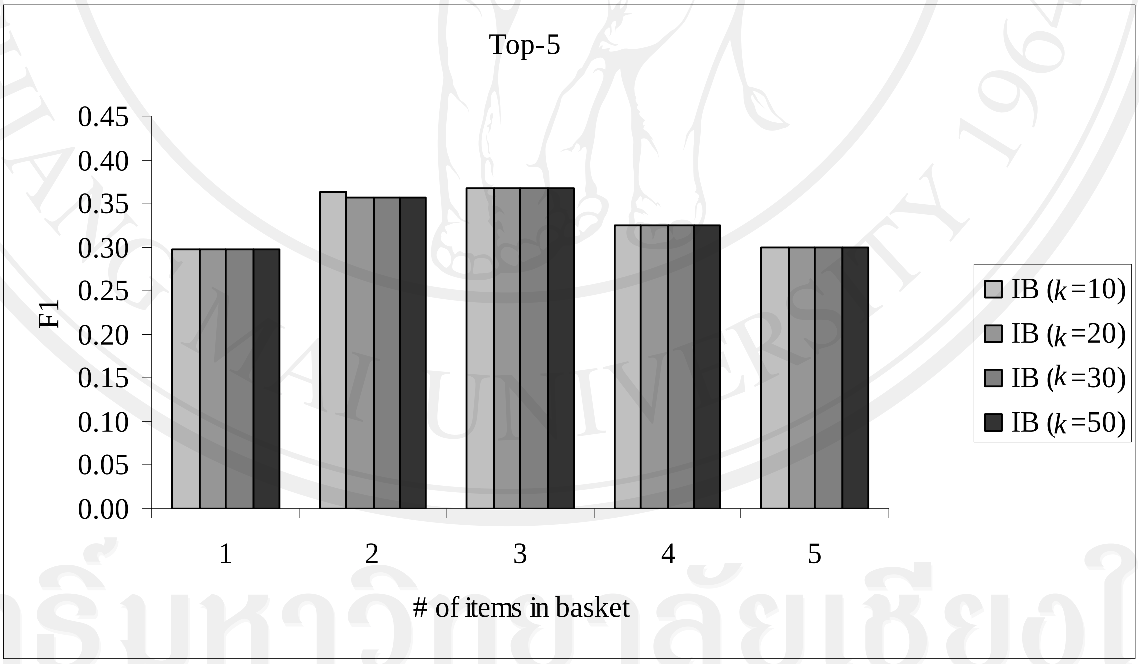

Figure 4.45 Average F1 sensitivity of neighborhood size using item-based recommendation system on ECR data set.

In this section, we compare the proposed methods with the common methods (i.e., VCR-KM, UB, IB, and FB) on the ECR data set. In the top-5 recommendation systems, the results of the average precision are shown in Figure 4.46. In the common methods, the results show that increasing the number of items in the basket increase the performance of the VCR-KM. The VCR-KM method has higher performance than the UB, IB, and FB. In the proposed methods, the results show that the systems based on the GA and MA with the fitness function in eq.(3.4) work better than the systems based on the GA and MA with the fitness function in eq.(3.1). The VCR-MAs work better than the VCR-GAs. The results also show that the hybrid methods increase the performance of the RS. In this data set, the results also show that the VCR-MA2 yields the best performance.

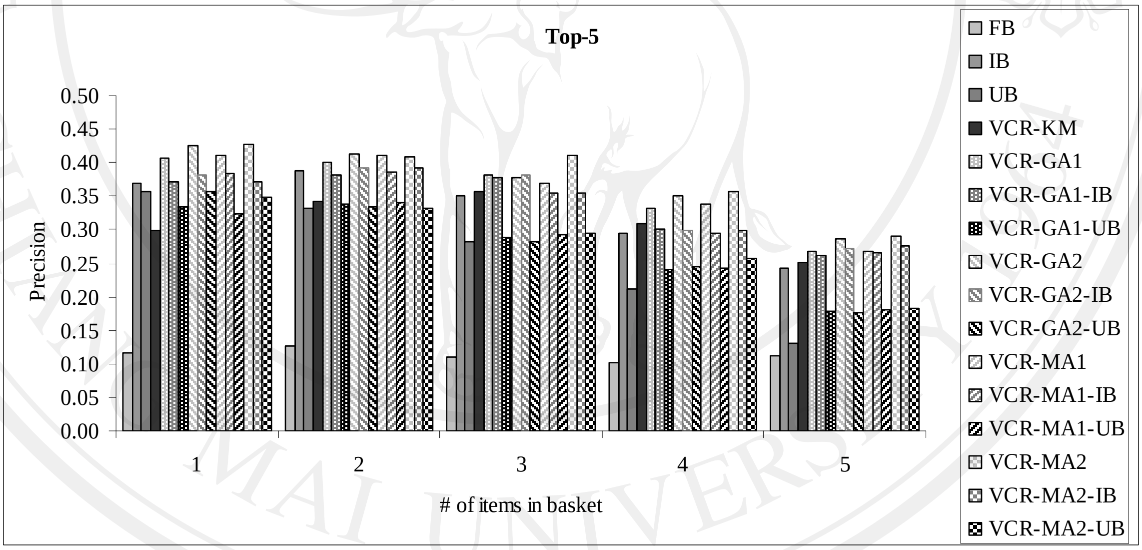

Figure 4.46 Average precision on ECR data set using the proposed methods and the common methods.

The results of the average recall are shown in Figure 4.47. In the common methods, increasing the number of items in the basket increases the performance of the VCR-KM. The VCR-KM method has better performance than the UB, IB, and FB. The VCR-KM method performs better than the IB when the number of items in basket is more than three. The systems based on the GA and MA with the fitness function in eq.(3.4) work better than that based on the GA and MA with the fitness function in eq.(3.1). When comparing the VCR-GAs and VCR-MAs, the VCR-MAs work better than VCR-GAs. The results also show that the VCR-MA2 yields the best performance.

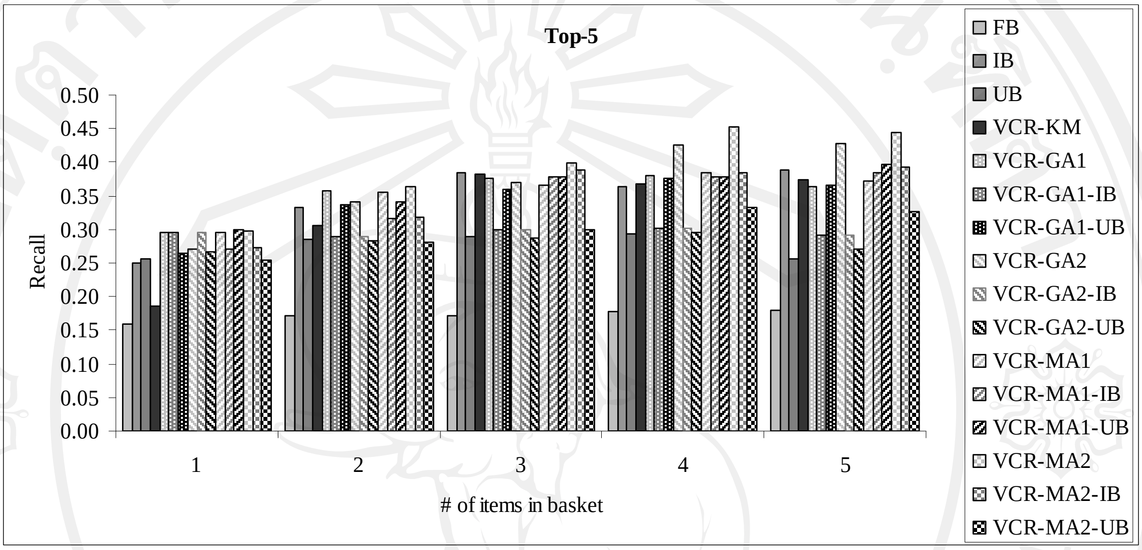

Figure 4.47 Average recall on ECR data set using the proposed methods and the common methods.

Figure 4.48 shows the results of the average F1. Among the common methods, the results show that the VCR-KM performs better than the UB, IB, and FB. The results also show that the hybrid methods help the performance of the common methods, i.e., UB, IB. The results also illustrate that the systems based on the GA and MA with the fitness function in eq.(3.4) perform better than the systems based on the GA and MA with the fitness function in eq.(3.1). Moreover, The VCR-MAs perform better than the VCR-GAs. The results also demonstrate that VCR-MA2 yields the best performance.

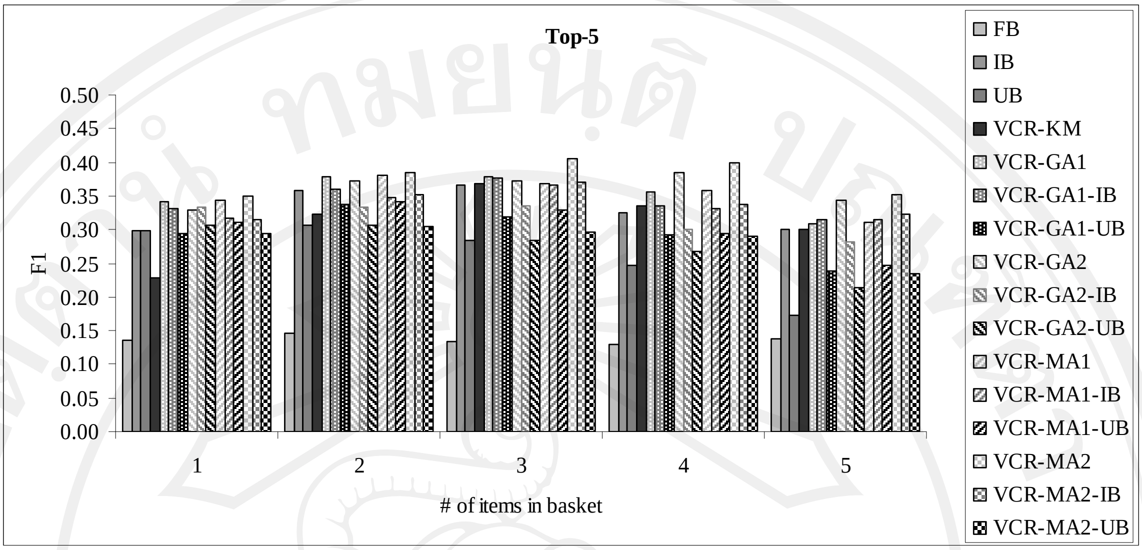

Figure 4.48 Average F1 on ECR data set using the proposed methods and the common methods.

4.5.3 Top-5 Recommendation Systems on RCM Data Set

In this section, we tested the neighborhood size on the user-based and item-based recommendation systems on the RCM data set. To determine the effect of neighborhood size, we performed the experiment by using the user and item-neighborhoods of sizes 10, 20, 30, and 50 (i.e., k = 10, 20, 30, 50). We also tested the effect of neighborhood size on the RCM data set for top-5 recommendation systems. In the test set, the items 1 to 5 were randomly selected into the basket. From the results of the neighborhood sensitivity as shown in Figures 4.49-4.54, we found that the user-neighborhood size = 30 and the item-neighborhood size = 10 are suitable.

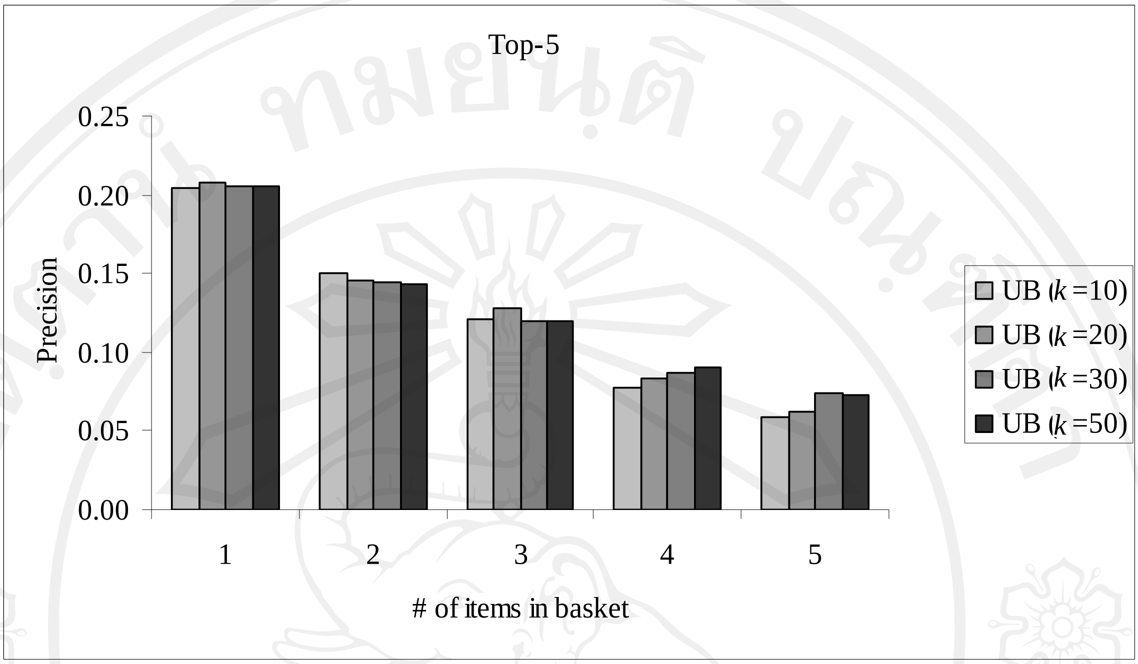

Figure 4.49 Average precision on RCM data set using the user-based methods.

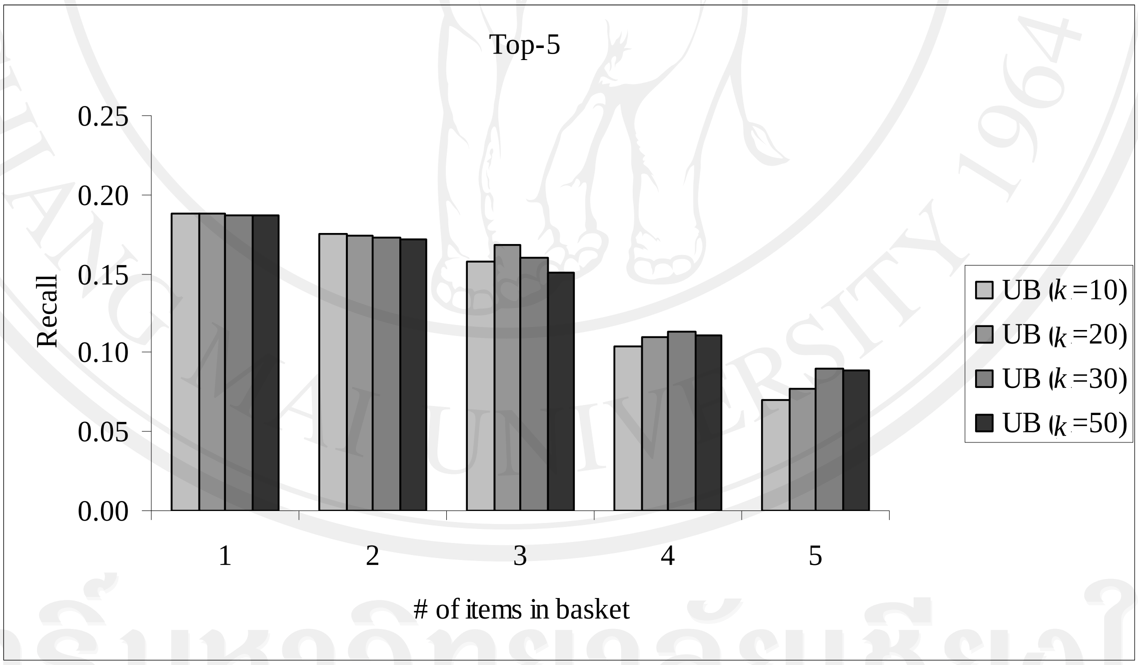

Figure 4.50 Average recall on RCM data set using the user-based methods.

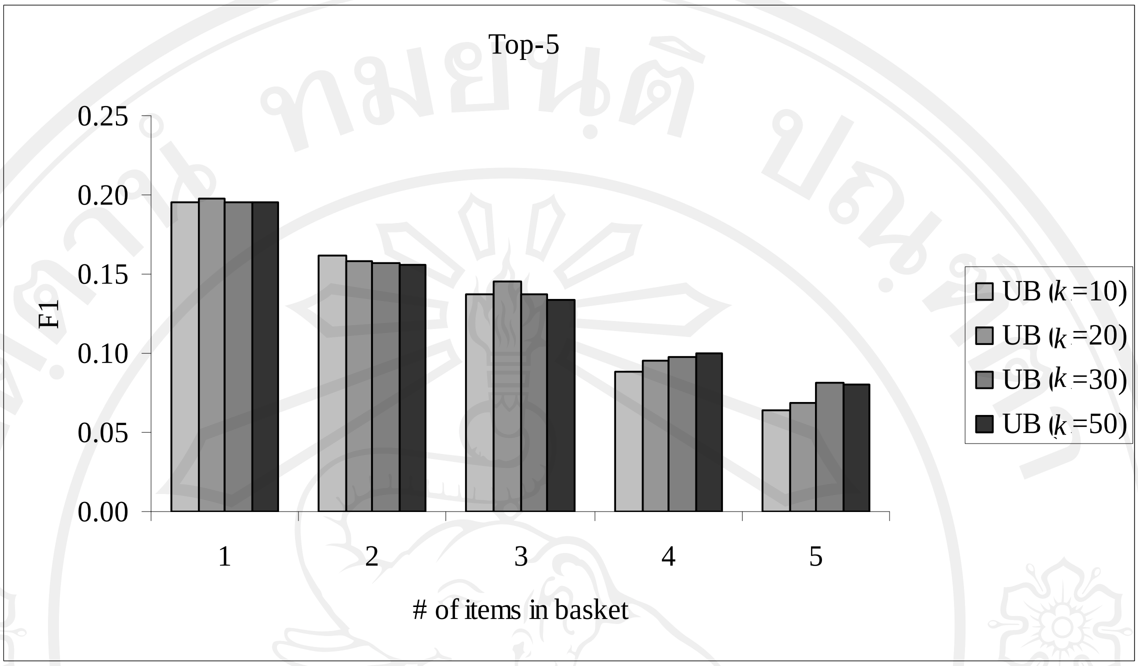

Figure 4.51 Average F1 on RCM data set using the user-based methods.

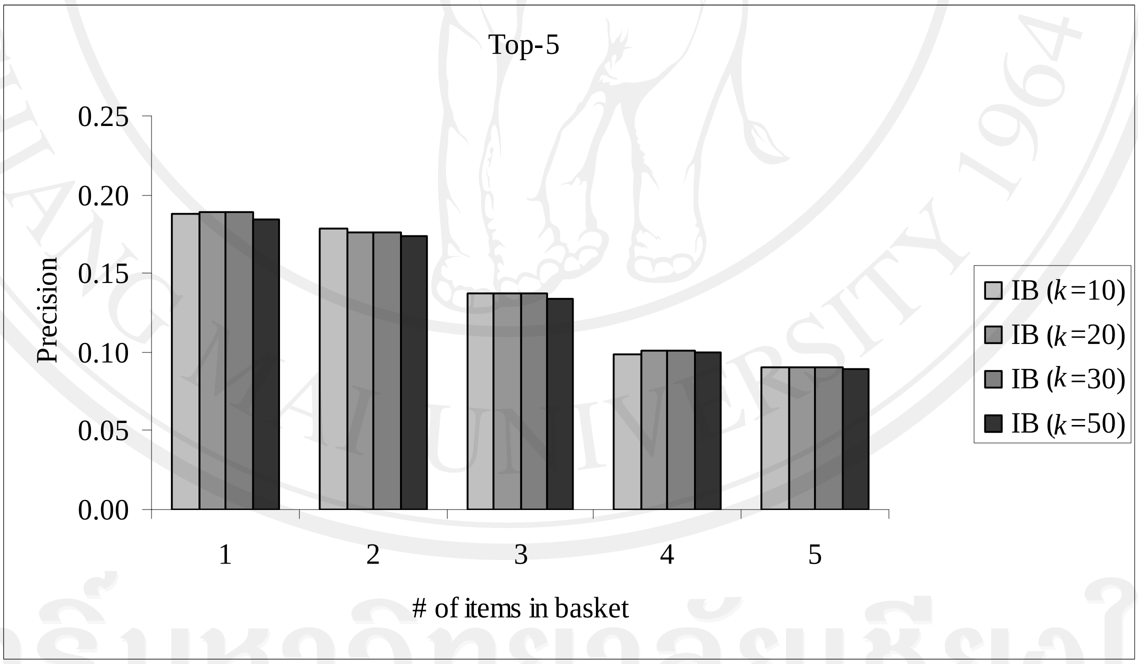

Figure 4.52 Average precision on RCM data set using the item-based methods.

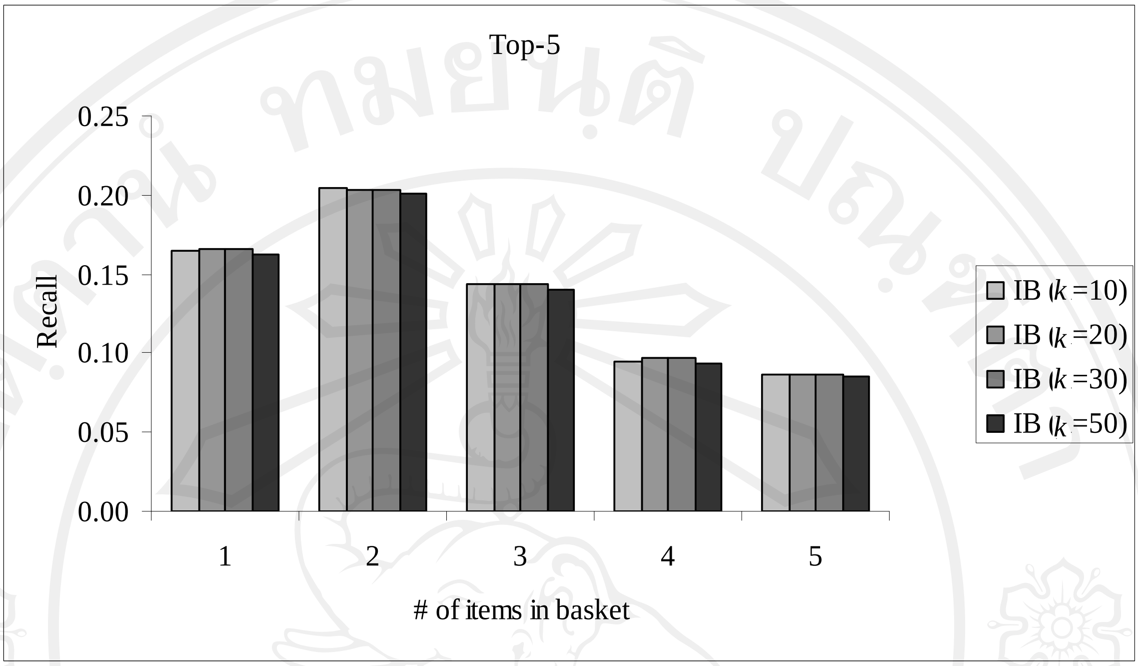

Figure 4.53 Average recall on RCM data set using the item-based methods.

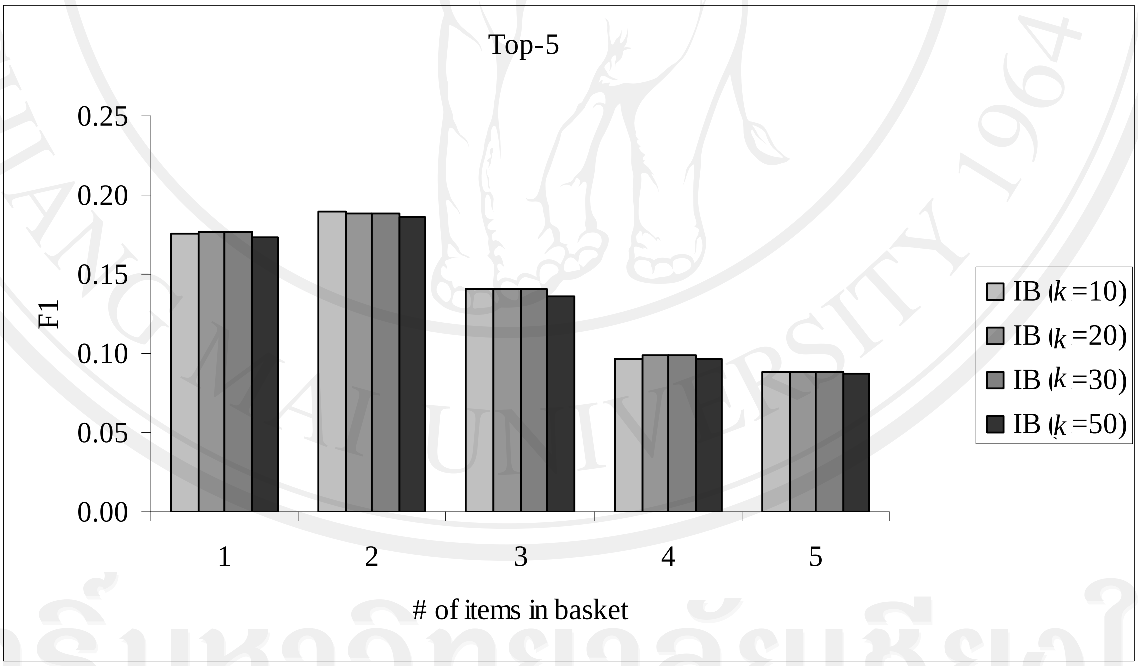

Figure 4.54 Average F1 on RCM data set using the item-based methods.

We compare the performance of proposed methods with the common methods (i.e., VCR-KM, UB, IB, and FB) on the RCM data set. In the top-5 recommendation systems, the average precisions are shown in Figure 4.55. In the common methods, the VCR-KM performs better than the UB, IB, and FB. In the hybrid methods, the hybrid methods increase the performance of the UB and IB. In the recommendation systems based on the GA with the fitness function in eq.(3.1), the VCR-GA1-IB yields better performance than the VCR-GA1 and VCR-GA2. In the recommendation systems based on MA with the fitness function in eq.(3.1), the VCR-MA1-IB yields better performance than the VCR-MA1 and VCR-MA1-UB. Moreover, the systems based on MA perform better than the systems based on GA. The systems based on GA and MA with the fitness function in eq.(3.4) perform better than the systems based on the GA and MA with fitness function in eq.(3.1). Moreover, the systems based on MA perform better than the systems based on GA with the fitness function in eq.(3.4). The results also show that the hybrid methods increase the performance of the recommendation systems of the UB and IB. In the all sixteen methods, the results show that the VCR-MA2 performs the best.

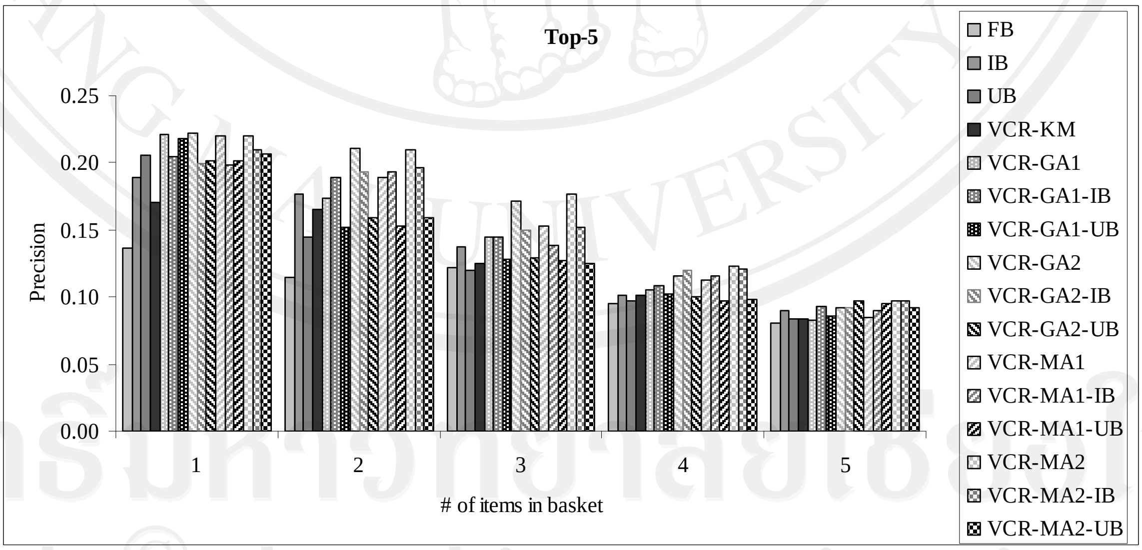

Figure 4.55 Average precision on RCM using the proposed methods and common methods.

The results of the average recall are shown in Figure 4.56. In the common methods, the VCR-KM performs better than the UB and FB. In the recommendation systems based on the GA with the fitness function in eq.(3.1), the VCR-GA1 performs better than the VCR-GA1-UB and VCR-GA1-IB. In the recommendation systems based on the GA with the fitness function in eq.(3.4), the VCR-GA2 performs better than the VCR-GA2-UB and VCR-GA2-IB. In recommendation systems based on the GA with the fitness functions in eq.(3.1) and eq.(3.4), the systems based on the GA with the fitness function in eq.(3.4) perform better. In the recommendation systems based on the GA and MA, the systems based on the MA perform better. The results also show that the hybrid methods help increasing recall performance of the UB and IB. The results show that the VCR-MA2 yields the best performance.

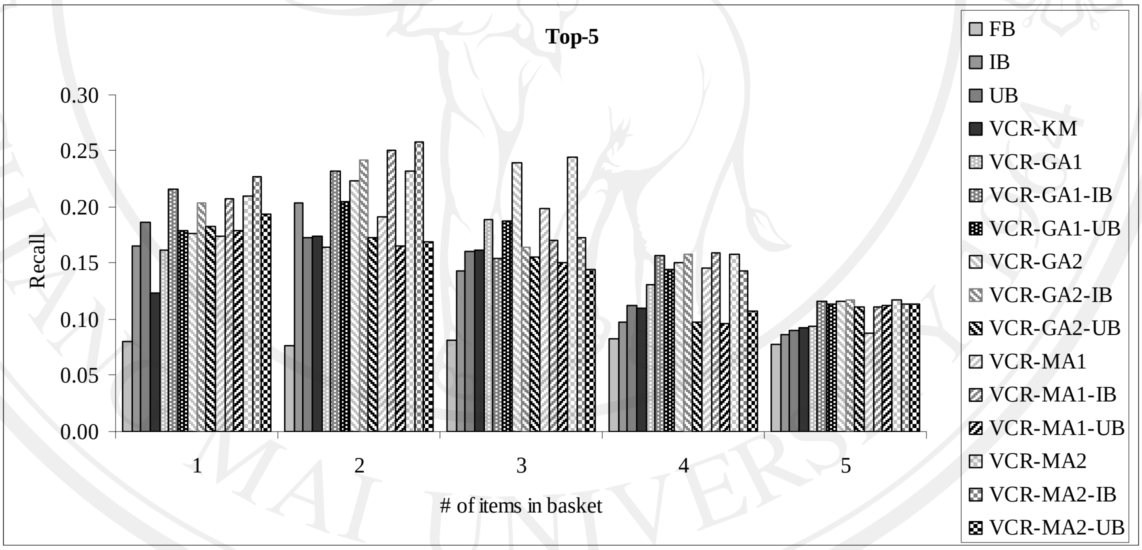

Figure 4.56 Average recall on RCM data set using the proposed methods and common methods.

The results of the average F1 are shown in Figure 4.57. In the common methods, the results show that the VCR-KM yields better performance than the UB, IB, and FB. In the recommendation systems based on the GA with the fitness function in eq.(3.1), the VCR-GA1-IB yields better performance than the VCR-GA1 and VCR-GA1-UB. In the recommendation systems based on the MA with the fitness function in eq.(3.1), the VCR-MA1-IB yields better performance than the VCR-MA1 and

VCR-MA1-UB. The systems based on the MA perform better than the systems based on the GA. In the recommendation systems based on the GA with the fitness function in eq.(3.4), the VCR-GA2 yields better performance than the VCR-GA2-IB and VCR-GA2-UB. In the recommendation systems based on the MA with the fitness function in eq.(3.4), the VCR-MA2 yields better performance than the VCR-MA2-IB and VCR-MA2-IB. The systems based on the GA and MA with the fitness function in eq.(3.4) yield better performance than the systems based on the GA and MA with the fitness function in eq.(3.1). The results also show that the hybrid methods increase the performance of the UB and IB. The results also show that the VCR-MA2 yields the best.

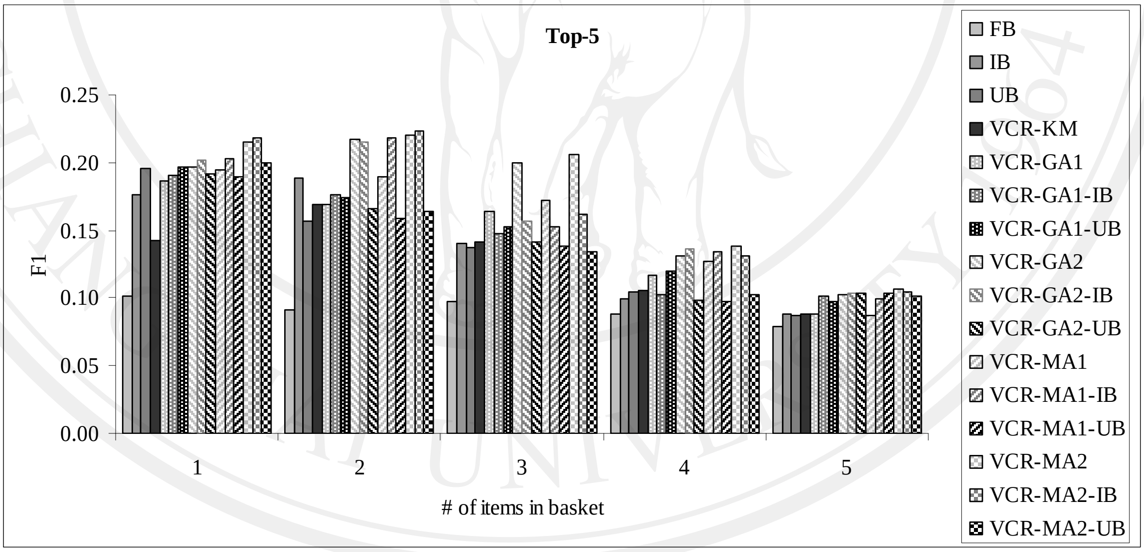

Figure 4.57 Average F1 on RCM data set using the proposed methods and common methods.

4.5.4 Top-5 Recommendation Systems on MovieLens Data Set

For the MovieLens data set, we tested the neighborhood size on the user-based and item-based recommendation systems [12]. We performed the experiment by using the user and item-neighborhoods of sizes 10, 20, 30, and 50 (i.e., k=10, 20, 30, 50) to determine the effect of neighborhood size. We also tested the effect of neighborhood size on the MovieLens data set for top-5 recommendation system. In the test set, the items 1 to 15 were randomly selected into the basket. From the results of the neighborhood sensitivity as shown in Figures 4.58-4.63, we found that the user-neighborhood size = 50 and the item-neighborhood size = 50 are suitable.

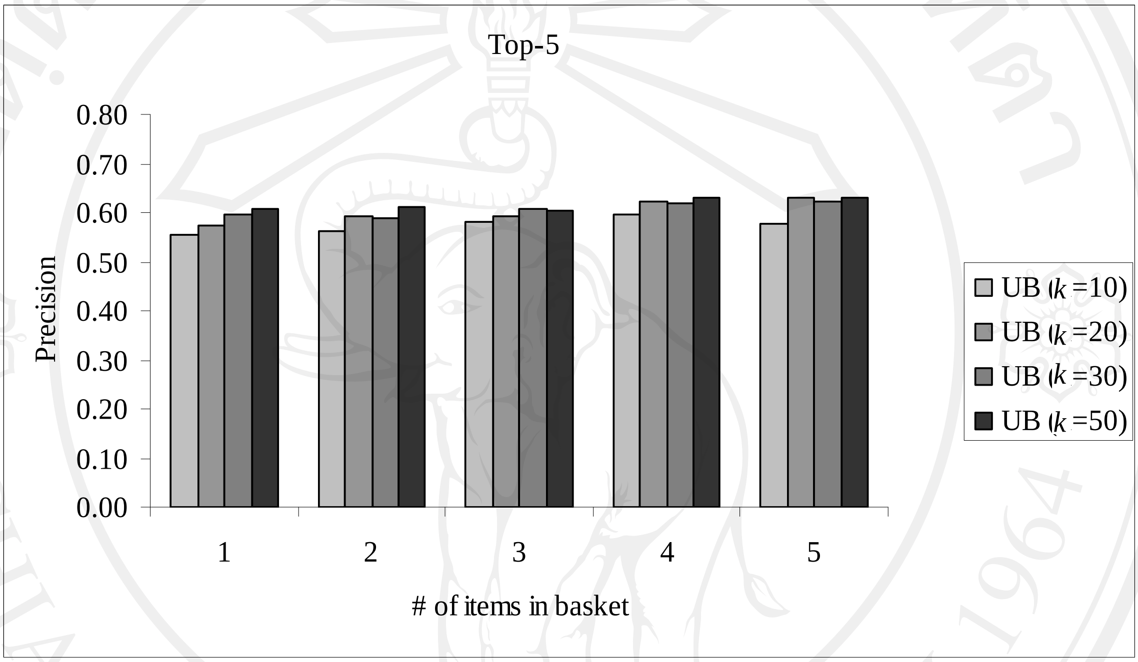

Figure 4.58 Average precision sensitivity of neighborhood size using user-based recommendation system on MovieLens data set.

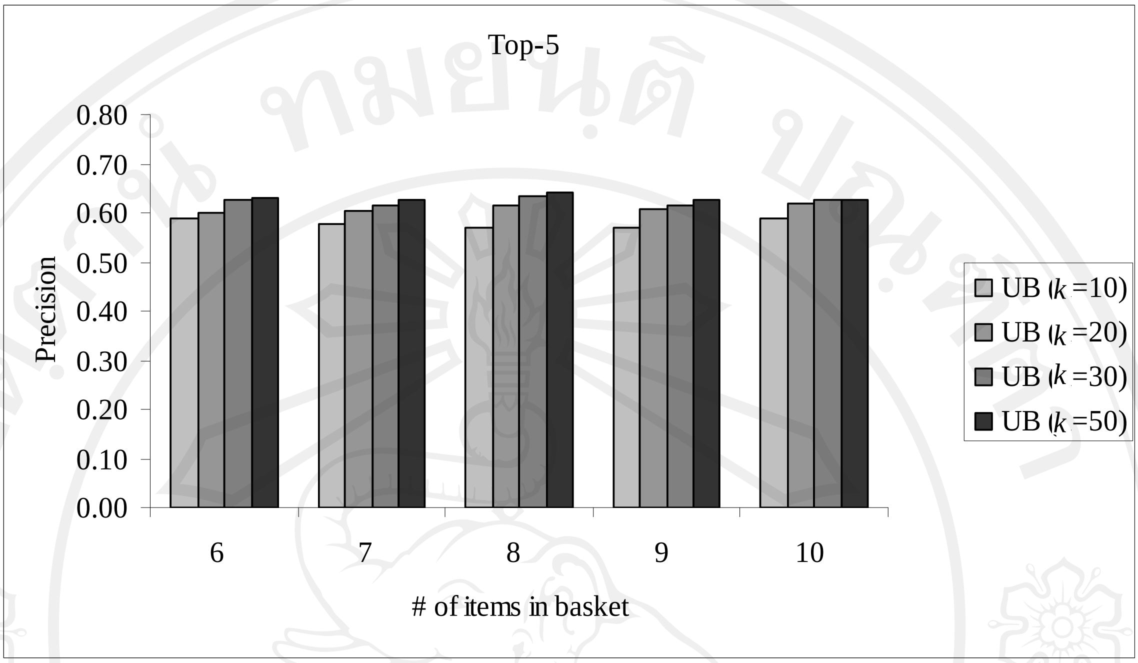

Figure 4.58 Average precision sensitivity of neighborhood size using user-based recommendation system on MovieLens data set (Continued).

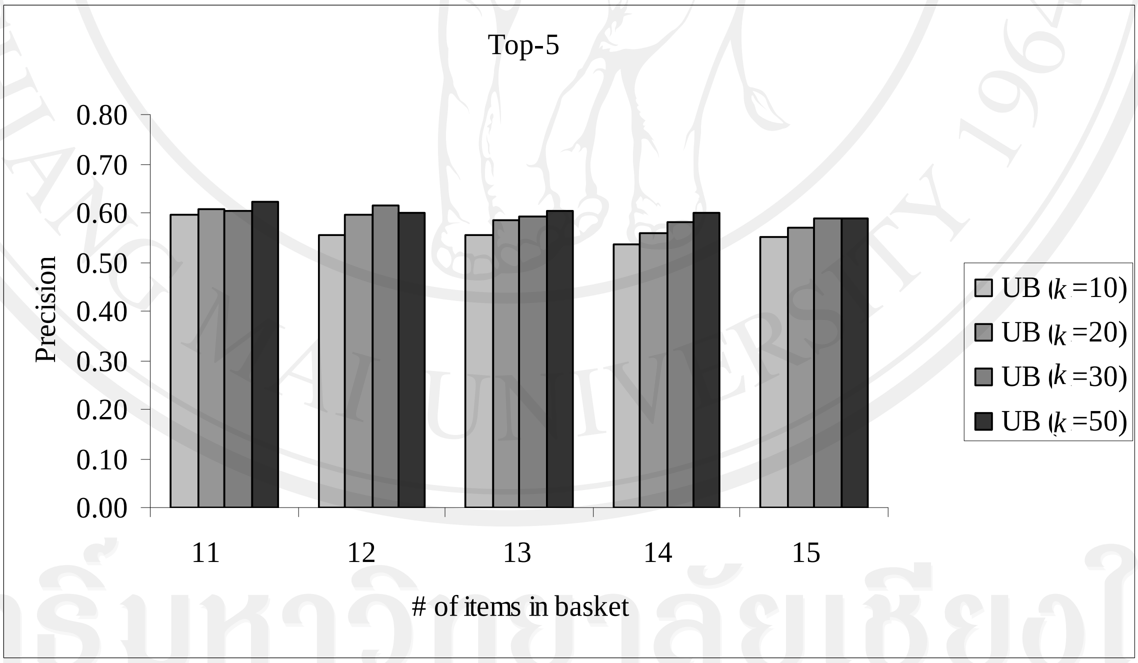

Figure 4.58 Average precision sensitivity of neighborhood size using user-based recommendation system on MovieLens data set (Continued).

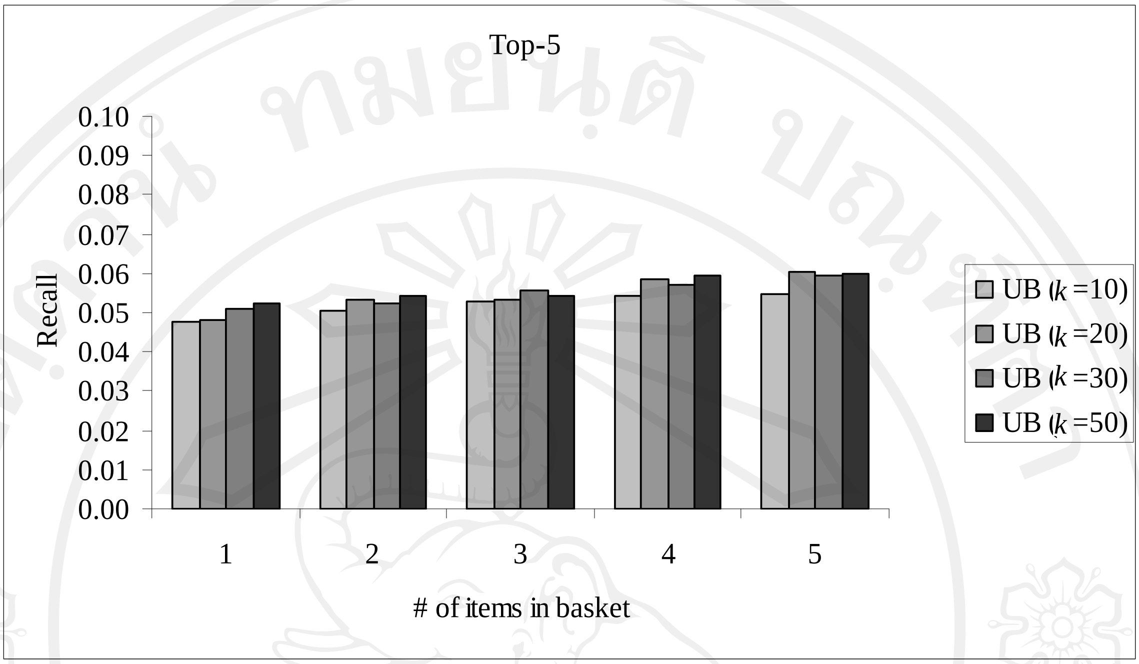

Figure 4.59 Average recall sensitivity of neighborhood size using user-based recommendation system on MovieLens data set.

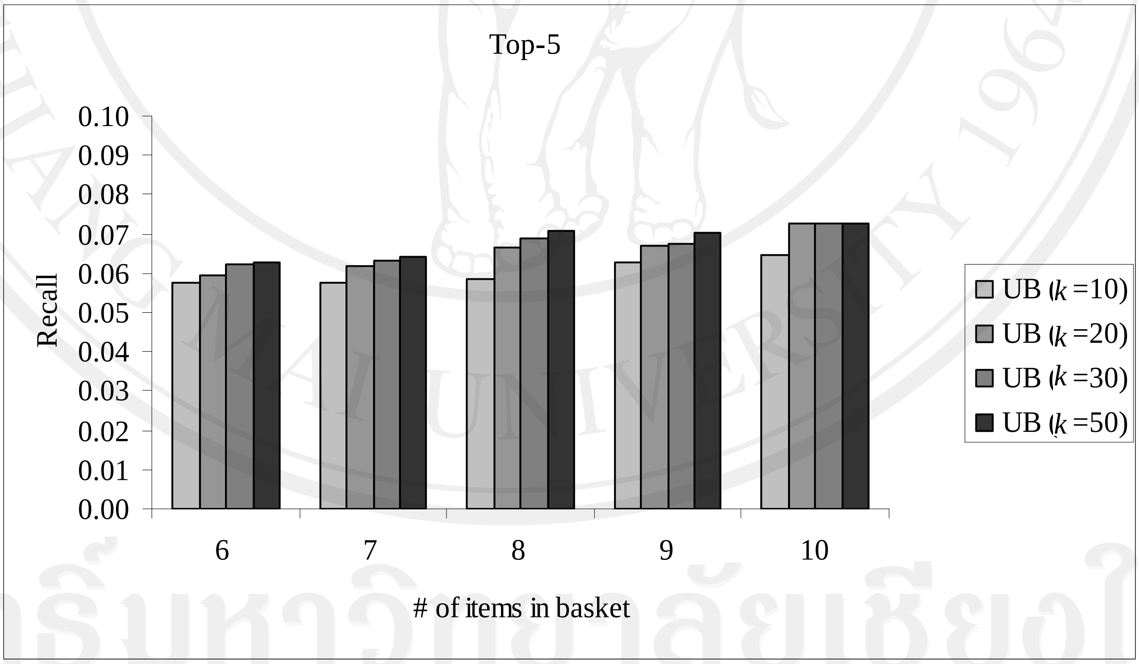

Figure 4.59 Average recall sensitivity of neighborhood size using user-based recommendation system on MovieLens data set (Continued).

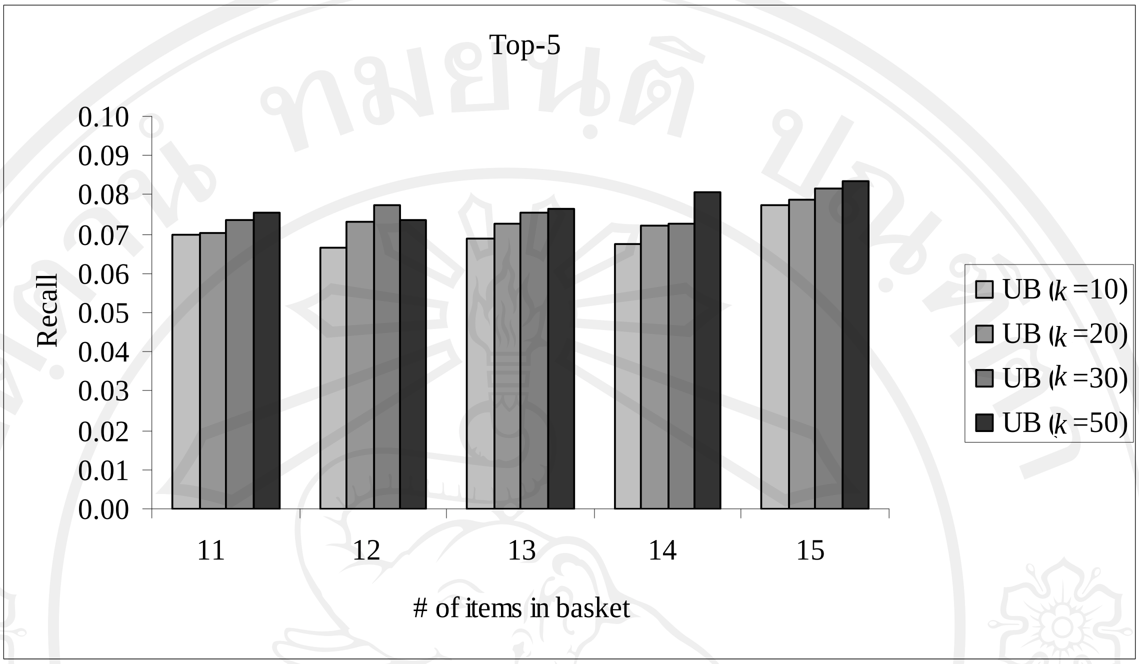

Figure 4.59 Average recall sensitivity of neighborhood size using user-based recommendation system on MovieLens data set (Continued).

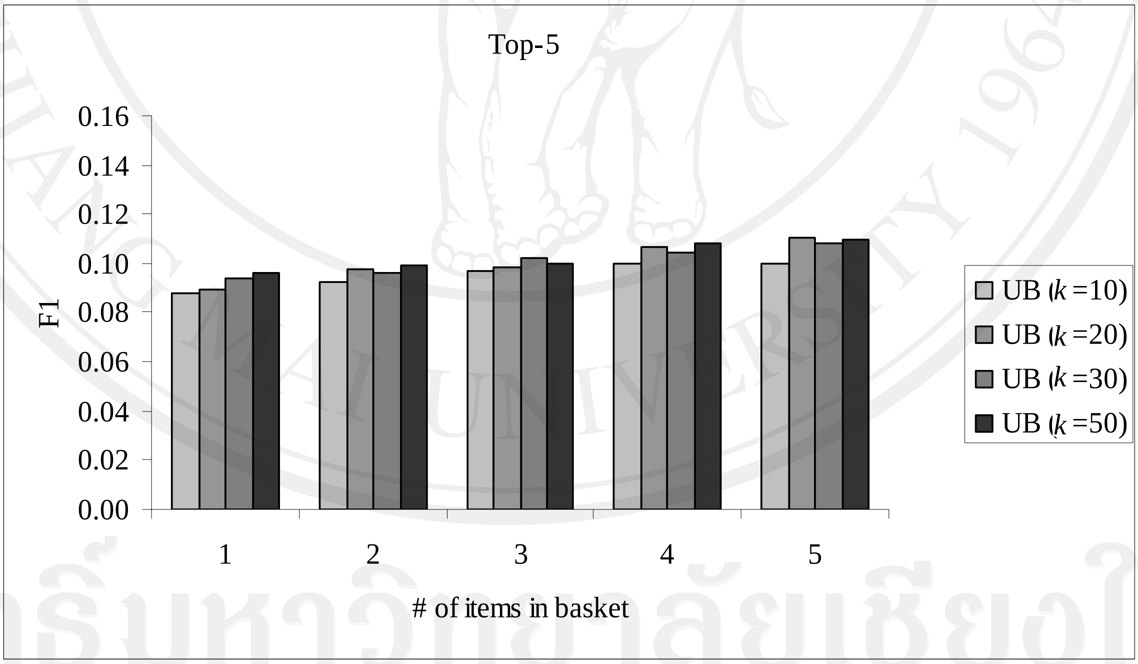

Figure 4.60 Average F1 sensitivity of neighborhood size using user-based recommendation system on MovieLens data set.

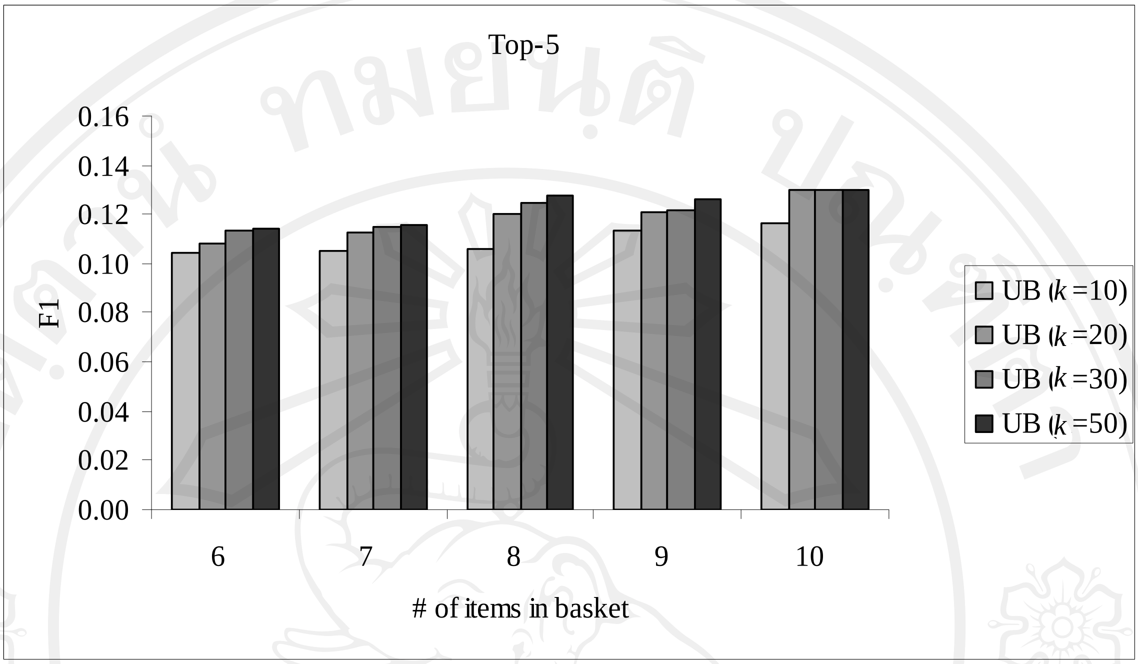

Figure 4.60 Average F1 sensitivity of neighborhood size using user-based recommendation system on MovieLens data set (Continued).

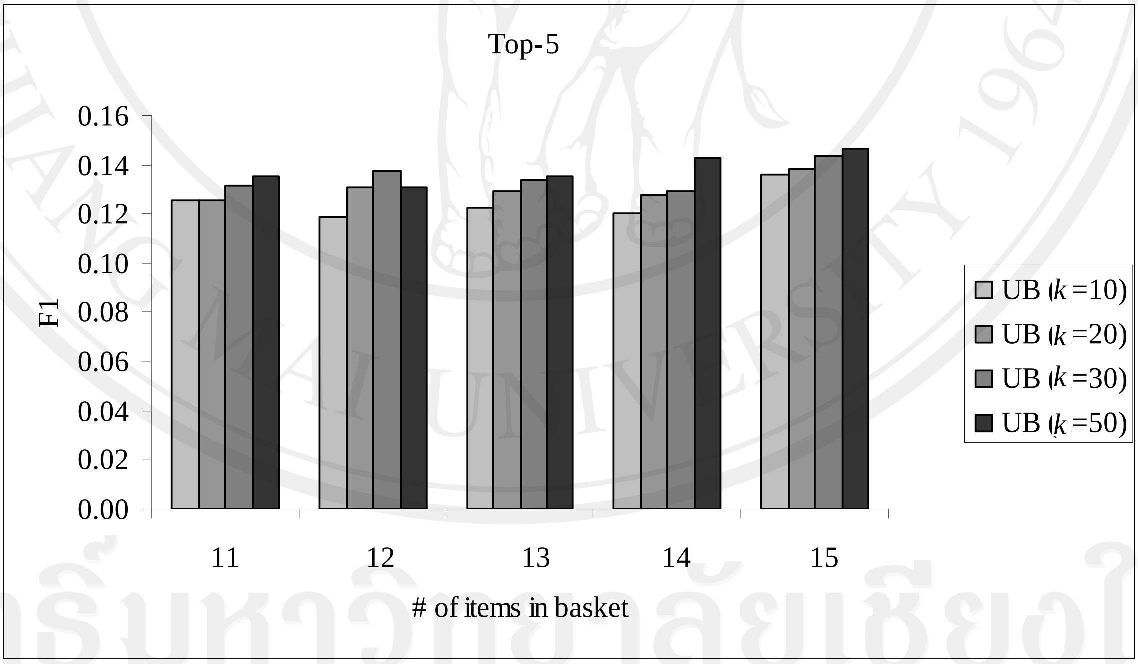

Figure 4.60 Average F1 sensitivity of neighborhood size using user-based recommendation system on MovieLens data set (Continued).

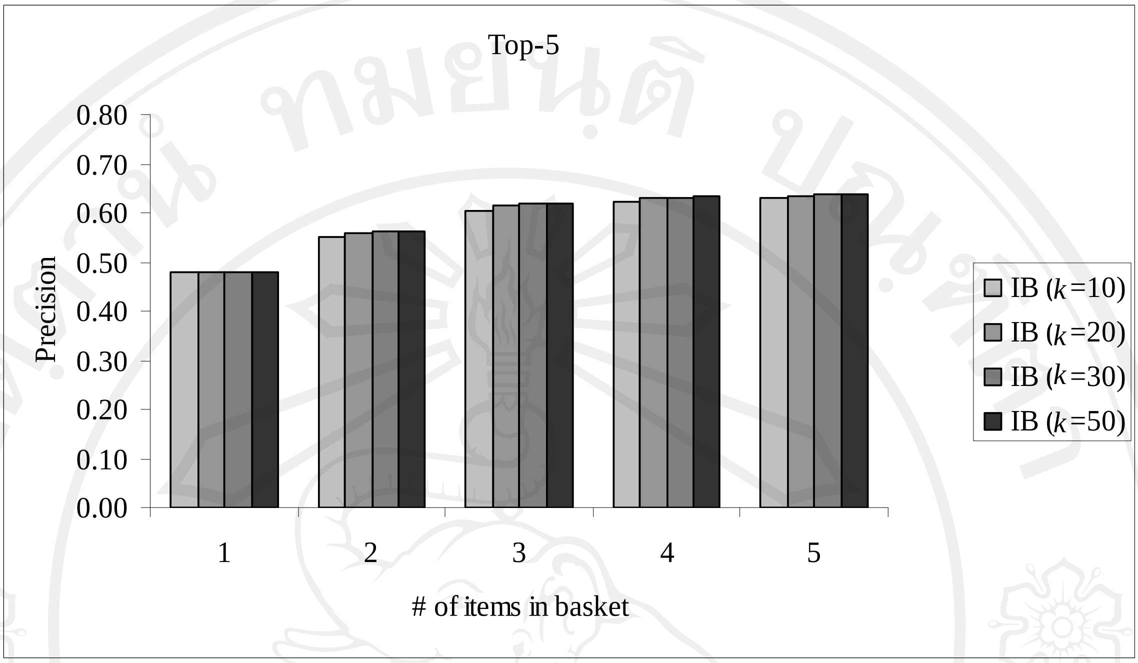

Figure 4.61 Average precision sensitivity of neighborhood size using item-based recommendation system on MovieLens data set.

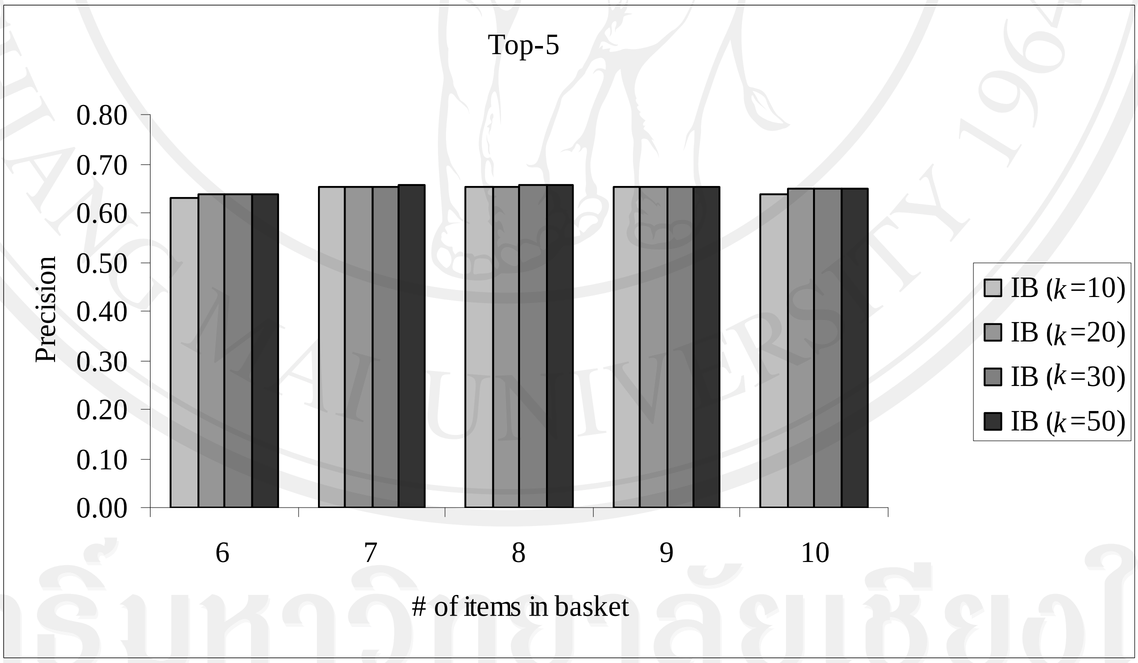

Figure 4.61 Average precision sensitivity of neighborhood size using item-based recommendation system on MovieLens data set (Continued).

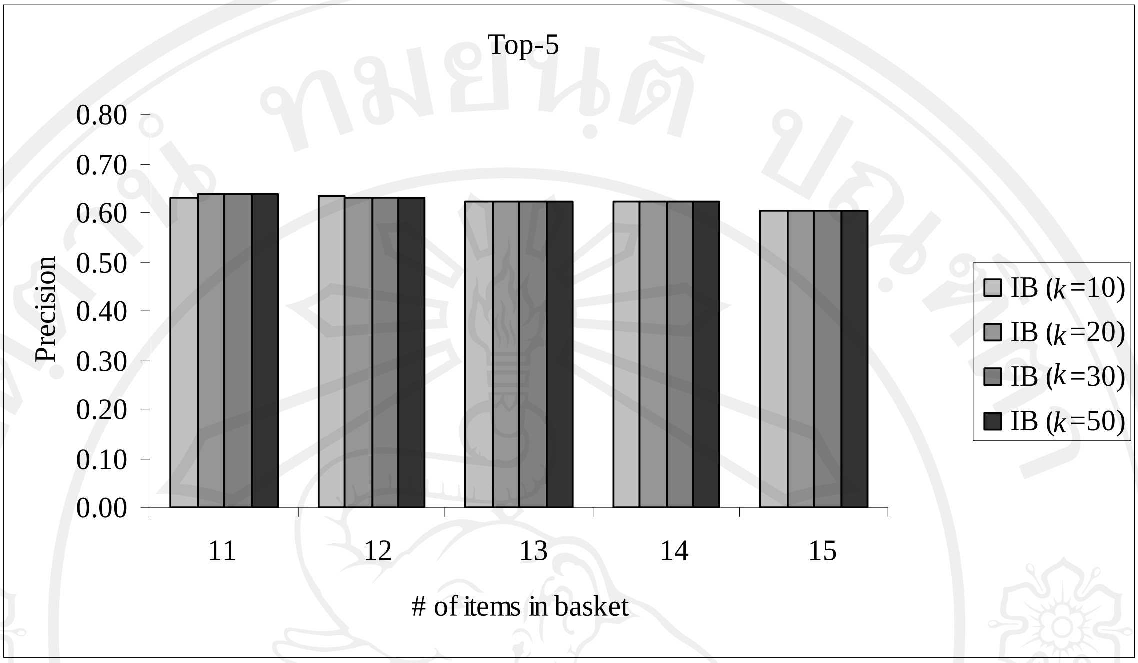

Figure 4.61 Average precision sensitivity of neighborhood size using item-based recommendation system on MovieLens data set (Continued).

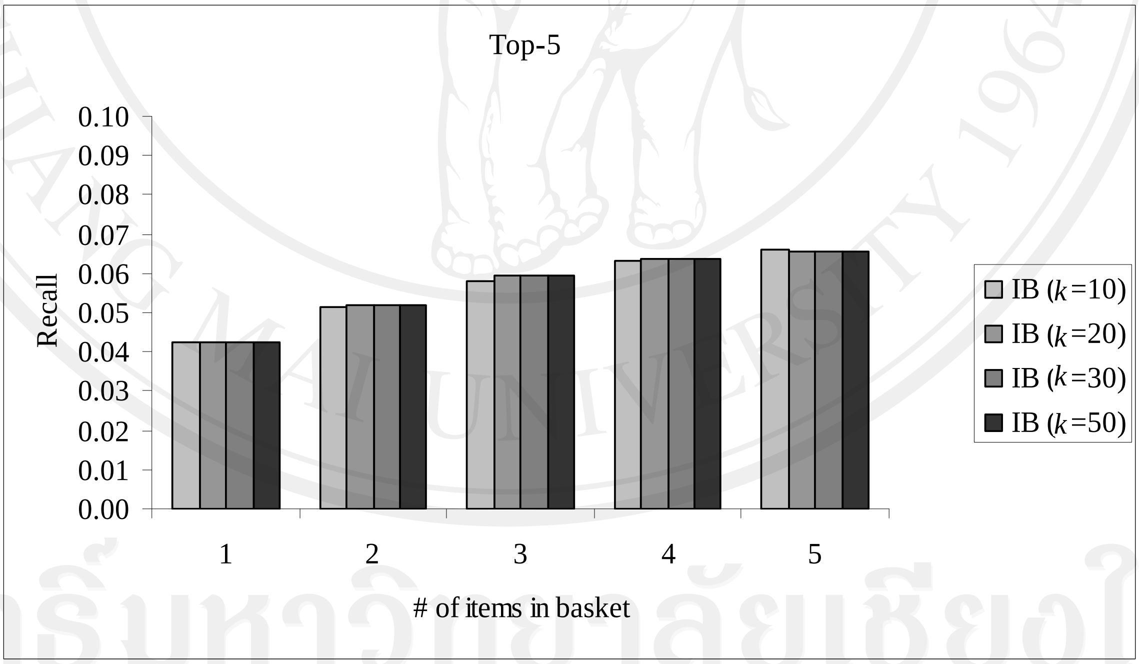

Figure 4.62 Average recall sensitivity of neighborhood size using item-based recommendation system on MovieLens data set.

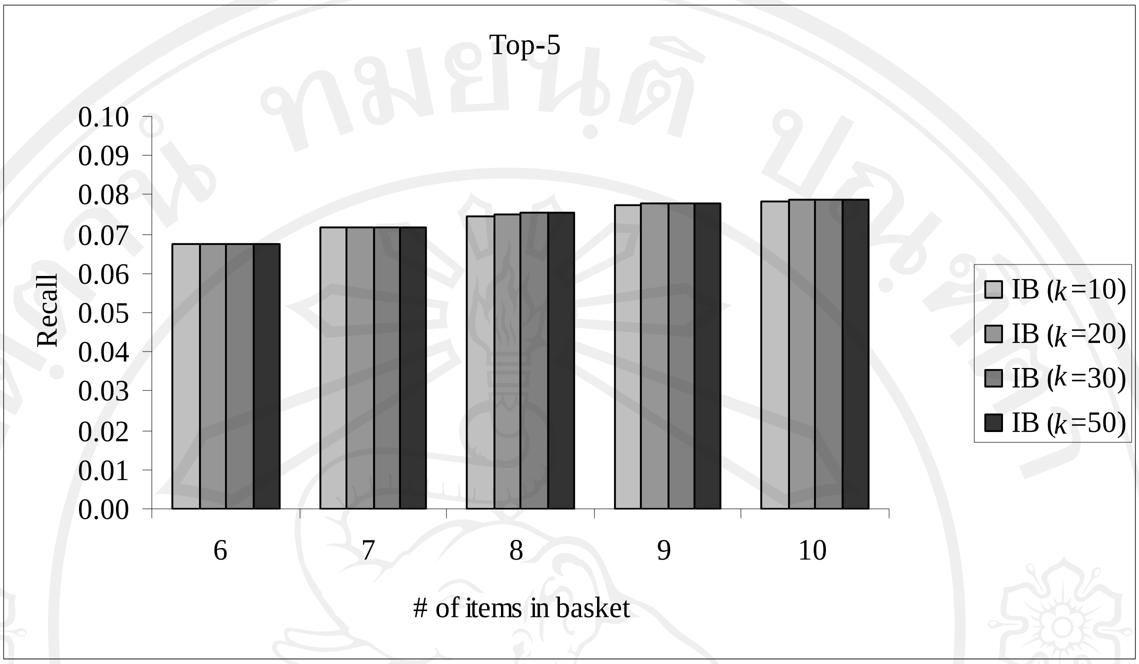

Figure 4.62 Average recall sensitivity of neighborhood size using item-based recommendation system on MovieLens data set (Continued).

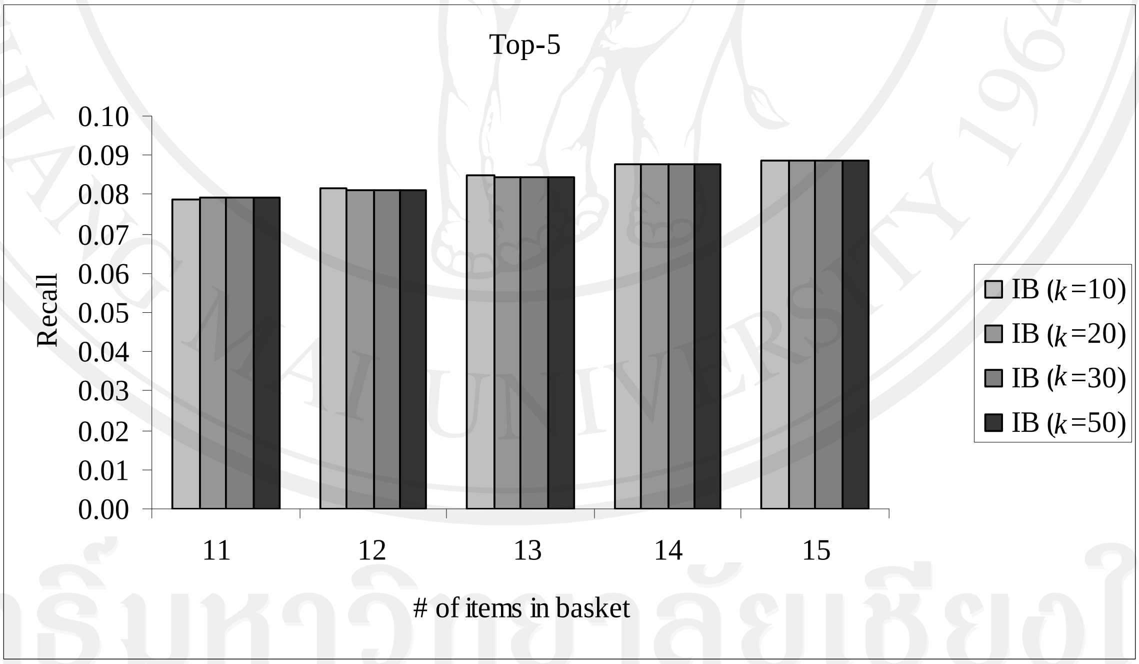

Figure 4.62 Average recall sensitivity of neighborhood size using item-based recommendation system on MovieLens data set (Continued).

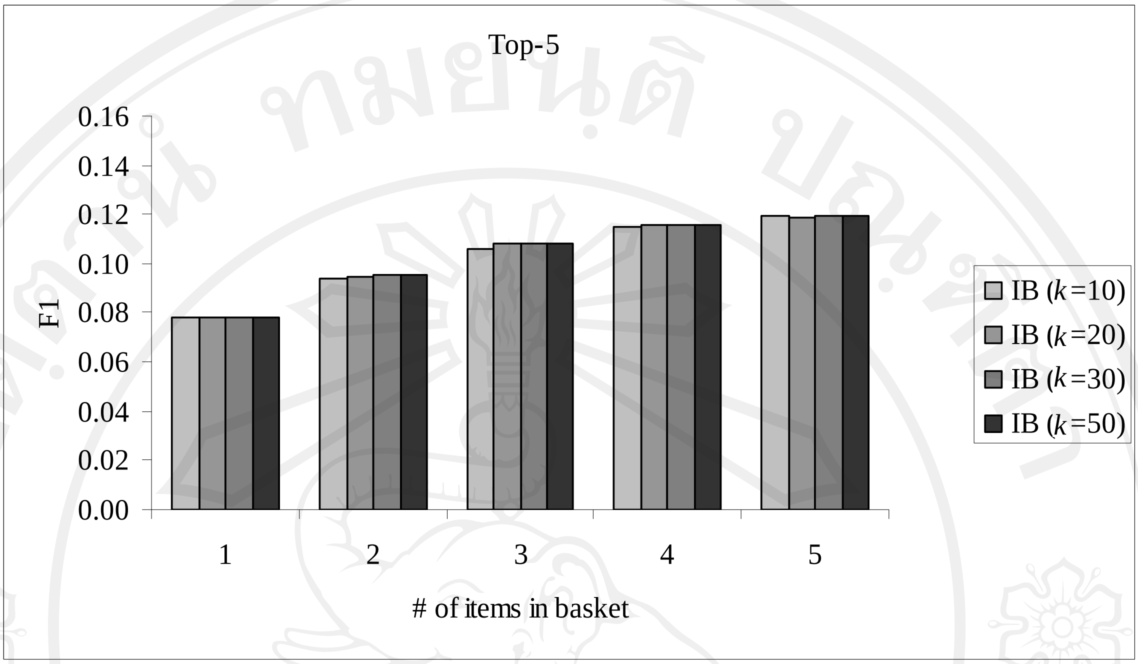

Figure 4.63 Average F1 sensitivity of neighborhood size using item-based recommendation system on MovieLens data set.

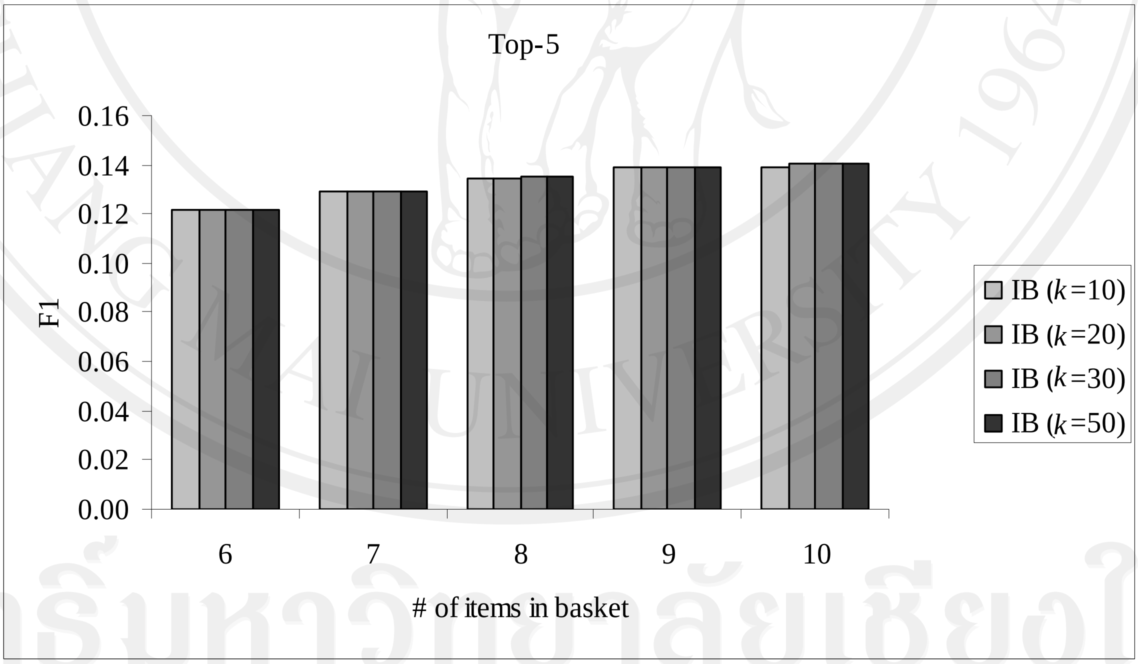

Figure 4.63 Average F1 sensitivity of neighborhood size using item-based recommendation system on MovieLens data set (Continued).

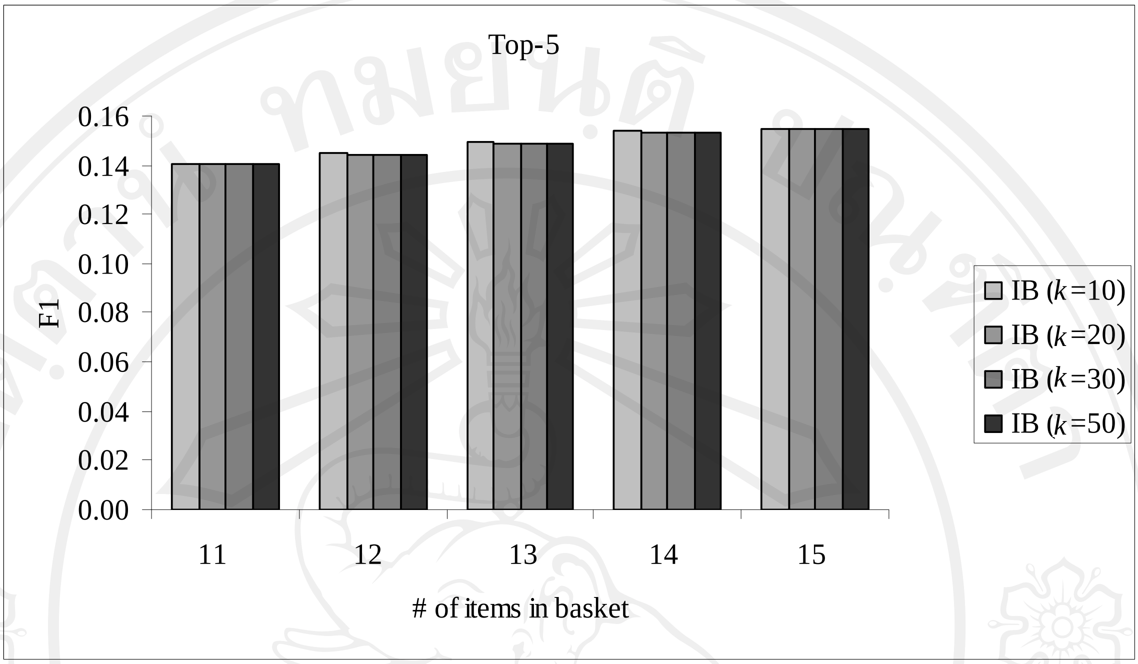

Figure 4.63 Average F1 sensitivity of neighborhood size using item-based recommendation system on MovieLens data set (Continued).

In this section, we compared the proposed methods with the common methods (i.e., VCR-KM, UB, IB, and FB). In the top-5 recommendation systems, the average precisions are shown in Figure 4.64. For comparing the common methods, the IB yields better performance. In the recommendation systems based on the GA with the fitness function in eq.(3.1), the VCR-GA1-IB performs better than the VCR-GA1 and VCR-GA1-UB. In the recommendation systems based on the MA with the fitness function in eq.(3.1), the VCR-MA1-IB performs better than the VCR-MA1 and VCR-MA1-UB. For comparing the systems based on the GA and MA with the fitness function in eq.(3.1), the systems based on the MA perform better. In the recommendation systems based on the GA with the fitness function in eq.(3.4), the VCR-GA2-IB yields slightly better performance than the VCR-GA2 and VCR-GA2-UB. In the recommendation systems based on the MA with the fitness function in eq.(3.4), the VCR-MA2-IB yields slightly better performance than the VCR-MA2 and

VCR-MA2-UB. The systems based on the GA and MA with the fitness function in eq.(3.4) yield better performance than the systems based on the GA and MA with the fitness function in eq.(3.1). To compare the systems based on the GA and MA with the fitness function in eq.(3.4), the systems based on the MA yield better performance than the systems based on the GA. the results also show that the hybrid methods increase the performance of the recommendation systems of the UB and IB.

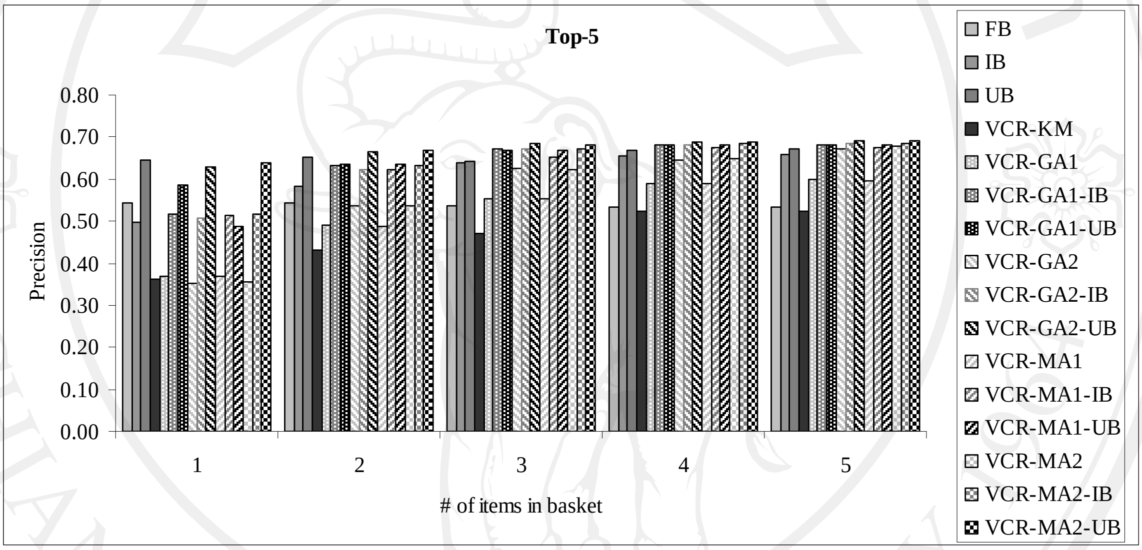

Figure 4.64 Average precision on MovieLens data set using the proposed methods and the common methods.

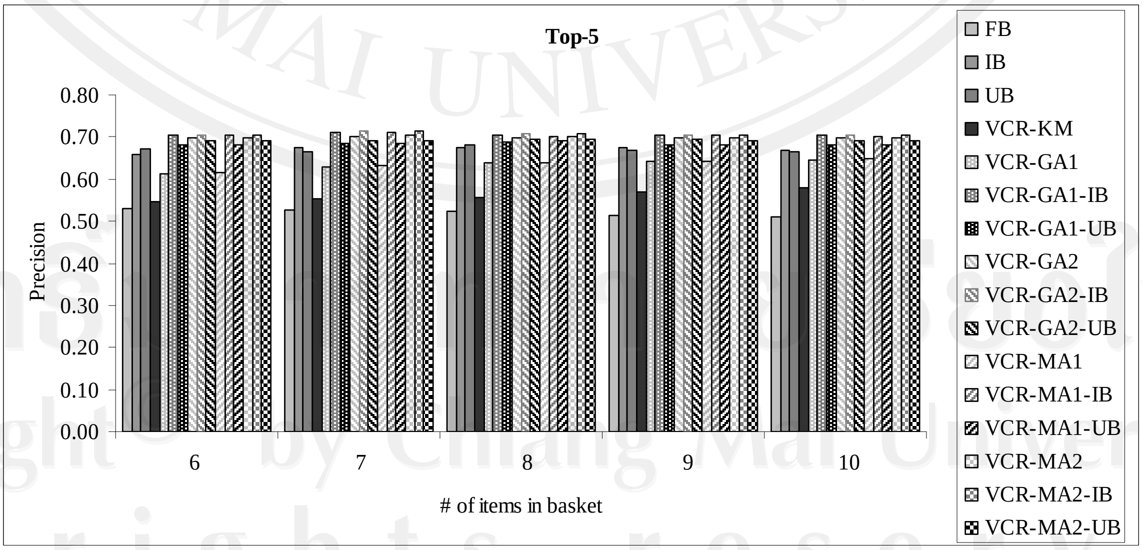

Figure 4.64 Average precision on MovieLens data set using the proposed methods and the common methods (Continued).

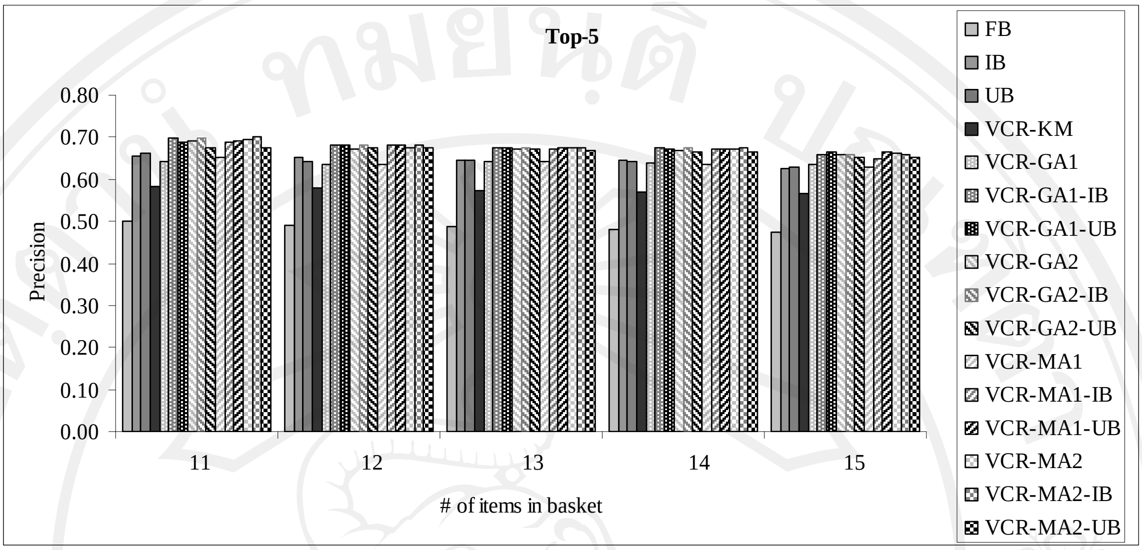

Figure 4.64 Average precision on MovieLens data set using the proposed methods and the common methods (Continued).

The results of the average recall are shown in Figure 4.65. In the common methods, the IB performs better. In the recommendation systems based on the GA with the fitness function in eq.(3.1), the VCR-GA1-IB yields slightly better performance than the VCR-GA1 and VCR-GA2. In the recommendation systems based on the MA with the fitness function in eq.(3.1), the VCR-MA1-IB yields better performance than the VCR-MA1 and VCR-MA1-UB. To compare the systems based on the GA and MA with the fitness function in eq.(3.1), the systems based on the GA performs better. In the recommendation systems based on the GA with the fitness function in eq.(3.4), the VCR-GA2-IB yields slightly better performance than the VCR-GA2 and VCR-GA2-UB. In the recommendation systems based on the MA with the fitness function in eq.(3.4), the VCR-MA2-IB yields slightly better performance than the VCR-MA2 and VCR-MA2-UB. The results also show that the hybrid methods help the increasing recall performance.

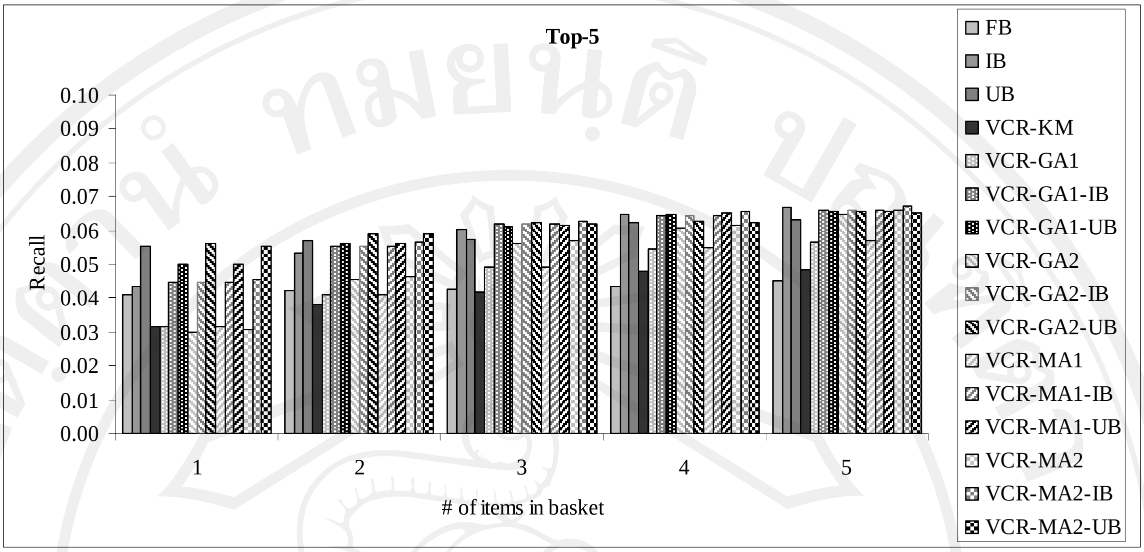

Figure 4.65 Average recall on MovieLens data set using the proposed methods and the common methods.

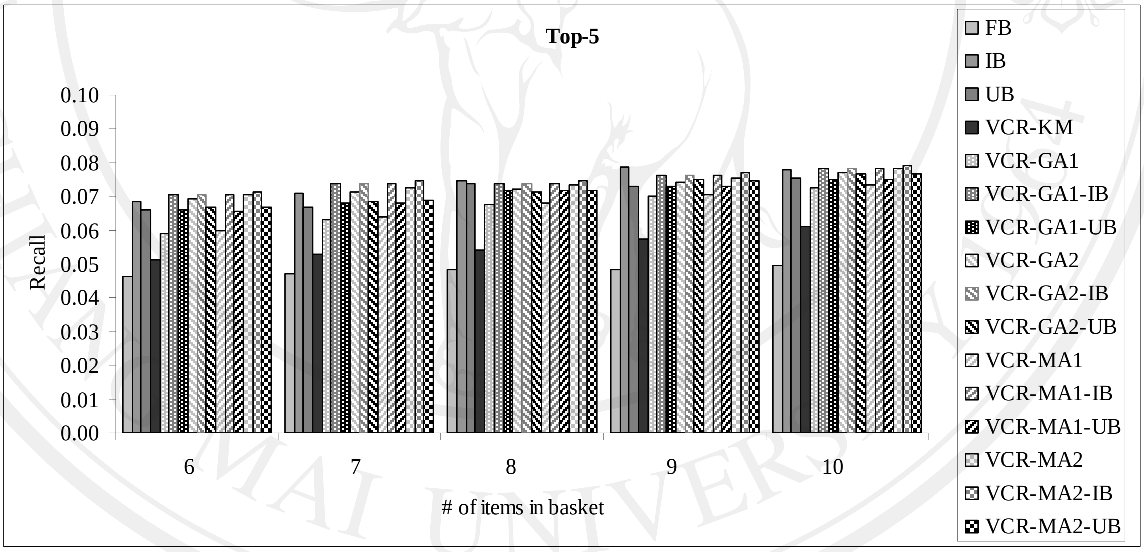

Figure 4.65 Average recall on MovieLens data set using the proposed methods and the common methods (Continued).

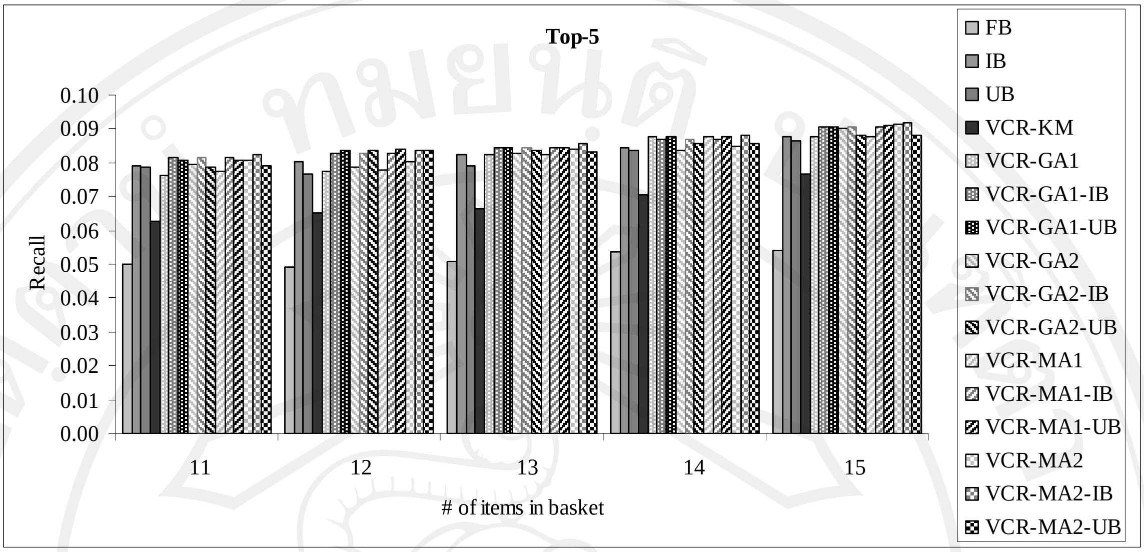

Figure 4.65 Average recall on MovieLens data set using the proposed methods and the common methods (Continued).

The results of the average F1 are shown in Figure 4.66. The results show that increasing the number of items in the basket increases the performance of the RSs. In the common methods, the results show that IB has the best performance. The F1 results also show that increasing the number of items in the basket increases the performance of the VCR-GAs and VCR-MAs. The systems based on the GA and MA with the fitness function in eq.(3.4) work better than the systems based on the GA and MA with the fitness function in eq.(3.1). The systems based on the MA perform better than the systems based on the GA. The results also show that the systems based on the GA and MA with the fitness function in eq.(3.4) yield better performance than the common methods.

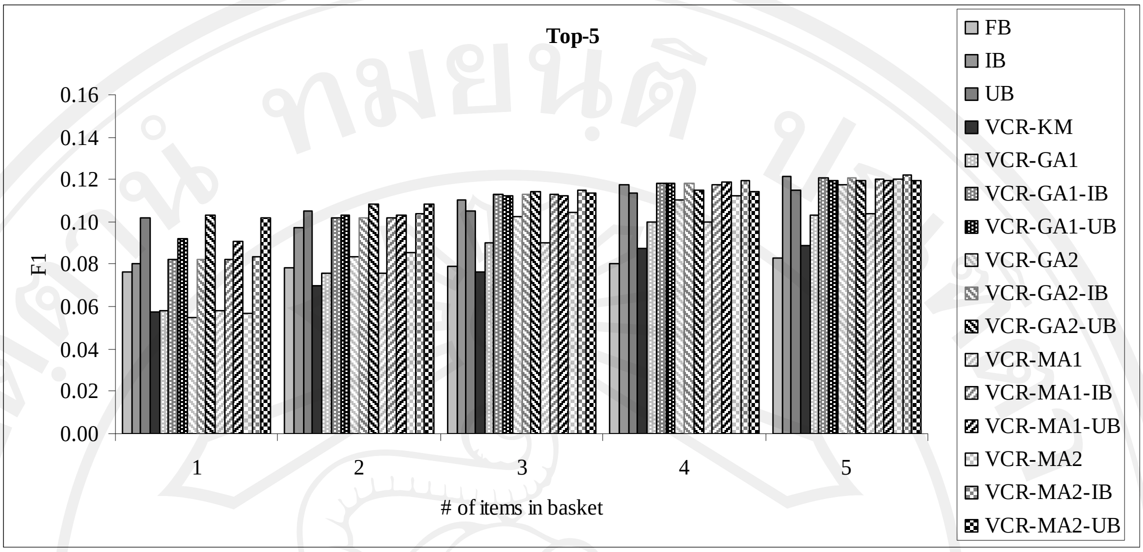

Figure 4.66 Average F1 on MovieLens data set using the proposed methods and the common methods.

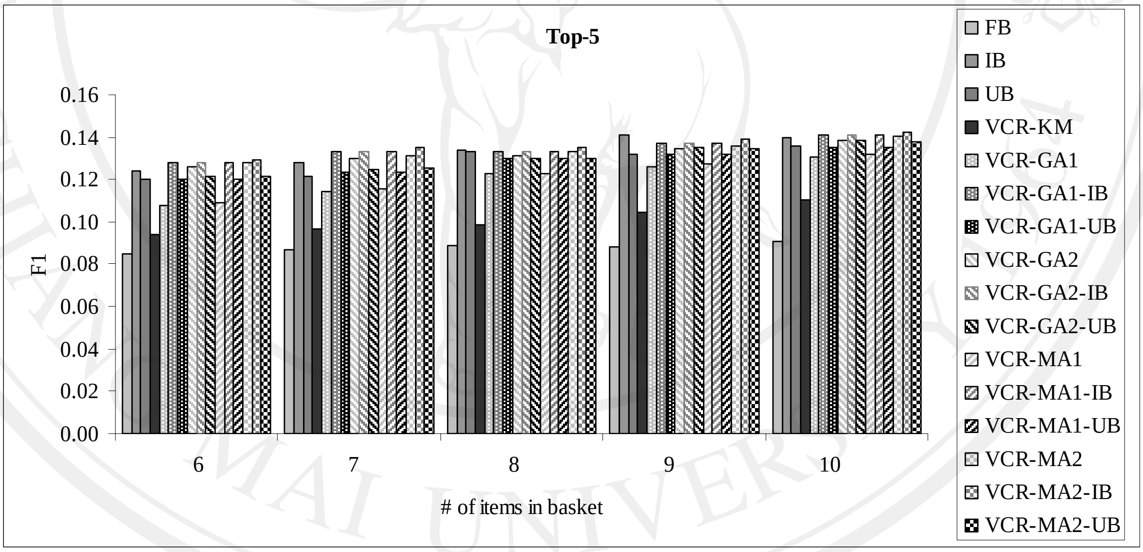

Figure 4.66 Average F1 on MovieLens data set using the proposed methods and the common methods (Continued).

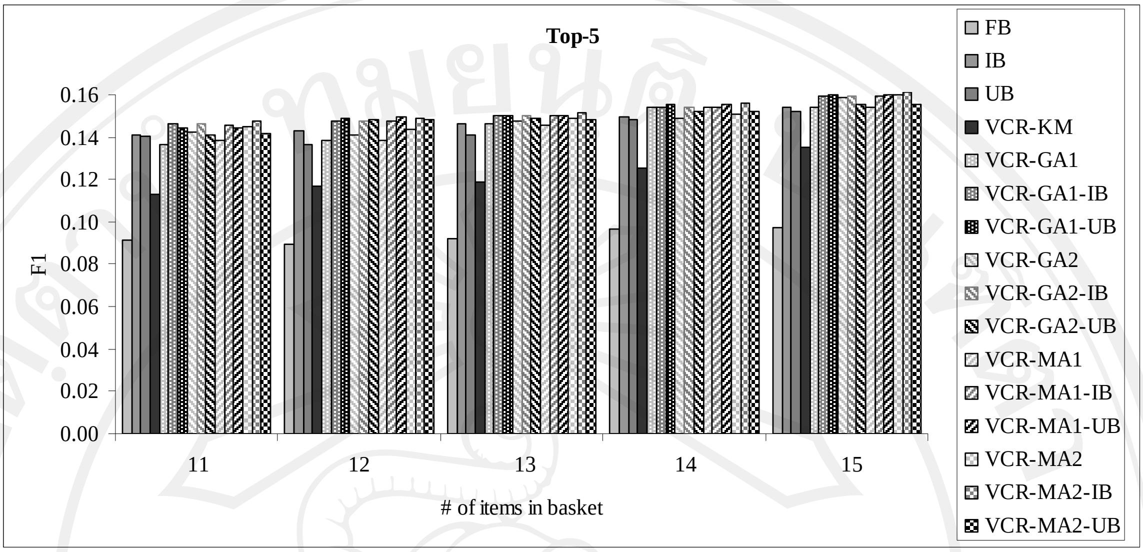

Figure 4.66 Average F1 on MovieLens data set using the proposed methods and the common methods (Continued).

4.6 Discussion

The objectives of our research are to solve the cold-start and sparsity problems in the recommendation systems. There are two main processes in this research. The first process is to develop the new clustering methods to cluster the users and items. The second process is to apply the clustering methods to generate the top-*N* recommendation. In the sparsity problem, we tested the sparsity level on the real-world data sets. The results (see Table 4.1) show that the sparsity-levels of the KDD, TTS, RCM, ECR, and MovieLens are 0.9855, 0.9692, 0.9502, 0.9832, and 0.9369, respectively. From the results of the sparsity-levels, it is hard to determine the sparsity problems in each data set. So, we also tested the number of items purchased or visited in each data set. The results (see Figure 4.1) clearly show that the KDD, TTS, RCM, and ECR have higher sparsity level than the MovieLens. In the cold-start problem, we added each item into the basket to evaluate the recommendation systems. To evaluate our proposed clustering methods, we tested the methods on three synthetic and five real-world data sets. In the real-world data sets, it is extremely impossible to know that our proposed methods are able to properly cluster the users and items. For this reason, we created three synthetic data sets to present the data sets with known ground truths. The results (see Figure 4.2, 4.6, and 4.7) show that our proposed clustering methods are able to properly cluster the users and items. Although the position and shape of each cluster are different from the original image, the elements in each cluster image are the same as the original image. It is clearly that our proposed clustering methods are able to property cluster the users and items. The results of clustering five real-world data sets (see Figure 4.13–4.37) also show that our proposed clustering methods are able to cluster the users and items because the resulting images have fewer numbers of clusters than in the corresponding original images. However, it is extremely to determine the actual number of clusters from a binary image. The sizes of clusters on the five real-world data sets were compared before and after clustering. The results (see Table 4.2–4.7) show that the total number of clusters before clustering is fewer than the total number of clusters after clustering. It is also difficult to identify the well-clusters in our proposed methods. For this reason, we also applied the Dunn’s index to identify the compact and well-separated clusters. Unfortunately, the Dunn’s index is not able to indicate the compact and well-separated clusters in our proposed clustering methods. The Dunn’s index uses the largest diameter of cluster to normalize the distance between clusters. But the diameter of cluster before clustering is smaller than the diameter of cluster after clustering in the original images. Moreover, the distance between clusters is not able to represent the well-separated clusters. Hence, we evaluated our proposed clustering methods with the top-*N* recommendation systems on the five real-world data sets. We compared the top-*N* recommendation systems based on our proposed clustering methods (i.e., VCR-GAs, VCR-GAs-IB, VCR-GAs-UB, VCR-MAs, VCR-MAs-IB, and VCR-MAs-UB) with the common methods (i.e., FB, UB, VCR-KM). We found that our proposed methods yield better performance than the common methods. We also found that our proposed methods help to improve the performance under the cold-start and sparsity problems. We also found that the clustering methods based on the MA and GA with the fitness function in eq.(3.4) perform better than that in eq.(3.1) because the fitness function in eq.(3.1) determines the compactness and number of clusters only. When the clustering methods based on the GA and MA were compared, we found that the methods based on the MA perform better than that on the GA because the MA is the extension of the GA. Hence, all results clearly show that our clustering methods are well-clusters from the results of the top-*N* recommendation systems.