Chapter 3

Research Designs and Methods

This chapter describes the research designs and the proposed methods. There are four sections. Section 3.1 describes the visual clustering method based on the genetic algorithm. Section 3.2 explains the visual clustering method based on the memetic algorithm. The model of the top-*N* recommendation system including the recommendation engines is described in section 3.3. Section 3.4 describes the evaluation measure for the top-*N* recommendation system.

3.1 Model of the Visual Clustering Method Based on the GA (VCM-GA)

The idea behind the proposed method is the interchanging position of rows and columns for searching the clusters on the binary image [25]. In the proposed method, the process can be divided into the three parts of the recommendation system. Firstly a users-items table is created from the purchased items as shown in Figure 3.1(a). Then the users-items table is mapped into a binary image as shown in Figure 3.1(b). Secondly we apply the genetic algorithm and binary image manipulation method to cluster users and items. Finally, we use the information in the clusters to recommend items to the active users.

Items

Users	1	2	3	…		*n*
1	0	1	0	0	0	0
2	1	1	0	0	0	0
3	0	1	1	0	0	0
⋮	0	1	1	0	0	0
	0	0	1	0	1	1
m	0	0	0	0	0	1

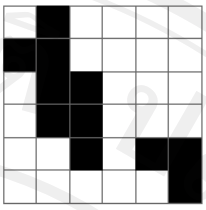


Figure 3.1 (a) Users-items table, (b) Binary image.

The genetic algorithm is used to find the optimized clusters in the binary image. Figure 2.6 shows the process of the genetic algorithm. It can be divided into five parts: initial population, evaluation, reproduction, crossover, and mutation.

3.1.1 Initial Population

In the process of initial population [25, 37], each chromosome is created by randomly interchanging the position of rows and columns. A chromosome can be divided into two genes: the first gene represents the users (rows) and the second gene represents the items (columns). The initial population is shown in Figure 3.2.

	Gene 1				Gene 2				
	R1	R2	R3	R4	C1	C2	C3	C4	C5
Image 1	1	2	3	4	5	1	2	3	4
Image 2	2	3	1	4	3	4	2	5	1
Image 3	4	1	2	3	1	2	4	3	5
…									
Image *n*	2	3	1	4	3	2	1	5	4

Figure 3.2 Example of initial population.

3.1.2 Evaluation

Evaluation is natural selection [25, 37]. The chromosomes with high fitness values are selected into a mating pool. In evaluation, we develop two fitness functions. The first fitness function calculates the fitness value. The number of the

clusters and the compactness of objects in the image are used to determine the fitness value, i.e.,

$$Fitness_i = a_1(1-\alpha_i) + a_2\beta_i, \tag{3.1}$$

$$\alpha_i = \frac{N_i}{\max_j(N_j)}, i = 1,2,...n, \tag{3.2}$$

and

$$\beta_i = \frac{C_i}{\max_j(C_j)}, i = 1,2,...,n, \tag{3.3}$$

where $Fitness_i$ is the fitness function calculated for the ith binary image. Parameters a_1 and a_2 are weight parameters. α_i is the normalized number of clusters, β_i is the normalized compactness, N_i is the number of the clusters, and C_i is the average compactness in the ith binary image. n is the number of the binary images, i.e., the number of chromosomes in population. This fitness function is high for the image with small number of clusters and the shape of each cluster is close to circle.

The second proposed fitness function proposed improves the weakness of the first fitness function for a recommendation problem. We found that compactness in the first fitness might not be a good indicator of a well-grouped cluster in this problem. For the recommendation problem, a well-grouped cluster can be of any shape, either circle, rectangle, elongated, etc., with many four-connected pixels. Therefore we discard the compactness in this fitness function, but add three more factors. This fitness function is

$$Fitness_i = a_1(1-\alpha_i) + a_2(1-\beta_i) + a_3\gamma_i + a_4\delta_i, \quad (3.4)$$

$$\beta_i = \frac{N_{s,i}}{\max_j(N_{s,i})}, i = 1,2...,n, \quad (3.5)$$

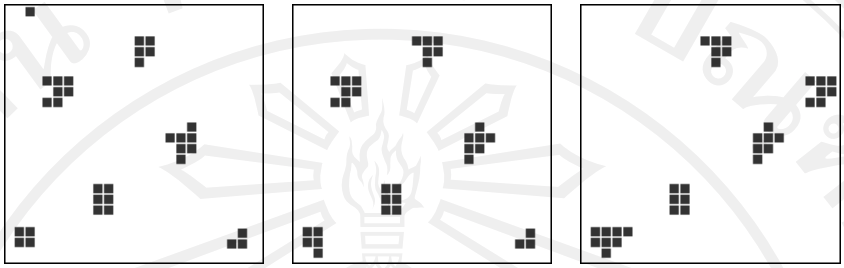
$$\gamma_i = \frac{P_{l,i}}{\max_j(P_{l,i})}, i = 1,2...,n, \quad (3.6)$$

$$\delta_i = \frac{P_{s,i}}{\max_j(P_{s,i})}, i = 1,2...,n. \quad (3.7)$$

α_i is defined as in eq.(3.1) to deal with the number of clusters. β_i is used to deal with the number of small clusters. The number of small clusters in the ith image is denoted by $N_{s,i}$. The clusters with less than four pixels are considered as small clusters. The size of large cluster (but does not need to be the largest cluster) is taken into account in γ_i. $P_{l,i}$ denotes the number of pixels of the third largest cluster in the image. Similarly, the size of small cluster (but does not need to be the smallest cluster) is taken into account in δ_i. $P_{s,i}$ denotes the number of pixels in the small cluster in the ith image. In this fitness function, $P_{s,i}$ is the number of pixels of the smallest cluster with larger than three pixels. The bottom line is that the fitness value will be large when pixels in the image are well-clustered (small α_i), the number of small clusters is small (small β_i), the third largest cluster is large (large γ_i), and the smallest cluster with more than three pixels is large (large δ_i).

Figure 3.3(a)-(c) are the example of three binary images. The fitness value of each image (chromosome) is calculated the fitness values the fitness function. Then,

the fitness value is filled into the Table 3.1. These fitness values are used in the reproduction process.



(a) (b) (c)

Figure 3.3 (a), (b), and (c) Example of the different clusters in binary image.

Table 3.1 Example of three chromosomes in population.

i	Chromosome	$Fitness_i$	P_i	$\sum_{i=1}^{n} P_i$
1	2 3 1 4 7 6 5 | 1 3 2 5 6 4 7	0.45	0.232	0.232
2	1 4 2 3 6 7 5 | 7 1 4 3 2 6 5	0.68	0.351	0.583
3	1 4 2 3 6 7 5 | 7 1 4 3 2 6 5	0.81	0.417	1

3.1.3 Reproduction

Reproduction is a process of matching in with individual chromosomes which are copied to the objective function [25, 37]. The chromosomes with higher fitness values have higher probability of being selected. Two chromosomes are selected from the mating pool of n_k until $n_k - n_{pop}$ to produce two new chromosomes. n_k is the number of the surviving chromosomes that is defined by user.

3.1.4 Crossover Operation

Crossover operation is the creation of children (offspring) from the parents which have been selected in the pairing process [25, 37]. We develop the crossover operation based on an encountering missing method. We design the crossover operation to only cross the value within the gene. For example, consider chromosome

1 (Parent 1) and chromosome 3 (Parent 2) if these are selected from the population in Table 3.1:

Parent 1 = 2 3 1 4 7 6 5 | 1 3 2 5 6 4 7

Parent 2 = 1 4 2 3 6 7 5 | 7 1 4 3 2 6 5

then, the chromosomes are randomly selected to process number 3, 1, 2, and 4 from gene 1 and 3, 2, 5, and 7 from gene 2, i.e.,

Parent 1 = 2 3 1 4 7 6 5 | 1 3 2 5 6 4 7

Parent 2 = 1 4 2 3 6 7 5 | 7 1 4 3 2 6 5

Child 1 = _ _ _ _ 7 6 5 | 1 _ _ _ 6 4 _

Child 1 = 1 4 2 3 7 6 5 | 1 7 3 2 6 4 5

Child 2 = _ _ _ _ 6 7 5 | _ 1 4 _ _ 6 _

Child 2 = 2 3 1 4 6 7 5 | 3 1 4 2 5 6 7

Finally, the parents are replaced with new children, for example, Child 1 is _ _ _ _ 7 6 5 | 1 _ _ _ 6 4 _. Based on encountering missing process number, the missing data from Parent 2 are filled in Child 1. The outcomes of the crossover process are Child 1 = 1 4 2 3 7 6 5 | 1 7 3 2 6 4 5.

3.1.5 Mutation

In the process of the mutation, we designed the mutation operators to allow interchanging of the position only within the same gene [25]. The selected point is replaced by a random point within the same gene. For example, consider chromosome 2 (Table 1) with the value in the column 5 (value is 6), after mutation, the values of the new chromosome is 2 6 1 4 3 7 5 1 3 4 2 5 6 7.

In this research, there are two fitness functions. Hence, the visual clustering based on the genetic algorithm with the fitness function in eq.(3.1) is called the VCM-GA1. The visual clustering based on the genetic algorithm with the fitness function in eq.(3.4) is called the VCM-GA2.

3.2 Model of the Visual Clustering Method (VCM) Based on the MA

In this section, we improve the performance of the VCM-GA1 by using the memetic algorithm to achieve a new clustering method namely VCM-MA1. We also improve the performance of the VCM-GA2 by using the memetic algorithm to achieve the VCM-MA2. The memetic algorithm (MA) is the extension of the GA [6]. It is available in the area of the search optimization [6, 22, 23, 38]. The main process of the MA is the same as GA process, i.e., initial population, evaluation, reproduction, crossover, and mutation. Moreover, the chromosome in the GA is called a genotype. The gene in the chromosome is called a meme. The detail of the GA is described in section 3.1. The extension process of the MA is a local search method. The local search method was added at the end of the mutation operation. The objective of the local search process in the MA is to refine individually [6, 39], i.e., improve their fitness by using the local search method. There are many local search methods such as

hill climbing, simulated annealing, tabu searches [22, 38, 40]. The standard of the local search method in the MA is the hill climbing algorithm [22, 38-40]. In this research, we selected the hill climbing method. We have described the details of the hill climbing method here. Figure 2.8 shows the process of the memetic algorithm.

Figure 3.4 shows the hill climbing local search algorithm. The process begins with the new population. The new population created after the mutation process is derived from the input of the hill climbing algorithm. Then, a neighborhood for each genotype, actual genotype, is created for selecting the best genotype, i.e., new genotype. The new genotype is compared with the actual genotype. If the fitness value of the new genotype is better than the actual genotype, the actual genotype is replaced with the new genotype. The iteration of this process is defined by the user who creates the system. In this research, ten percent of the row and column sizes are the input genotype of the hill climbing algorithm. The ten percent of the row and column sizes are interchanged within the same meme to create the neighborhood. The best genotype in the neighborhood is selected and replaces the old genotype.

```
Hill Climbing Local Search (actual genotype):
Begin
        While (termination condition)
                New genotype ← neighbors (actual genotype);
                If new genotype is better than actual genotype then
                        actual genotype ← new genotype
                End if
        End while
End
```

Figure 3.4 The hill climbing local search algorithm.

3.3 Model of the Visual Clustering Method (VCM) Based on the *K*-means Algorithm

The model of the visual clustering by using the *k*-means algorithm (VCM-KM) is to cluster the users and items. There are three processes. The first process is to cluster the users (rows). The features are items (columns). The second process is to cluster the items (columns). The features are users (rows). The third process is to group the elements in the same cluster, i.e., group users (rows) and group items (columns). A cluster is a group of elements using the 4-connected neighborhood.

3.4 Models of the Top-*N* Recommendation Systems Using VCM-GAs, VCM-MAs, and VCM-KM

The models of the top-*N* recommendation systems using VCM-GAs, VCM-MAs, and VCM-KM consists of two processes. The first process is to cluster the users and items in a binary image using the VCM-GAs ,VCM-MAs, and VCM-KM. In this process, the users-items table is created from the transaction records (purchasing records). The users-items table is then mapped into a binary image. After creating the binary image, the VCM-GAs, VCM-MAs, and VCM-KM are applied to find the optimized clusters in the binary image. The extra details of the first process were

described in sections 3.1 and 3.2. The second process is to create the recommendation engines for the top-*N* recommendation systems. The following detail of the recommendation engine is described. Figure 3.5 shows the model of the proposed top-*N* recommendation systems.

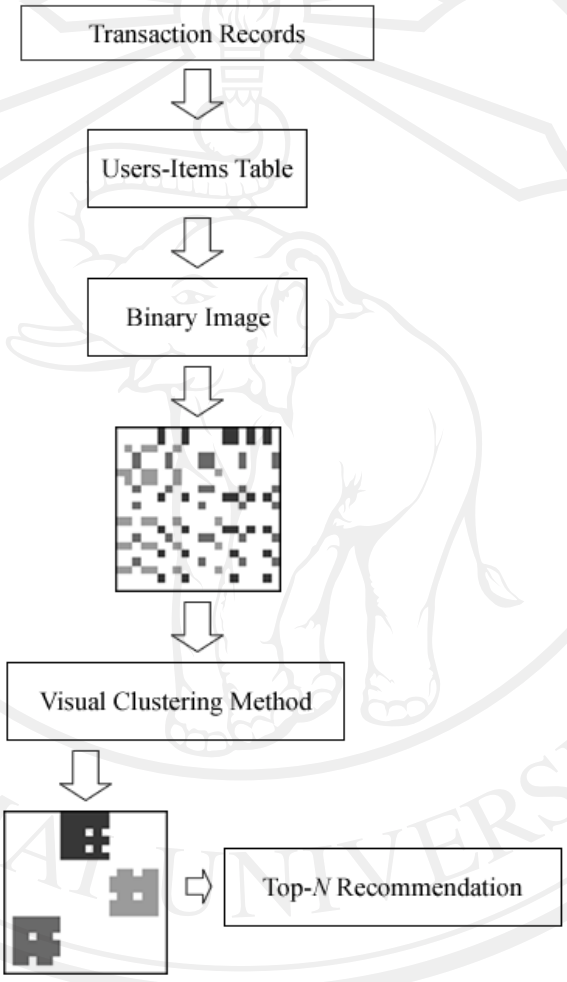


Figure 3.5 Model of the proposed top-*N* recommendation systems.

The recommendation engine is the main process of the RS, we have developed thirteen recommendation engines. The first recommendation engine, namely VCR-GA1, is designed to directly use the information in the derived clusters to generate the top-*N* items to an active user. This recommendation engine used the visual clustering method (VCM) based on the GA with the fitness function in eq.(3.1) to derive the

clusters. The process of all of the generating top-*N* items is the same as the first recommendation. The top-*N* most frequency items which will be recommended to the active user come from the union of the element in the clusters that the purchased items of the active user belong to. The second recommendation engine, namely VCR-GA-UB, is combination of the VCR-GA1 and UB. The third recommendation engine, namely VCR-GA1-IB, is the combination of the VCR-GA1 and IB. The fourth recommendation engine, namely VCR-GA2, used the VCM based on the GA with the fitness function in eq.(3.4) to derive the clusters. The fifth recommendation engine, namely VCR-GA2-UB is the combination of the VCR-GA2 and UB. The sixth recommendation engine, namely VCR-GA2-IB, is the combination of the VCR-GA2 and IB. The seventh recommendation engine, namely VCR-MA1, used the VCM based on the MA with the fitness function in eq.(3.1) to derive the clusters. The eighth recommendation engine is the combination of the VCR-MA1 and the UB namely, VCR-MA1-UB. The ninth recommendation engine, namely VCR-MA1-IB, is the combination of the VCR-MA1 and IB. The tenth recommendation engine, namely VCR-MA2, used the VCM based on the MA with the fitness function in eq.(3.4) to derive the clusters. The eleventh recommendation engine, namely VCR-MA2-UB, is the combination of the VCR-MA2 and UB. The twelfth recommendation engine, namely VCR-MA2-IB, is the combination of the VCR-MA2 and IB. The thirteenth recommendation engine, namely VCR-KM, used the VCM based on the *k*-means clustering method to derive clusters.

3.5 Evaluation Measure for the Top-*N* Recommendation System

The F-measure is used to evaluate the performance of the top-*N* recommendation systems [6, 12, 15, 43-45]. There are three parameters on F-measure: precision, recall, and F1. Precision is defined as the ratio of the number of elements in the hit set to the number of elements in the recommendation set. Calculating the precision is given by

$$\text{precision} = \frac{\#\text{of hits}}{\#\text{of recommended items}}. \tag{3.8}$$

Recall is defined as a ratio of the number of elements in the hit set to the number of purchased items. Calculating the recall is given by

$$\text{recall} = \frac{\#\text{of hits}}{\#\text{of purchased items}}. \tag{3.9}$$

F1 is a parameter combining precision and recall. Calculating F1 is given by

$$\text{F1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precison} + \text{recall}}. \tag{3.10}$$

In the experiments, we evaluate the performance of the recommendation systems by using the F-measure.