Chapter 2

Principles and Theories of the Study

This chapter describes the principles and theories of the study. There are four sections. Section 2.1 describes the fundamental of clustering method. Section 2.2 describes the top-*N* recommendation systems including frequency-based method, user-based collaborative filtering method, and item-based collaborative filtering methods. Section 2.3 describes the genetic algorithm (GA). Finally, section 2.4 describes the memetic algorithm (MA).

2.1 Fundamental of Clustering Algorithm

Clustering algorithm is the process of grouping a set of objects into the classes of similar objects [24]. A cluster is a group of objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. The clustering algorithm has been widely applied in many applications such as data mining, machine learning, and information retrieval [3, 16, 17, 25-30]. The basic types of data in clustering algorithm can be classified into (1) interval-scaled, binary, categorical, ordinal, and ratio-scaled variables, (2) variables of mixed type, (3) vector objects.

Interval-scaled variables are continuous measurements of a roughly linear scale such as weight and height, latitude and longitude coordinates. The data should

be standardized. Standardizing measurements attempt to give all variables an equal weight.

A binary variable has only two states: 0 or 1, when 0 means that the variable is absent and 1 means that it is present. A binary variable can be classified into symmetric and asymmetric.

Table 2.1 shows the 2-by-2 contingency table, where q is the number of variables that equal 1 for both objects i and j, r is the number of variables that equal 1 for object i but that are 0 for object j, s is the number of variables that equal 0 for object i but equal 1 for object j, and t is the number of variables that equal 0 for both objects i and j. The total number of variables is p, where $p = q + r + s + t$.

A binary variable is symmetric if both of its states are equally valuable and carry the same weight. For example, *gender* is a binary variable that have two states: male and female. The symmetric binary dissimilarity between two objects i and j is defined as

$$d(i, j) = \frac{r + s}{q + r + s + t}. \tag{2.1}$$

A binary variable is asymmetric if the outcomes of the states are not equally important such as the positive and negative outcomes of a disease. The asymmetric binary dissimilarity between two objects i and j is defined as

$$d(i, j) = \frac{r + s}{q + r + s}. \tag{2.2}$$

Table 2.1 A contingency table for binary variables.

object $j\rightarrow$ / object $i\downarrow$	1	0	Sum
1	q	r	$q+r$
0	s	t	$s+t$
Sum	$q+s$	$r+t$	P

A categorical variable has more than two states. Let the number of states of a categorical variable be *M*. The states can be denoted by letters, symbols, or a set of integers. For example, *m_color* is a categorical variable that may have three states: red, yellow, and green. The dissimilarity between two objects *i* and *j* is defined as

$$d(i, j) = \frac{p - m}{p}, \tag{2.3}$$

where *m* is the number of matches (i.e., object *i* and object *j* are in the same state), and *p* is the total number of variables.

An ordinal variable has an inherent order to the relationship among the different categories. Examples of ordinal variables include education level (e.g. elementary, secondary, college), satisfaction level (very dissatisfied, dissatisfied, neutral, satisfied, very satisfied). The treatment of ordinal variables is quite similar to that of interval-scaled variables when computing the dissimilarity between objects.

A ratio-scaled variable makes a positive measurement on a nonlinear scale. The data examples include the growth of a bacteria population or the decay of a radioactive element. There are three methods to compute the dissimilarity between objects. The first method is to treat ratio-scaled variables like interval-scaled. The second method is to apply logarithmic transformation to a ratio-scaled variable *f*

having value x_{if} for object i by using the formula $y_{if} = \log(x_{if})$. The y_{if} values can be treated as interval-valued. The third method is to treat x_{if} as continuous ordinal data and treat their ranks as interval-valued.

A variable of mixed types can contain all of the six variable types listed above. The dissimilarity $d(i,j)$ between objects i and j is defined as

$$d(i, j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}, \qquad (2.4)$$

where the indicator $\delta_{ij}^{(f)} = 0$ if either (1) x_{if} or x_{jf} is missing, or (2) $x_{if} = x_{jf} = 0$ and variable f is asymmetric binary; otherwise, $\delta_{ij}^{(f)} = 1$. The contribution of variable f to the dissimilarity between i and j, that is, $d_{ij}^{(f)}$, is computed dependent on its type:

- If f is interval-based: $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}}$, where h runs over all nonmissing objects for variable f.
- If f is binary or categorical: $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$; otherwise $d_{ij}^{(f)} = 1$.
- If f is ordinal: compute the ranks r_{if} and $z_{if} = \frac{r_{if} - 1}{M_f - 1}$, and treat z_{if} as interval-scaled.
- If f is ratio-scaled: either perform logarithmic transformation and treat the transformed data as interval-scaled; or treat f as continuous ordinal data, compute r_{if} and z_{if}, and then z_{if} as interval-scaled.

The major clustering algorithms can be divided into four groups, i.e., partitioning, hierarchical, density-based, and grid-based algorithms. The partitioning algorithm is the similar idea to that in the research. So, only the partitioning algorithm is described in this section. The most commonly used partitioning algorithms are

k-means and *k*-medoids. Each cluster's center of the *k*-means algorithm is represented by mean value of the objects in the cluster. Figure 2.1 shows an example of the clustering a set of objects based on *k*-means algorithm. The symbol, "+", is the mean of each cluster.

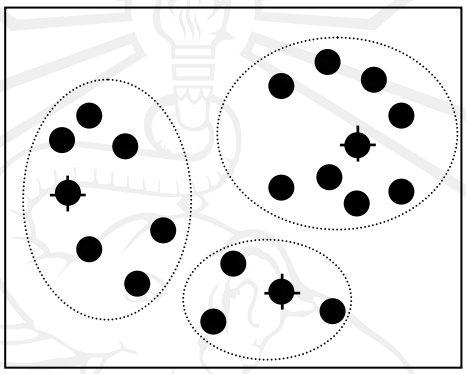

Figure 2.1 Clustering of a set of objects based on the *k*-means algorithm.

The process of *k*-means algorithm starts with randomly selecting *k* of the object. The mean of each cluster is then initialized. Each object is assigned to the cluster which it is most similar based on the distance between the object and the cluster mean. Then, the new cluster mean is computed for each cluster.

The process of computing new cluster iterates until the criterion function converges. Commonly, the square-error criterion is used. It is defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2 , \qquad (2.5)$$

where E is the sum of the square error for all objects in the data set. In the data set p is the object in space representing a given object and m_i is the mean of cluster C_i.

2.2 Top-*N* Recommendation Systems

The methods of the recommendation system (RS) in the literature reviews can be categorized as: (1) content-based (2) demographic-based algorithms, (3) feature-based, and (4) collaborative filtering. In the content-based algorithm [1, 2], the RS recommends items to an active user based on a description of the item and the user profile. The active user is the user who is using the RS. The user profile is based on items that user has liked in the past. The content-based recommendation system compares the profile of the item to the user profile to decide on its relevancy to the user. In the demographic-based algorithm [9, 10], the recommendation system aims to categorize the user based on personal attributes and makes a recommendation based on demographic classes. The user profile can be created from the demographic data such as age, gender, education, location, etc. The demographic data can be used to identify the type of users that likes a certain item. In the feature-based algorithm [7, 8], the recommendation system recommends items to an active user based on the active user profile and product profile. The product profile is created from the product features. Then, the product profile is used to create the user profile. The system generates items that have not yet been purchased to the active user by comparing the profile of the active user to the product profile. In the collaborative filtering algorithm [1-6], it is wildly applied in the area of the RS. It can be divided into the user-based and item-based algorithms. However, the basic of the recommendation system is the frequency-based method. The frequency-based, user-based, and item-based methods are described in the next subsections.

2.2.1 Frequency-Based Method

The frequency-based method (FB) is a basic method for the RS. The RS generates the top-*N* items to an active user by sorting the frequency count of the purchased items and produces the *N* most frequent items that have not yet been purchased to the active user. The basic of the frequency-based recommendation system is shown in Figure 2.2.

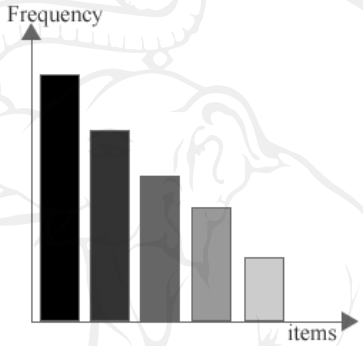

Figure 2.2 Example of the sorted histogram used top-*N* frequency-based recommendation system.

2.2.2 Item-Based Top-*N* Recommendation Algorithm

In the item-based collaborative filtering algorithm (IB), the recommendation system produces the top-*N* items to an active user by calculating similarity between items [4, 5, 31]. This algorithm uses item-to-item similarity to compute the relations between the items. All of the application in the RS, the similarity measure is the cosine measure. The cosine measure is shown in eq.(2.6). In the process of recommendation, the similarities of items are computed. Their corresponding similarities are recorded. Then, for each user who has purchased a set *U* of items, this information is used to compute the top-*N* recommendation. The process of the top-*N* recommendation can be divided into three steps. The first step, the set *C* of candidate items is identified by taking the union of *k* most similar items for each item $j \in U$.

The set *C* removes from the union any items which are already in *U*. The second step, for each $c \in C$ its similarity is computed to set *U* as the sum of the similarities between all the items $j \in U$ and *c* using only the *k* most similar items of *j*. Finally, the items in set *C* are sorted in non-increasing order with respect to that similarity. Then the first *N* items are selected as the top-*N* recommendation items set. Figure 2.3 shows the concept of the item-based top-*N* recommendation system. Figure 2.4 shows the process of the item-based top-*N* recommendation system.

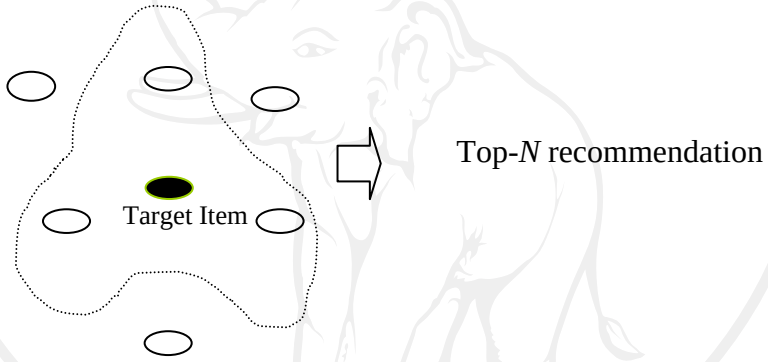

Figure 2.3 Concept of the item-based top-*N* recommendation system.

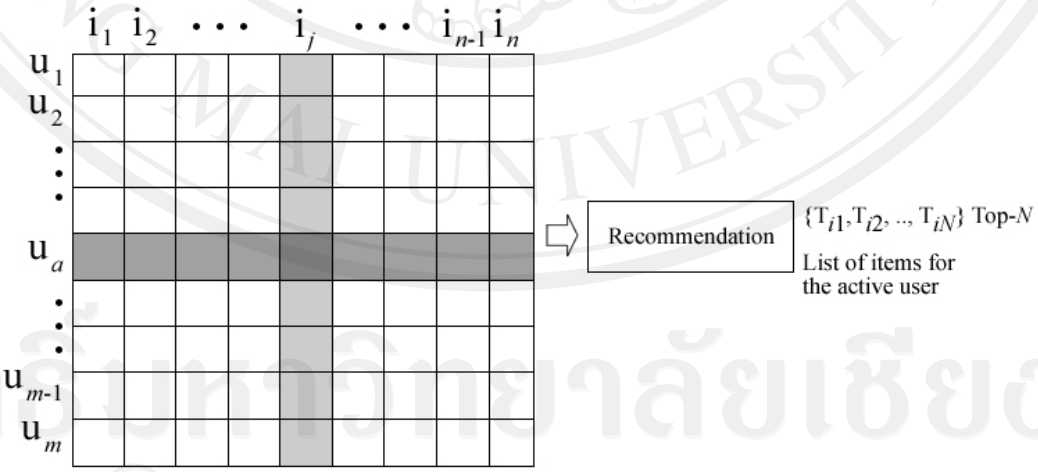

Figure 2.4 Item-based collaborative filtering process for the top-*N* recommendation.

$$Similarity = \frac{A.B}{\|A\|\|B\|}. \quad (2.6)$$

2.2.3 User-Based Top-*N* Recommendation Algorithm

In the user-based collaborative filtering algorithm (UB), this method produces the top-*N* items by calculating similarity between active users with other users who had made similar purchases. Figure 2.5 shows the concept of the user-based top-*N* recommendation system.

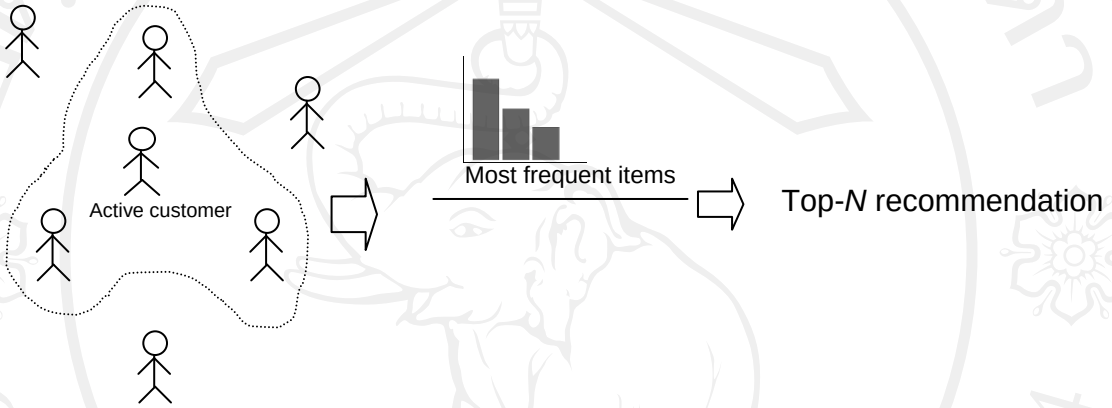

Figure 2.5 Concept of the user-based top-*N* recommendation system.

To generate a recommendation, the user-based collaborative filtering algorithm is to create the top-*N* recommendations from neighborhood of users by using the most-frequent items recommendation technique. This technique looks into the neighborhood *k*. The neighborhood of the active user is computed by using similarity measure. Mostly, the similarity measure is cosine. After all neighbors are accounted, the recommendation system sorts the items according to their frequency count and produces the *N* most frequent items as shown in Figure 2.2. Finally, the recommendation system recommends the top-*N* items that have not yet been purchased by the active user.

2.3 Genetic Algorithm (GA)

Genetic algorithm (GA) is a search of heuristic algorithm that mimics the process of natural evolution. It is widely applied with many applications including machine learning, data mining, and information retrieval [32-35]. In the recommendation systems, the GA is widely applied to improve the performance [3, 16, 19, 20, 36]. The process of GA can be divided into five parts: (1) initial population, (2) evaluation, (3) reproduction, (4) crossover operation and (5) mutation operation. The population of the GA is a group of chromosomes consisting of genes (an array of values). By mimicking the natural selection, the chromosomes with a high fitness value are selected into a mating pool. The reproduction process occurs in the pool by copying individual chromosomes to the next generation. The crossover operation creates children from the parents based on the paring process. The mutation operation aims to maintain genetic diversity from one generation of the population to the next.

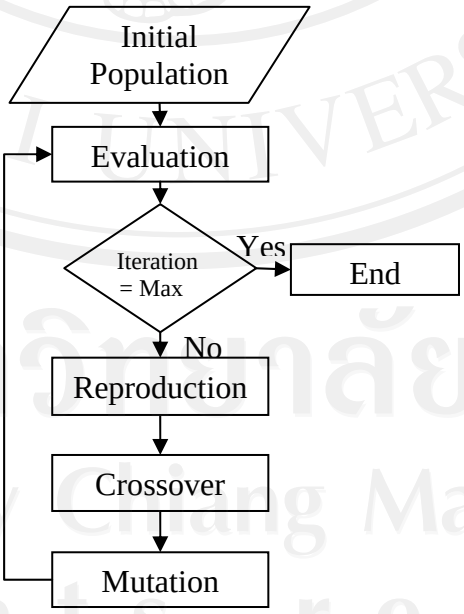

Figure 2.6 Process of the genetic algorithm.

Initial population is a group of chromosomes which are encoded to an optimization problem. n_k is the number of chromosomes in the population. The basic of a chromosome is binary. A chromosome consists of the gene which is represented with bit strings. Table 2.2 shows the example of the initial population containing four chromosomes.

Table 2.2 Example of the initial population.

n	Chromosome
1	00110010011000
2	00110110011110
3	10110110011001
4	11110010011000

Evaluation is natural selection that occurs in the iteration of the algorithm. To produce two new offspring, two chromosomes are selected from the mating pool of n_k until $n_k - n_{pop}$ chromosomes. n_{pop} is the number of chromosomes in the population. The basic selection methods are: pairing from top to bottom, random pairing, weight random pairing, cost weighting, and tournament selection [37]. The weight random pairing, roulette wheel weighting, are widely applied in the GA applications.

Table 2.3 is an example of the roulette wheel weighting. The rank weighting finds the probability from the rank, *n*, of the chromosome:

$$P_n = \frac{N_k - n - 1}{\sum_{n=1}^{N_k} n}. \qquad (2.7)$$

Reproduction begins by selecting of parents. The parents are selected from the mating pool depending on their fitness values. The selected parents are used to create the children to the next generation.

Crossover operation is the creation of children (offspring) from parents which are selected in the paring process. The simple crossover can be divided into two steps. Selecting the parents, i.e., is the first step. The second step is to cross the information between the parents to generate the children. Figure 2.7 shows a simple crossover operation.

Table 2.3 Example of the roulette wheel weighting.

i	Chromosome	P_i	$\sum_{i=1}^{n} P_i$
1	00110010011000	0.4	0.4
2	00110110011110	0.3	0.7
3	10110110011001	0.2	0.9
4	11110010011000	0.1	1.0

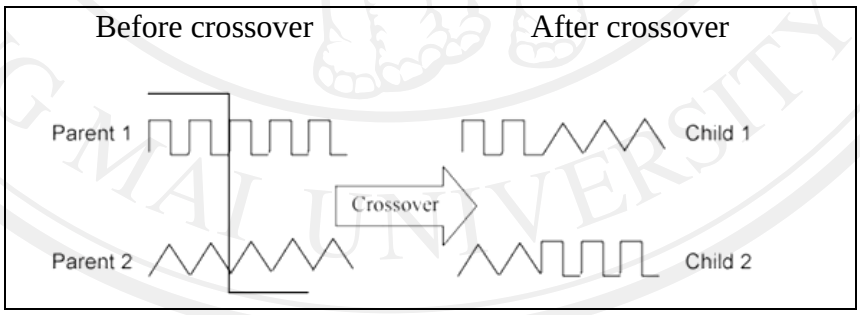

Figure 2.7 A simple crossover schematic.

Mutation operation is a genetic operator used to maintain genetic diversity from one generation of a population of chromosomes to the next. In other word, the mutation operation is local modification which restores lost information to the population. The underlined bit string is the example of the mutation operation.

Before: 00110010011000

After: 00110010011001

2.4 Memetic Algorithm (MA)

Memetic algorithm (MA) is an evolutionary algorithm that is an extension of the traditional genetic algorithm [6, 22, 23, 38, 39]. The MA is widely applied in many area including data mining, machine learning, and information retrieval [40-42]. A local search process is the extension of the genetic algorithm to improve the problem solutions. The process of the MA can be divided into six process including initial population, evaluation, reproduction, local search, crossover operation, and mutation. The five main processes, i.e., initial population, evaluation, reproduction, crossover, and mutation, are the same as in the GA process. The population of the MA is a group of genotypes (chromosomes). A genotype consists of memes (gene). In the natural selection, some of the genotypes with the better individuals are selected into a mating pool. In the reproduction process, the selected individuals are copied to the next generation. The recombination operation is applied to two selected individuals (parents) to create two new offspring genotypes (children). The goal of the mutation operation is to maintain genetic diversity from current generation of the population to the next. The local search is the main process which is added to the GA. Hence, only the local search is described in this section. The process of the MA-based algorithm is shown in Figure 2.8.

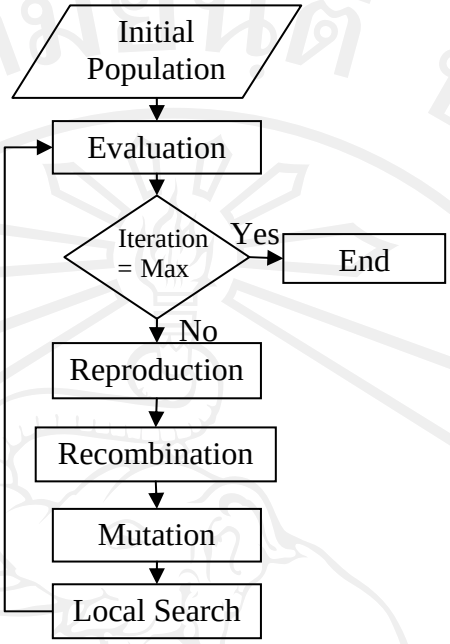

Figure 2.8 Process of the MA-based algorithm.

The hill climbing, simulated annealing, and tabu search algorithms are the basic of the local search of the MA. The hill climbing search is the standard local algorithm [38, 39]. Creating neighborhood of each individual and identifying the better individual in the neighborhood are the concept of the hill climbing. The simulated annealing algorithm starts with a random initial placement. The move operator is the placement through a defined move. The fitness value is calculated for the change in the score due to the move made. The selected individual depends on the change in the score, accept or reject the move. The final process of the simulated annealing algorithm is updated and repeat processes. The tabu search algorithm is the meta heuristic. It guides a local search produce to explore the solution space. The memory-based algorithms are the hallmark of tabu search algorithm. The tabu search begins with generating the initial population and the initial memory structures. Then,

the neighborhood is generated and modified to select the best solution to update the tatu list in the memory structures.