A recommender system based on interactive evolutionary computation with data grouping

Hyun-Tae Kim, Jong-Hyun Lee, Chang Wook Ahn*

*School of Information & Communication Engineering, Sungkyunkwan University (SKKU) Cheoncheon-dong, Suwon 440-746, Republic of Korea

Abstract

Nowadays, recommender systems are widely applied in e-commerce websites to help customers in finding the items they want. A recommender system should be able to provide users with useful information about the items that might be interesting to them. The ability of immediately responding to changes in users preferences is a valuable asset for such systems. This paper presents a novel recommender system that combines two methodologies, interactive evolutionary computation and content-based filtering method. Also, the proposed system applies clustering to increase the time efficiency. The system aims to effectively adapt and respond to immediate changes in users preference. The experiments conducted in an objective manner exhibit that the proposed system is able to make recommendation with ensuring quality and speed.

Keywords: recommender system; information filtering; interactive evolutionary computation

1. Introduction

In daily life, many people often encounter the situations that they have to make a decision to purchase some products without enough personal experience of various alternatives. In some cases, customers are likely to inquire someone’s help, such as talking with a friend who has a plenty knowledge of the product, consulting an expert, or reading a magazine and article related to the decision. Such various references may be helpful to the users in making an appropriate decision. On this point, a recommender system has same usefulness but provides the user with refined recommendation of alternatives which are tailored to users preferences [1], [2].

Recommender systems are helpful to the people living in these days. After the 1990s, there were impressive changes in technologies related to information and entertainment. With this change, numerous resources, such as TV channels, books, music and interactive documents on the World Wide Web, were exposed to the people [3]. The technologies have grown with remarkable speed, and many people still have information overload problems. In this regard, many of e-commerce site have applied recommender systems in response to this problem, such as Amazon.com, eBay and CDNow [3], [4].

In this paper, we propose a new recommender system by combining content-based filtering with interactive evolutionary computation, and apply clustering to the system to improve the quality of recommendation. We expect that our proposed system will provide more appropriate information.
This paper is organized as follows. Section 2 reviews the related work. Section 3 describes the structure of our proposed system and explains how to operate each component in this system. In Section 4, experimental results and analysis are provided. Finally, Section 5 concludes this paper.

2. Related Work

2.1. Recommender System

The purpose of recommender systems is to provide information fulfilling users’ need. In recommender system, a variety of methods have been emerged as the basis for recommender. There are two main methods: collaborative filtering and content-based filtering [1], [6].

In collaborative filtering method, the recommender system discovers relations between each profile stored in its system. After identifying correlation of each profile, the system classifies user profiles. The system provides recommendation derived from other user profiles having similar ratings in the past [5], [6].

Otherwise, in the content-based filtering method, the recommender system extracts a set of features from items rated by users. After this step, the system analyzes the similarity between processed items and all of remaining items. The system recommends items that have similar features which the user preferred in the past [5], [6].

However, existing recommendation methods have the limitation; in content-based filtering, the recommender systems have the overspecialization problem such that it can only make the recommendation biased toward items which are too well-tailored to a user’s preference. To overcome this limitation, we proposed new recommender system by combining the content-based filtering method and interactive evolutionary computation.

2.2. Feature extraction

The feature extraction is a technique that derives properties from specific data, such as document, music and photos. As mentioned earlier, the extracting features from items are an essential step in the content-based filtering method. In our proposed system, we employ a feature extraction tool (i.e., CLAM [16]) to analyze the properties of items. The CLAM is a software framework for research and application development on the audio and music field.

2.3. Interactive Evolutionary Computation (IEC)

Evolutionary Computation (EC) is the general term for several computational techniques that are based on the evolution of biological life in the nature, among which Genetic Algorithms (GAs) are most widely used. GAs work on a population of candidate solutions; each solution has a fitness value indicating its closeness to the optimal solution of the problem. The solutions having higher fitness values than others are selected, and survive to the next generation. GAs produce better offspring (i.e., new solution) by the combination of selected solutions. The methods can discover, preserve, and propagate promising sub-solutions [7], [8].

Interactive Evolutionary Computation (IEC) is also an optimization method as the genetic algorithm. However, a user takes charge of the evaluation of fitness value for candidate solutions [9], [10].

2.4. Data grouping: K-mean clustering algorithm

In our proposed system, we apply data grouping (i.e., clustering) to improve the computational efficiency in terms of accuracy and quality of recommendation. We employ $K$-mean clustering algorithm [11], which is a technique commonly used to separate a data set into $k$ groups. This technique divides data set based on the features of data set. It starts by choosing $k$ initial cluster centroid, and iteratively refines the given data set [12].

3. System Overview

The recommender system proposed in this paper is based on the interactive evolutionary computation (IEC) to achieve main goal of this system, which are recognition of users preferences and ability to response quickly for users’ behaviors. Our proposed system is composed of three phases as follow: preprocessing, user evaluation and IEC phase.
3.1. Preprocessing phase

In this phase, we first perform feature extraction to each music track using CLAM. It produces XML files as a result; the proposed system then parses the XML files to initialize individuals for IEC. We consider seven extracted features: Tempo, Pitch, Octave, Root, Mode, Desc and Simil, which are real numbers. Also, as shown in Fig. 1, the features are divided into three parts according to their usage.

Once feature extraction is completed, the proposed system makes some groups from the given music data using $K$-mean clustering algorithm. The system refers grouping features of each individual to classify items. The system then stores each item’s group information; the system uses such group information to assist finding similar items.

3.2. User evaluation phase

The proposed recommender system allows its users to evaluate the fitness value of each music track. A user can assign his or her own rating score, which is the most exact way for users to represent their subjective preferences. After having the given items rated by users, the proposed system evolves a population based on the evaluated data.

3.3. Interactive Evolutionary Computation (IEC) phase

The IEC phase is a fundamental element for making recommendations since the system produces promising items to users based on their own evaluations. We consider two genetic operators, selection and crossover; we exclude the mutation operator because it has the potential to make the population deviate from the common patterns of candidate solutions discovered by the evolutionary process. Fig. 2 shows how to operate IEC phase in the proposed system. As shown in Fig 2, IEC works based on user evaluation and it divided by three steps: Selection, Crossover, and Matching.

3.3.1. Selection

We apply the truncation selection method to IEC phase of the proposed system because the item rated lower scores by users should not be considered to make recommendations. The method has the strength of elitism to impose high selection pressure to prefer the top $T\%$ candidate solutions having higher fitness values. The remaining items rated lower scores are then discarded [13], [14]. Once the selection procedure is finished, half of the selected items would be applied the crossover operator in a probabilistic manner.
3.3.2. Crossover

In this step, the proposed system applies the BLX-α crossover method [15] to the selected items from the previous step. As indicated in the preprocessing phase, because the extracted features of each individual are real numbers, we use BLX-α crossover on the selected items.

3.3.3. Matching

This step aims to calculate similarity between music features of items resulted from the previous steps and the remaining items in our proposed system. When the system conducts this step, the system calculates similarity with specific items based on group information. This system calculates the similarity by checking the distance of the features of each item as shown in equation (1).

$$
dist (s, t) = \sqrt{\sum_{j=1}^{m} \frac{1}{n} \sum_{i=1}^{n} (s_j - t_j)^2}
$$

where $dist(s, t)$ is the Euclidean distance between two items. The variable $n$ is the number of music features that each items have; the variable $m$ indicates the length of each features in the individual representation.

4. Experiments and Results

4.1. Implementation

In order to verify the performance of our proposed system, we design a test-agent, which is performed experiment automatically. In our previous work [10], we had tried to perform the experiment by building a test website. It was hard to collect valuable result even though it has better accessibility than agent based experiment as shown in the Fig. 3. Because, it takes long time to gather result so that some users might not concentrate on their experiment steadily. On this point, we decided to apply test agent based experiment. Because, the agent based experiment, it can save time to collect and analyze the result.

4.2. Experiment results

To estimate the performance of our proposed system, we conducted experiments on the test-agent application; the agent processes a list that contains 10 items (i.e., music tracks) for each generation. For each experiment, the agent randomly chooses an item which is a criterion to evaluate fitness values of items in a list as generation goes by. All experiments were run 100 times and 50 generations for each run. We thus measured the change of average fitness values and execution time to verifying the performance of our proposed system. As seen in the Fig. 4, the average fitness value gradually increases with the number of cluster. However, the execution time exponentially decreases as the number of cluster increases.

It denotes that the proposed system can reduce the execution time as the result from making groups (i.e., clustering). Also, the difference of average fitness values between the case using only one cluster and the cases using two or three cluster is enough small considering other results using more than three clusters. In other words, it means that our proposed system can guarantee enough quality of recommendation with efficient execution time.

Fig. 3. The test agent application for experiment
5. Conclusion

In this paper, we presented a new recommender system based on interactive evolutionary computation. Also, we tried to apply data grouping towards our proposed system. The proposed system is able to recognize the trend of user’s preference and then adaptively provide appropriate recommendation with having time efficiency.

To design the system, we incorporated the interactive evolutionary computation with the content-based filtering method. First, the system extracts unique features of each item and it operates based on extracted features. The system grants its user the role of evaluating the fitness value of each item. We then apply the interactive evolutionary computation to obtain the most appropriate items to recommend; we also apply data grouping to the system in order to reduce computational time in finding the candidate item.

The experimental results exhibited that the average fitness values, which are objectively collected by mean of evaluations, increase by degrees as the number of clusters increases. However, the execution time is exponentially decreased within the case of using three clusters. It denoted that the proposed system can produce the appropriate items with saving time. Therefore, we believe that our recommender system has enough potential to be applied to other fields.

Reference