LETTER

A New Evolutionary Approach to Recommender Systems

Hyun-Tae KIM[†], **Jinung AN**[††], ***Nonmembers*, *and* Chang Wook AHN**[†a)], ***Member***

SUMMARY In this paper, a new evolutionary approach to recommender systems is presented. The aim of this work is to develop a new recommendation method that effectively adapts and immediately responds to the user's preference. To this end, content-based filtering is judiciously utilized in conjunction with interactive evolutionary computation (IEC). Specifically, a fitness-based truncation selection and a feature-wise crossover are devised to make full use of desirable properties of promising items within the IEC framework. Moreover, to efficiently search for proper items, the content-based filtering is modified in cooperation with data grouping. The experimental results demonstrate the effectiveness of the proposed approach, compared with existing methods.

key words: *recommender systems, user's preference, interactive evolutionary computation, content-based filtering, data grouping*

1. Introduction

Since the 1990s, information technologies have considerably advanced, and thus a variety of resources have been available on the World Wide Web [1], [2]. However, this change has caused an information overload problem that forces users to examine a vast number of alternatives (i.e., items) in order to make their decision [1], [3]. In this sense, recommender systems have emerged as a promising solution to this problem. The aim of recommender systems is to provide users with information that fulfills their demands or needs. In general, conventional recommender systems have attempted to employ 'information filtering' as a core technique, by which suggestions appropriate to the user's preference can be produced [1]–[3].

Interactive evolutionary computation (IEC) inspired by natural selection and genetic inheritance is a population-based search optimization method [4]. However, IEC makes use of human knowledge or intuition in the course of optimization; IEC let users evaluate individuals (i.e., candidate solutions) by themselves. Thus, users can directly assign their preference information to each individual. This aspect is suitable to trace the user's uncertain and time-varying preference. Also, IEC has been successfully applied to diverse real-world applications, such as music, art, computer-aided design, and recommender systems [3]–[5]. However, most research has been conducted by merely applying an IEC instance to the problems of interest. For example, the conventional iGA (interactive genetic algorithm) was used to recommend products in the E-commerce market [5]; however, its contribution lies in not iGA itself but an automatic extraction of design variables for iGA.

Two information filtering techniques have been widely used in the recommender systems: *content-based filtering* and *collaborative filtering*. The former provides items similar to the things that the user preferred in the past, the latter items that other users with similar preference liked [1]. However, these methods have severe limitations. For instance, when a set of new items is added to the recommender system, adequate recommendations related to the new items cannot be made at that juncture; it is known as a *sparse problem*. Moreover, the content-based filtering suffers from an *overspecialization problem* such that items having features similar to the things that the user liked before are only recommended. To resolve these problems, various recommendation techniques have been developed so far [1]–[5].

This paper presents a new evolutionary approach to recommender systems by combining the content-based filtering with IEC. In relation to this, new genetic operators are developed to improve the search capability of IEC and the content-based filtering is modified, in concert with data grouping, to rapidly discover suitable items from potential alternatives.

2. Proposed Approach

The process of making proper recommendations can be defined as a search problem. After rating/scoring items from user side, the recommender systems attempt to search for suitable items among numerous alternatives. It denotes that the ability to recognize and respond the user's preference is very crucial for recommender systems. On the other hand, the content-based filtering is robust and efficient to make suggestions in agreement with the user's preference. However, the filtering method often encounters the sparse/overspecialization problem; this method cannot respond to the sudden change of the user's preference. The fusion of content-based filtering and IEC to come up with an efficient recommendation method is suggested in this regard. The proposed method is briefly outlined in Algorithm 1. It is characterized by three phases: *preprocessing*, *IEC*, *information filtering*. For convenience, music items are employed for all necessary examples in the following subsections.

Manuscript received May 28, 2013.
Manuscript revised September 25, 2013.
[†]The authors are with the Department of Computer Engineering, Sungkyunkwan University, Suwon 440–467, Republic of Korea.
[††]The author is with Robot Research Division, DGIST, Daegu 711–873, Korea.
a) E-mail: cwan@skku.edu (Corresponding author)
DOI: 10.1587/transinf.E97.D.622

Algorithm 1 Proposed Recommendation Method

```
/* Preprocessing phase */
Perform feature extraction on each item
Classify all items by applying data grouping to the extracted features
while 'termination' do
  /* IEC phase */
  Encode individuals of IEC by the features (only on the first visit)
  Apply fitness-based truncation selection and feature-wise crossover
  if 'overflow' in the list then
      Randomly discard a proper number of offspring
  end if
  if 'underflow' in the list then
    Fill the shortage with randomly chosen parents
  end if
  /* Information filtering phase */
  if 'makeup item' then
    Search for the closest item within its own group
  end if
  if 'virtual item' then
    Find the closest group between the item and the other groups
    Discover the closest item within the found group
  end if
  Add the found items into the (recommendation) list.
end while
```

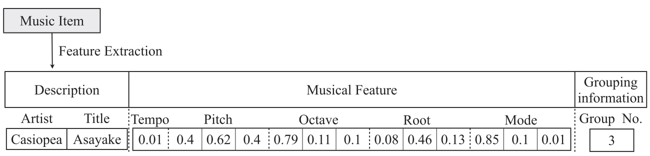

Fig. 1 An example of encoding individuals based on extracted features.

2.1 Preprocessing Phase

In this phase, feature extraction and data grouping are performed on all items; this is necessary to arrange a working environment of the proposed method. Firstly, the feature extraction is conducted to acquire unique attributes of items, which compose each individual of IEC. Figure 1 illustrates an example of encoding individuals in terms of music features. Next, the data grouping is applied to all the extracted features. The aim is to gather the items with similar features together and process all the groups independently, thereby enhancing the speed and quality of recommendation. The group information of items is attached to their corresponding individuals (see Fig. 1).

2.2 IEC Phase

After completing the preprocessing step, an initial (recommendation) list of items is provided to the user. The list is prepared by randomly choosing alternatives. Then, the user assigns scores (i.e., fitness values) to all items of the list. Generally, the user would not be disturbed by this rating process; for instance, the score of each music item can be automatically computed by the user's behavioral pattern, such as fullback time. The initial list is evolved by repetitively applying genetic operators: *selection* and *crossover*. Note that mutation is not employed here since its negative effect can be overstressed within the IEC framework. In other words, if the desirable features of candidate items are destroyed by mutation, it is hard to rediscover them in an interactive way.

The ordinal selection is preferable due to its consistent selection pressure regardless of fitness distribution [3], [7]. Considering the nature of IEC (i.e., interaction between human and system), satisfactory recommendations should be made as quick as possible. In relation to this, the truncation selection is most promising; however, this scheme is not suitable for the recommender systems since its selection pressure is too strong to avoid premature convergence. Thus, a new fitness-based truncation selection is proposed. Since the range of possible fitness values is available in the recommender systems, all the fitness values (of items) assigned by the user can be readily normalized between $[0, 1]$. For instance, if the user listens to a music item by half, its fitness value would be 0.5. Then, the fitness-based truncation selection chooses all candidate items whose fitness values are better than $\tau \in [0, 1]$. The strength of this method lies in rapidly making recommendations corresponding to the user's preference without having premature convergence due to its adaptive selection pressure; a smaller number of items would be selected at early generations (i.e., a diverse search is conducted) while higher selection pressure is supplied in later generations (i.e., promising items are preserved).

In principle, crossover plays an important role in creating new offspring which retain the properties of their parents. This can be achieved by intermixing good partial-solutions, called building-blocks (BBs), without breaking them. However, the conventional crossover has a critical BB-disruption problem [7]. A new crossover operator, termed 'feature-wise crossover', is designed in this regard. The proposed crossover produces new candidate items (i.e., offspring) by exchanging entire components (i.e., genes) of the parents at the level of features. Only the positions between features become potential crossover points. Note that the offspring cannot be actual recommendation items since the role of crossover is to merely intermix the desirable features of the parents; thus, the offspring is called 'virtual items'. At this juncture, the information filtering is required to come up with actual items with regard to virtual ones, which is described in the next subsection. Figure 2 illustrates an example of the 2-point feature-wise crossover about music items.

When creating the offspring, two problems may take place due to the fitness-based truncation selection: *overflow* and *underflow* in the recommendation list. Figure 3 exemplifies the two problems in detail. Assume that the list consists of N items. If the number of selected items (i.e., parents) is greater than $N/2$, the overflow problem occurs since the crossover produces more than $N/2$ offspring (i.e., virtual items). In this case, a proper number of virtual items are randomly discarded to resolve this problem. Meanwhile, if the number of parents is less than $N/2$, the under flow problem happens due to the generation of less than $N/2$ offspring.

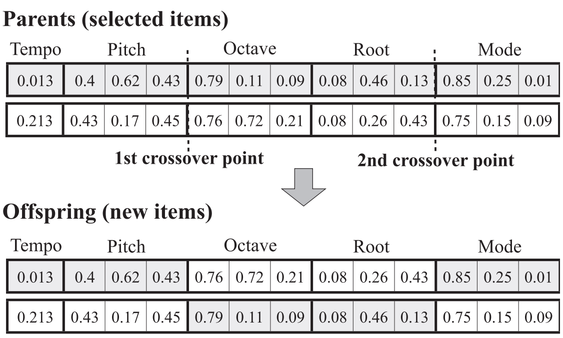

Fig. 2 An example of the 2-point feature-wise crossover.

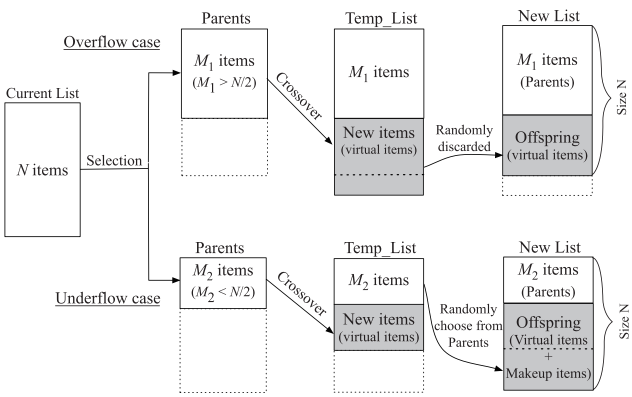

Fig. 3 An example of resolving the overflow/underflow problem.

In this instance, the shortage of offspring is filled with the items randomly chosen from the parents, which are termed 'makeup items'.

2.3 Information Filtering Phase

In order to produce actual recommendation items, an information filtering technique is applied to the new candidate items (i.e., offspring) obtained in the IEC phase. This filtering method discovers actual items which are most similar to the offspring. This task is performed by measuring the similarities between the features of offspring and those of potential items. For the sake of convenience, the Euclidean distance is employed as a metric of similarity.

$$dist(x, y) = \sqrt{\sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} (x_{i,j} - y_{i,j})^2} \tag{1}$$

where $dist(x, y)$ is the Euclidean distance between two items, x and y, and $x_{i,j}$ ($y_{i,j}$) denotes the jth component of the ith feature of the item x (y). Moreover, n is the number of extracted features of each item and m is the number of components of each feature. Prior to computing the distance between two items, the extracted features should be normalized between $[0, 1]$.

In this study, the content-based filtering technique is modified to further enhance the speed of computation. Obviously, the offspring consist of two kinds of items: *makeup* and *virtual*. Since the former is comprised of the randomly chosen parents, each makeup item has its group information. As to each makeup item (i.e., those filled in the underflow case), the closest item based on the similarity metric is found within its own group. This actual item is then added into the recommendation list. Meanwhile, no information about group is available for the virtual items. For each virtual item (i.e., those generated by crossover), the distance (i.e., similarity) between the virtual item and each group's centroid is computed, and the closest group is chosen. Then, the most similar item is searched from the alternatives within the chosen group, and the found item is added into the recommendation list.

3. Experimental Results

The proposed method was evaluated on a set of music items, which consists of a total of 400 MP3 music files. Prior to the experiment, unique music features of each item were extracted by using CLAM [8], which is a widely used extraction tool for music data. A test agent was employed to effectively perform the experiment since consistent data are hard to be collected from an actual environment where real users are directly involved. In the agent-based test, a music item is randomly selected from the music data set. This item acts as a reference of the user's preference. Then the distance between the reference and each item in the list is measured, and this value is utilized as the rating information assigned to the music item by the user. In addition, a change of the user's preference can be emulated by randomly choosing another music item as the reference. Furthermore, the proposed method was compared with other recommendation schemes: an existing interactive evolutionary approach [3] and the conventional content-based filtering [9]. Each method processed 50 lists at every run, and each list was comprised of 10 music items. The quality of recommendation was taken as the performance metric; a smaller similarity value indicates a better recommendation. Due to its simplicity and efficiency, the k-means algorithm [6] was employed as a grouping method in this experiment. In addition, the parameter τ of the fitness-based truncation selection was set to '0.5' and the 2-point feature-wise crossover was employed. All the results were averaged over 200 runs.

Prior to comparative studies, a preliminary experiment was conducted to investigate the effect of the grouping method on the recommendation performance. It was observed that the execution time was remarkably enhanced when two or three groups were used, while the average similarity value deteriorated slightly. This denotes that the data grouping can considerably reduce the computation time without much compromising the recommendation quality. In the following experiments, the number of groups would be set to '2' in view of the quality and speed of recommendation.

Figure 4 compares the average similarity values achieved by the three methods during the process of 50 lists.

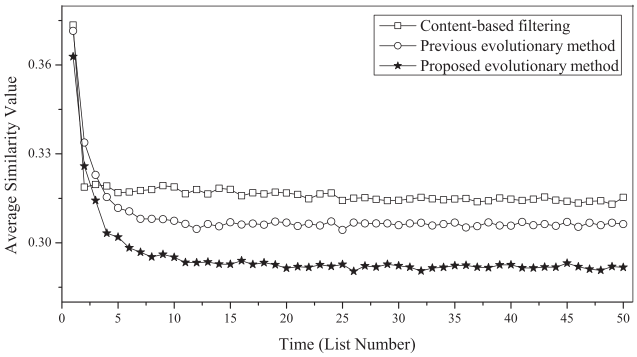

Fig. 4 Performance of the proposed approach and the existing methods.

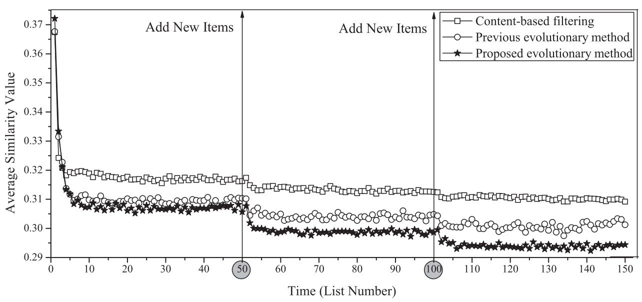

Fig. 5 Comparison of three methods when inserting new music items.

It is shown that all the methods started with a high similarity value since each method created an initial list by randomly choosing music items. As time passed, the proposed method produced music lists with better similarity values than the other references. It denotes that the proposed method makes the most suitable recommendations all the time. This improvement mainly results from the innovation of genetic operators: the fitness-based truncation selection guarantees fast and stable convergence due to its adaptive nature of selection pressure and the feature-wise crossover helps maximally traverse the search space without destroying any desirable feature (i.e., BB).

Figure 5 exhibits the average similarity values with respect to the insertion of new music items. In this experiment, we investigated how efficiently the proposed method treats the newly added items, in which conventional recommendation methods suffer from the sparse problem. The initial data set was composed of 200 music items, and new 100 items were inserted at 50th and 100th time instant. The figure shows that the average similarity values were further improved after adding new music items. This denotes that the proposed method is apt to keep the trend of the past similarity values. Moreover, the proposed method made better recommendations than the references. This result is due to the improved search capability of carefully designed genetic operators and the synergistic effect of content-based filtering and data grouping. Consequently, the proposed approach can effectively handle sudden changes in the environment of recommender systems, such as the user's preference or the addition/deletion of music items.

4. Conclusion

In this paper, a new evolutionary recommendation method that is able to adapt the user's preference and provide appropriate suggestions was proposed. To achieve this goal, the proposed method brought the strengths of content-based filtering and IEC. In this regard, genetic operators were innovated to optimize the exploratory power of IEC, and the content-based filtering was modified to cooperate with the grouping method. The experiments were carried out with a music data set. It was proven that the proposed approach has the capability to produce better recommendations than the existing methods. In addition, a situation of suddenly adding new data sets was investigated by experiment. The results demonstrated the effectiveness of the proposed approach in resolving the limitations of existing recommendation methods.

Acknowledgments

This work was supported by the NRF grant funded by the Korean government (MEST) (no. 2012-013-735) and also by the DGIST R&D Program of the MSIP (14-RS-02).

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE Trans. Knowl. Data Eng., vol.17, no.6, pp.734–749, 2005.
[2] R. Burke, "Hybrid web recommender systems," Lect. Notes Comput. Sci., vol.4321, pp.377–408, 2007.
[3] H.T. Kim and C.W. Ahn, "An interactive evolutionary approach to designing novel recommender systems," Int. J. Physical Sciences, vol.7, no.15, pp.2327–2338, 2012.
[4] H. Takagi, "Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation," Proc. IEEE, vol.89, no.9, pp.1275–296, 2001.
[5] M. Tanaka, T. Hiroyasu, M. Miki, Y. Sasaki, M. Yoshimi, and H. Yokouchi, "Automatic generation method to derive for the design variable spaces for interactive genetic algorithms," Proc. IEEE CEC'10, pp.1–8, 2010.
[6] A.K. Jain, "Data clustering: 50 years beyond k-means," Pattern Recognit. Lett., vol.31, no.8, pp.651–666, 2010.
[7] D. Schlierkamp-voosen and H. Mühlenbein, "Predictive models for the breeder genetic algorithm," Evolutionary Computation, vol.1, no.1, pp.25–49, 1993.
[8] X. Amatriain, J. Massaguer, D. Garcia, and I. Mosquera, "The clam annotator: a cross-platform audio descriptors editing tool," Int. Symp. Music Information Retrieval, pp.426–427, 2005.
[9] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. Okuno, "An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model," IEEE Trans. Audio Speech Language Process., vol.16, no.2, pp.435–447, 2008.