Adaptive Music Recommendation Based on User Behavior in Time Slot

Ning-Han Liu, Szu-Wei Lai, Chien-Yi Chen and Shu-Ju Hsieh

Department of Management Information Systems
National Pingtung University of Science & Technology, Taiwan, R.O.C.

Summary
The digital music is booming through rapid expansion of Internet. Users can listen to their favorite songs via web all the time. The electronic commercial leads to development of the recommendation system, which enhances the desire of music buying for customers. The online music recommendation system usually grabs the historical record from past listeners. With extensive data analysis or statistical means, the system recommends the popular music to others. However, users' demand is far beyond that because their choice or change of listening behaviors is often affected by various factors, such as time or place. If the system only considers the type of favorite songs for users, it seems not a comprehensive service for personal recommendation system. So this research will add time scheduling to the music playlist, and combines classification technology of decision tree to suggest users the suit music more precisely. Eventually, the accuracy of recommend results achieved in our anticipate result after implementation and analysis.

Key words:
Music recommend system, decision tree, temporal continuity, content-filtering, collaborative-filtering.

1. Introduction

The flourishing popularity of digital music has brought about the development of customized recommendation technology. When a considerable amount of information is processed, the filter system is often applied to help filter required information. Such an example is Pandora [14], which is a typical music recommendation system based on the music itself. It divides music into different categories based on various music features and recommends music with similar features to interested users. Other music recommendation systems, such as iLike [9], differentiates itself by helping users who share similar interests form music communities based on the users' favorite singers selected on the web. Another system, Last.fm [12], recommends music by calculating the similarity among users based on the music selected by the users, it finds common-interest groups to recommend music items for the user. Hence, recommendation services have recently become a growing trend.

Even though many music recommendation systems have solved most of the customer recommendation issues, there are problems have been ignored, such as whether the interests of users have changed over time; such factors should also be investigated. In order to let the recommendation follow the habits of users, our research proposes to add a time sequence into the playlists to provide a better recommendation, by applying feedback from users and learning their music listening behaviors. Such a system is able to automatically adjust the customized playlist to obtain the most appropriate music information for users. Figure 1 shows the differences of two user's favorites over time.

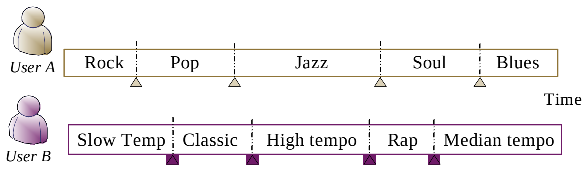

Fig. 1 An example of different users' music listening behaviors at different time intervals (▲: time division point).

Figure 1 demonstrates the results after time division has been applied. It is clear that users have different music preferences at different time sequences. Some users prefer Rock at a certain time while others prefer slow temp. If we only categorize music into types such as Pop, Rock, Jazz, etc., the different preferences over time will be overlooked. On the other hand, when a time division change occurs, the material contained in the same time interval changes as well. These changes might affect the result and accuracy of the categorization model. Therefore, in terms of selecting time division points, the selection must be of significant time points. This research is to provide an appropriate algorithm to address this problem. In order to increase the accuracy of the recommendation, the algorithm generates a decision tree to represent the user's preference after a certain time division.

The remainder of this paper is organized as follows. Section 2 discusses current recommendation systems and

Manuscript received February 5, 2009
Manuscript revised February 20, 2009

related technologies. Section 3 states the problem to be solved in this research and the proposed methods. Experimental results are shown and discussed in Section 4. Finally, the conclusion and suggestions for future work are presented in Section 5.

2. Related Works

2.1 Recommendation Systems

Generally, a recommendation system serves as an information filtering and customization tool. The recommendation system in our context sorts the music playlist based on the user's rating of music items, similarity of item contents or user's browsing behaviors. According to different algorithms, concurrent recommendation systems can be categorized into three categories: content based filtering (CBF) recommendation, collaborative filtering (CF) recommendation, and combined recommendation.

2.1.1 Content Based filtering

Many researchers have already conducted research work on the external features of music to implement effective music management and categorization, such as Meta Data. Other scholars proposed content based methods [1] [21], which generate recommendations by categorizing a user's preferred content features and then extracting similar items according to the user's previously preferred items. Daniel M. [3] used lyric feature analysis to find similar items based on lyrics that describe race conflicts and social issues. Steffen P. [17] proposed to generate a content feature analysis based on customer playlists according to user profile and feedback information. Figure 2 illustrates the base process of content feature analysis.

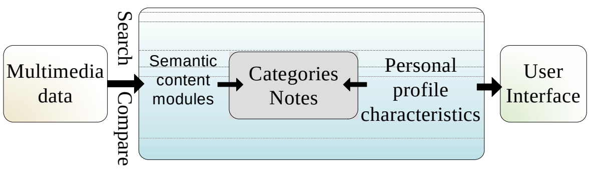

Fig. 2 Basic Content based information filtering.

The advantages of CBF are: 1. It is capable of classifying items and recommending similar music tracks based on content features of the recommended items. 2. It is suitable for users with special preferences. 3. It is capable of recommending new tracks or tracks that have never been listened to before. 4. No requirement of other user's profile information, which solves the cold-start and sparse information problem. The disadvantage is that the content features of the user's preferred music cannot be displayed.

2.1.2 Collaborative filtering

The collaborative filtering (CF) mechanism uses the correlation between users based on their rating or user profile to find users with similar interests. Then users with common-interests recommend items to each other. The CF mechanism is the most widely applied recommendation technology [1] [6] [10], which generally obtain user's preferences from visiting records or browsing behaviors.

The advantages of the CF mechanism are: 1. It is capable of filtering information whose contents are not easily analyzed, such as music or video clips. 2. Avoiding inaccurate content analysis by relying on other user's experiences. 3. Capability of recommending new items, which is the distinguishing feature of CF compared to other filtering methods.

Although the CF mechanism is widely used, many unsolved problems still remain, such as the cold-start problem and sparse information, which results in bad performance during the system initialization stage.

2.1.3 Hybrid Recommendation

Due to the advantages and disadvantages of different recommendation methods, several researchers proposed a combined recommendation method, which combines both CBF and CF approaches to construct a recommendation system. The combinational approach is able to partially solve certain recommendation loss problems. For example, the systems of Clauadui S. [2], Wang H. [21] constructs user collaborative groups based on users' tagging labels. Other users who have tagged the same label are included into the same group. The initial grouping can reduce the calculation time, and further analysis is conducted based on music contents for the purpose of mutual recommendation

In summary, the music recommendation systems mentioned before use CBF or CF technologies. However, with regard to customized music recommendation, these systems rarely take time parameters into account, which should not be neglected when dealing with personal preference.

2.2 Playlist Generator

Generally, playlist generators use traditional methods. Playlists are generated by randomly dragging music to the playlist or by manual selection. However, Rob Van G. [16] employed visualization technology, which automatically locates a position selected by the user on the map according to property tags (such as mood, genre, year, and tempo). Similar music tracks can be easily found at the given position. Not only can new tracks or track never listened to before be found easier, but it also increases the interaction between users and systems. In recent years, Nuria O. [13] proposed the PAPA method. This method sorts custom playlists according to Purpose-Aware and user's physiological responses when the user is doing exercises. For example, the user's pace increases, slow songs are played and vice-versa. For the purpose of assisting a user to adjust cardiac rhythm, music categorization analysis is always involved while generating playlists. Category learning will be further discussed in the following section.

2.3 Category learning

Category learning is a frequently used learning method within the machine learning domain. A Decision tree is a category learning tool that is usually used in data mining. The so-called category is built is built according to classification models based on data collection with known classification attributes. These models are used to predict new incoming data. Among the more well-known classification learning methods, the ID3 category learning algorithm proposed by Quinlan in 1986 performs well in the application of the decision tree. In 1993, Quinlan proposed another category learning algorithm, C4.5, to further improve continuous value processing, in which ID3 is incapable of doing. Both algorithms are discussed in the following sections.

2.3.1 ID3 Category learning

Due to ID3's compact and low calculation costs, it is widely used in many areas. Its core algorithm employs the information gain as an optimal selection criterion.

ID3 is advantageous for its clear algorithm theory and simple mechanism. However, ID3 intends to choose small categories or even a single category when processing attributes with multiple values or continuous values. In addition, ID3 is relatively sensitive to noise. In the case of zero sparseness, each subset contains only a single item. It can easily result in oversized decision trees and causing the over-learning problem. Regarding this issue, Quinlan proposed C4.5 to further improve ID3.

2.3.2 C4.5 Category learning

The C4.5 category learning method originates from ID3. C4.5 is an information-oriented supervised learning method. The Gain Ratio bifurcation guideline is employed by C4.5 to select the optimal attribute, and a binary segmentation approach is used to build the root and internal nodes of the decision tree [15]. In order to avoid noise and incomplete information, the error rate of each sub-tree node and leaf node is estimated using the predicted error rate. The decision tree is pruned according to the predicted error rate. A compact and low error rate decision model is available after pruning.

C4.5 inherits the advantages of ID3 and further improves ID3 in the following aspects:

1. Gain Ratio is used to select the optimal attribute, which overcomes the problem that ID3 faces in selecting an attribute with more values during Information Gain attribute selection.
2. Capable of processing both discrete and continuous attribute values.
3. Capable of processing incomplete data collections.

2.4 Music Features Extraction

In a traditional music database, music categorization is based on artists, styles, years and so on. Normally, the information is obtained manually, which results in low efficiency especially for large music databases. With the automatic classification and embedded search method, an appropriate feature extraction method is vitally important.

Music features are an important component of music. The meaning of a song can be expressed by combining multiple features. In this study, 23 attributes with different dimensions are extracted from the four major components of music [6], including sound quality, pitch, duration, dynamics. In this study, the uploaded MP3 songs are segmented by using Echo Nest Analyze API [19]. Attributes with different dimensions are calculated from each segment.

3 System construction and method

The research presented in this paper applies decision-tree classification learning as its core infrastructure, in which a music recommendation system with Hybrid time scheduling is developed within a web environment.

During the initial recommendation period, the lack of relevant preferences regarding music tastes of new users caused poor recommendation performance. Therefore, our study is meant to generate an initial list by means of Collaborative filtering to solve the Cold-Start problem to an appropriate degree. Moreover, user feedback was later acquired to determine preference of music taste, thus giving the ability to build a user behavior model. In the end, the system, by comparing models, presents music in conformity with user interests at the designation time point through the list, and in experiments, conducts the user modeling and data analysis from the feedback which causes related phenomena.

The core of the recommendation system establishes the users' behaviors model by applying decision tree learning and can be essentially divided into three phases, as shown in Figure 3.

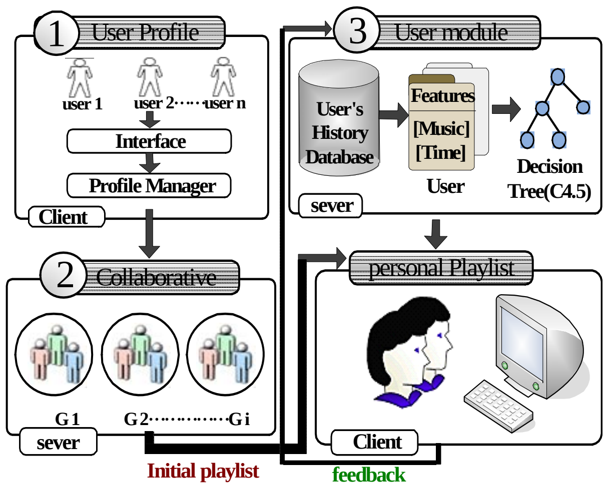

Fig. 3 System framework of the Smart Music Schedule recommendation system.

The first phase: Gathering preference information from users

We designed a recommendation system platform which asks users to fill in personal information while registering, such as age, gender, career and preferred music types, as the basis for grouping users sharing similar interests.

Then, the system can record feedback from users, including marked time, content features and evaluation information (ranked into 10 grades), which are saved together in the user browsing behavior database.

The second phase: Cooperative recommendation module, which generates initial music lists for new users

Grouping similar users: the system is unable to do any music recommendation when users rarely listen to music or there is no feedback from new users. In order to solve this problem, we locate groups who share similar interests by means of collaborative filtering based on the basic user profile information gathered at the first phase. We set a threshold for this grouping method. When ***K*** users are found to be similar, they are considered as sharing similar interests. The formula is shown in Eq. (1):

$$New\,users\ similarity\ is\ defined\ by: U_{ij} = \frac{|C_i \cap C_j|}{|C_i \cap C_j|} \qquad (1)$$

Where C represents music type, $|\boldsymbol{C_i} \cap \boldsymbol{C_j}|$ represents the shared interests of two users and $|\boldsymbol{C_i} \cup \boldsymbol{C_j}|$ represents all categorizes.

Generating initial list: through the results of such grouping, we are able to further arrange music into the initial playlist in a descending ranking order, in which the shared music represents shared interests. By applying this list, individual tastes can be obtained within the short-term, allowing the quick gathering of feedback information. In doing so, the behavioral model can be established more smoothly.

The third phase: Personalized recommendation module

This module is the kernel of this study. We designed an exclusive music intelligent scheduling agent for users. Through the feedback information available in records, by building the personalized behavior model using decision tree classification learning when new music items are added into the back-end database, the system is able to make the right decision by comparing users online immediately through this module.

Compared to other learning algorithms, such as neural network learning which has good performance in regards to accuracy and fault tolerance, it has the difficulty of dealing with categorized variables. Regarding the interpretation of the learning process, it is less easily understood compared to decision-tree classification learning which limits its application. Thus, when solving a complex problem, the C4.5 classification learning algorithm becomes very useful. Due to its self-purification capability, it is able to easily categorize either discrete or continuous numerical. It is easy for people to understand the decision rule generated from a complex problem, thus, the fast adjustment and data build up during the learning process makes the decision tree classification learning method more adaptable in our system.

In the next section, we demonstrate the process of applying the decision tree classification learning on continuous numerical attributes and discuss how we employ C.45 to build a user behavior model, as well as how to arrange a personalized music list.

In the built model, if we divide the time information into a fixed-length sequence, such as a full-day (24 hours), none of the time concepts are included, which leads to the result that the recommended music stays the same, no matter whether it is recommended in the morning or afternoon. If it is divided within smaller units, such as per hour or minute, by using decision tree classification learning, the system will handle such problems in the same manner even within different behaviors of users. This then leads to the incorrectly categorization of users' taste preferences. In such situations, additional load may be forced on the system; therefore, we consider time as continuous attribute, to establish a decision tree together with other fields.

3.1 Decision Tree Generation

When dealing with objects marked as 'time mark' and 'music feature', which have high dimensional values, it is required to consider the vast amount of information and the problems of sparse information or noise, when performing categorization. If we only apply the ordinary ID3 decision tree on a segmentation point search, there might be only a single numerical value after information is segmented. However, the calculation of information Gain prefers to be the larger value, which leads to the excessive adaptation within the categorized rule. Thus, Quinlan [15] thereafter proposed C4.5 Category learning, in order to solve the disadvantages associated with ID3.

This system uses C4.5 category learning as the basis to build up the preference model, of which, the algorithm for dealing with continuous numerical attributes is shown in Figure 4. We will discuss how to establish the user model in each step by employing this algorithm.

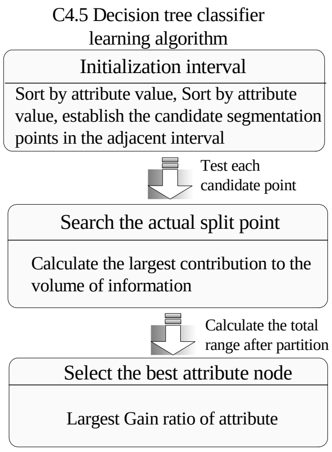

Fig. 4 The process procedure of C4.5 for continuous value processing.

Step 1: Interval initialization

During initialization, according to the order of **F** from small to large, e.g. **F** = {$\boldsymbol{A_1}$, $\boldsymbol{A_2} \ldots \boldsymbol{A_n}$}, we can establish a candidate cut point, ***CP***. This point is obtained through the calculation between adjacent numbers, defined as Eq. (2):

$$CP = (A_i, A_i + 1)/2 \quad (2)$$

Where A_i stands for the continuous numerical values within the interval, $\boldsymbol{A_i}$ +1 is the adjacent number of $\boldsymbol{A_i}$, ***CP*** is the value of the cut point.

Step 2: Search the real cut point position

The candidate cut point, obtained through the search, is used to calculate the maximum information Gain in order to determine the position of the real cut point. It is calculated as the 'entropy before cut' minus the 'entropy after cut', this is shown in Eq. (3):

$$GainRatio(F) = Entropy_{before}(S) - Entropy_{after}(S) \quad (3)$$

If it is required to calculate the information gain of **F**, the entropy after cut is required to be calculated, such as that shown in Eq. (4), when n results are included in ***S*** incidents, $\boldsymbol{C_p}$ is the probability of each result corresponding to a type. The entropy in ***S*** is:

$$Entropy_{after}(S) = \sum_{k=1}^{n} -C_p log_2 C_p \quad (4)$$

which is used to calculate the comparison between the cut point and information gain, and find out a point ***K*** as the position of the real cut point. Then with the real cut point we are able to divide the numerical attribute information into two sections, '<= ***K***' and '> ***K***', by means of binary segmentation.

Step 3: Select the best features of the attribute nodes

The results after segmentation are handed over to the Gain Ratio function to calculate the importance of this attribute. If an attribute **F** is included in ***S***, the value of the gain ratio is required to be calculated. This is calculated as the 'entropy before cut' minus 'entropy after cut', and then divided by "entropy of after segmentation", as shown in Eq. (5)

$$GainRatio(F) = \frac{Entropy_{before}(S) - Entropy_{after}(S)}{Entropy_{after}(S)} \quad (5)$$

Different attributes with different dimensions will be applied to calculate and compare with the gain ratio; the larger value will be chosen to be considered as the root node. Recursively repeating Step 1 ~ Step 3 of the calculus from the subset such an attribute, we are able to determine the next attribute node and branch, until either every subset belongs to the same category, or there is no information to be used for classification.

3.2 Building the user behavior model

Practice is to give reliability, applying the upper limit value of the binomial probability distribution as the value of the estimated error. ***N*** represents the number of sub-trees within this training material, while K is the number of incorrectly categorized information in N, the number of forecast errors is ***N***×Ucf(***K***,***N***). When sub-tree nodes are substituted by leaf nodes, and the forecast error rate is low, the sub-tree node will be pruned into the leaf node and replaced by the sub-tree within the major category after pruning, otherwise, the original sub-tree is kept [15].

After the decision tree is built, besides the expectation of high accuracy, the ability of generating simple and easy to interpret rules is also necessary. Therefore, after pruning the decision tree, the program rules will become easy to understand, which can be located between the root node and leaf nodes. Such as If (Time) < = 6 ∶ 00 And (Avg_Pitch) < = 302 And (Time) < = 3 ∶ 25 Than Class = 6 Else Class = 4, eventually, the transformed program can reduce the complexity of classification.

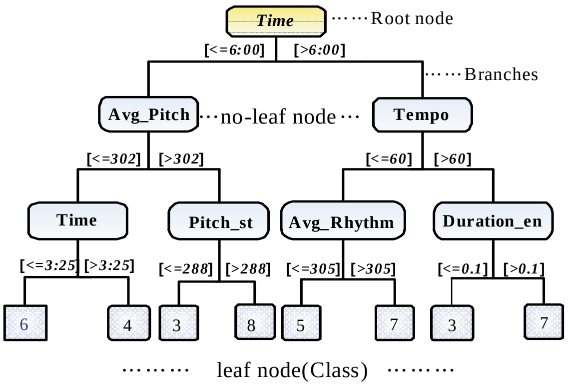

Fig. 5: A Single user behavior model.

3.3 Produce a list of the results of scheduling music

After the user preference model is established, the system is able to locate music which is suitable for the user at the current time by filtering the music information database. Then, a song is randomly selected from the database. The reason we apply randomness is to avoid the possibility of selecting the same music at same time period. At the end of the first song, the music information database will be filtered again to find out the follow-up order of the next song. In this way we can generate a list of the results of the scheduled music.

4 Experimental Results

4.1 System Environment Establishment and Estimation

Our player platform was developed using the C# programming language under a web environment. The back-end database was developed using Microsoft Office Access 2003. The man-machine interface of the player is Macromedia Flash 8.0. Users can use our customized prototype service via their internet browser. Finally, users' feedback information was analyzed to find recommendation.

The source of songs comes from MIDI songs collected by background music makers on the internet. The collected songs are converted by programs into the comparable format and saved in the music database. Currently, there are 115 songs in the database.

4.2 System Interface

For our system we used Flash to build the platform of the MuPA recommendation system. Information about the current status is obtained from the front end and its parameters are transmitted to the kernel at the back-end for processing. The results are transmitted back to the user interface at the Client end.

Fig. 7 User registration interface.

Figure 7 shows a screen shot of the user registration interface. Users fill in some basic information through this interface and then leave the feedback according to the related recommended music they have listened to, which is depicted in Figure 8. Finally, the personalized recommendation list is provided in the Menu after the user's preferences model is generated.

Fig. 8 MuPA recommendation list and evaluation screen.

4.3 Experimental Results and Analysis

From observing the database, we obtained the following statistical results. 56 people participated in this study. The age distribution is between 23-year-olds and 35-year-olds. These people use internet frequently.

Generally, in order to test the recall of the system, users need to go through all samples in the music database. However, due to individual differences of users, we are unable to require all users to do so. Therefore, recall is beyond the scope of this paper. In order to verify the "time scheduling" based system structure proposed in this study, the results are obtained with solid evidence. The evaluation system employs two evaluation indicators, Mean Absolute Error (MAE) and precision, which are widely used in other system evaluations, such as Herlocker [6], Sarwar [18], Vucetic [20], and Yu [23]. We will firstly introduce our experiments according to the indicators mentioned above, and then discuss the experimental results.

Experiment 1: Model system error rate estimation. This is a pre-test experiment.

Generally, a model will be built to estimate the error rate of the system. MAE is used as an evaluation indicator to calculate the error rate.

The error rate is a proportion of the error classification made on the original training data by the model after being analyzed by different categorization methods and model construction. In order to evaluate the average error between the predicted ratings and real ratings, we use the MAE indicator to verify the system. This definition is given by the following formula:

$$MAE = \frac{\sum_{u=1}^{N} |P_{u,i} - R_{u,i}|}{N} \quad (6)$$

Where $\boldsymbol{N}$ is the number of samples in the database, $\boldsymbol{P_{u,i}}$ is the predicted rating from user $\boldsymbol{u}$ on item **i** and $\boldsymbol{R_{u,i}}$ is the real rating from user $\boldsymbol{u}$ on item **i**.

First, users were required to listen to more than 15 songs to establish user profiles. 3/5 songs were selected as training data and the rest of the songs were used as test data for a system pre-test. The MAE was calculated every two days during the one-month experiment period. Five user's feedback samples were selected randomly. The error evaluation was conducted on the feedback samples three times. The experimental results are illustrated in Figure 9.

According to the MAE values given in the diagram, MAE decreases during the early stages, which indicates that the initial error is large. As time goes by, user feedback information keeps increasing and the MAE values decreases significantly. Finally, the error tends to be stable and constant. The experimental system reached a stable state in 19 days after experiments began. The amount of time needed to obtain precise recommendation calculations and verification is a rough estimate in this experiment.

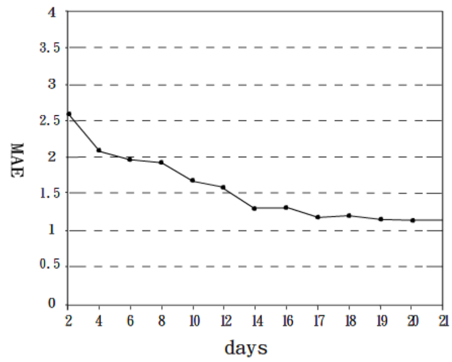

Fig. 9 MAE values of time scheduling based recommendation.

Experiment 2: Recommendation precision comparison between C-baseMR recommendation system proposed by Yoshinori Hijikata [22], iLike recommendation system [9], Pandora recommendation system [14], random recommendation and our approach.

Recommendation precision is defined as the rate that the system provides precise recommendations after the model is generated and analyzed by different categorization algorithms. The definitive formula of recommendation is given by Eq. (7):

$$Precision = \frac{Number\ of\ accept\ \ items}{Number\ of\ recommend\ \ items} * 100\% \quad (7)$$

It is the ratio between the number of accepted items and the number of recommended items.

The experiment procedure is discussed as follows. The experiment duration was 35 days. Firstly, music items are automatically allocated to current users. The evaluation focused on the last 15 items of user feedback. In order to obtain objective results, the average precision was calculated three times. The maximum rating is set to 10 marks. The feedbacks with one to five marks were labeled as "0", which stands for "dislike", while feedbacks with six to ten marks were labeled as "1", which stands for "like". Therefore the higher the precision value is the more accepted the recommended item is. The experimental results are illustrated in Figure 10.

According to the bar graph, both C-baseMR and our approach achieved over 70% recommendation precision. In addition, our approach obtains the best performance with precision at 78.33%, which relatively outperforms Pandora and iLike. Random recommendation achieves below 50% accuracy, which is worse than all the other methods. Based on the experimental results, our time scheduling based system performs well in terms of recommendation precision.

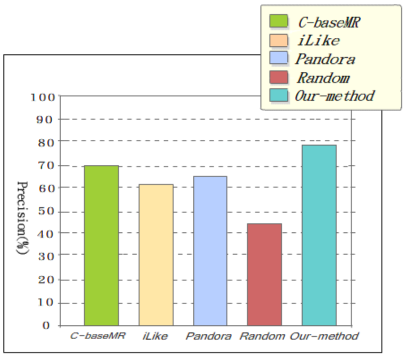

Fig. 10 Accuracy of the Recommender System Comparison.

Experiment 3: Recommendation precision comparison between users with and without regular rest.

The purpose of this experiment is to analyze if our time scheduling based recommendation system can appropriately recommend music according to a user's situation by analyzing time and place differences while listening to music.

To verify our approach, we sampled undergraduate and postgraduate students from different grades. These sampled students were grouped into two groups according to their different work and rest patterns. Twenty students were randomly selected and their feedback was used to calculate the recommendation precision. The calculation formula is similar to the one used in experiment 2. The experimental results are given in Figure 11.

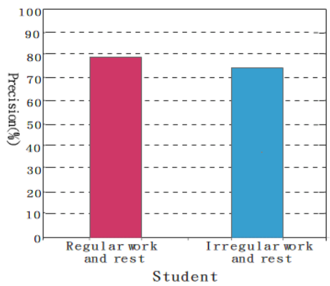

Fig. 11 A comparison of recommendation precision between users with and without regular rest.

According to the bar graph, the recommendation precision is higher than 70% in both the regular work-and-rest pattern group and the irregular work-and-rest pattern group. Both groups are satisfactory. The experimental results indicate that our time parameter based model is insensitive to a user's work-and-rest patterns. It also means that in most cases, a user's personal preference is the dominant factor, whereas time parameters are rarely used in model construction.

5 Conclusions and Suggestions

This research is based on the methodology of recommendation systems and related theory, with the addition of a creative time-ranging recommendation, which is unique in comparison to other music recommendation systems. The system is capable of providing suitable music recommendations according to different individual preferences whereby the overall precision currently reaches 78.33%. Although the entire collection of music samples only reaches a small amount of 115, which is not considered an abundant database, the current achievements indicate that the future development of this research would be more comprehensive with larger music samples. Generally speaking, the main contribution of the research includes:

1. Solving the Cold–start dilemma: It is a common problem that occurs among other music recommendation systems, especially for new users. Normally the system cannot provide relevant recommendations with insufficient estimation data and evaluation. Our system can analyze the different preferences of users when they first register and share that information with other users who have similar preferences. With this approach, the disadvantage of ineffective recommendation can be overcome.

2. Real-time recommendation: Traditional recommendation speculates the users' preferences according to their browsing behaviors. However, it can only search music with similar styles. This research is based on advanced recommendation techniques, which provide more suitable and accurate music information through the unique self-learning process, thus making the whole recommendation system more dynamic.

According to the results from the analysis, the proposed customized music scheduling agent has achieved it intended preliminary effect. The future direction of this research, should endeavor to make further improvements, these include:

1. System performance: The current music recommendation library has been sampled and established manually. The system will behave more effectively if an auto sampling module is involved.

2. Concept extension: music therapy is widely carried out

in foreign countries. Such therapy can effectively reduce the pain of a patient, relieve pressure and stress, and help them recover quicker based on the original prescription. Our future plan will focus on using this kind of music recommendation system to hospitals and designing a patient targeted music selection system.

Acknowledgements

This work was partially supported by the NSC under Contract NSC97-2218-E-020-002.

References

[1] M. Balabanovic, and Y. Shoham, , "Learning Information Retrieval Agents: Experiments with Automated Web Browsing.", in *AAAI Spring Symposium on Information Gathering from Heterogeneous*, Distributed Environments, Stanford, 1995.

[2] S. Claudiu, W. N. Firan and P. Raluca, "The Benefit of Using Tag-Based Profiles," in *Proceedings of the 2007 Latin American Web Conference (LA-WEB)* 32-41, 2006.

[3] M. Daniel and J. C. Sally, "Sociology and Music Recommendation Systems," in Proceedings of *International Conference on Music Information Retrieval (ISMIR)*, 2007.

[4] Y. Fu, "Data Mining: Tasks, Techniques, and Applications," in *IEEE Potentials*, 18-20, 1997.

[5] D. Goldberg, D. Nichols, and D. Terry, "Using Collaborative Filtering to Weave an Information Tapestry," in *Communications of the ACM*, 1992.

[6] J. L. Herlocker, and J. A. Konstan, "Content-Independent Task Focused Recommendation," in *IEEE Internet Computing*, 40-47, 2001.

[7] E. R. Hammer and M. S. Cole, "Guided Listening: a Textbook for Music Appreciation by Hammer and Cole," *WCB*, 1992.

[8] J. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An Algorithmic Framework for Performing Collaborative Filtering," in *Proceedings of SIGIR 99*, 230-237, 1999.

[9] iLike.com, http://www.ilike.com, Retrieved date: 2007/11/5. (2007)

[10] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl, "GroupLens: Applying Collaborative Filtering to Use Net News," in *Communications of the ACM*, 1997.

[11] K. Katsuhiko, H. Keiji and N. Katashi, "A Music Recommendation System Based on Annotations about Listeners," in *Proceedings of Automated Production of Cross Media Content for Multi-Channel Distribution*, 2005.

[12] Last.fm, http:///www.last.fm, Retrieved date: 2007/11/6. (2007)

[13] N. Oliver and L. Kreger-Stickles, "PAPA: Physiology and Purpose-Aware Automatic Playlist Generation," in Proceedings of *International Conference on Music Information Retrieval (ISMIR)*, 2006.

[14] Pandora.com, http://www.pandora.com. Retrieved date: 2007/10/20. (2007)

[15] J. R. Quinlan, "C4.5: Programs for Machine Learning," *Morgan-Kaufmann*, San Francisco, 1993.

[16] G. Rob van and V. Fabio "Visual Playlist Generation on the Artist Map," in *Proceedings of ISMIR*, 2005.

[17] S. Pauws, and S. van de Wijdeven, "User Evaluation of a New Interactive Playlist Generation Concept," in Proceedings of *International Conference on Music Information Retrieval (ISMIR)*, 2005.

[18] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Application of Dimensionality Reduction in Recommender System-A Case Study," in *Proceedings of ACM Web KDD Web Mining for E-Commerce Workshop*, 2000.

[19] The Echo Nest Analyze, API, http://developper.echonest.com/ docs/analyze/xml, xml description, downloaded March 31, 2008.

[20] S. Vucetic, and Z. Obradovic, "A Regression-base Approach for Scaling-up Personalized Recommender Systems in E-commerce," in *Proceedings of ACM Web KDD Web Mining for E-Commerce Workshop*, 2000.

[21] H. Wang, W. Wang, J. Yang and P. S. Yu, "Clustering by Pattern Similarity in Large Data Sets," in *Proceeding of ACM Special Interest Group on Management of Data*, 2002.

[22] Y. Hijikata, K. Iwahama, S. Nishida, "Content-based Music Filtering System with Editable User Profile," in Proceedings of *ACM symposium on Applied computing*, 1050-1057, 2006.

[23] K. Yu, X. Xu, M. Ester and H.P. Kriegel "Selecting Relevant Instances for Efficient Accurate Collaborative Filtering," in *Proceedings of the 10th International conference on Information and knowledge management*, 239-246, 2001.