International Journal of Advanced Technology in Engineering and Science
Vol. No.4, Issue No. 06, June 2016
www.ijates.com
ijates
ISSN 2348 - 7550

MUSIC PLAYER FOR JOGGERS USING ANN AND HMM

Mayuri Mohan Tharval

Information Technology Department, Terna Engineering College, Mumbai, (India)

ABSTRACT

In recent years the physical inactivity in human beings results in numerous chronic diseases. To reduce the risk of illness many people maintain a habit of exercising daily. Jogging is a cheap and common form of exercise which helps to improve the health. It also helps to reduce the risk of illness. Joggers often carry mobile devices, such as iPods or smart phones, to listen to music in order to control their mood or enhance their thoughts. But a jogger usually spend some time preloading and arranging the music that will be played throughout the exercise session. A novel service introduced in this paper is that it provides the jogger with a smart music player agent to control the music on mobile device without any effort. This system involves pace prediction and music filtration technologies. To filter out undesired music an artificial neural networks(ANN) is used as the kernel of the music filter. Hidden Markov Model(HMM) is used for pace prediction and searching techniques of similar pattern are used to select music with a suitable tempo. This unique service is attractive to the joggers.

Keywords: Intelligent music player, music database, Artificial Neural Networks(ANN), Hidden Markov Models(HMM).

I INTRODUCTION

A jogger usually shuffles the playlist so that the activity does not become tiresome by listening to the same music over and over again. With the help of a traditional music player, the jogger need to preload desired music files onto the device before starting the activity of jogging. The quantity of music that a person can listen and store has been increased dramatically. Therefore, the jogger may spend a lot of time searching for and loading preferential and suitable pieces of music.According to observations and studies [1], [2], music with a fast tempo is suitable for a fast pace and, similarly, slow music is appropriate for a slow pace.

The adoption of a G-sensor in most modern mobile phones makes it possible to detect the jogger's speed. Music with a fast tempo is suitable for a fast pace and, similarly, slow music is appropriate for a slow pace. By using this easily accessible information, we are not only able to plan the playlist so as to improve the user's enjoyment without buying an additional sensor device .A music recommendation system serves to retrieve music that the user may be interested in. The music playlist generator in the system sorts out the music playlist according to the current situation and user's preferences. A playlist generator predicts new music that may be appreciated by the user and also arranges the user's music collection. Artificial neural networks (ANN) are adopted to filter out the user's undesired or inappropriate music.

A Hidden Markov Model (HMM) and pattern searching techniques help to predict a jogger's pace which forms the basis for the selected music.This system helps us to promote the user's motivation to jog. In addition, most of the experimental subjects agreed that the automatic music player system would be useful for the running exercise.

In this system, artificial neural networks (ANN) [11] are adopted to filter out the user's undesired or inappropriate music. We use a hidden Markov model (HMM) [12] and similar pattern searching techniques to predict a jogger's future pace which forms the basis of the selected music. According to experimental results, this system can promote the user's motivation to jog. In addition, most of the experimental subjects agreed that the automatic music player system would be useful for their running exercise.

II. MUSICAL INFLUENCE AND PERSONAL INTERESTS

In this section the design of the music filtering modules and the prediction of a jogger's pace using adopted techniques are presented respectively.Music features are extracted from audio waves and symbolic form data. Each user has personal filtering modules that select the user's desired and suitable music. The details are discussed below.

2.1 Features Extraction

The system extracts values for musical features from symbolic form [13] and raw audio data. For the purpose of simplifying the system design and serving the preferences of the majority of people, the database in this system only contains songs under the mantle of pop music, including hip hop, rock & roll, blues, jazz and so on, as opposed to classical music, which has fewer listeners.

There are several symbolic forms of digital music.The MIDI features have been proposed in [13] that include mean and standard deviation of the pitch values, pitch density, pitch entropy, tempo degree, and loudness.

Mel-scale frequency cepstral coefficients (MFCC) is the well known feature in the area of speaker and speech recognition, which defines the human perception sensitivity with respect to frequencies.The input vocal signal is segmented into frames of 20 milliseconds, with an optional overlap of half of the frame size. The selected twelve parameters of MFCC as features in order to avoid high dimensionality in the data. Moreover, to reduce the complexity of this data, the calculated averages and standard deviations of each dimension, i.e. a piece of music is represented by a twenty four dimensional vector. In order to simplify the system design and provide a higher accuracy of vocal features, manually identified segments with the significant singer's voice is used.

2.2 Music Filtering Modules Mechanism

The input data for the music filtering modules includes the attributes of the music features. After the modules evaluate these inputs through an intelligent mechanism, they output the predicted value of the user's preference of the musical piece. ANNs were adopted as the kernels of the filtering modules due to their ability to classify continuous values, such as the value of features input. In order to enable the neural networks to learn the user's preference in music, the system collects the scores and date with the associated music file name, known as ID, after the user has listened to the music through the web based music player in the user client. The user's preference is scored within a range of one to ten, where a higher score implies that the user feels the music being played is suitable for listening.

The ANN training time can be higher than normal in the following two cases. Firstly, when the training set data for an ANN has a higher dimensions or large amount of training samples, so more time is required to complete the training. Secondly, if the prediction does not satisfy the user's need, the user's new feedbacks are recorded to retrain the ANN. Retraining the system requires more time if all of the user's data is used as training samples. Below is the structure of the mixed artificial neural network which helps to reduce the training time required for the artificial neural network. This helps in the generation of music playlist.

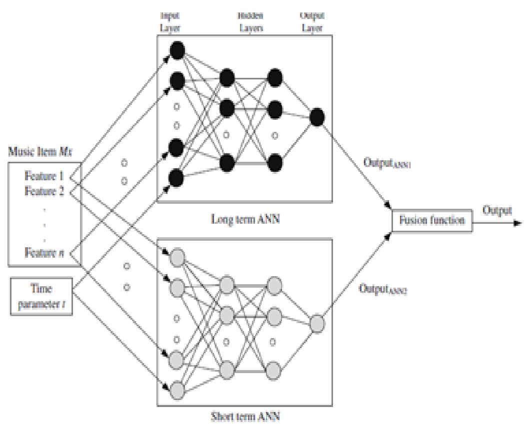

Fig. 1 The Structure of Mixed Artificial Neural Network

To reduce the training time, we propose an innovative structure for Music Filtering Module I as shown in Fig. 1. This structure consist of two ANNs: one is called as the short term ANN, which indicates the user's recent behavior to the music selection, and the other is called as the long term ANN, which is used for record the user's preference in music for long term. By using the two ANNs, two output values are obtained, namely OutputANN1 and OutputANN2.

In the surviving pieces of music, there may be songs that are not suitable for listening to when the user is running. Therefore, a second filtering module based on ANN for each user is constructed . Keeping in mind that the running jogger is not able to easily rate the score precisely, the user has to only touch the screen of the mobile device to denote that the music is negative or inappropriate during a jogging activity. Songs that are played without the user touching the screen are treated as positive. This divides the music into two classes, suitable and unsuitable. The music that has been played during the user's jogging activity becomes training data. Moreover, since the amount of music that can be listened to during the jogging activity is quite small so single feed-forward ANN can be used.

The previous Music Filtering Module I is trained with music that the user has rated ordinarily and is used to filter out music that the user dislikes. However, in the surviving pieces of music, there may be songs that are not suitable

ijates
ISSN 2348 - 7550

for listening to when the user is running. Therefore, we constructed a second filtering module based on ANN for each user. Keeping in mind that the running jogger is not able to easily rate the score precisely, the user has to only touch the screen of the mobile device to denote that the music is negative or inappropriate during a jogging activity. Songs that are played without the user touching the screen are treated as positive. This divides the music into two classes, suitable and unsuitable. The music that has been played during the user's jogging activity becomes training data. Moreover, since the amount of music that can be listened to during the jogging activity is quite small, we only use a single feed-forward ANN.

III. ADAPTIVE PLAYLIST GENERATION AGENT

In a previous system [10], the recommended music was used to point out to the jogger whether they needed to accelerate, decelerate, or maintain their current pace. However, the joggers may be unable to reach the pace suggested by the system, and so a wide gap is formed between the tempo of the music and the jogger's actual pace, reducing the user's satisfaction with the system. The system detailed in [9] retrieves the music by simply using the jogger's present pace, thus decreasing the smoothness of the music playing. This is due to the music length possibly not being consistent with the user's ability, and so the music may be changed even if a song is not completed.

The Markov process is usually 'memoryless' i.e the process should satisfy the Markov property to make predictions for the future of the process is entirely based on its present state. The hidden markov model is basically a statistical Markov model in which the system modelled is assumed to be the Markov process with hidden states. In Markov model the states are directly visible to the observer, but in hidden markov model the state is not directly visible the output of the state is visible.The states of hidden markov model has a probability distribution for the generated output tokens. The sequence by which the tokens are generated by the hidden markov model gives the information of the sequence of states.The parameter of the model is not hidden but the state sequence is hidden through which the model passes. So even if the model parameters are known the model is still hidden. A hidden Markov model observes a sequence of emissions, but do not know the sequence the model went through to generate the emissions of various states. Analyses of hidden Markov models help to recover the sequence of states from the observed data.The HMM is used to predict the jogger's future pace which forms the basis of the selected music.

HMM can be used in many fields where goal is to recover a data sequence that is not immediately observable. Applications include:

•Speech recognition

•Machine translation

•Activity recognition

•Cryptanalysis

•Alignment of biosequences.

Example:

Consider two friends, Ram and Sham, who live far apart from each other and who talk together daily over the telephone about what they did that day. Sham is only interested in three activities: walking in the park, shopping, and cleaning his apartment. The choice of what to do is determined exclusively by the weather on a given day. Ram has no definite information about the weather where Sham lives, but he knows general trends. Based on what Sham tells him he did each day, Ram tries to guess what the weather must have been like.

Ram believes that the weather operates as a discrete Markov Chain. There are two states, "Rainy" and "Sunny", but he cannot observe them directly, that is, they are hidden from him. On each day, there is a certain chance that Sham will perform one of the following activities, depending on the weather: "walk", "shop", or "clean". Since Sham tells Ram about his activities, those are the observations. The entire system is that of a hidden Markov model (HMM). Ram knows the general weather trends in the area, and what Sham likes to do on average.

```
states = ('Rainy', 'Sunny')
observations = ('walk', 'shop', 'clean')
 start_probability = {'Rainy': 0.6, 'Sunny': 0.4}
transition_probability =
  {
   'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
   'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},
   }
 emission_probability =
  {
   'Rainy' : {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
   'Sunny' : {'walk': 0.6, 'shop': 0.3, 'clean': 0.1},
   }
```

In this piece of code, start_probability represents Ram's belief about which state the HMM is in when Sham first calls him (all he knows is that it tends to be rainy on average). The particular probability distribution used here is not the equilibrium one, which is (given the transition probabilities) approximately {'Rainy': 0.57, 'Sunny': 0.43}.

The transition_probability represents the change of the weather in the underlying Markov chain. In this example, there is only a 30% chance that tomorrow will be sunny if today is rainy.

The emission_probability represents how likely Sham is to perform a certain activity on each day. If it is rainy, there is a 50% chance that he is cleaning his apartment; if it is sunny, there is a 60% chance that he is outside for a walk. Thus, HMM gives us the probability of a particular scenario.The Fig.2 helps us to understand the example of HMM in a diagrammatic manner.

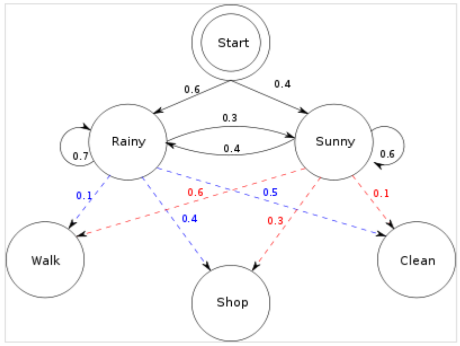

Fig.2 Example of HMM

IV. JOGGERS HMM MODEL

The mobile device used can only provide the jogger's pace. However, the runner has three different states, warm-up or initial, normal, and tired even at the same running speed. These states cannot be adequately identified by the jogger's pace. Therefore, HMMs used to depict the user's jogging characteristics or properties. A HMM with three states was adopted as shown in Fig. 3

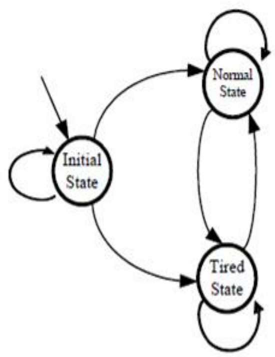

Fig. 3 State Model

There are three states in this model. The initial state identifies the warm up state of a jogger and the normal state means that the jogger has a general vigor. The tired state implies that the jogger feels tired. The set of states is denoted as State={State0, State1,State2}.

V. CONCLUSION

The impact of music in exercise performance is very positive in relation to mood and improvement of a user's confidence and motivation. The playlist generator is based on artificial neural networks(ANN). In this generator, the generation of the playlist is not limited to the user's preference in music but also to the changes in user preferences over time. ANN model is basically used so that the songs can be filtered out properly because there may be songs that are not suitable for listening to when the user is running. The music that has been played during the user's jogging activity becomes training data. Moreover, since the amount of music that can be listened to during the jogging activity is quite small, we only use a single feed-forward ANN.

In order to satisfy the needs of different users, multiple features extracted from both the symbolic form and the wave data for classification . For the purpose of simplifying the system design and serving the preferences of the majority of people, the database in the system only contains songs under the mantle of pop music, including hip hop, rock & roll, blues, jazz and so on, as opposed to classical music, which has fewer listeners.

The runner has three different states, warm-up or initial, normal, and tired even at the same running speed. These states cannot be adequately identified by the jogger's pace. Therefore, HMMs are used to depict the user's jogging characteristics or properties. It is time-consuming to train the artificial neural networks and a real time system not possible to built easily when the user needs to retrain the ANN.

REFERENCES

[1] N. Oliver, and F. Flores-Mangas, "MPTrain: A mobile, music and physiology-based personal trainer," in Proc. of the 8th Conference on Human-computer Interaction with Mobile Devices and Services, 2006.

[2] S. E. Schwartz, B. Fernhall, and S. A. Plowman, "Effects of music on exercise performance," Journal of Cardiopulmonary Rehabil, vol. 10, pp.312-316, 1990.

[3] K. S. Park, W. J. Yoon, K. K. Lee, S. H. Oh, and K. M. Kim, "MRTB framework: A robust content-based music retrieval and browsing, " IEEE Trans. On Consumer Electronics, vol. 51, no. 1, pp. 117-122, Feb. 2005.

[4] X. Zhu, Y. Y. Shi, H. G. Kim, and K. W, Eom, "An integrated music recommendation system," IEEE Trans. On Consumer Electronics, vol. 52, no. 3, pp. 917-924, Aug. 2006.

[5] J. C. Platt, C. J. C. Burges, S. Swenson, C. Weare, and A. Zheng, "Learning a Gaussian process prior for automatically generating music playlists," in Proc. of Advances in Neural Information Processing Systems, 2002.

[6] A. Flexer, D. Schnitzer, M. Gasser, and G. Widmer, "Playlist generation using start and end songs," in Proc. of the 9th International Conference on Music Information Retrieval, 2008.

[7] K. Kaji, K. Hirata, and K. Nagao, "A music recommendation system based on annotations about listeners' preferences and situations," in Proc. of First International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS'05), pp.231-234, 2005.

[8] A. Andric, and G. Haus, "Automatic playlist generation based on tracking user's listening habits," Multimedia Tools and Applications, vol. 29, number 2, 2006.

[9] J. T. Biehl, P. D. Adamczyk, and B. P. Bailey, "DJogger: A mobile dynamic music device," in Proc. of CHI '06 Extended Abstracts on Human Factors in Computing Systems, pp. 556-561, Montreal, Quebec, Canada, ACM Press, 2006.

[10] R. de Oliveira, and N. Oliver, "TripleBeat: Enhancing exercise performance with persuasion," in Proc. of the 10th Conference on Human-computer Interaction with Mobile Devices and Services, 2008.

[11] I. P. Guyon, "Applications of neural networks to character recognition," International Journal of Pattern Recognition and Artificial Intelligence, 5, pp.353-382, 1991.

[12] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in Proc. of the IEEE, vol. 77, no.2, Feb. 1989.

[13] N.H. Liu, S. J. Hsieh, and C. F. Tsai, "An intelligent music playlist generator based on the time parameter with artificial neural networks," Expert Systems with Applications, doi: 10.1016/j.eswa.2009.09.009, 2009.

[14] J. Han, and M. Kamber, Data Mining Concepts and Techniques, Morgan Kaufmann, 2001, pp.418-419.