An Algorithm for Automatic Playlist Generation in Radio Stations

I.M.K. Kahanda* and U.G. Kanewela

Department of Computer Engineering, Faculty of Engineering, University of Peradeniya

Introduction

In a typical radio station, DJs play songs using pre-generated playlists. A playlist is a set of music titles which should be played in a given order. Composing high quality playlists for radio stations is very demanding because of the large number of songs available for selection. Therefore the person responsible for composing playlists have to either memorize or listen to the excerpts of the songs in order to choose the right song for the right situation.

All the radio stations in Sri Lanka build their playlists manually. In today's competitive world, even though maintaining a high standard largely defines the popularity of the station, the above situation makes it difficult for a human to achieve this consistently. This is the motivation for this research which examined the feasibility of using a computer to generate music playlists for radio stations using a newly developed Automatic Playlist Generation (APG) algorithm.

Related work

APG, which is formally defined by Pauws *et al.*, (2006) has enjoyed lots of research recently and the researchers have come up with many different approaches depending on different applications, such as similarity based techniques for Personal music players to Constraint Satisfaction Programming (CSP) approach for Electronic Music Distribution (EMD) systems. Similarity based techniques are specially used for recommender systems, in which the listener selects one or more of her favorite songs to be used as seed songs and the generator will pick the rest of the songs one by one for the unordered playlist. Collaborative filtering (French and Hauver, 2001; Pestoni *et al.*, 2001), timbre similarity measures (Logan and Salomon, 2001; Aucouturier and Pachet, 2002) and machine learning using existing playlists (Platt *et al.*, 2002) are the popular similarity based techniques. The major advantage of using these techniques is that their sensitivity in capturing the changes in the choices of a single listener.

In CSP approach, instead of selecting individual music titles, one by one, like above, the idea is to produce a collection of music titles, in an order, restricted to a set of global constraints (Pachet *et al.*, 1999). One of the better CSP approaches to APG, with regard to scalability is proposed by Aucouturier and Prachet (Pachet, 2002), which starts by having a random set of songs from the whole set, and iteratively tries to improve the playlist by making small changes to it, using adaptive search (an adaptation of the local search) algorithm (Codognet and Diaz, 2001).

Both these approaches so far have looked at APG for playlist generation mainly for a set of very few listeners or a large set of listeners which can provide immediate feedback. This paper, for the first time, presents an APG technique for radio stations (a large global set of listeners from which continuous immediate feedback cannot be obtained), *PlayGen*, which is a hybrid of the aforementioned techniques and enjoys the benefits of both the schemes.

Methodology

The APG algorithm presented in this paper, *PlayGen*, uses the algorithm presented in (Prachet, 2002) as a starting point due to its scalability. The main idea of that the algorithm presented is that the qualities/characteristics which a playlist should posses are described as constraints.

Problem statement

The playlist generation problem is defined as follows in (Prachet, 2002). A playlist S is a sequence of variables $v_1, v_2..., v_n$ whose values x_i can be taken from a catalogue of music titles D. The problem is to assign values to variables, so that the resulting sequence satisfies constraints $c_1, c_2,...,c_m$.

Constraints

Constraints are seen as simple cost functions. The cost represents how 'bad' the constraint is satisfied, for the given assignment of variables. Constraints may hold on attributes of music titles and these are typically metadata of songs. Therefore, a playlist will have a total cost associated with it. The algorithm starts with a certain initial playlist (having a certain total cost) and iteratively refines the solution until a playlist satisfying all the given constraints (having a total cost less than a given threshold) is found. Authors have developed the following three compulsory constraints for radio stations along with several other optional ones.

- C-ALL-DIFFERENT: Two songs of the same album/artist should not be included.
- C-DISTRIBUTION: Genre (e.g. pop), type of artist (e.g. female), release year and tempos (in beats per minute) of two consecutive songs should be different.
- C-VARIETY: Number of different genres of songs should exceed a certain threshold.

For example, the cost associated with C-VARIETY is quantified as follows.

Cost(C-VARIETY):

Return the length of the playlist – number of genres of music titles divided by length of the playlist

PlayGen algorithm

PlayGen possesses two major features which are not found in the APG algorithm presented in (Pachet, 2002). First, as shown in Figure 1, the *PlayGen* algorithm searches only through a small subset of songs termed request neighbor set (RNS), as opposed to the whole set D. RNS is the set of similar songs to the recent listener song requests, which is generated by the nearest neighbour algorithm utilizing the metadata and calculated attributes of music titles, and specially helps when the collection is very large like in a typical radio station, since it narrows the search space effectively. Formal definition of RNS is given below.

RNS = N songs with min(*TotalDistance*(x_i,x_r))
Where, x_r=requested song and x_i=all other songs.

TotalDistance(x_1,x_2)=$\sum$*distance*($a1_j$,$a2_j$)
Where $a1_j$ and $a2_j$ are the j^{th} attribute of x_1 and x_2 respectively.

Secondly the algorithm introduces a new attribute of a song, *playability*, which derives the suitability of a song to be played at the moment of time considered. Playability function *P* is defined as follows.

P=*P*(release year, number. of times requested/ recommended, last time played, time of the day).

Further, the algorithm addresses a set of global constraints that would cover various requirements of any radio station as detailed under *constraints*. Therefore, this is the first time that these two (CSP and similarity based) techniques are combined to develop a hybrid APG algorithm for radio stations.

```
Let S ≡ playlist, P ≡ set of playable songs and dj ∈ D

Assign random songs from P∩RNS to S
        Until cost(S) is less than threshold
         For each vi
                 Calculate cost vi
         Find vw whose cost is the worst
         For each song dj ∉ S and dj ∈ P
                 vw ← dj
                 Calculate cost(vw)
         Locate dL with the lowest- cost(vw←dj)
         vw ← dL
         Find cost(S)
```

Figure 1. PlayGen algorithm

Experimental results

(0) bonnie & clide - beyonze - rap - 2002
(1) give it to me - nelly furtado - rock - 2007
(2) you all dat - baha men - world - 2000
(3) all rise - blue - rock - 2001
(4) smack that - eminem - rap -2006
(5) feel good inc. - gorillaz - rock -2005
(6) with oive - hilary duff - pop - 2007
(7) because of you - ne-yo - rnb - 2007
(8) coco jumbo – mr president - pop - 2000
(9) real girl - mutya buena - rock - 2007

Figure 2. Example playlist

The evaluation method of the algorithm was to provide a collection of playlists generated by *PlayGen* to two different sets of testers, to be rated in a scale of 5 where score of 1 means "completely unsuitable to be played on radio" and a score of 5 means "perfectly suitable to be played on radio". Firstly, a professional playlist composer (E) at a radio broadcasting corporation rated them by estimating the suitability of them to be played on their radio station. Secondly, a set of frequent radio listeners (L) of the above station rated them considering their own satisfaction derived out of listening to them, relating to qualities like variety etc. In a set of five playlists, the playlists 1-4 were generated using all three compulsory constraints (of which playlist 3 is depicted in Figure 2) while playlist 5 was randomly generated using only C-ALL-DIFFRENT constraint. Figure 3 depicts the playlist ratings from the test. All five playlists were generated from a database containing song attributes of 200 music titles. Most of the attributes were metadata of songs like title, artist, album, and tempo, which were extracted from allmusic.com and id3v2 tags of the songs.

Discussion

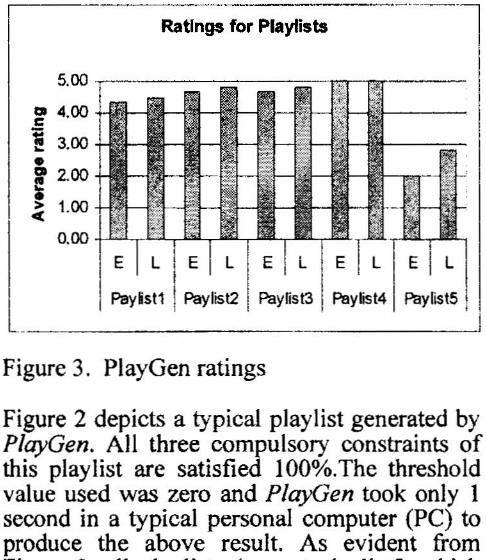

Figure 3. PlayGen ratings

Figure 2 depicts a typical playlist generated by *PlayGen.* All three compulsory constraints of this playlist are satisfied 100%.The threshold value used was zero and *PlayGen* took only 1 second in a typical personal computer (PC) to produce the above result. As evident from Figure 3, all playlists (except playlist5, which was randomly generated) have received very good average ratings (higher than 4.3 out of 5) from both E and L, which concludes that *PlayGen* is capable of generating quality playlists that significantly satisfies the radio station requirements and listener preferences at the same time, even with just three carefully developed constraints.

Conclusions

In this paper the authors considered the automatic playlist generation problem. For the first time, two most popular methods, namely constraint satisfaction approach and the similarity based techniques, of playlist generation, are combined to develop an automatic playlist generator, which would suit radio stations. The developers were able to test and prove that it is feasible to generate quality music playlists using a computer.

References

Aucouturier, J.J. and Pachet, F. (2002) Finding songs that sound the same, *Proceedings of IEEE Benelux Workshop on Model based Processing and Coding of Audio*, University of Leuven, Belgium.

Codognet, P. and Diaz, D. (2001) Yet another local search method for constraint solving, *Proceedings of the AAAI Fall 2001 Symposium*, Cape Cod, MA.

French, J.C. and Hauver, D.B. (2001) Flycasting: On the fly broadcasting, *Proceedings of International Conference on Web Delivering of Music (WedelMusic)*, Firenze, Italy.

Logan, B. and Salomon, A. (2001) A music similarity function based on signal analysis, *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, Tokyo, Japan.

Pachet, F., Roy, P. and Cazaly, D. (1999) A Combinatorial approach to content-based music Selection, *Proceedings of IEEE Multimedia Computing and Systems International Conference*, Firenze, Italy.

Pachet, F. (2002) Scaling up music playlist generation, *Proceedings of IEEE ICME, Swiss Federal Institute of Technology*, EPFL, Lausanne, Switzerland.

Pauws, S., Verhaegh, W. and Vossen, M. (2006) Fast generation of optimal music playlists using local search, *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, Victoria, Canada.

Pestoni, F., Wolf, J., Habib, A. and Mueller, A. (2001) Karc: Radio research, *Proceedings of International Conference on Web Delivering of Music (WedelMusic)*, Firenze, Italy.

Platt, J.C., Burges, C.J.C., Swenson, S., Weare, C. and Zheng, A. (2002) Learning a Gaussian Process Prior for automatically generating music playlists, *Proceedings of Neural Information Processing Systems (NIPS)*, Vancouver, Canada.