RESEARCH ARTICLE OPEN ACCESS

Enhancing Probability of Music Selection Using Smart Shuffling Policy

Suraj Rasal [1], Paras Gandhi [2], Shraddha Shelar [3]
Assistant Professor [1]
Dept. Computer Engineering [1] & [2]
Bharati Vidyapeeth Deemed University's
College of Engineering Pune
Assistant Professor [3]
Dept. Information Technology [3]
D Y Patil College of Engineering, Akurdi
Pune -India

ABSTRACT

When someone plays music in "Shuffled Mode", the entire system is based on randomness [1]. The chances of same song repetitions or a song of your choice not playing are high [1], [2]. This paper deals with this problem and introduces Smart Shuffling concept based on controlled randomness. Classification algorithm of Machine Learning is used for the calculations. Application of this paper covers smart music shuffling that aims to deliver you with next song of your liking on shuffled mode.

Keywords:- Shuffling Policy, Smart Shuffling.

I. INTRODUCTION

Technology always has a room for improvement. Just improving rocket science isn't growth in technology; instead even blinking of eye leading to something brilliant is technology.

There are billions of machine users in the world, it's impossible to provide each of them with a machine tuned up to their needs and choices [1]. However we can provide them with a machine that can tune itself up accordingly. This paper is based on a music shuffling concept that provides user with a dynamic shuffling in which user gets a song of his liking on shuffle mode. The only input user gives is pressing next song button and it's used as a parameter/feature for training the system.

II. RESEARCH WORK

Conventionally, *when* a user plays music on shuffled mode each song has its index and to get the next song

i = Round off [Random ()*T]

where 0 < Random() < 1

T = Total no. of songs

i = Index for next song

This system is completely random; instead this paper provides a smarter algorithm for music shuffling

III. SMART SHUFFLING

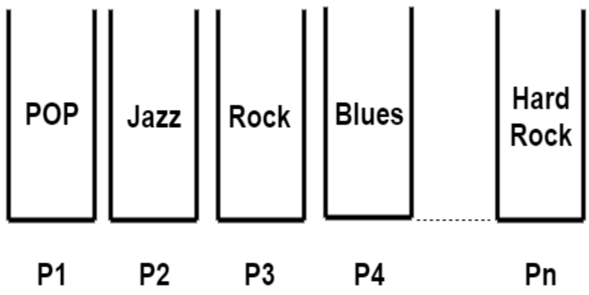

Fig. 1.Types of stacks considered

n = number of stacks

$$\mathbf{P_n h(i_n)} = \frac{1}{1+e^{-i}}$$

initially $i_n = 0$,

so $P_n h(i_n) = 0$

- User listens to a song from stack 'P_n', for playtime 'P_t', where total playback time is

'T_p'. Like Index 'l' for the song is calculated as,

$$l = \frac{2P_t - T_p}{T_p}$$

P_t = Play Time in Seconds

T_p = Actual Length of song in Seconds

From this we can extrapolate that, worst case P_t = 0 and best case P_t = T_p i.e.

$$-1 \leq l \leq 1$$

- The song was from stack P_n so, stack hypothesis [3]**$P_nh(I_n)$** changes as following

$$\mathbf{i_n = i_n + l},$$

$$\mathbf{P_nh(i_n) = P_nh(i_n + l)},$$

where, $P_nh(i_n) = \frac{1}{1+e^{-i}}$

$\frac{1}{1+e^{-i}}$ is called *Sigmoid Function[4]*

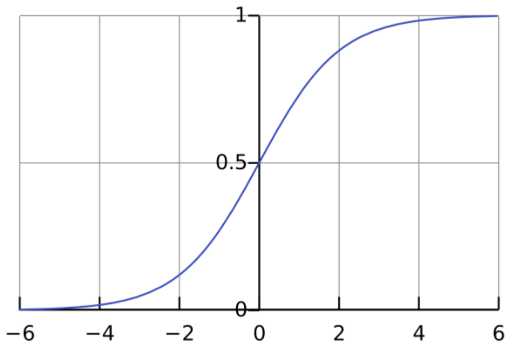

Fig.2.Sigmoid Curve

- For a positive like index l, hypothesis for the stack P_n i.e. **$P_nh(i_n)$** will have a growth and similarly for a negative like index l, it will have a shrink

- Hypothesis ratio:

$$\mathbf{P_1:P_2:P_3......:P_n == P_1h(i_1) : P_2h(i_2) : P_3h(i_3) ....: P_nh(i_n)}$$

- Percentage Ratio **P_nR** for each stack **P_n**, calculated from hypothesis ratio

$$\mathbf{P_nR} = \frac{P_n(i_n)}{T} \times 100$$

$$T = P_1h(i_1) + P_2h(i_2) + P_3h(i_3) + ... P_nh(i_n)$$

Controlled Randomness:

Assumed Data,

hypo stack **H[]** of length 10

- Slots $S(P_n)$ for stack P_n

$$\mathbf{S(P_n)} = \text{Round Off}\left(\frac{P_nR}{10}\right)$$

- Slot Placement in H[] such that,

$$\mathbf{S(P_1) + S(P_2) + S(P_3) ....+ S(P_n) = 10}$$

for Each stack P_n,

let no. of slots $S(P_n)$ be x ,

than slot index s_x is calculated as

$$\mathbf{s_x = Round\ Off[Random() \times 10]}$$

$$\mathbf{x = 1, 2,...., X}$$

Note: If slot s_x is free on stack H[] , P_n is placed at s_x otherwise s_x is recalculated.

ex -

$$\mathbf{H[\ ] = [\ P_1, P_2, P_1, P_n, P_3, P_n, P_2, P_n, P_1, P_1\ ]}$$

- Next Stack **N_P** is picked from **H[ ]** *randomly* by calculating Next Index **N_i** as,

$$\mathbf{N_i = Round\ Off[Random() \times 10]},$$

$$\mathbf{N_P = H[\ N_i\ ]}$$

- Finally, Next Song **N_S** is picked from the selected stack **N_P** by calculating its index randomly(we will maintain randomness but from a selected stack) as

$$\mathbf{N_S = Round\ Off[Random() \times L_P]}$$

$$\mathbf{L_P = Length\ (N_P)\ i.e.\ Length\ of\ stack\ N_P}$$

- *N_S* is the next song that will be played from the stack *N_P*.

IV. CONCLUSION

With this concept we can get a music shuffle tuned up exclusively for each user. The concept of controlled Randomness provides us with a system in which randomness is based on Probability. With time accuracy of the model will rise significantly. This system can be applied to select most significant and user required music. By using this policy user will be able to select his or her most probably music with interested musical properties.

REFERENCES

[1] Markus Schedl · Arthur Flexer · Juli´an Urbano. (2012). The Neglected User in Music Information Retrieval Research. Journal of Intelligent Information Systems. 1 (1), 1-12.

[2] Dmitry KUCHARAVY and Roland DE GUIO. (7th November 2007). APPLICATION OF S-SHAPED CURVES. ETRIA TRIZ Future Conference 2007, Frankfurt. 1 (1), 1-30.

[3] Tamás Jónás. (2007). Sigmoid functions in reliability based management. Rperiodica polytechnica Social and Management Sciences. 15 (2), 67-72.

[4] M.T. Tommiska. (November 2003). Efficient digital implementation of the sigmoid function for reprogrammable logic. Reprinted with permission from IEE Proceedings – Computers and Digital Techniques. 150 (6), 403-411.