Music Playlist Shuffling Evaluation

Umang Jain(17074016)
Anuraj Singh(17074003)

OBJECTIVE

we have a list of songs, lets say 100 songs, and we need to shuffle them such that we can get the permutation of songs that's looks random to humans.

Problem at hand :-

Let’s say we have our algorithm and we got shuffled playlist, Now we need to evaluate how much random (ACC.TO HUMANS) our playlist is?

Example:-

p1 :- m1 m1 m2 m2 m3 m3 m5 m5 m6 m6

p2 :- m1 m2 m3 m4 m2 m5 m6 m1 m2

p1 is **bad** (should have low score on our evaluation matrix) because it doesn’t look random to Humans

and

p2 is relatively better

Features On Which we think Algorithm should be evaluated

1. Uniformly Distributed

Uniform distribution means if we have list of **size 20 consisting of 10 different songs** then ideally each song should be present 2 times in list.

So if we have a list with song1 appearing 7 times then its a very poor shuffling

Here we have a decision to make:

> so if we do not allow repeating means we will have n songs and playlist contain exactly n songs each exactly once, then if don’t need it.
>
> but we allow playlist size to be m (>n) then we need to address make sure our distribution is uniform.

Souce :- http://www.randomwalking.com/cms/cmsfiles/mp3_shuffle.pdf

Solution

we could use variation of chi-square test :- chi-square test is a test for uniformity

x = average ((abs((o[i] - e[i])) / e[i]]) over all i)

Where o[i] = number of times ith song is observed

E[i] = expected value of occurence of ith song (m/n)

source:- https://www.eg.bucknell.edu/~xmeng/Course/CS6337/Note/master/node43.html

2. AUTO-CORRELATION

Playlist items should not be correlated means if last 50 songs is somehow correlated to first 50 songs

then our algorithm not ideal as it is supposed to be random.

Souce :- http://www.randomwalking.com/cms/cmsfiles/mp3_shuffle.pdf

Solutions

we could use

Autocorrelation :- Basically it tells How much current value is correlated to previous values.

Partial-autocorrelation :- Basically it tells How much current value is correlated to previous K values. Where K is lag interval;

Sir, we are Planning to use it like a black box (don't actually understand the maths behind it).

Source :-
https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/

Quality Estimation (Most Important)

Quality estimation :- How much random (according to human) is our algorithm?

Now, this becomes quite hard to do because we want to know how much humanize random our algorithm is.

Solution

We tried

1. Music Shuffling Algorithm Evaluation
2. Machine Translation Quality Estimation
3. https://stackoverflow.com/questions/63903975/how-to-evaluate-a-humanized-music-shuffling-algorithm

Solution

So we gonna go ahead with this solution-

So we will have a list of 100 songs and we will shuffle it with our algorithm and then ask user to judge it on 1-10 scale

Imp :- user will judge on the randomness he/she perceived, not on how much he/she liked the songs in playlist

Example

List after shuffling with algorithm, L

I will listen to all 100 songs and after every 10 songs we will rate previous 10 songs on 1-10 scale

let it be :- 5 8 2 6 4 7 8 5 8 9

Then we can take its average and get the overall rating of playlist

Also we think we should give more importance to first value as compared to second value, and more importance to second value as compared to third value and so on

so instead of average we could use weighted average:-

(10*5 + 9*8 + 7*2 + ...)/(10 + 9 + 8 + ..);

Also we think that for a list a human evaluator will not have confidence in giving rating first time (and also for same list and for same evaluator and same condition rating can be change), so we think that a same list should be evaluated two times by a human(evaluator) and then we will take average.

This way we can have better evaluation.

Overall Evaluation Metric

Overall rating = a(user rating) + b*(uniform distribution score) + c*(auto correlation score)*

Where a,b,c are to number between [0-1] such that a+b+c=1

We we thinking to give more importance to user rating

So a = 0.6, b = 0.2 ,c = 0.2

Things to Discuss:-

1. In Uniform Distribution, should we allow repetitions or not. I think we should.
2. In Quality Estimation, few decisions to be made.
3. Next week plan :-
    a. we could build the GUI and evaluate previous two algorithm.
    b. Parallel we will be looking for some more evaluation techniques.

Thank You