Pandora’s Jukebox

Slides explanation

Features and USP of our model

We have created a personalized recommendation system based on multi arm bandits with a flavour of content based recommendation.

1. The recommendations are based on the user’s choice of genre and artist. These two are our arms (genre) and subarm (artist) that the user pulls.
2. The recommendation system is capable of finding songs based on your past history and preferences (basically for each genre we maintain a list of three songs that he played or played and liked recently and based on this we recommend similar songs).
3. The recommendation system helps you explore difference music. It is based on a modified epsilon greedy policy (explained later) which seldomly provides the user with songs of different genres.
4. The recommendation system is also capable of handling the cold start problem. When the user logs in to the system for the first time, he is asked with his genre preferences which help us provide better recommendation from the very beginning.
5. The recommendation system also takes into account the current mood of the user by updating the arms values accordingly.

Dataset

We had taken data from the million songs dataset (~50K entries). It contained the following features for the songs:

1. Genre
2. Track-ID
3. Artist
4. Title
5. Tempo
6. Loudness
7. Time signature
8. Key
9. Mode
10. Duration
11. Avg-timbre (total of 12 values)
12. Var-timbre (total of 12 values)

Features from 5-12 are related to the core attributes of songs. These particular features were used to find songs similar to the user preferences (content based recommendation mentioned earlier).

Data Preprocessing

1. Two separate text file where converted into csv files and then were concatenated on basis on common field track-id.
2. Artist which had less than 10 songs were filtered out.
3. Songs which had duration less than a minute were filtered out.

Decision Parameters

1. Genre arms: These are the main arms that the user call pull.
2. Artist arm: Each genre arm (main arm) has subarms within it for artist, also called as subarms which the user can pull.
3. Current user preferences: As mentioned earlier we have maintained a separate list (of length 3) of currently played or played and liked songs of the user and this helps up to find similar songs
4. History of ignored recommended item: We also mention a separate list (of length 10) which stores the recommended songs which were ignored by the user. This helps us in not showing the user the ignored songs again, but since its a list of finite length, as and when it gets full, ignored songs get popped out of it and again enter the item pool. Hence it will never happen that the ignored songs won't be ever shown again.

Basic notations

1. a_i : ith genre arm
2. b_j : jth artist sub arm
3. $E(a_{i\_t'})$: estimated value of ith genre arm after t' pulls (similarly for bj)
4. α : weight for genre arm
5. β : weight for artist sub arm
6. R : Reward
7. ε : probability of exploration
8. t : round of recommendation
9. A : user action
10. t' : tth pull of arm

Model

1. $\boldsymbol{\varepsilon}_t = 1/\sqrt{(t+1)}$

   This is the modified epsilon policy we follow. (more on this in the next section)

2. $E(a_{i\_t'}) = \boldsymbol{\alpha}.E(ai\_t'-1) + (1-\boldsymbol{\alpha}).R(A)$

   This is how the new estimated value of a arm is calculated. We give a weight of $\boldsymbol{\alpha}$ to the previous estimate and a weight of $(1-\boldsymbol{\alpha})$ to the current reward. This reward is calculated as the cumulative reward for the subarms that fall under this arm divided by three.

3. $E(b_{j\_t'}) = \boldsymbol{\beta}\ .E(bj\_t'-1) + (1-\boldsymbol{\beta}\ ).R(A)$

   This is how the new estimated value of a subarm is calculated. We give a weight of $\boldsymbol{\beta}$ to the previous estimate and a weight of $(1-\boldsymbol{\beta})$ to the current reward. This reward is calculated based on the actions that the user takes on the respective subarms (the reward policy has been discussed in the further slides).

4. $a_i = \text{argmax}(E(a_{i\_t'}))$

   In every round we select genres that belong to the arms (top 3 for going greedy and top 2 while going exploratory) with maximum estimated value

5. $b_j = \text{argmax}(E(b_{j\_t'}))$

   Once the arms are selected the artists are selected based on the subarms with the maximum estimated value.

6. $\cos(S_i.S_j) = S_i.S_j/(||S_i||.||S_j||)$

   To find similarity between the songs we use the cosine similarity (based on the song features 5-12 discussed above).

Modified $\boldsymbol{\varepsilon}$-greedy policy

We have kept $\boldsymbol{\varepsilon}$ as a function as number of rounds (t). This helps us in establishing an initial exploratory phase when $\boldsymbol{\varepsilon}$ is > 0.3. $\boldsymbol{\varepsilon}$ starts with a value of around 0.7 and in around 10 rounds falls to 0.3. After it reduces below 0.3 we stop the initial exploratory phase and keep $\boldsymbol{\varepsilon}$ constant as 0.3 from thereon (normal phase).

Actions

Once the songs have been recommended to the user the user can do the following actions

1. Played: This means that the user has completely played the song
2. Stopped: This means that the user played the song but stopped it in between
3. Liked: This means that the user liked the song.
4. Disliked: This means that the user disliked the song.
5. Ignored: This means that the user did not do anything with the song.

Reward Policy

For the artist subarms we followed the following reward policy

1. Played got a reward of 0.5
2. Played and Liked got a reward of 1
3. Played and Disliked got a reward of -1
4. Stopped got a reward of 0.25
5. Stopped and Liked got a reward of 0.75
6. Stopped and Disliked got a reward of -1
7. Ignored got a reward of 0

New estimated value of the arm was calculated according to point 3 in Model section. β was kept as 0.6.

For the genre arm we took the cumulative reward from all the songs that fall within this genre and divided it by 3 to obtain the total reward of the genre (3 because we at max showed 3 songs from any particular genre, more about it in the next section). New estimate value of the arm was calculated according to point 2 in Model section. α was kept as 0.8.

We kept different values for α and β, particularly more for α because if a user did not like a particular artist from the genre does not mean that he dislikes the whole genre. So we retain much of the previous estimate of the genre by keeping α high. And keeping β low ensures that the artist gets penalised a bit heavily and does not occur again much.

Learning algorithm

0. For new user, take input from the user regarding his preference of genres and give those genre arms an initial estimated value of 0.5.

1. Check if exploratory phase is on

1.1 If yes then calculate epsilon according to the formula 1/sqrt(rounds + 1)
1.2 Else set epsilon as 0.3

2. Randomly generate a number and see if it is greater than epsilon

2.1 If yes go greedy
2.1.1 Find the top three arms (which represent genre) with maximum estimated value

2.1.2 For the first genre get three songs of that genre (matching the user preference list for that genre)
2.1.2.1 Two of this should belong to the subarm (artist within that genre) which has the maximum estimated value.
2.1.2.2 One of this should belong to any artist but should same genre.

2.1.3 For the second genre get two songs of that genre (matching the user preference list for that genre)
2.1.3.1 One of this should belong to the subarm (artist within that genre) which has the maximum estimated value
2.1.3.2 One of this should belong to any artist but should same genre

2.1.4 For the third genre get one songs of that genre and this should belong to the subarm (artist within that genre) which has the maximum estimated value. (matching the user preference list for that genre)

(Above we exclude the songs in the ignore list)

2.1.5 Recommend these 6 songs to the user

2.2 If no go exploratory

2.2.1 Find the top two arms (which represent genre) with maximum estimated value

2.1.2 For the first genre get two songs of that genre (matching the user preference list for that genre)
2.1.2.1 One of this should belong to the subarm (artist within that genre) which has the maximum estimated value.
2.1.2.2 One of this should belong to any artist but should same genre.

2.2.3 For the second genre get two songs of that genre (matching the user preference list for that genre)
2.2.3.1 One of this should belong to the subarm (artist within that genre) which has the maximum estimated value.
2.2.3.2 One of this should belong to any artist but should same genre.

2.2.4 Next randomly choose a genre and select two songs from this genre (matching the user preference list for that genre).

2.2.5 Recommend these 6 songs to the user

(Above we exclude the songs in the ignore list)

3. Take input from the user on how he/she reacted to the recommended songs.

4. Calculate rewards for these songs (artist subarms) according to the reward policy for artist subarms (as discussed above).
4.1 If any song was ignored then add it to the ignored queue (if full pop a song hence remove it from the ignored queue)
4.2 Update artist subarms.

5. Calculate reward for the genre arms according to the reward policy for genre arms (as discussed above).
5.1 Update genre arms.

6. Update the user preference history (last 3 items played or played and liked for particular genres).

7. Back to step 1

Accuracy calculations?

Accuracy calculations were not possible with our model because:

1. We had ourselves defined the actions that the user could do with the recommended songs (like play, stop, ignore etc).
2. It was not possible to get test data for the same.
3. Even if we relied on practical testing of the model (like requesting our friends to use the model) then the obtained accuracy results would not have been satisfactory given the inherent bias.