

YouTube Radio Bot

By Marlon D'Ambrosio

Structure:

1. Introduction

1.1 Motivation

1.2 Objective

2. Background/History of the Study

2.1 What are bots?

2.2 Bots in the music industry

3. Approach and Implementation

3.1 Simulating human input with PSO

3.2 Selenium and AutoIT

3.3 Supervised learning for determination of fitness of video

4. Experiment Results and Discussion

5. Conclusion

6. References

1. Introduction

I always found the thought of bots mimicking human behaviour very intriguing, companies on various platforms are fighting against artificial clicks/streams and accounts that are not managed by humans. So, generating clicks through bot accounts is rather easy, but the difficulty in that project is going to move around certain parameters and to completely mimic the behaviour of a person. So that the AI will develop its own preferences in music, taste and it's going to simulate its listening behaviour accordingly. Meaning at the end we would have a fully functional radio of randomized songs based on songs found and associated with the initial artists given. By better understanding how to mimic normal behaviour, you can better protect the application through against the usage of those kind of bots. The emerging behaviour patterns of an AI are easier to analyse and to discover if you are looking from the side of the Bot and not from the other side.

The objective of the project is to simulate a human-like behaviour on music streaming

services to better understand how the already existing bots are able to not get removed, so that the application can improve their bot protection system. Not only that, but also creating a viable option for the user to listen to their favourite artists radio on YouTube without ads and with the accompanying video.

2. Background/History of the Study

With all the positive aspects and all the incredible useful tools the internet has given us, there are always those who abuse that power and are not using it for their intended usage. This paper is going to delve deeper into one tool that has a very negative connotation due to its usage, named bots. A bot is “a computer program that runs automated tasks over the internet”(Oxford Dictionary). There are “good” and “bad” bots, the good ones are obeying the site’s owners intentions, ToS and their robots.txt file, which indicates the sites a bot is allowed on. Application for those can be web crawling for example by google, chatbots or socialbots. On the other hand, there are “bad” bots, that do violate at least one of the criteria given above. These are used for spamming content, email harvesting or generating fake clicks. [2]

Since fake streams are now available more than ever, streaming services like sportify and tidal are trying their best to crack down on those. But creating an effective and efficient bot protection is not very simple, because differentiating between a human and bot can be very difficult [3]. To improve that detection system, you need to know how to do a bot yourself first. Since there needs to be a pattern or a logical sequence the songs are being played at, to seem less suspicious. It is built like a radio station of that artists to play their songs and similar artists.

3. Approach and Implementation

To seem more like a human being navigating through the website and not a computer program, I simulated a natural seeming human mouse movement with the particle swarm optimization. To surpass pattern recognition of the bot protection it needed the randomness factor of the algorithm.

To optimize the runtime of the algorithm, I split algorithm into two independent parts. One is for finding the correct x coordinates and the other one is for finding the correct y coordinate. Splitting them up, allowed the program to stop finding one of the coordinates if it is sufficiently close and therefore optimize the total time of the task. The fitness function is defined by the absolute distance of the current position and the target position.

After testing out many different values for the constants c1 and c2, they are both set at 1.2. Further, to not find the element too fast, not to overshoot the target coordinates too much and seem more natural, the weight constant in the velocity function is determined by the function: $0.0719644 * (x^{0.30423}) + 0.468071$. That function returns values from 0.6 at 10 to 0.9 at 600. I needed that function to regulate the velocity of the individual particles. The closer it gets to the target the more accurate it also need to be. Is the weight too high, the target gets overshooted by quite a lot and the jumps between iterations are too high. Is the weight too low, the algorithm would take too long to solve. The function determines a perfect balance between too high and too low depending on the fitness of the current element.

The terminating condition of the algorithm is fairly simple and straight forward. The algorithm stops if difference of both x and y is smaller than 3. In this case the absolute difference is not used, because the target coordinates are from the top left corner of the element. That means if x or y is lower than the target is not be able to be clicked on properly.

To determine attributes of elements of the DOM from the current site, I needed to use the selenium library with a chrome web driver. To further enhance the user experience, the program contains an AdBlocker, so the YouTube videos run smoothly without any unnecessary interruptions. Selenium does not have a possibility to move the mouse independently but rather just jumps to elements. For that reason, the program uses another module called AutoIT, which is originally a script language to automate mouse movement/behaviour and keyboard input.

The program now is able to search for an inputted artist, but to determine which of the songs are falling into the criteria of being a song of that artists or similar ones is done with supervised learning. First the song goes through multiple criteria like length of video, certain keywords or the structure of the title, which is almost identical for every proper Song. After the song determination, the song is given a fitness of either 0 (No

song or a song by an unknown artist), 3 (produced from someone the artist was produced by or someone the artist had a feature with), 7 (the artist is a feature on the song) and 10 (The song is by the artist). If the artists is on the same song with other artists, then those ones are getting saved for next evaluations(fitness=3). The decision to make more than the main artist play is made for diversity reasons and to enhance user experience as much as possible while listening to the “radio”. The fitness also determines the probability that the associated song is going to be played next.

4. Experiment Results and Discussion

After thorough testing of the PSO, it is determined to get the element at nearly 100 % efficiency, this is due to a purposefully chosen lack of the accuracy with the accommodating range of 3.

The learning criteria on the other hand, is less accurate. This is mainly due to a lack of consistency in the music title the artists choose. There is a general unofficial consensus on the formatting of a title, e.g. TRETSMANN - GRAUER BETON (prod. KITSCHKRIEG) (OFFICIAL VIDEO). There should always be a “–” between artist and title, unfortunately this is sometimes not the case. Those songs fall through and are not being considered to be played.

5. Conclusion

To conclude, the project did manage to fulfil its proposal to a certain extend, it is able to navigate through YouTube, play songs of a specific artist and simulate human behaviour, not only in the mouse movement but also in playing behaviour with the help of the supervised learning. The more time spent on this project the more fascinating the topic became for me and the more I realized how difficult Bot protection for such big platforms are. Nevertheless, I would say that my knowledge and level of awareness regarding botting has increased tremendously.

6. References

Provide proper reference you have used.

- [1] Bot. bot noun - Definition, pictures, pronunciation and usage notes | Oxford Advanced Learner's Dictionary at OxfordLearnersDictionaries.com. (n.d.). Retrieved May 7, 2022, from <https://www.oxfordlearnersdictionaries.com/us/definition/english/bot>
- [2] Cloudflare, C. (n.d.). What is a bot? What is a Bot. Retrieved May 7, 2022, from <https://www.cloudflare.com/en-gb/learning/bots/what-is-a-bot/>

[3] King, A. (2021, October 4). What are fake streams? Spotify Issues Warning, may withhold royalties. Digital Music News. Retrieved May 7, 2022, from <https://www.digitalmusicnews.com/2021/10/03/what-are-fake-streams-spotify-for-artists/#:~:text=Spotify%20is%20cracking%20down%20on,song%2C%20%E2%80%9CA%20Lot.%E2%80%9D>