Ray Butler

11/6/2022

YouTube’s Recommendation Overview
New Content VS High Popularity Content

YouTube is one of the largest recommendation systems in use. Not only in the number of users it works with but also in the sheer size of its data set. It also faces a common problem for social media platforms. The data it is working with is rapidly changing and growing. Many hours of video are uploaded per second. The recommendation system needs to work just as well with the new data as it does with the old data. This is especially true for regular YouTube users who want to see new videos on topics they would be interested in with out having to search for them.

In conjugation with other product areas across Google, YouTube uses deep learning as a general-purpose solution for nearly all learning problems. The system is built on Google Brain which was open sourced as TensorFlow.

The overall structure of the recommendation system is 2 neural networks. The first one taking a much larger set of inputs and then feeding a smaller set of inputs into the second one. The first network is for candidate generation and the second for ranking. The candidate generation network takes events from the user’s YouTube activity history as input and retrieves a small subset (hundreds) of videos from a large corpus. These candidates are intended to be generally relevant to the user with high precision. The candidate generation network only provides broad personalization via collaborative filtering. The similarity between users is expressed in terms of coarse features such as IDs of video watches, search query tokens and demographics.

Presenting a few “best” recommendations in a list requires a fine-level representation to distinguish relative importance among candidates with high recall. The ranking network accomplishes this task by assigning a score to each video according to a desired objective function using a rich set of features describing the video and user. The highest scoring videos are presented to the user, ranked by their score.

The two-stage approach to recommendation allows YouTube to make recommendations from a very large corpus (millions) of videos while still being certain that the small number of videos appearing on the device are personalized and engaging for the user. Furthermore, this design enables blending candidates generated by other sources.

During development, YouTube makes extensive use of offline metrics (precision, recall, ranking loss, etc.) to guide iterative improvements to the system. However, for the final determination of the effectiveness of an algorithm or model, they primarily rely on A/B testing via live experiments. In a live experiment, they can measure subtle changes in click-through rate, watch time, and many other metrics that measure user engagement. This is important because live A/B results are not always correlated with offline experiments.

One of the interesting things YouTube discovered through A/B testing is that users don’t want to have their homepage recommendations overly weighted by what the recently searched or watched. But when watching a video, the recommendation for which video to watch next should be weighted by this information. This basically led to 2 slightly different recommendation systems. The one for the homepage that takes the collection of user history as a bag or words style approach converted to n-grams and bigrams does know the source of each word and doesn’t give as much weighting to how recent it was. The one recommending which video to watch next while watching a video is much more heavily weighted especially to the current video or a recent search, if any, that was used to find the video.

In conclusion, YouTube is trying to solve some of their unique problems in recommendation through the use of two neural networks instead of just one. The feedback to how well these are working from a user perspective comes primarily from live A/B testing of user responses to small changes.

Source: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45530.pdf

Source: https://daiwk.github.io/assets/youtube-multitask.pdf