A Brief Review of YouTube Video Recommendation System

Jiecheng Zhao
Department of Computer Science
University of Illinois, Urbana-Champion
Urbana, IL, USA
jz109@illinois.edu

ABSTRACT

Video recommendation system is a major role in attracting users. Meanwhile, it faces challenges such as scale, high dynamics, and noisy data. In this paper, we reviewed the technologies used in YouTube recommendation system in the past few year.

KEYWORDS

Multi-task learning; mixture of experts; recommendation system

1 Introduction

As probably the largest video website of the world, YouTube is watched over 1 billion hours every day, by more than 2 billion human users [1, 2]. The recommendation system is a critical part of YouTube, for over 70 percent of watch time on YouTube was spent on the videos recommended by the system [3].

YouTube recommendation system provides personalized recommendations that help users find high quality videos relevant to their interests. The goal is to attract the users to watch more videos, find useful videos, and ensure that good quality videos are watched by more users.

Like most recommendation systems, the YouTube recommendation system generally includes two components: candidate generation and ranking. The candidate generation forms a subset of videos from all the available videos according to the features of video contents and the user's past activities. The ranking system scores this subset of videos based on specific objects and features and select a port of them as output.

The realization of YouTube recommendation system is extremely challenging from the following aspects [5, 7].

- Scale: with billions of users and videos, the scalability of recommendation algorithm is extremely important. Many proposed algorithms work well on small number of contents, but their performance degrades or become inefficiency when serving in such a huge system. The recommendation system needs to be both effective and efficient in dealing with large amount of data.
- Freshness: YouTube has very dynamic corpus. Many hours of new videos are uploaded every second. Meanwhile, the users' activities update over the time. It is critical for the recommendation system to be responsive to new contents and users' new actions, but also considers the well-established videos.
- Noise. Videos may have less or no meta data for the system to classify. Although users provide explicit feedback by rating videos, the major users' activity is implicit, such as clicks, watching time and comments. These implicit feedbacks provide noisy signals for the recommendation system the predict user's behaviors and satisfactions.

From technology perspective, the ever-developing text mining, machine learning and natural language processing methods provide powerful tools for the YouTube video recommendation system. As the core technology evolves from statistic-based algorithm to deep-learning-based algorithm, the architecture of the recommendation system becomes more complex and can fusion more features of both videos and users. The goal of the recommendation system is better achieved thanks to these technologies.

This paper briefly reviewed the algorithms used by the YouTube video recommendation system in the past few years. The key parts of the algorithms are analyzed and compared from the perspectives of objective, input data, candidate generation, ranking, implementation and performance.

2 Technology Review

In this section, several key technologies used in the YouTube video recommendation system from 2010 to 2019 are reviewed [4-7].

2.1 Objective

In general, the objective of the YouTube video recommendation system is to provide a list of videos the users may be of interest. However, the quantitative standard of this objective changes over the time. Because of this change, the recommendation algorithms are designed and optimized from different perspectives.

In the early stages, the objective is the increase the clicks of the users, i.e. the views of the videos [4]. This partially led to clickbaits, i.e. videos with misleading titles and thumbnails to attract users' clicking.

Then the objective evolves to the watch time. The target is to increase the watch time of each user, indicating the users enjoy the contents. However, a long watch time does not necessarily mean quality time spent. Uses' satisfactory is then added to the objective.

2.2 Input Data

The input data of the recommendation system generally includes two parts: the content data and user activity data.

The content data includes the raw video streams, video metadata such as title, description, user ratings, clicks, etc. The user activity data can be further divided into explicit and implicit categories. The former includes the user's rating to a video, favoring a video, and subscribing to an uploader. The implicit activities are generated from user's watching and interactions, such as start to watch a video, watched a big portion of a video, and completing watching a video. It could also include the user's geographic region, device, gender, logged-in state, and age [5].

All the input data provides features of videos and users and are input into the algorithms to generate a list of recommended contents.

2.3 Candidate Generation

Candidate generation is to form a subset of all the videos available in YouTube site, based on the user's activities and videos' features. In the early years, this is achieved by finding the relations between to videos. The higher the relatedness score is, the higher probability that these two videos are co-watched within sessions [4]. The score is generally based on the normalized co-watch counts. With these scores, the videos highly relate to the user's favorited videos can be collected. Furthermore, those videos highly to these collected videos can be added to the collection. This can iterate several times, to collective vides reachable within a distance of n from any videos the user favorites. This greatly increased the videos new to the users.

On the other hand, the recommendation can be deemed as a multiclass classification problem. E.g., the videos can be classified as 'interest/positive' or 'not interest/negative' to the user. Based on this idea, Deep Neural Networks (DNN) is used to generate the candidate set. The features of videos and users are input into DNN for training [5]. The features are concatenated into a first layer, and then followed by several layers of fully connected Rectified Linear Units (ReLU).

Machine learning system exhibits a tendency towards the part because they are trained from historical examples. Example age is fed into the model for correction. With this feature, the model can accurately accommodate time-dependent popularity.

The recommendation system is sometimes an optimization problem, and the recommended videos gain the highest score in the objective function. However, as discussed in 2.1, the objective of the recommendation system may not be single. E.g., the system would like to recommend a video the users would like to watch completely, and rate highly, and share with their friends. The different objectives could sometimes conflicting. To address this, a multi-task learning approach, Multi-gate Mixture-of-Experts (MMoE) is used to learn the model tasks relationships from data [6, 7]. In MMoE, the labels and features are input into a group of networks, each of which is called an expert. Then a gating network for each task is introduced. The gating networks take the input features and output softmax gates. These softmax gates assembles the experts with different weights, to achieve different tasks. The results of assembled experts are then passed into the task-specific tower networks.

2.4 Ranking

The generated candidates cannot be displayed to the users in all, due to the space of the webpage and the information acceptation ability of users. Typically, only the most related ones are selected. Furthermore, they should be sorted so the users will likely see the most related one in the first place. This is realized by the ranking procedure. Ranking scores the candidate videos.

Early ranking method scores the candidate videos based on two criteria: video quality and user specificity [4]. Both scores are then linearly combined to form a rank list of the candidate videos. At this step, diversify is considered, since users tend to watch different types of videos instead of only one or two types. Typically, the number of each topic or the number from the same uploader is limited so that the final recommended videos are diversified. Another method is to limit the number of videos generated from the same related video.

DNN can also be used for ranking [5]. Compared with candidate generation, in the ranking step, the video set is smaller, so more features can be used for the model training, without significantly increase the computation effort. The score/probability given by the DNN is used for ranking.

When multiple ranking objectives exist, MMoE can be used to generate a multitask ranking model [6]. For each candidate, the score of each task is generated, and these scores are weighted and combined, to generate a final score for ranking. The weights can be manually tuned to achieve the best performance.

The position of a video in a query could introduce bias to the ranking system [7]. This is because the users are inclined to clicking and watching videos displayed closer to the top of the list, even though they are not quite interest to the user. To tackle this issue, the positions of all videos are used, to prevent the model from over-relaying on the position feature. This is implemented by introducing a shallow tower to the model. The input of this shallower tower is the ranking order decided by the current system, and the output is added to the final logit of the main model, to eliminate the bias.

2.5 Implementation

The implementation of the recommendation system mainly considers the large scale of YouTube videos and users. In the statistics-based approaches, the relations between videos can be pre-computed [4]. This enables the access to large amount of data with ample amounts of CPUs. Low latency is therefore achieved in generating the recommendation list. As the dataset (videos) are generating from time to time, the computation is executed several times per day for updating.

MapReduce technique is also used, to fully utilized the distributed computation resource, and achieve reliability and low latency [4].

In the multi-task recommendation system, a shared-bottom DNN structure is used [6]. It allows parameters to automatically allocated to capture either shared task information or task-specific information, avoiding adding many new parameters for each task.

2.6 Performance

All recommendation systems we reviewed achieved excellent performance improvement compared to their peer algorithms. The statistical-based recommendation system achieve almost twice click through rate than the most viewed videos [4].

The DNN-based system increased the watch time dramatically on recently uploaded videos in A/B testing and improved the holdout mean average precision (MAP) [5].

The MMoE-based system with shallow tower increased users' engagement and satisfaction when comparing with the shared-bottom model with the same computation resources [7].

3 Conclusion

The design and implementation of a recommendation system for a large video site such as YouTube is very challenging, due to the characteristics such as scalability, high data dynamics, and noisy date. Meanwhile, the multiple objectives exist simultaneously, as the users' objectives on each video are not necessarily identical.

Deep-learning based method is the state-of-art in realizing this recommendation system. Multiple objects are achieved in candidate generation and ranking. Large number of features are input into the model to improve the accuracy and eliminate the biases.

In implementation, based on the cloud-computing system, distributed computation technology is used to achieve low latency and high reliability. The computation algorithm is also designed to maximize the resource sharing.

In the future, the following topics could be of interest in video recommendation system development.

1. Furtherly improve the accuracy by including more features, such as user's activities from sites other than YouTube.
2. Recommend a specific section of a video instead of a whole video.
3. Optimized the weights of different tasks in MMoE automatically.

REFERENCES

[1] https://backlinko.com/youtube-users#youtube-statistics/

[2] https://blog.hootsuite.com/how-the-youtube-algorithm-works/

[3] https://www.cnet.com/news/youtube-ces-2018-neal-mohan/

[4] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. V. Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath, "The YouTube video recommendation system," in Proceedings of the fourth ACM conference on Recommender systems, Barcelona, Spain, 2010.

[5] P. Covington, J. Adams, and E. Sargin, "Deep Neural Networks for YouTube Recommendations," in Proceedings of the 10th ACM Conference on Recommender Systems, Boston, Massachusetts, USA, 2016.