

# Youtube Recommendation System

Ashish Sharma, Anirudh Kamat, Jyoti Mudkanna  
(Project Guide)  
MIT ADT University

**Abstract:-** YouTube is one of the most sophisticated and large-scale industrial recommendation systems available. The system is described at a high level in this work, with a focus on the huge performance increases brought on by deep learning. The study is divided into two sections based on the conventional two-stage information retrieval dichotomy: first, we describe a deep candidate generation model, and then we explain a separate deep ranking model. We also share practical takeaways and insights gained from creating, iterating, and maintaining a large-scale recommendation system with significant user impact.

## INTRODUCTION

Every major e-commerce or entertainment website makes product recommendations based on a variety of criteria. Like how YouTube and Netflix recommend movies and shows, or how Amazon recommends goods that it believes we will like. All of these rely on some sort of recommendation system. They strive to make the experience more personalised and geared towards you based on your past history and how you have engaged with the service or how similar people have interacted with the service. This results in a more pleasant and efficient experience for the customer, as well as a significant time and energy savings for the firm.

A recommendation system is an algorithm that may be used to propose relevant material to the user. We can't picture life without a recommendation system now that it's become such an integral element of online content consumption. Every day, 720,000 hours of content are uploaded to YouTube alone. Amazon's websites have an inventory of over 12 million products. How can one search and determine what to watch or buy when there are so many options? If this ever-growing list of options is shown to the user in its raw form, it will confuse and frustrate them, ultimately leading to a negative service experience.

YouTube is the most popular video-creation, sharing and discovery site on the internet. Over a billion YouTube users rely on recommendations to find personalised content from a vast collection of videos. In this study, we'll look at how deep learning has lately influenced YouTube's video recommendation algorithm.

## TECHNIQUES USED FOR RECOMMENDATION SYSTEM:

### 1. Content-Based Filtering:

A content-based recommender system proposes products that have features in common with goods the user has previously liked. A typical CB recommender starts by creating a user profile based on user comments and item evaluations. After that, the user profile is compared to item features, and the items that match are recommended. For example, it recommends novels based on extracted book descriptions from the Amazon.com website. It learns about the user's preferences and suggests books based on those preferences. Different profile-item matching approaches are utilised in the CB approach to match aspects of new items with user profiles and determine whether or not a given item is attractive to the user. Utilities are assigned to items based on utilities previously assigned to observed items, and user profiles are either intuitively or explicitly changed.

The attributes and descriptions of these objects are described by item profiles, which are represented by Boolean values or integer values reflecting term frequency (TF) or term frequency-inverse document frequency (TF-IDF). By comparing item representations with representations of user interest and preferences utilising keyword matching, identifying nearest neighbour approach, cosine similarity, and typical categorization, items are automatically identified as relevant or non-relevant.

### 2. Collaborative Filtering:

Collaborative filtering matches users with other users who share similar interests to propose things. It takes user feedback in the form of ratings for specific items and looks for patterns in rating behaviour among users to identify groups of users with similar preferences. A user profile is a collection of user preferences that the user has submitted either directly or implicitly. Amazon, for example, employs the CF method to offer things based on user purchasing trends as well as user ratings.

In a normal CF setting, there is a list of users and things represented by the numbers  $[u_1, u_2, u_3, \dots, u_n]$  and  $[i_1, i_2, i_3, \dots, i_n]$  respectively. Each user has a list of objects to rate, which can be done directly or implicitly. In this method, a user-item rating matrix 'R' is created, which represents user preferences about products. Different strategies are used to identify missing ratings,

such as finding "nearest-neighbour" for new users in recommending things to them based on ratings provided by their nearest-neighbours.

### 3. Hybrid Filtering:

Two or more filtering methods are integrated in a hybrid approach to improve performance above CB and CF procedures when used separately. Several researchers combined CB and CF methodologies in order to improve results and address the weaknesses of both approaches. Burke [6] divides hybridization approaches into seven categories:

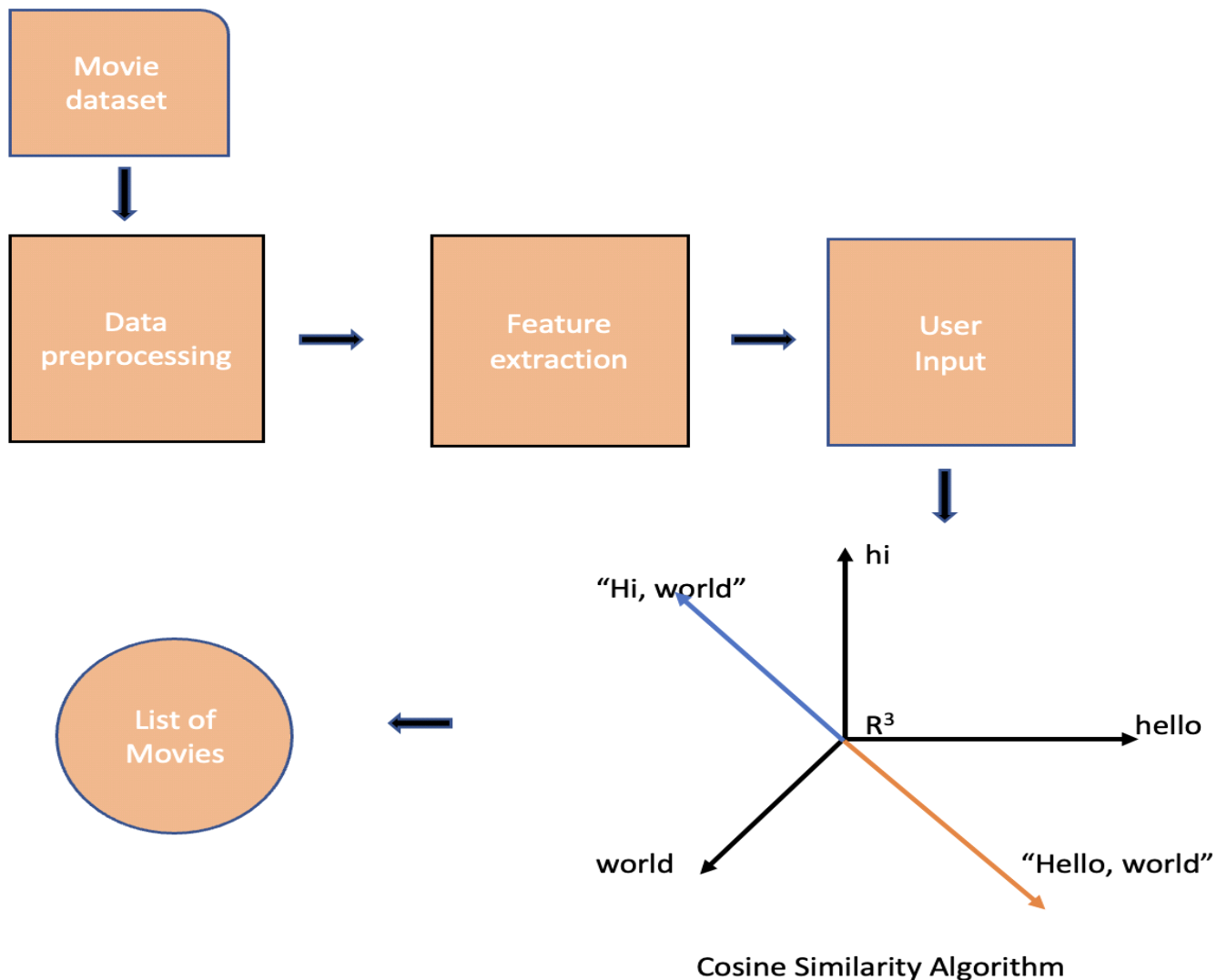
(1) weighted, (2) switching, (3) mixed approach, (4) feature combination, (5) cascade, (6) feature, and (7) meta-level hybridization technique. Burke claims that combining several filtering methods achieves peak performance and solves difficulties that individual filtering approaches suffer when used separately.

### MODEL ARCHITECTURE

We learn high dimensional embeddings for each video in a set vocabulary and feed these embeddings into a feedforward neural network, inspired by a continuous bag of word language models. The embeddings translate a variable-length series of sparse video IDs to a dense vector representation of a user's watch history. The network requires fixed-size dense inputs, and among different ways, simply averaging the embeddings performed best (sum, component-wise max, etc).

Importantly, the embeddings are learned in tandem with the rest of the model parameters using standard gradient descent backpropagation updates. A wide first layer of features is concatenated, followed by numerous layers of completely connected Rectified Linear Units (ReLU)

### YOUTUBE RECOMMENDATION SYSTEMS WORKING:



Two neural networks are used in the system: one for candidate generation and the other for rating. The candidate generation network uses events from a user's YouTube activity history to get a tiny subset (hundreds) of videos from a big corpus. These candidates are meant to be highly precise while yet being broadly relevant to the user. Only collaborative filtering allows the candidate generation network to deliver extensive customisation. Users are grouped together based on coarse characteristics such as video watch IDs, search query tokens, and demographics.

To identify relative importance among candidates with strong recall, a fine-level representation is required when presenting a few "best" recommendations in a list. The ranking network does this goal by assigning a score to each video based on a specified objective function and a large number of characteristics that describe the video and the user. The user is shown the highest-scoring videos, which are ranked by their score.

The two-stage approach to recommendation allows us to make recommendations from a big corpus of videos (millions) while being confident that the limited number of videos that surface on the device are personalised and entertaining for the user. This architecture also allows for the blending of candidates provided by other sources, such as those outlined in a previous paper.

Offline measurements (precision, recall, ranking loss, and so on) are heavily used during development to steer iterative system improvements. However, we rely on A/B testing via live experiments for the final conclusion of an algorithm's or model's performance. We can track small changes in click-through rate, watch time, and a variety of other user engagement data in a live experiment. Because live A/B results aren't always connected with offline studies, this is significant.

#### Cosine Similarity Algorithm:

The similarity of two vectors in an inner product space is measured by cosine similarity. It detects if two vectors are pointing in the same general direction by measuring the cosine of the angle between them. In text analysis, it's frequently used to determine document similarity.

*Cosine similarity:* This measures the similarity using the cosine of the angle between two vectors in a multidimensional space. It is given by:

$$\text{similarity}(x,y) = \cos(\Theta) = x \cdot y / \|x\| * \|y\|$$

#### Output:

```
Movies suggested for you :  
1 . Iron Man  
2 . Iron Man 2  
3 . Iron Man 3  
4 . Avengers: Age of Ultron  
5 . The Avengers  
6 . Captain America: Civil War  
7 . Captain America: The Winter Soldier  
8 . Ant-Man  
9 . X-Men  
10 . Made  
11 . X-Men: Apocalypse  
12 . X2  
13 . The Incredible Hulk  
14 . The Helix... Loaded  
15 . X-Men: First Class
```

### RESEARCH OPPORTUNITIES IN ALLEVIATING ISSUES AND CHALLENGES:

We give some design principles for fine-tuning recommenders that will perform better in mitigating difficulties such as latency, cold-start, scalability, context-awareness, grey-sheep, and sparsity by carefully studying the issues and challenges presented in Section 3.

These guidelines include the following:

- A recommender system that employs demographic filtering and clustering can group users with similar preferences and demographic characteristics so that the system can focus on the right user group rather than the entire dataset. This will reduce latency, improve speed, and address the sparsity and grey sheep issues.
- Newly registered users' personal information can be retrieved through registration. Contextual information such as location, time, and other factors can be gleaned from their IP address, and things that have been most frequently seen, downloaded, and purchased by other users with comparable contextual information are recommended. This is a simple way to avoid the problem of a cold start.
- For a user whose preferences change frequently, two recommendation lists should be kept. The first should be kept up to date with the user's current preferences, while the second should be kept up to date with the user's long-term preferences so that the system can offer things that fit the user's previous transaction history.
- Items that are obsolete or older should be filtered out by the recommender system. To find such products, a time threshold should be employed, and as a result, newer items should be displayed to consumers along with accurate suggestions. begin the issue.

### CONCLUSION:

Thus the architecture and reasonings for the youtube video recommendation system can be understood. And now we know how it works we can build our own recommendation system for youtube.

By using content-based filtering in the movie recommendation system, we were able to demonstrate how to build a movie recommendation system. In the realm of the internet, recommendation systems have become the most important source of useful and trustworthy information. Simple ones take into account one or a few parameters, whilst more complicated ones employ multiple parameters to filter the results and make them more user-friendly. A good movie recommendation system can be constructed using advanced deep learning and various filtering approaches such as collaborative filtering and hybrid filtering.

This might be a significant step forward in the development of this model, as it will not only become more efficient to use, but will also boost the business value.

### REFERENCE:

- [1] Cristos Goodrow's Inside Youtube : On Youtube recommendation system. 2021.
- [2] Collaborative Filtering in Recommender Systems: Technicalities, Challenges, Applications, and Research Trends, Pradeep Kumar, Pijush Kanti Datta Pramanik, Avick Kumar Dey Parsenjit Chaudhory ResearchGate 2020.
- [3] Recommender System: Issues , Challenges, and Research opportunities, Shah Khusro Zafar Ali, Irfan Ullah , IEEE 2020.
- [4] Recommender Systems: An Overview, Research Trends, and Future Directions, Pradeep Kumar, Pijush Kanti Datta Pramanik, Avick Kumar Dey, Parsenjit Chaudhory. IEEE 2021.
- [5] M. Muozorganero, G. A. Ramezgonzlez, P. J. Muozmerino, and C. D Kloos, "A collaborative recommender system based on space-time similarities," vol. 9, no. 3, pp. 81-87, 2010.
- [6] Suvir Bhargav. Efficient features for movie recommendation systems. 2014.
- [7] G. Wang, "Survey of personalised recommendation system," Computer Engineering & Applications, 2012.
- [8] Kelvin Luk, Introduction to TWO approaches of content-based recommendation systems.