Netflix Recommendation System

How does Netflix recommend movies?

Parth Patel - 02/22/2022

Reference - M. Chiang, *Networked Life.* New York, NY: Cambridge University Press, 2012.

Netflix Recommendation System

Overview

- It uses the data related to user behaviours to profile each user's taste in movies.
- Among the inputs to this system is the data of ratings given by users to the movies. A data point consists of 4 fields - userID u, movie index i, ratings r_{ui} and the date when ratings where given.
- The output is the prediction of ratings for each user u and movie i which is not watched by that user. It can be limited to few numbers of movies having highest prediction values which indicates that user is likely to watch these movies more.

The Netflix Prize

Competition

- In October 2006, Netflix organised international contest to create recommendation algorithm that can achieve 10% more accuracy than the in house “Cinematch” recommendation system.
- Around 5000 teams participated and 44000 submissions were made by those teams throughout the contest period of 3 years.
- Netflix made 2 dataset public - training set of 100M reviews and probe set of 1.4M reviews. Additionally it had 2 data sets of 1.4M reviews quiz and test set which were hidden and were used for testing.
- The Cinematch algorithm had RSME of 0.9563 and the final best 2 teams achieved RSME of 0.8567 on test dataset.

Netflix Recommendation System

Key Ideas

- The dataset can viewed as very large and sparse matrix ***R***, having row for each user and column for each movie.

Movies

Users	1	2	3	4	5	6	7	8
1		5		2	4			
2	4		3	1			3	
3		5	4		5		4	
4						1	1	2
5	3		?		?	3		
6		?	2		4		?	

- The value in a cell represents the rating given by a user for a movie. Empty cell means either user has not yet watched the movie or not given the rating for it.

Netflix Recommendation System

Collaborative filtering

- Collaborative filtering finds the hidden structures in the pattern of data in ***R*** and leverages them to predict ratings. There are two methods for collaborative filtering
    - Neighborhood Model - In this method the goal is to find similarity between either users or between movies. The items having more similarity are considered as neighbors.
    - Latent-factor model - In this method the aim is to find hidden key factors(latent factors) that represent the user interaction with movie. This is done by factorizing ***R*** in two short vectors of latent factors.

Netflix Recommendation System

Baseline Predictor

- The baseline predictor just uses global data knowledge to recommend movies . It considers average of all the ratings r and two biases bu - factor by which a particular user's rating behaviour differs then the average rating and bi - the quality of movie i as compared to the global average of ratings.

- The ratings can be predicted as $r^{-} = r + b_u + b_i$

- We could have used $b_u = \sum_u \frac{r_{ui}}{M_u}$ where M_u is number of movies rated by user *u* and $b_i = \sum_u \frac{r_{ui}}{M_i}$ where M_i is the number of users rated the movie *i*. But these averages might not decrease the RMSE of the model.

- To improve the accuracy, we choose $\{b_u, b_i\}$ as model parameter so that RMSE can be reduced during training mode. The cost function then would be

$$min_{(b_u,b_i)} \sum_{u,i} (r_{ui} - r_{ui}^{-})^2$$

Netflix Recommendation System

Neighborhood Model

- The baseline model can be improved by using collaborative filtering. The aim is to find *cosine* similarity co-efficient among the movies and use it while predicting the ratings.

- Each pair of movies *i, j* - the column data is considered as a vector in Euclidean space and find the angle between those vectors. Here the users which gave ratings to both the movies are taken into consideration. The cosine similarity co-efficient can be calculated as under

$$d_{ij} = \frac{\sum_u r_{\bar{ui}} r_{\bar{uj}}}{\sqrt{\sum_u r_{\bar{ui}}^2 r_{\bar{uj}}^2}}$$

- Then consider nearest(most similar) L_i neighbors only while predicting a rating for movie *i* for a particular user *u* who has not rated it. Now we can include collaborative filtering into the baseline predictor as

$$r_{ui}^N = (r + b_u + b_i) + \frac{\sum_{j \in L_i} d_{ij} r_{\bar{uj}}}{\sum_{j \in L_i} |d_{ij}|}$$

Netflix Recommendation System

Incorporating temporal dynamics

- The main idea that helped to reduce the RMSE by 10% was about including the temporal dynamics in the model. The winning teams considered the fact that the popularity of a movie changes with the time and also the users taste in the movie and rating standards also changes. With this the biases $\boldsymbol{b_u}$ and $\boldsymbol{b_i}$ becomes time dependent.

- To incorporate this fact in the model, they divided the days in five years of dataset into 30 bins each about ten weeks long. So a change in movie trend will be represented as $b_{i,bin(t)}$ and change in user's taste in movie as $\sigma_u(t)$ and rating standards as $b_{u,t}$. After this above two biases are represented as

$$b_i(t) = b_i + b_{i,bin(t)}$$

$$b_u(t) = b_u + \sigma_u(t) + b_{u,t}$$

- This introduced 90 new parameters in the model one each for bin of time and particular time dependent variables.