## 2.0 Introduction

The recommendation system is the product of the rapid development of the Internet (especially the mobile Internet). With the rapid development of today's technology, the amount of data is increasing day by day. People feel more and more helpless in the face of massive data. It is to solve the problem of information overload that people have proposed a recommendation system. The recommendation system is essentially a technical means for users to find the information they are interested in from the massive amount of information when the user needs are not clear. The recommendation system combines the user's information (region, age, gender, etc.), the user's historical behavior, the user's interest preferences, and the user's past behavior on the item (whether to buy, whether to click, whether to play, etc.), using a recommendation algorithm to generate A list of items that users may be interested in, providing users with accurate personalized recommendations.

With the progress of society and the improvement of material living conditions, people no longer have to worry about survival, so people have more and more needs for non-survival needs, such as reading books, watching movies, video games, etc., and these non-survival needs are often In many cases it is uncertain, unconscious, and I don’t know what I need. The need for survival is very strong and obvious to people. For example, when a person is dying of thirst, the person's first need must be water. Unlike survival needs, people are actually more willing to accept passively recommended good items in the face of non-survival needs, such as recommending a video game to the user. If they meet the user's taste, the user may like it very much.

Today's video game industry is developing rapidly, the amount of video game information on the Internet is quite large, and people's demand for obtaining video game information of interest is increasing. The management and operation of a large number of video games becomes more and more complicated as the amount of data grows. Therefore, how to integrate the advantages of various algorithms can provide users with reliable video game recommendation results and ensure that users can access the correct Recommendation data has become an important issue to be solved in the recommendation system design.

## 3.0 Data Collection

The datasets for a commercial web application have almost never been exposed for public usage or research, since they are private and strategic assets for the business. Normally they are generated through

Fortunately, there is an existing web source regarding recommendation engine study for us to play with, which is Amazon product data prepared by Associate Professor Julian McAuley, UCSD at http://jmcauley.ucsd.edu/data/amazon/.

Typically, a user activates for a commercial web app could be concluded as viewing a specific web page, purchasing products, comments, and rank items, content, or news. It is quite difficult to summarise a standard data structure for a recommendation engine due to the variety of demands and requirements for various business activities.

However, a classic data structure format has been widely used after long term experimentation and operation with major internet giant companies such as Amazon and Facebook:

- \*\*\*user id\*\*\* : unique user id

- \*\*\*item id\*\*\* : unique item id

- \*\*\*behaviour type\*\*\* : type of behaviour , i.e. purchase or view an item

- \*\*\*context\*\*\* : behaviour context, including location and time etc.

- \*\*\*behaviour weight\*\*\* : weight could be the viewing length for a video or rank for an item

- \*\*\*behaviour content\*\*\* : if a user comments something, the content could be saved as a text file. If user click an item, the content could be a binary input.

### 3.1 Web API

Amazon Instant Video review data ,

http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews\_Amazon\_Instant\_Video\_5.json.gz ,

has been selected as the dataset for this project.

### 3.2 Web Crawling

Nine feature values are stored in this data set, and there is only one item id for product-related information. We need to obtain more product information through the technical means of web crawlers.

According to the item id, we can see the details page of the item on the Amazon official website so that we can crawl to the corresponding product name. Amazon has a powerful anti-crawler mechanism to prevent users from maliciously accessing websites. So it is not easy to get all the data in this project. This project only gets the name information on some items.

In order to improve the page loading speed and improve the user experience, Amazon uses dynamic loading JavaScript to insert HTML DOM elements. It makes capture information more difficult. The way we adopt is through the item name obtained on Amazon, and then through the third-party platform to obtain more relevant information based on the item name. From reaching the relative integrity of the product's information data.

## 4.0 EDA

## 5.0 Algorithm Research

There are many approaches to implement recommender engines, such as Collaborative filtering, Content-based filtering, Multi-criteria recommender systems, etc. Their brief concept and implementations could be referring to as <https://en.wikipedia.org/wiki/Recommender_system>.

### 5.1 Input Data Structure

The input data structure is straightforward. It is a big User-Item with rating matrix that is represented as a specific user ranks an item with a certain rating as per the code below:

### 5.2 Performance measures

The commonly used metrics are the \*\*\*Mean Absolute Error(MAE)\*\*\* and \*\*\*Root Mean Squared Error(RMSE)\*\*\*, \*\*\*precision\*\*\* and \*\*\*recall\*\*\* will also be used to evaluate the quality of a model for comparison.

However, these classic evaluation measures are highly criticized. It has been seen that results of offline measures have a low correlation with results from user activities or online tests (A/B tests). It is probably because the online recommender engine is a static model as the user behavior being updated continuously.

In this project, we would just use classic evaluation measures to compare different machine learning models since we do not have a real online environment to implement A/B tests.

### 6.1 SVD

The famous SVD algorithm, which was popularized by Simon Funk during the Netflix Prize in 2006. Its documentation and reference could be referred to links as below:

https://sifter.org/~simon/journal/20061211.html

https://surprise.readthedocs.io/en/stable/matrix\_factorization.html#surprise.prediction\_algorithms.matrix\_factorization.SVD

SVD stands for singular value decomposition mathematically, which is a factorization of a real or complex matrix that generalizes the eigendecomposition of a normal square matrix to any m × n matrix via an extension of the polar decomposition.

Actually, it is dry and headache to focus on the mathematical details for this algorithm, \*\*\*SurPRISE\*\*\* lib has encapsulated ready to use functions to implement this approach.

The code below has followed a standard machine learning process, which has iterated through all combinations of parameters in \*\*\*param\_grid\*\*\* variable with K folds method (3 folds), so as to find the best prediction model based on RMSE and MAE score.

\*\*\*{'n\_epochs': 15, 'lr\_all': 0.01, 'reg\_all': 0.2}\*\*\* parameters have been found as the best score model for this approach

### 6.2 SVD++

The SVD++ algorithm, an extension of SVD taking into account implicit ratings. When a user rated a product, it means that he has used the product. Such behavior contains specific information. So we can understand the problem as follows: the behavior of the rating reflects the user's preferences from the side, Such a reflection can be reflected in the model in the form of implicit parameters, thereby obtaining a more refined model, which is SVD++. Study shows that social popularity based SVD++ Recommender System may be such a technology in most of the cases.

<https://research.ijcaonline.org/volume87/number14/pxc3894033.pdf>

### 6.3 NMF

Non-negative matrix factorization algorithm is very similar to SVD. The application of non-negative matrix factorization is more and more extensive, and the scope of application includes text dimensionality reduction, topic extraction, image processing, and so on. By limiting the matrix decomposition (cannot be a negative matrix. It leads to the classic matrix decomposition method in the recommendation system that can achieve excellent prediction performance. However, It can not make recommendation interpretations that are consistent with people's habits like User-based Collaborative Filtering ( (Even people with similar tastes also bought this product).

http://papers.nips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.pdf

## 7.0 Similarity Modules

### 7.1 Cosine

Since there are not many parameters to be tested for a similarity module, we can simply set up a model and cross-validate them with K folds. The python code below computes the RMSE and MAE result plus their fir and test running time.

As RMSE and MAE results are similar for 5 folds, we can specify that there is not much skewness for the dataset.

The documentation for Cosine Similarity could be referred to as <https://surprise.readthedocs.io/en/stable/similarities.html>.

## 8.0 Top N Recommender

### 8.1 Get Top N Items

This approach is based on the top-10 items with the highest rating prediction for each user in the dataset. The input could be user rating to different items (i.e. 20 items). Then, it will return the top 10 best prediction items.

Below is a simple example showing the input and output for the top 10 items with ID and prediction value recommended for a user based on its prediction model