Machine Learning Engineer Nanodegree Capstone Project

Author: Mario Perales Pina

Domain Background

Recommender Systems

Recommend products is a key challenge in different kind of business .E-commerce and retail companies are leveraging the power of data and boosting sales by implementing recommender systems on their websites. Nowadays, Recommender Systems are increasingly used in a lot of well-known webs and it's a great opportunity to explore this technique deeply.

Recommender systems can be also applied in a multiple uses cases: retail cases, e-commerce, bank products. Improving forecast quality is of great advantage to make accurate business decisions or provide to the client a new and personalized product or service for them

State of the art

How recommender system works

Recommender systems function with two kinds of information:

- Characteristic information. This kind of recommender systemns uses information about items (keywords, categories, etc.) and users (preferences, profiles, etc.).
- User-item interactions. Ratings, number of purchases, likes, etc. given by a user for the items

Based on this, we can distinguish between three algorithms used in recommender systems:

- Content-based systems, which use characteristic information.
- Collaborative filtering systems, which are based on user-item interactions.
- Hybrid systems, which combine both types of information with the aim of avoiding problems that are generated when working with just one kind.

Next, we will dig a little deeper into content-based and collaborative filtering systems and see how they are different.

In [1]:
```
### Talk here about Content based and collaborative filtering
```

Personal Motivation

Lately, I have started working for a job where we use recommender systems. Since, this project has already started a couple of years ago, I did not have the opportunity to develop the algorithm and my tasks are defined in other Machine Learning related problems.

Therefore, I have recently decided to do this Capstone Project with a related problem developing a Recommender System project.

Problem Statement

Nowadays, there are a lot of companies that recommend services and products in their websites. Products like movies and online ecommerce website are offering products with a recommender system algorithm in backend.

The problem here to solve is to recommend books using Goodreads database. In this website reviews and ratings are available there for the books in their database.

Datasets and Inputs

As I mentioned previously, we are going to use a Goodreads database from this Kaggle Database available in this link

This database contains information about the books and the interactions between users and those books.

In this database there are 1636235 distinct books. Books dataset has information about the pages in the books, the ratings that the books have received, the names and the author. In the interactions database, we have 362596 interactions with 103533 distinct books and 8919 distinct users.

Algorithm benchmark

Algorithm 1 - Matrix Factorization:

A matrix factorization is a way of reducing a matrix into its constituent parts.

It is an approach that can simplify more complex matrix operations that can be performed on the decomposed matrix rather than on the original matrix itself.

An analogy is the factoring of numbers as we studied in elementary school, such as the factoring of 30 into 2 x 3 x 5. However, there are many ways to decompose a matrix, hence there are a range of different matrix decomposition techniques.

One of the most used methods in Collaborating Filtering using Matrix Factorization technique is Singular value decomposition (SVD) algorithm. The aim is to provide users with books' recommendation from the latent features of item-user matrices. Here the original matrix A have dimensions number of users (N_u) times number of items (N_i) (N_u x N_i). A is factorized in two matrices; the first one, U, will have N_u x M dimensions and the second matrix will, V, will have N_i x M being M a parameter of the algorithm.

The final matrices will be $A = U V^T$.

Finally, we can see U as a matrix with users' features and V a matrix with items' features. To build these two matrices we use the registers we have in A, since this matrix will usually be very sparse (a user only reads/buys/uses a few products) and U and V are build iteratively to minimize the error when these two matrices are multiplied.

In this first algorithm, we use A simply as the user-item interaction.

Algorithm 2 - Combined matrix with Matrix Factorization (possible extension of the project if there is enough time left):*:

For this algorithm, We want to introduce bias for A adding the author information. To do that I am going to compute this

1. Let us call A to the matrix with user_id, item_id and ratings matrix. A (N_u x N_i)
2. And A_2, to the matrix with user_id, author_id and ratings matrix. This matrix will have the same dimension as A, there will be duplicates with user_id and author_id because we want to sum the same values. This is going to be computed with an Window function to maintain the same elements.
3. A' will be the weighted sum of A and A_2. A' = k A + *(1 - k)* A_2 for k in [0,1]

A' will be the weights used for training, but the error will be computed with A.

Algorithm 3 - Softmax model (possible extension of the project if there is enough time left):*

Here we want to make things a little bit hard, making a artificial neuronal network to make predictions and a part with user embeddings based on the movies a user rates. This method will be expensive since we have a N_i neurons output layer and in our case, we have 103533 distinct books, which is almost infeasible here. Maybe in the industry with available resources will be possible process that amount of information

Evaluation metrics

I will use the RMSE for the predicted rating compared to the real rating using a 5 fold cross validation. Then I will be able to compare the algorithms proposed

Project Design

- Step 1 (Data processing) - Preprocess the data. Reading all the users and items archives and joining them. For the second algorithm several joins and data processing techniques will be neccesary.
- Step 2 (Data modeling) - AWS SageMaker to train the models, since they are scikit based model, on the subsets of the goodreads data and get forecast estimations
- Step 3 (Inference) - Evaluate the forecasts using the outlined accuracy measures.

References

- Database. https://www.kaggle.com/bahramjannesarr/goodreads-book-datasets-10m
- Sheng Zhang, Weihong Wang, J. Ford, F. Makedon and J. Pearlman, "Using singular value decomposition approximation for collaborative filtering," Seventh IEEE International Conference on E-Commerce Technology (CEC'05), 2005, pp. 257-264, doi: 10.1109/ICECT.2005.102.
- Hug, N., (2020). Surprise: A Python library for recommender systems. Journal of Open Source Software, 5(52), 2174. https://doi.org/10.21105/joss.02174