Advanced Machine Learning Final Project Report

Manthan Shah nalostta@vt.edu

Mrigank Dhingra dmrigank@vt.edu

Pratijay Guha pratijayguha@vt.edu

Abstract

The project involves building a recommendation model for the clothing brand H&M. While there are numerous open-source resources on models which recommended products like movies using data based on user-ratings, our recommendation model uses purchase-data habits to better push relevant products and improve sales. This poses a challenge as the current rating models do not directly translate to using purchase data. We also explored the various algorithms and parameters to take into account considering the compute-resources and the type of data available to us. In this project we have used a baseline, ALS, BST and retrieval model for recommendations.

1 Introduction

There has been a continuous trend in the past decade with respect to product consumption. Video on-demand and online shopping has seen huge jump ever since the onset of Covid-19. More and more people prefer the convenience of online shopping and virtual experience in their fast-paced lives. Ever since Netflix pivoted to a subscription-based video on demand model, its revenue has grown from 1.36 billion to around 25 billion in a span of 12 years. Netflix has over the years spent heavily researching to push movies relevant movies to its customers. This in essence improves the overall user experience as the more relevant the recommendation is, more likely is the user going to enjoy. Similar such models are being heavily employed in sectors such as e-commerce and retail where relevance of the pushed product improves the chances of sale, thus improving revenue.

To better understand such practices, our project implements various Machine-Learning based recommender-models that recommend clothing-articles and other products from the HM retail company. These models are built using data that ranges from past transactions, article features, pricing as well as customer demographics over a span of years find products that the customer is most likely to buy within a week of the last purchase date.

1.1 Problem Statement

H&M Group is a family of brands and businesses with 53 online markets and approximately 4,850 stores. The online store offers shoppers an extensive selection of products to browse through. But with too many choices, customers might not quickly find what interests them or what they are looking for, and ultimately, they might not make a purchase.

To enhance the shopping experience, product recommendations are key. More importantly, helping customers make the right choices also has a positive implication for sustainability, as it reduces returns, and thereby minimizes emissions from transportation. In this project, the aim is to develop product recommendations based on data from previous transactions, as well as from customer and product meta data. The available meta data spans from simple data, such as garment type and customer age, to text data from product descriptions to image data from garment images.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

1.2 Data

The data obtained from Kaggle consists of the purchase history of customers across time, along with supporting metadata. Specifically, there are 3 files in the 'csv' format focused on the 3 aspects of data:

- **Customers**: This file had information associated to a customer like customer-id, age, zip-location, membership status etc. To maintain privacy, any critical information which can used to identify any individual like name, data-of-birth etc. is removed, thus maintaining anonymity. Each customer is associated with a unique customer id which is used as an index to identify respective transactions.
- **Transactions**: This dataset consists of the date, article-id, and the customer-id of the corresponding transaction. The transactions provided happen over a period of approx. 2-year, ranging from 2018 to September 2020. Duplicate rows correspond to multiple purchases of the same item. This set forms the primary training data-set.
- **Articles**: This dataset covers the features and meta data of each article in the HM store, covering features like color, perceived color, garment type, graphical appearance. It also has a text based description.

2 Methodology

We introduce a few terms here that will be very important going forward. One of the terms is co-clustering which is a term in data mining that relates to the simultaneous clustering of rows and columns of a matrix. Co-clustering can be seen as a method of co-grouping two types of objects based on similarity of pairwise interactions (1).

2.1 Baseline

The baseline in this case was a primitive implementation of a co-clustering algorithm based on the discussions in the Kaggle Community. We firstly created a dictionary of top k items that were bought after item i. This dictionary was mapped back to the last purchase of a user to get k recommendations. For customers that did not have any prior purchases, we recommend the top 12 articles from last week. While this is a great method for a baseline, the data size makes it difficult to compute these on the go. However, the overall computation requirement in this scenario is much lower than Deep or other ML models.

2.2 Matrix Factorization

One of the most popular algorithms to solve co-clustering problems (and specifically for collaborative recommender systems) is called Matrix Factorization (MF). In its simplest form, it assumes a matrix $R \in R^{(m \times n)}$ of ratings given by musers to nitems. Applying this technique on R will end up factorizing R into two matrices $U \in R^{(m \times k)}$ and $P \in R^{(n \times k)}$ such that $R = U \times P$ (their multiplication approximates R) (1).

While Matrix Factorization is well suited for use in recommendations, especially given that we can scale it up since it uses sparse matrices, it is still faced with the inability to add external customer and item features and instead rely on the ALS algorithm to generate user and item embeddings.

2.3 Deep Neural Network Retrieval Model

The Deep Neural Network Retrieval model, just like the method of ALS, understands and learns the interactions between the customers and articles while taking several article and customer features into the account. The main difference between the approach of ALS and the Retrieval model is the fact that the Retrieval model employs string lookup and embedding layers. There exist 2 stages in a very basic Retrieval based model:

- **Retrieval**: The retrieval stage is mainly responsible for selecting a few hundreds or even thousands of articles for each customer out of the entire articles dataset. It does it eliminating all the articles that a particular customer is not interested in

- **Ranking**: Then the remaining articles are ranked based on the likelihood that the customer would be interested to purchase that article.

When compared with the BST model, it is easier to make changes the model like adding further embedding layers without affecting the rest of the model.It has an extremely strict data input requirements and would not function properly if the data has not been pre-processed according. When compared with the method of ALS, the Retrieval model is more complex and difficult to understand.

2.4 Behavior Sequence Transformer (BST)

Behavior Sequence Transformer is an attention-based transformer which leverages off the time-series nature of our dataset. One of the main reasons for experimenting with another deep neural network model was the fact that the Retrieval model didn't consider the sequence in which each customer makes their purchases over a period. It makes use of the provided customer and article features and a score/rating parameter to generate ratings for each article for each customer. The topmost rated articles are recommended for each customer. Our model is a sequence-to-vector implementation which accepts the following as its inputs:

- A fixed-length sequence of article_ids purchased by a customer.
- A fixed-length sequence of ratings, which we defined as how frequently an article was purchased, for each article bought by a customer.
- Customer feature which we decided to include was the age of the customers when they made the purchases. Other customer features such as membership status and postal code could also be considered.
- Article feature which we decided to include was the Product_group_type. Other features such as article appearances and color could also be considered.
- A target article_id for which to predict the rating.

With post-processing, we were able to fetch the top 12 rated articles for each customer. Takes the time-series nature of the dataset into the account which a traditional deep neural network cannot. It is easy to add other customer and article features for a better model training. It is a resource demanding algorithm when compared with something like the method of ALS for generating recommendations. The data must be pre-processed in a particular manner in order for the model to generate article purchase sequences. It has a complex model architecture, and it is quite difficult to make changes to the layers without affecting the rest of the model.

2.5 Complement Chart Approach

This is novel approach was designed based on feature-engineering of the articles. CC is a 2D symmetric matrix of size equal to the number of features of a particular feature type (example=color, garment type).

The approach follows the philosophy of recording 'complements', which is essentially an article that goes well together with another article to complete a set or an 'attire'. This is done by obtaining the frequency of articles that are bought together within a purchase window. We then generate complement charts of the features of these complementing articles. This process is synonymous to training a model.

Once CC are obtained, we look at the features of articles purchased by a customer and use it to select relevant rows from the complement chart. By adding these, we get feature scores which signify the measure of 'desirability' of the features based on the user's purchases. We can obtain the final score of an article based on a weighted sum of the feature scores for a customer.

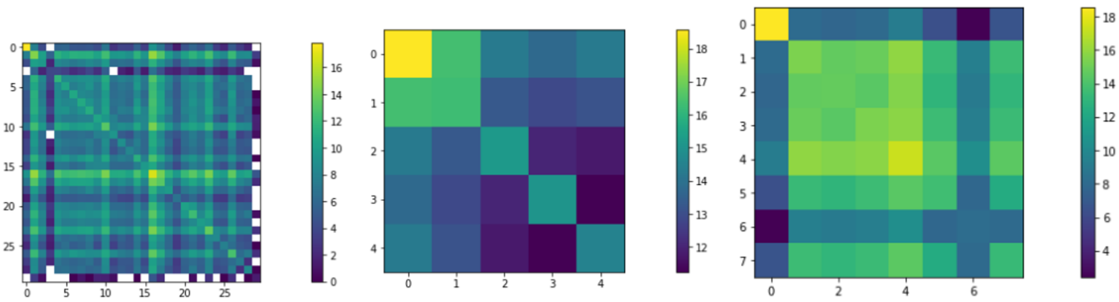

Figure 1: complement charts for 3 feature types :- color,garment type and graphical appearance

3 Results

3.1 Baseline

The baseline model resulted in an eye-popping MAP@12 score of 0.0215. This is pretty good given that the top public score in the Kaggle Competition is 0.0364. Since we do not have many parameters to tweak in this model, we find this score to be satisfactory.

3.2 ALS

The ALS model gave us a best score of 0.0054 which is an order of magnitude lower than the baseline. This can be expected since we were not able to send in external customer and item features into the model. The two parameters that were tunable were the number of iterations and factors. On doing a grid search, the best model was found to be using 100 factors and 1000 iterations.

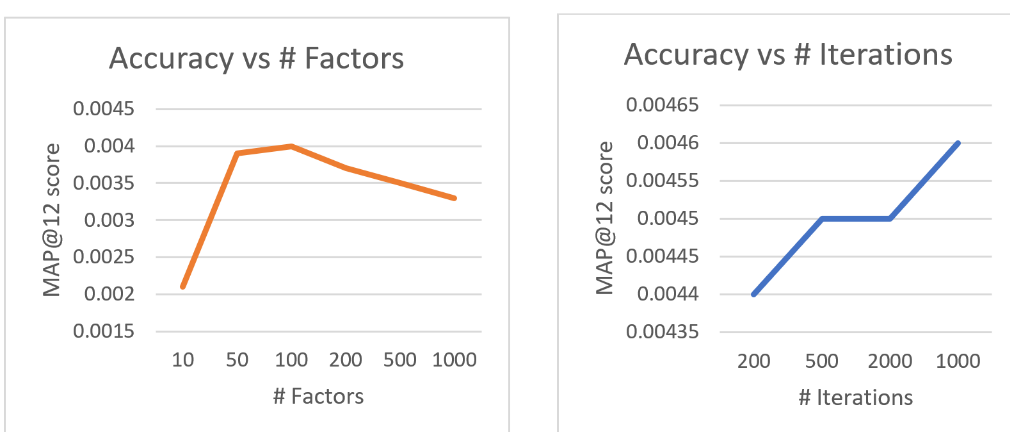

Figure 2: model optimization results

3.3 Behavior Sequence Transformer (BST)

The BST with a single customer and article feature got 0.0199 as its MAP@12 score according to the kaggle's evaluation metric. This still does not beat the baseline score, but addition of several customer and article features would boost the overall score.

3.4 Deep Neural Network Retrieval Model

The retrieval model scored 0.0063 according to the kaggle's MAP@12 evaluation criteria. This score does tell us that it can be further improve if we build interactions between other features and include them in the interactions dictionary. This would also lead to the addition of other embedding layers to the model architecture.

4 Conclusion

Upon observation we found that the performance of all of the above models is strongly correlated to the definition of rich customer interaction. In essence we found that the models based on the approach of frequently purchased together performed much better than the machine learning informed models which were more computational than feature-engineering based. Surprisingly, simpler machine learning informed models ended up outperforming computationally intensive models except for the DNNRM, where the Deep learning-based architecture outperformed traditional matrix factorization techniques. The complement-chart method, being inhouse is a proof of concept. Given more time, we can make this algorithm scalable by using graph database like neo4j.

While the BST model showed high promise, it did not perform as well as expected and this is probably because the number of features in the model were not enough. Due to limited resources and size of data, we were unable to optimize the BST mode. As a further extension to this project, this model can be improved upon by increasing the number of the features,hyper-parameter optimization and more interaction-experimentation. An ensemble method of baseline for new and BST for recurring customers would give us the best results.

5 Contributions

Mathan Shah : EDA, Complement Chart, ALS model implementation, model evaluation.

Pratijay Guha : EDA, baseline model implementation, ALS model implementation.

Mrigank Dhingra : EDA, BST model implementation, Deep Neural Network model.

References

[1] `https://www.kaggle.com/c/h-and-m-personalized-fashion-recommendations`

[2]`https://developers.google.com/machine-learning/recommendation/collaborative/basics`

[3] `https://github.com/microsoft/recommenders`

[4] `https://datasciencemadesimpler.wordpress.com/tag/alternating-least-squares/`

[5] `https://keras.io/examples/structured_data/movielens_recommendations_transformers/`

[6]`https://github.com/fickaz/TFRS-on-Retail-Data/blob/main/src%20a.%20Retrieval.ipynb`