CMPT310 Group 10 Project Report

Brandon Zhang
Department of Computing Science
Simon Fraser University
bza57@sfu.ca

Japneet Singh
Department of Computing Science
Simon Fraser University
jsa395@sfu.ca

Li Zidai
Department of Computing Science
Simon Fraser University
zee_li@sfu.ca

Chang-Hsiu Tsai
Department of Computing Science
Simon Fraser University
cta106@sfu.ca

Perapat Arerassadorn
Department of Computing Science
Simon Fraser University
parerass@sfu.ca

Abstract

Machine learning models were explored to solve the H&M Personalized Fashion Recommendations problem on the Kaggle data science competition website for this class project. Transaction history, customer and article metadata, as well as article image data was provided for this competition, but a simplified dataset based on the available data was used for the purposes of this project. The models tested include Markov Process, Term-Frequency, and a custom algorithm that ranked weekly trends. Ensemble models of these algorithms were also tested. The maximum prediction rate across all models was 13%, achieved by the weekly trends algorithm and all ensemble models. An inverse correlation between model complexity and prediction performance was observed for this dataset, though a different distribution of performance ranking is suspected for larger datasets. Demographic and seasonal factors are highly impactful factors to account for, and a customer's recent purchases can be used to infer immediate or future purchases in recommendation systems.

1 Introduction

H&M offers an extensive choice of online items, and to facilitate the shopping experience, recommendation systems are a crucial technology that allows customers to spend less time browsing for items by presenting to them directly the products they might be interested in. The problem of learning a customer's preferences can be intractable due to the inexhaustible range of factors that may influence a customer's decisions. As such, an apriori model of customer behaviour is impractical, if not impossible. Instead, using data of past purchase behaviour presents an efficient if imperfect model to use for making product predictions. This lends itself well to training unsupervised learning methods that can quickly adapt to nuanced patterns as they happen in real time. Among unsupervised learning techniques, clustering is one of the most powerful models used for recommendation systems. For example, items that are frequently purchased together imply some common feature between them, and a clustering model will allow those other items to be prioritized for recommendation when one of them is purchased. All the models we explore in this project will make use of some form of clustering model.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

2 Experimental Setup

2.1 Data

Data was provided by H&M Group as a part of a Kaggle data science competition. The data includes a history of over 31 million transactions over a 2-year time period, a list of customers with associated metadata, a list of purchasable products with product information, and a database of product images. Due to limited resources, we decided to simplify the dataset to reduce the required computation time and control for confounding factors such as demographic and season. We shortened the transaction history range to dates between 2020-04-24 and 2020-09-22, a period of just under 5 months, and focused our prediction models for customers of age 25 who made the largest number of purchases among all ages within this time period. This left us with a data set of 339233 transactions and 30669 customers. We also excluded customer and product metadata from our training models, focusing solely on purchase patterns. The data was split into training and test sets by taking every customer's last transaction as the test set and all other transactions as the training set. We grouped the training data by customer ID and put them in chronological order. This gave us training data for individual customers' purchase behaviour.

2.2 Prediction Models

We used the following prediction models. Markov process, Exponentially-Weighted Term-Frequency vector product (EXP-TF), and a default prediction based on recent purchase trends. We also combined these in ensemble models to produce consensus predictions. Each model produces up to 12 predictions for a customer and if it contains their corresponding test-set product the prediction is considered correct.

2.2.1 Markov Models

A Markov process takes an input corresponding to a state and will output the most likely states to occur next according to past data. Our input state for this problem are the product Id's of a sequence of transactions, and the output state is the product ID of the next following transaction. Every purchase in a customer's purchase history constitutes an input to output mapping for the Markov process to train on, including an input of 0 previous product purchases. We implemented Markov processes with multiple variations on the parameters of the input. The length of the input sequence was tested from 1 to 20, and the ordering of the purchases was either chronological or lexicographic. The models with lexicographic ordering are meant to eliminate the confounding factor of an arbitrary chronological order in a set of truly correlated items, and the models with chronological ordering is the control.

2.2.2 Exponentially Weighted Term-Frequency

EXP-TF was implemented by constructing a matrix where rows and columns are article IDs. Multiple EXP-TF matrices were constructed for two families of weight decay functions $(1-a)^{i-1}$ (EXP-TF1) and $2-(1+a)^{i-1}$ (EXP-TF2) for various values of decay rate $a \in [0,1]$. The number of preceding transactions weighted m was set to where the exponential weight dropped below a threshold of 0.1. If there were fewer than m previous transactions, dummy product Ids corresponding to the position of the null value were incremented with the positional weight. For each product a_n in a list of customer transactions, and for each product at position a_{n-i} for $i \in [1..m]$ a weight is added to the row corresponding to the product at a_n and column corresponding to the product at a_{n-i}. Predictions were made by taking a customer's last m purchases and selecting the products with the top 12 sums of weights corresponding to each of the m product purchases.

2.2.3 Weekly Trending Purchases

Weekly trend predictions can be used to make predictions for customers who have no purchase records, particularly new customers or customers with a new account. Our trending model calculates weights for an item as its proportion of weekly sales. Weights for each item are calculated for each week of the 5 month period and summed. A larger sum indicates a more popular item. We take the 12 products with the largest sums and use them as predictions for all customers.

Table 1: Model Prediction Results

Model	Correct Predictions	Accuracy Rate
Markov-Chro-1	3966	0.129
Markov-Chro-2	829	0.027
Markov-Chro-3	484	0.015
Markov-Chro-4	401	0.013
Markov-Chro-5	380	0.012
Markov-Chro-6	365	0.012
Markov-Lex-1	3966	0.129
Markov-Lex-2	890	0.029
Markov-Lex-3	530	0.017
Markov-Lex-4	432	0.014
Markov-Lex-5	418	0.013
Markov-Lex-6	396	0.013
EXP-TF1/1.0/1	2993	0.097
EXP-TF1/0.8/2	2843	0.093
EXP-TF1/0.7/2	2848	0.093
EXP-TF1/0.6/3	2480	0.081
EXP-TF1/0.5/4	2002	0.065
EXP-TF1/0.4/5	1658	0.054
EXP-TF1/0.3/7	1137	0.037
EXP-TF2/1.0/1	2993	0.097
EXP-TF2/0.5/2	2831	0.092
EXP-TF2/0.4/2	2828	0.092
EXP-TF2/0.3/3	2448	0.080
EXP-TF2/0.2/4	2011	0.066
EXP-TF2/0.15/5	1674	0.055
Trending	4005	0.131
Trending + Markov-Lex-1	4005	0.131
Trending + EXP-TF1/1.0/1	4005	0.131

Notes: Numbers listed next to EXP-TF models from left to right are weight decay rate and input size.

2.2.4 Ensemble Models

We combined the trending predictions with the best of the Markov and EXP-TF models to generate ensemble models. Predictions for a customer from both models were pooled and a new selection of the 12 most common predictions was chosen.

3 Results

The most correct predictions of 4005 came from the Weekly Trending algorithm and each of the ensemble models. Markov models of input length 1 performed almost as well, but performance worsened with increasing input length. Lexicographically ordered Markov inputs performed consistently better than chronologically ordered models of the same input length, and approached a limit of 362 and 356 correct predictions respectively at higher lengths (not shown in this table). EXP-TF models also exhibited inverse correlation of prediction accuracy with input length but decreased at a slower rate as compared to the Markov models.

4 Discussion

4.1 Markov Models

A unique thing about the Markov process models is they could fail to produce predictions for inputs due to not seeing the pattern before. There were only 737 bad inputs in the length-1 models, but exploded quickly to over 20000 for length-2 models, approaching a limit of approximately 25000 for both. The prediction rate followed a similar pattern of the inverted direction. Considering the training data only consisted of around 339233 transactions and the size of the input space

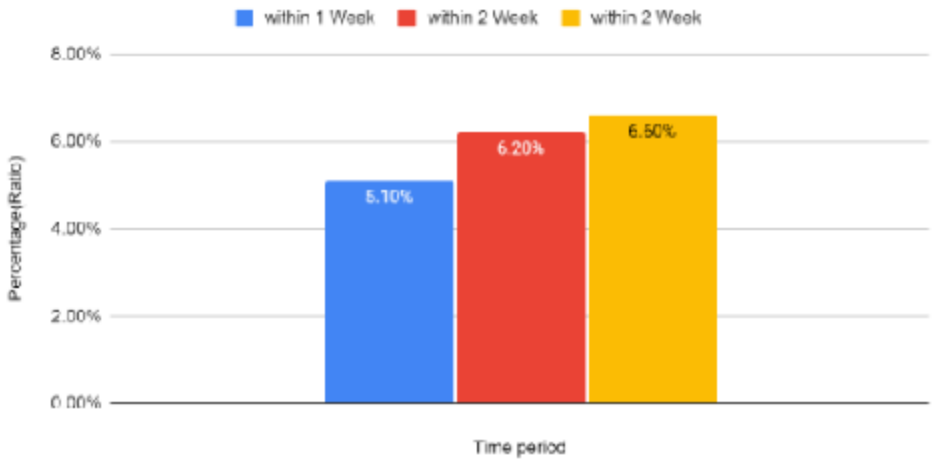

Figure 1: Cumulative percentage of customers who bought the same item again within 1, 2, and 3 weeks.

increases exponentially, this is expected. For the same reason, the lexicographic ordering achieved consistently better prediction rates by reducing the differentiation of inputs by order permutations and consolidating more training data into a smaller set of inputs. However, this is likely not the only factor affecting the prediction performance of the Markov models. With more transaction data, input combinations of the exponentially growing output domain of longer input lengths may be able to accumulate enough instances to produce statistically significant output distributions that have the benefit of the higher input precision. The two contrary forces, a higher accuracy threshold but also diverging statistical significance with increasing input length, gives Markov processes the property that the optimum input size for sequential prediction increases with the amount of available training data. The full transaction history data may have been a good opportunity to test this hypothesis, but it was beyond our time and resource constraints to do so.

4.2 Exponentially Weighted Term Frequency

Unlike the Markov model, EXP-TF does not have the same fragility to input length, yet still exhibits the same pattern of lower prediction accuracy as it increases. What accuracy it maintains is consistent with the proportional influence of the weight of the most recent purchases. The prediction count drop off of -150 is smallest going from length 1 to 2, and then accelerates to -400 up to length 5. There was not much different between each weight decay function, the difference in prediction accuracy was negligible and inconsistent for either one. The difference between them was a decelerating decay vs. an accelerating decay, which are both only slightly deviated from a linear decay with the same input length.

4.3 Weekly Trending Purchases

A result of the way the weekly trending suggestions was calculated is that customers would be recommended products they already bought. One would expect that this would produce ineffective recommendations, but counter-intuitively these predictions performed the best, if only by a small margin. Analysing the data showed that a non-negligible number of customers did in fact repurchase previously purchased items (Figure 1). 5.1% of customers purchased the same item again within a week, 6.2% within two weeks, and 6.6% in three weeks. This would make sense for items that are naturally purchased in multiples like socks or underwear, and less so for larger stylistically relevant items such as shirts or pants. We did not look into which items were actually repurchased but it would have been simple to do so.

4.4 Ensemble Models

Curiously, each ensemble model had the same prediction rate as the Weekly Trending algorithm, despite having significantly different prediction sets for each customer. The similarity between the results hints either that the reduced dataset might not be diverse enough to accurately compare these algorithms' performance at full scale, or the algorithms are performing similar computations and constructing similar models.

4.5 Data Context

As stated before, these results are from a reduced data set that reduced demographic and seasonal factors. The highest publicised score in the Kaggle competition leaderboard for the full data set is 0.0358. Compared to our maximum of 0.131 this suggests that demographic and/or seasonal factors may be a significant factor to account for, and that even a simple disjoint categorization along these parameters might significantly improve the performance of a naive recommendation model.

5 Conclusions

We observed the reliability of a number of heuristics for designing recommendation systems. Our initial intuition to control for demographic and seasonal factors resulted in a surprisingly high prediction accuracy relative to the scores observed in the Kaggle competition leaderboard, suggesting that they are indeed important factors to account for in models. A customer's most recent transactions are on average more predictive than earlier ones, and as always, more data is always better to have.

Contributions

Brandon Zhang: Powerpoint, presentation script, coding, report, live presentation
Japneet Singh: coding, report, live presentation
Zidai Li: Powerpoint, presentation script, report, live presentation
Chang-Hsiu Tsai: presentation script, coding, report, live presentation
Perapat Arerassadorn: Powerpoint, presentation script, report, live presentation

References

Byfone. (2022, February 27). H&M Trending Products Weekly. Kaggle. Retrieved April 21, 2022, from https://www.kaggle.com/code/byfone/h-m-trending-products-weekly

KARPOV, DANIIL. (2022, February 27). H&M Eda First look. Kaggle. Retrieved April 21, 2022, from https://www.kaggle.com/vanguarde/h-m-eda-first-look

Lichtlab. (2022, February 21). Do customers buy the same products again?? Kaggle. Retrieved April 21, 2022, from https://www.kaggle.com/code/lichtlab/do-customers-buy-the-same-products-again/notebook

Lichtlab. (2022, March 8). [0.0226]Byfone&Chris Combination Approach. Kaggle. Retrieved April 21, 2022, from https://www.kaggle.com/code/lichtlab/0-0226-byfone -chris-combination-approach

NGUYEN, TUANANH. (2022, March 20). LSTM/sequential model with item features tutorial. Kaggle. Retrieved April 21, 2022, from https://www.kaggle.com/code/ astrung/lstm-sequential-modelwith-item-features-tutorial#3.-Create-dataset-and-train-model-with-Recbole