Recommendation system for the fashion company

Yejin Kim Sehwa Jeong

Abstract

This study is proposes recommendation for the fashion company. We discuss the recomendation model using RNN units while considering many kinds of features including image, language and categorical data. We explore the possibility of the RNN based recommendation model for fashion company.

1 Introduction

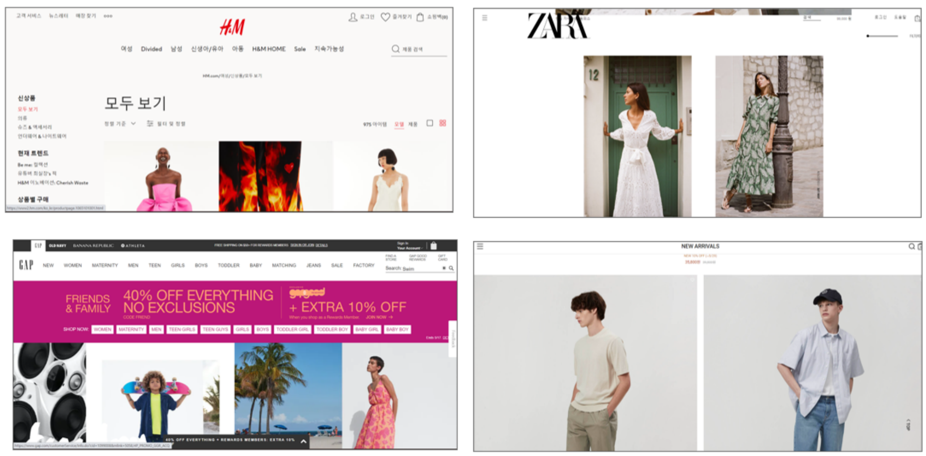
A recommendation system is a system that provides personalized recommendations to users. A person's activity data (e.g. items they have shared) is analyzed to understand the items within it and the relationships between them. This understanding forms the algorithm's personalized recommendations to a user. The notion of a recommendation system is always one of introducing new items to the user, and therefore the algorithm should be constantly learning and improving its understanding of the user's interests.

There are different types of recommendation systems, and the one that is most appropriate for a given situation depends on the type of data being analyzed and the type of recommendations that are needed. The most common types of recommendation systems are content-based, collaborative filtering, and hybrid.

Content-based systems recommend items to a user based on the similarity between the items and the user's past activity. In other words, if a user has shared items that are similar to Item A, the content-based system will recommend Item A to the user. Collaborative filtering systems recommend items to a user based on the similarity between the user and other users. In other words, if User A and User B have shared similar items in the past, and User A has shared Item C, the collaborative filtering system will recommend Item C to User B. Hybrid systems are a combination of content-based and collaborative filtering. They recommend items to a user based on the similarity between the items and the user's past activity, as well as the similarity between the user and other users.

The strength of a recommendation system lies in its ability to accurately understand the user's interests and provide relevant recommendations. The accuracy of the recommendations is constantly being improved as the system learns more about the user. Based on this strength, the recommendation system is widely used in many areas, such as online shopping, music platform, social networking, and streaming service.

Among them, we will focus on fashion companies, especially in e-commerce. There are numerous choices are available so an efficient recommendation system is required in the fashion industry. But compared to other commodities, building effective recommendation systems for the fashion domain is challenging for a number of reasons.



First, it is different from the conventional recommendation system method. A commonly known recommendation system will be a content-based approach and collaborative filtering. These algorithms are based on the consideration that users' past trends will remain the same in the future. This assumption does not fit in the fashion recommendation system. So the frequently praised matrix factorization approaches are not accurate. In addition, we have to use the users' and product features in the model, so it is necessary to use an algorithm that is different from the previous one. Second, the data generated by fashion companies are too diverse. Image data such as clothes photos may exist, and language data such as describing clothes or the reviews of consumers who bought clothes may also exist. Therefore, not only basic recommendation system algorithms but also deep learning techniques like Computer vision or Natural Language Processing may be required. In addition, even if the data are well-established, they will be categorical, so it is hard to handle. Third, users' preferences, product styles, and trends change over time. You know, the word "fashion" means both clothing style and trend. Therefore, this characteristic should be reflected in problem setting and model selection.

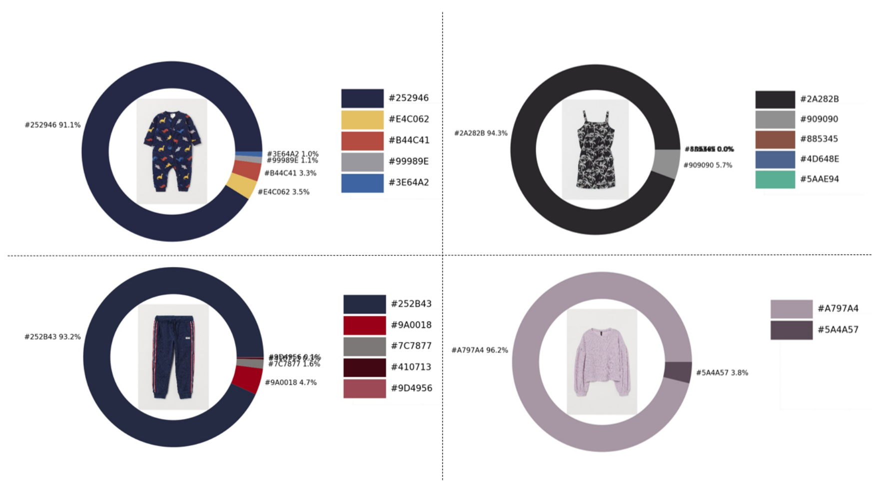
So we set the fashion recommendation problem reflecting these characteristics. The problem is this. predict the items that a customer will buy in the next n-day period. N is the parameter. To solve this problem, we design a new recommendation architecture, and we call it FOR-system(fashion online recommendation system).

2 Model Design and Architecture

2.1 Feature extraction

2.1.1 Image

Image data is extremely important for fashion companies when it comes to developing a recommendation system. By understanding how customers interact with different images of clothing items, fashion companies can develop a better understanding of customer preferences and trends. Additionally, image data can help identify which items are being viewed the most frequently and which items are being ignored. This information is critical for fashion companies in order to make informed decisions about what products to promote and how to target customers with specific ads. Overall, image data is a powerful tool that can help fashion companies improve their recommendation system and better understand their customers. So in this chapter, we describe how to extract important features in image data assuming that fashion companies have them.[1][2]We note two features: Color, Shape. Colors help get people's attention at first sight. So color is very important when a customer

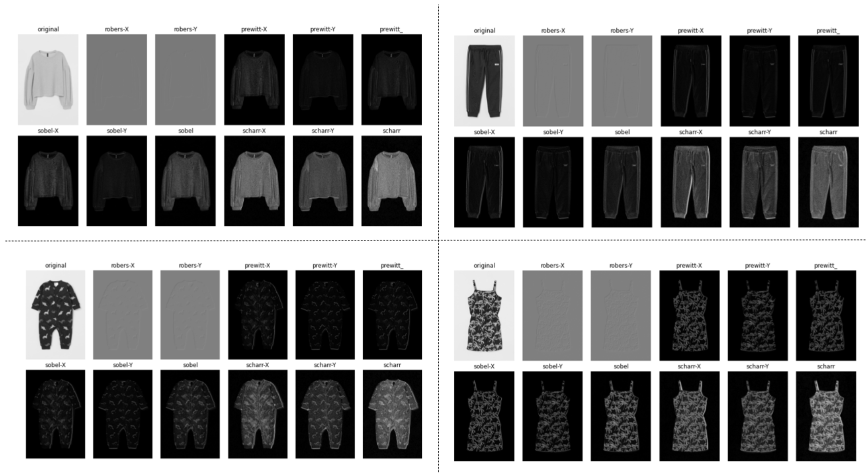


buys clothes. Therefore, it is necessary to extract the color feature from image data. Color feature extraction is a process of extracting certain features from an image based on its color content. This can be done using some kinds of color space transformation to extract the desired features.

Image data is usually represented as a grid of pixels, with each pixel having a specific color. The color of each pixel is typically represented as a combination of red, green, and blue (RGB) values. However, since the RGB value has three values per color, so the data is too wide and difficult to understand at once. So we changed this RGB value to HEX color code and converted it to one value.

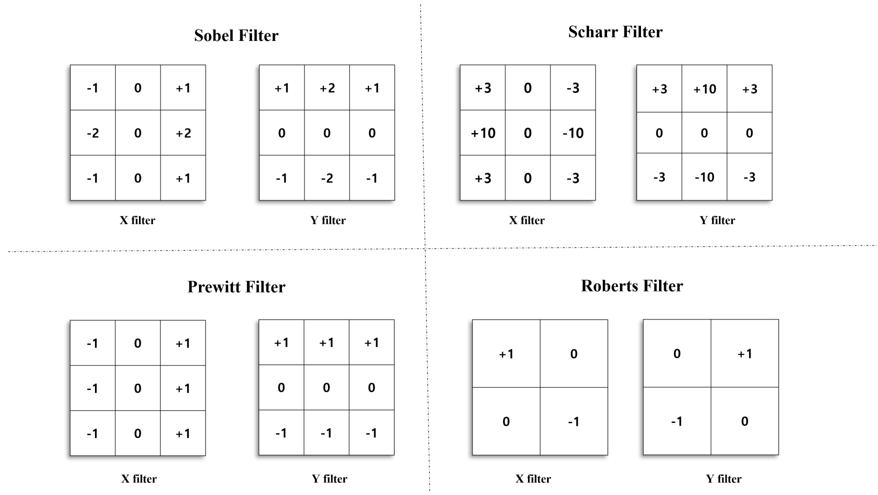
To extract the various color, we have to set up some parameters. One is tolerance and the other is limit. Tolerance represents the group colors to limit the output. It gives a better visual representation by grouping up similar colors. Limits represent the upper limit to the number of extracted colors presented in the output.

When we buy clothes online, the shape of the image data is important. By looking at the image



data, we can get an idea of what the clothing will look like on us. This is especially important when we buy clothes online because we can't try them on before we buy them.

Shape feature extraction from image data is the process of identifying and extracting certain shapes from an image. The process of shape feature extraction can be done by image gradient methods. In computer vision, image gradients are changes in brightness over adjacent pixels. They are useful for edge detection and other segmentation tasks.[3] The gradient of an image is a vector field that points in the direction of the greatest change in brightness. The magnitude of the gradient is the rate of change in brightness. To detect the edge and point of the image, we present first-order derivatives filter of the image. The first-order derivative of an image is a two-dimensional array, with one element for each pixel in the image. There are many types of filters, but we used four types of edge detection filters here: Sobel, Prewitt, Roberts, and Scharr filters. The Sobel filter is the most commonly used of the four filters. It works by convolving the image with two 3x3 kernels. One kernel is used to detect horizontal edges and the other is used to detect vertical edges. The Prewitt filter is very similar to the Sobel filter. It also uses two 3x3 kernels to detect horizontal and vertical edges. However, the Prewitt filter uses different values for its kernels. The Roberts filter is another edge detection filter. It works by convolving the image with two 2x2 kernels. One kernel is used to detect horizontal edges and the other is used to detect vertical edges. The Scharr filter is the most sensitive of the four filters. It uses a 3x3 kernel to detect horizontal edges and a 3x3 kernel to detect vertical edges.



2.1.2 Language

For the item data, there is a column named detail_desc which contains data with natural language format. To use this data, we tried to extract the feature from this text file.[4] We used tools that can extract the keywords form the text file. We used feature extraction package called YAKE(Yet another keyword extractor)[5]. Yake is a model using unsupervised approach that doesn't need a training step. It can extract a keyword in single document without using external document.

This module considers casing, word positional, word Frequency, Word relatedness to context, and word DifSentence. Word Relatedness to Context, computes the number of different terms that occur to the leftand right side of the candidate word. The more the number of different terms that co-occur with the candidate word (on both sides), the more meaningless the candidate word is likely to be. Finally, Word DifSentence quantifies how often a candidate word appears within different sentences. Similar to Word Frequency, Word DifSentence values more those words that often occur in different sentences. And then after running the model, this module give us keywords with ranking as an output.

Using this method to our text data, we can get keywords for each items. We can extract total top n keywords for whole data then we can assign each sentence with keyword. In our study, we used this module with ngram size 1 and window size 1. Then we printed out the Top 1 key words from each row of our article data. The example of keyword extraction is shown below.

Jersey top with narrow shoulder straps → "Jersey"

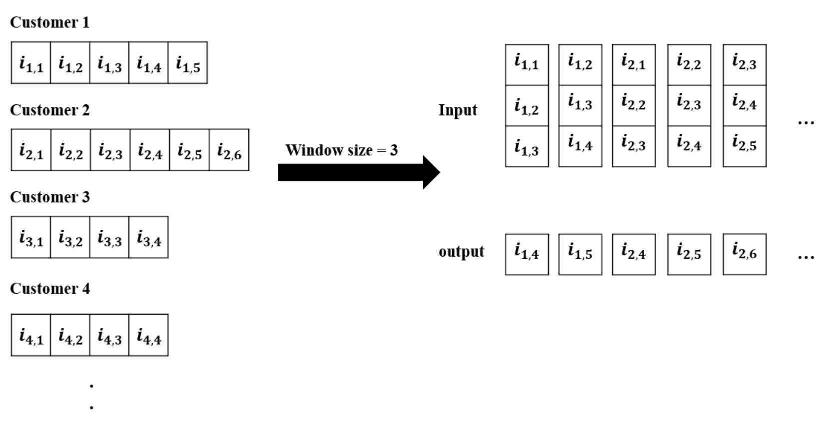
Tights with built-in support to lift the bottom. Black in 30 denier and light amber in 15 denier. → "Tights"

After the keyword extracting process, we could get total 105542 keywords from 105542 articles. This data is used as feature to our recommendation model.

2.2 Sequence mini-batches

A time series is a series of data points, typically measured at regular intervals, that is used to track changes or trends over time. Time series data is often used in statistical analysis and forecasting and it can be used to track changes or trends over time. Si Under this definition, we think that the history of users purchasing clothes was a kind of time-series data and it is very reasonable to use the RNN model in the fashion company recommendation system.

RNN tasks usually use in-sequence mini-batches. So it is common to use a sliding window over the sequence and put these windowed fragments next to each other to form mini-batches. Therefore, we need to change the data to this mini-batch format. First, we create an order for each customer. Then, we set the window size and generate a mini-batch as much as the window size for each user. See Figure for more details.



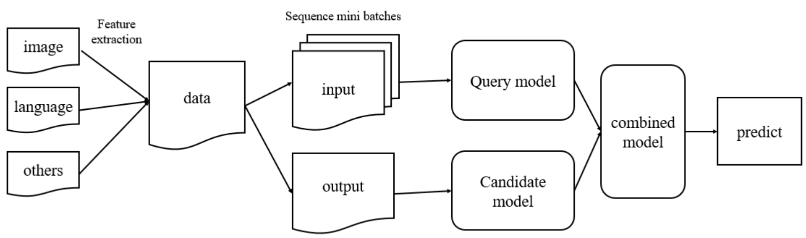
We will frame the RNN based supervised learning problem as predicting the next purchased items in the next purchasing given the previously purchased items and detailed information about them. So we have to transform the data into previously purchased clothes and context features as labels and the next purchased clothes as targets using sequence mini-batches.

2.3 Recommendation system with deep learning

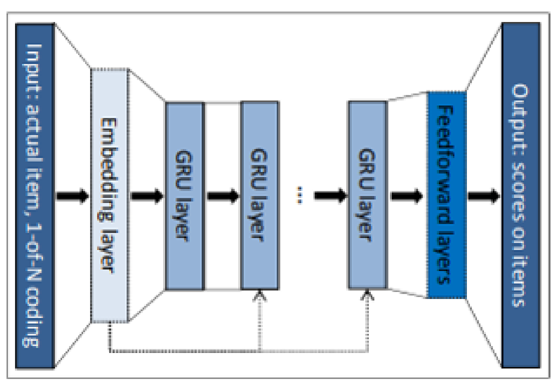
We use two tower architecture for our model. We use query tower and candidate tower. We build each tower separately and combine them in the final model.For query model, we use GRU layer to encode the sequential data, and keep the same candidate tower for the candidate model.

We use FactorizedTopK as metric. It computes metrics for across top K candidates surfaced by a retrieval model.And we use Retrieval task which is a Keras layer that takes the query and candidate embeddings as arguments, and returns the computed loss. The loss we used is Categorical Cross entropy

The full model is similar to the basic retrieval model. It pick out the watch history feature and pass in to the query model. And pick the next label feature then pass into candidate model then gets embeddings back. Then the task computes the loss and metrics.



2.3.1 General architecture of the network



Our model has input-embedding-GRUs-FCL(Fully connected Layer)-score structure. We can use other RNN layers instead of GRU. But in previous study, GRU showed best performance among various RNN layers.[6]

2.4 Evaluation criteria

After training the model, we check the accuracy through top K categorical accuracy. In a top-k categorical accuracy evaluation, the system only needs to correctly recommend the k most popular items to receive a perfect score. This evaluation is often used in recommendation systems, where the goal is to recommend a set of items to a user. The user may be interested in any of the items, so the recommendation system needs to recommend a set of items that contain the user's desired item. The top-k categorical accuracy evaluation metric is a way to measure the accuracy of the recommendation system.

To calculate the top-k categorical accuracy, we first need to calculate the number of items in the test set that are in the top-k recommendations. We then divide this number by the total number of items in the test set. This gives us the accuracy of the recommendation system. For example, suppose we have a test set of 10 items and a recommendation system that makes 5 recommendations. If the recommendation system correctly recommends the 3 most popular items in the test set, then the top-k categorical accuracy is 3/10, or 30

2.5 Comparison with prior models

Many recommendation models are constructed to use implicit data. Implicit data is data that is not explicitly provided by the user. It is data that is gathered through the user's interactions with the system. This data can be used to improve the recommendations that the system provides. Implicit data can be collected in many ways. One way is through the user's buying behavior. The system can track what items the user buys and use this information to improve the recommendations.

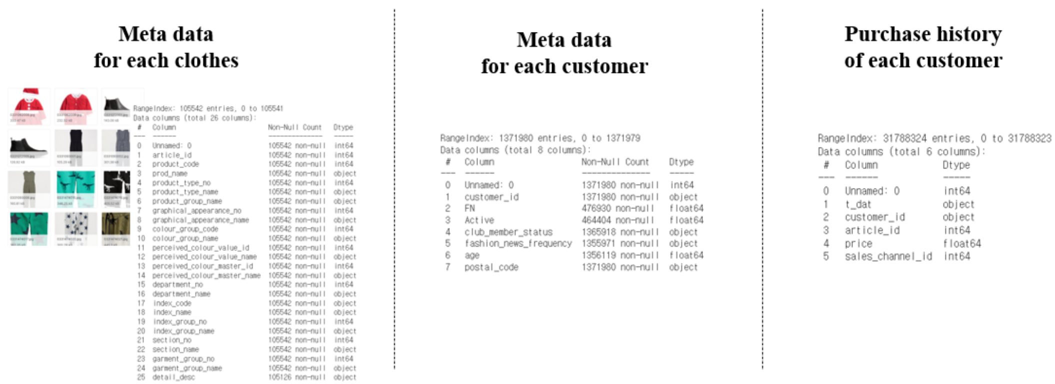
The 'implicit' library is a powerful tool for building recommendation systems for implicit datasets. It provides a fast and efficient way to find related items for a given user. So we used this to apply Alternating Least Squares, Bayesian Personalized Ranking, Logistic Matrix Factorization.

3 Data

H&M is one of the world's largest fashion retailers, with over 4,000 stores in more than 60 countries. H&M's business concept is to offer fashion and quality at the best price in a sustainable way for women, men, teenagers and children. H&M has been ranked the second most valuable global brand in the fashion industry, and is a major fashion company with over 4,000 stores in 62 countries.

H&M's online market is a growing part of its business, accounting for 10% of its sales in 2018. The company has invested heavily in its online presence, with a focus on mobile shopping and personalization. In 2017, H&M launched a new app that allows customers to shop by outfit, rather than by individual items. The app also provides recommendations based on past purchases and allows customers to save items for later.

Recently, H&M released its own dataset to the open-sourced which is called Kaggle. The H&M dataset in Kaggle contains data on over 200,000 items of clothing, including information on price, color, size, and more. Typically, it contains an image and language dataset so we can test it our FOR-online model. This data can be used to understand the trends and preferences of H&M's customers, as well as to track the success of individual items.



In order to measure the accuracy of our model for the fashion company, we test its overall performance using H&M dataset. It allows us to perform experiments on real data and measure the accuracy of the model.

4 Experiments

In this chapter, we illustrate the performance and accuracy of our FOR-system algorithm. The model is implemented in Tensorflow, and Keras frameworks. The experiment is performed on the google Colab platform.

We evaluate the accuracy of the model on the H&M data set and compare the performance of the FOR-system against the existing recommendation model. And we conduct the experiment with and without leveraging features. We also use different RNN-based deep learning algorithms with our FOR-system algorithm and evaluate the performance. Notice that in this experiment, we set the embedding dimension to 128, epochs to 10, window size to 10 and lag size to 1.

We present the table to compare all models with H&M dataset. In this experiment, We can see that the all of the FOR-system models scored higher than the prior recommendation model no matter of the value of k. Also, most of the FOR-system with leveraging features obtains higher score than FOR-system without features. According to this, we can emphasize the importance of leveraging features in fashion companies. Overall, FOR-system with LSTM and feature leveraging obtains the highest top k categorical accuracy in all k as shown in the table.

	Top 1	Top 5	Top 10	Top 50	Top 100
Alternating Least Squares	0.0005232	0.0009522	0.001152	0.002362	0.005122
Bayesian Personalized Ranking	0.0004253	0.0008637	0.0009621	0.002682	0.004687
Logistic Matrix Factorization	0.0007104	0.001256	0.0009735	0.003632	0.006132
FOR-system + Simple RNN	0.001276	0.003533	0.004364	0.009085	0.01389
FOR-system + LSTM	0.0009501	0.001692	0.002078	0.004245	0.006383
FOR-system + GRU	0.0009204	0.001811	0.002197	0.004453	0.006502
FOR-system + Simple RNN + leveraging features	0.0006235	0.002464	0.003325	0.008016	0.01196
FOR-system + LSTM +leveraging features	**0.001633**	**0.003652**	**0.004750**	**0.009501**	**0.01437**
FOR-system + GRU+ leveraging features	0.0008907	0.002256	0.003147	0.007690	0.01143

5 Conclusion

In this paper, we have proposed a new recommendation system for the fashion company. Fashion companies had several challenges to build effective recommendation systems compared to other commodities. But, existing recommendation system algorithms did not reflect these features. So we design a new recommendation architecture which is called FOR-system(Fashion Online Recommendation-system). This new algorithm leverage many kinds of features including image, language, and categorical data. And to reflect the changing trend in the fashion domain, we used the RNN-based model, so we need preprocessing to change data into sequence mini-batches. We use top-k categorical accuracy to compare the performance and FOR-system with LSTM and feature leveraging obtains the highest top-k categorical accuracy in all k.

Through the above processes, we have found that it is more effective to create a recommendation system according to the characteristics of a domain. Since recommendation systems receive less academic attention than other areas, we hope that research on these recommendation systems will continue.

References

[1] Wei-Ta Chen, Wei-Chuan Liu, and Ming-Syan Chen. Adaptive color feature extraction based on image color distributions. *IEEE Transactions on image processing*, 19(8):2005–2016, 2010.

[2] Gaurav Kumar and Pradeep Kumar Bhatia. A detailed review of feature extraction in image processing systems. In *2014 Fourth International Conference on Advanced Computing Communication Technologies*, pages 5–12, 2014.

[3] Zujun J Hou and Guo-Wei Wei. A new approach to edge detection. *Pattern Recognition*, 35(7):1559–1570, 2002.

[4] Mingxi Zhang, Xuemin Li, Shuibo Yue, and Liuqian Yang. An empirical study of textrank for keyword extraction. *IEEE Access*, 8:178849–178858, 2020.

[5] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289, 2020.

[6] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.