

A Stacking Recommender System Based on Contextual Information for Fashion Retails

Heitor Werneck[1], Nicollas Silva[2], Carlos Mito[1], Adriano Pereira[2], Elisa Tuler[1], Diego Dias[1(✉)], and Leonardo Rocha[1]

[1] Universidade Federal de São João del-Rei, São João del Rei, Brazil
{werneck,mito}@aluno.ufsj.edu.br, {etuler,diegodias,lcrocha}@ufsj.edu.br
[2] Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
{ncsilvaa,adrianoc}@dcc.ufmg.br

Abstract. The recent success of distinct e-commerce systems has driven many fashion companies into the online marketplace, allowing consumers to quickly and easily access a worldwide network of brands. However, to succeed in this scenario, it is necessary to provide a tailored, personalized, and reliable fashion shopping experience. Moreover, unfortunately, current solutions on marketing have provided a general approach to push and suggest the most popular or purchased items in most cases. Thus, this paper proposes a new **ensemble recommendation system** based on the stacking of classical approaches associated with contextual information about customers and products. Our idea is to incorporate user preferences and item characteristics to ensure a desirable level of personalization to commonly applied methods. Our method is a Neural Collaborative Filtering algorithm that can combine any recommendation system with contextual domain information. The results are promising, showing significant gains of up to 80% MRR, 70% NDCG, and 108% Hits when compared to popular baselines for this same scenario.

Keywords: Recommendation systems · Stacking · Word embeddings

1 Introduction

E-commerce applications are related to the use of digital communication applied to business. E-commerce is now present in the daily lives of businesses and consumers, affecting and changing how the parties involved in the transactions interact. In all these applications, there is a disproportion between the availability and the ability of people to evaluate the information [20]. This volume of data has led researchers to strive to analyze the information generated by users, modeling a personalized profile according to their preferences. These profiles are used to feed predictive models that aim to match users with available items

Supported by CAPES, CNPq, Finep, Fapesp and Fapemig.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
O. Gervasi et al. (Eds.): ICCSA 2022, LNCS 13375, pp. 560–574, 2022.
https://doi.org/10.1007/978-3-031-10522-7_38

in the domain. Such models are known as Recommendation Systems (RS) [16], whose goal is to identify relevant items according to user preferences, defined throughout interactions. To illustrate the importance of RS, we can mention the profit gains observed in several businesses, for example, 35% increase in sales at a DVD retailer [22] and at eBay, a 6% increase in profit was observed in the implementation of a new recommendation method [6].

More recently, we have observed a scenario in which online marketing has been growing at a swift pace in recent years – the luxury fashion [9]. This growth is associated with the behavioral change of consumers in the traditional fashion industry who now have quick and easy access to a worldwide network to retailers of major luxury fashion brands. It is necessary to provide a tailored, personalized, and reliable fashion shopping experience to succeed in this scenario, which makes the role of RS even more important in the user journey, allowing customers to discover products that suit their style, complement their choices, or challenge them with bold new ideas. Coupled with these issues, the sparsity of data in these applications is even greater. In addition to many existing products common to traditional recommendation scenarios, other characteristics significantly increased the number of possible options, making the scenario even more challenging.

The RS literature on the fashion domain was scarce for a long time compared to the literature in other traditional domains. However, recently we have observed a growing interest in this domain. Some of the existing proposals are adaptations of traditional collaborative filtering strategies that use implicit information from user/item interaction (i.e., clicks and navigation links) [23], or explicit (i.e., rate, purchase) information [10,12,15,17,27,31]. There are proposals that present solutions from other perspectives, such as integrating fashion and human perception themes about custom body shapes and knowledge of professional designers [28] and using 3D body shape information [8]. While these usually present exciting results, they require a set of information that the user is not always comfortable to provide. While all of these proposals present interesting results in particular domains (i.e., datasets), we have not found a method in the literature that can be generalized to different domains. Neither did we find a robust quantitative evaluation, comparing with state-of-the-art methods, to prove the good results.

In this paper, we propose a solution that aims to generalize to different fashion RS by combining different context information: (i) customer-product interaction (i.e., day, week); (ii) items characteristics (e.g., material, color, size, price); (iii) customer behavior (i.e., number of interactions, number of items consumed), with different traditional state-of-the-art RS. Our approach is a *stacking* that combines RS predictions with contextual meta-data. It enables the exploitation of pre-existing user-item interaction modeling using existing methods (e.g., LightGCN) with a context that can recognize new user-item interaction patterns. This work sets out to accomplish:

1. An evaluation of multiple competitive RS in the fashion recommendation domain;

2. A best-fit approach for the fashion domain, exploiting context information along with a state-of-the-art method in RS;
3. To demonstrate the importance of context in the fashion recommendation task;
4. To improve recommendations for different types of users through contextual information.

Our proposal is general enough to allow different RSs to be combined. Specifically, in the present work, we present a first instantiation that combines a collaborative filtering neural network method, a classical non-personalized method, and the domain context information. In our experimental evaluation, considering two Amazon data collections, the results showed significant gains of up to 80% MRR, 70% NDCG, and 108% Hits when compared to methods considered state-of-the-art for the fashion recommendation scenario.

The remainder of the paper is organized as follows. Section 2 describes some basic concepts related to RSs, the main works in the fashion domain, and the main associated challenges. In Sect. 3, we detail our proposal. In Sect. 4, we discuss the instantiation of the proposal and the achieved results compared to several existing proposals in the literature. Finally, Sect. 5 presents our conclusions and directions for future work.

2 Related Works

2.1 Recommendation Systems

Recommendation Systems (RSs) can be broadly divided into two classes, namely, personalized and non-personalized [3]. Non-customized techniques are a bit simpler, as they recommend items not based on the user's taste but rather on user preferences in general (i.e., most popular, highest rated, among others). On the other hand, personalized techniques can be considered more complex than non-personalized ones, as they need to extract from the users' information possible tastes and use them to recommend items that are more similar to the user's profile. Such information can be obtained through user feedback, a click on a website or a review of an item bought/watched on the internet. As we will discuss, our proposal allows combining approaches from both classes.

Regarding personalized RSs, the most traditional approaches are content-based filtering [25] and collaborative filtering [1]. In content-based strategies, the user will receive recommendations of items similar to those he has consumed in the past, i.e., the system will recommend the items whose attributes most closely resemble the items consumed by the user. On the other hand, in strategies based on collaborative filtering, users with similar preferences in the past tend to have similar preferences in the future. The system not only uses the items consumed by a user in the past but also acts by matching his preferences with the preferences of similar users. This approach ignores the notion of actions having a context [2] and that users interact with the system in a particular context, and

preferences related to items may be dependent on the context the user is in, a prevalent aspect in the fashion domain.

Another set of personalized RSs is the context-based ones that use various types of contextual information to make recommendations. Contextual information can be [11]: spatial (those contexts that define the geographic state or environment of users, or items), static (those contexts that do not change over time and affect the recommendation, such as age and gender), and temporal (those contexts that change temporally and are dynamic, such as social relationships). These RSs are very pertinent to the fashion recommendation domain since, for example, clothing can have seasons and can be recommended depending on the season and the country of the user. Contextual information can improve the effectiveness of the recommendation process and, in this domain, is often essential. These RSs proposals are the most suitable to be combined in our proposal.

2.2 General Challenges of Recommendation Systems

In the recommendation task, systems can suffer from several problems in domains, such as [18] cold start problem, which happens when a new user or item enters the system, or also in a lighter definition, these same elements have little history of interactions; explainability, which is commonly found to lack this same concept in neural network-based recommender systems; grey sheep, where one user does not agree with any other group (collaborative filtering methods usually suffer from this problem); sparsity, where users usually interact in a system with an extensive catalog of items and the distinction between interactions over user items makes the interaction matrix sparse, which can lead to less accurate recommendations. The fashion recommendation scenario also deals with all these problems. In this paper, we will focus on the problem of data sparsity.

2.3 Fashion Recommendation Systems

Finally, this section presents some of the main works related specifically to fashion RSs. In [4] the authors propose a method of stacking, used in more traditional domains, combining a few attributes (created from interaction data). The proposal is limited to simple models with little ability to combine attributes at a high level. [23] proposes a simple recommendation approach for the fashion domain in which it uses item price, popularity, and how recent the item is to aid in recommendations, being a simple strategy with very little diversification in exploiting attributes of the domain. In [17] the authors propose the insertion of visual attributes into RSs from adaptations to improve the recommendation of fashion items. [30] proposes a fashion compatibility knowledge learning method that incorporates visual compatibility relationships as well as style information. In [7] the authors propose a framework to jointly train feed-forward neural networks with embeddings and a linear model with attribute transformations. While the proposal is interesting, there is no discussion of using other recommendation models as input to the model. Other work explores the recommendation task generically but without the use of additional information beyond the [13,14,29]

interactions. With this, these models suffer from problems in domains where context is essential; however, they still describe an important part of modeling user-item interaction, and new ways of reusing pre-existing methods can be investigated.

Most approaches in the literature are concerned with specific challenges in recommending fashion items and do not address an evaluation with current state-of-the-art methods of traditional recommender systems. Furthermore, generic approaches that integrate context information are complex for hyper-parameter tuning and do not discuss the use of pre-existing systems for information aggregation. Our work goes in another direction and presents itself as an approach with the potential to be applied in practical recommender scenarios.

3 Stacking for Fashion Recommendation Systems

All contextual information is equally important for the fashion recommendation domain and must be considered when proposing solutions. A complete system should consider at least three types of information: (1) those related to the user-item interaction (i.e., day and week of interaction); (2) those related to the items (e.g., material, color, size, and price); and those related to the users (i.e., number of interactions and number of items consumed). In this sense, the main approaches related to fashion recommendation already use this information to model the users' interest in the available items and thus make their recommendations. Some of these approaches rely on non-personalized information, such as the popularity of their products, to achieve more sales. On the other hand, other approaches try to understand users' preferences better and recommend the products they are most interested in. The main hypothesis of this work is that both approaches are equally important for a scenario where the system has to please its users and constantly sell its products. For this reason, our proposal is an RSs stacking that aims to combine the predictions of distinct approaches with existing contextual information about users and items. We propose adopting models based on neural networks since these models can match attributes at multiple levels.

3.1 Our Neural Network

Our proposal consists of a simple and effective approach to combine multiple data instances known as Multilayer Perceptron (MLP). MLP is a class of feed-forward artificial neural networks that consist of at least three layers of nodes: (1) an input layer, (2) a hidden layer, and (3) an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. Its architecture is highly dependent on the problem to be studied, and its results can change sharply according to its variation. In this case, it is essential to perform a hyperparameter search method. Moreover, MLP uses a supervised learning technique called backpropagation. Its multiple layers and nonlinear activation

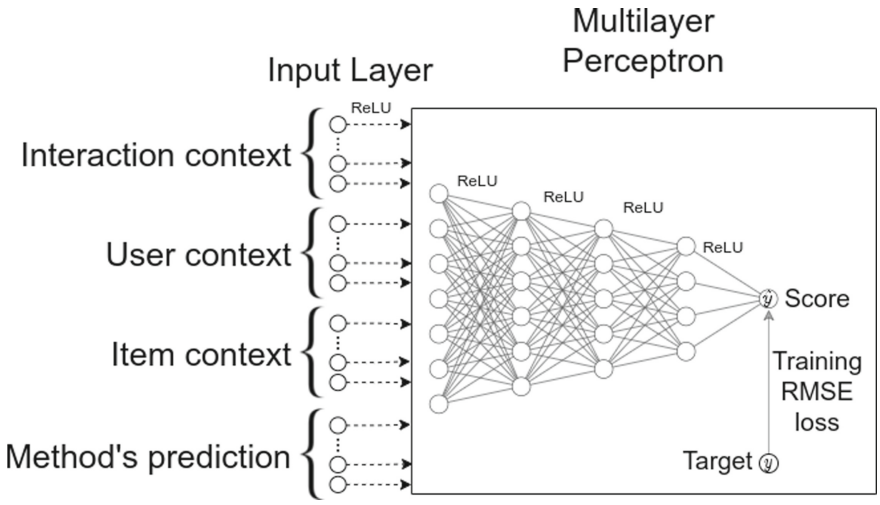

Fig. 1. Stacking created over distinct models and contextual information for fashion.

distinguish MLP from a linear perceptron [26]. Figure 1 depicts our proposal, combining different inputs to our neural network.

MLP has the following attributes as input: predictions from the recommendation models (e.g., SVD and SVD++); data about the user; data about the interaction; and data about the item. To use the contexts more appropriately with MLP, we proposed an initial pre-processing step on the data. This pre-processing transforms categorical attributes into vectors of Boolean attributes (one-hot encoding). For the attributes with many available categories, we propose to use only the 100 most popular ones to avoid the problem of high dimensionality in the recommendations. In terms of recommendation models, our initial proposal consists of combining only two proposals: (i) non-customized model named Popular that ranks the items according to their popularity, defined in this work as the number of different users who have clicked on the item; and (ii) LightGCN [13], a method that learns user and item embeddings by propagating them linearly in the user's interaction graph with the item, using the weighted sum of the learned embeddings across all layers as the final embedding. Thus, the combined information is:

1. Context: consists of the contextual information of users, items, and user-item interaction;
2. LightGCN item: consists of the information obtained from the user-item modeling generated by the LightGCN method (score assigned to each item for each user);
3. Popular item: score assigned by the non-personalized Popular method to each item (independent of the user).

3.2 Implementation Details

We chose two methods. One is a state-of-the-art collaborative filtering approach in the literature, which can model complex representations of the user interaction behavior; and the other is a non-customized method that proves to be very effective in various scenarios, being robust in some cases that are challenging for collaborative filtering methods (i.e., cold-start). Finally, to improve the network's learning, we proposed applying z-score normalization to the scores given by the methods.

In our work, the proposed neural network uses the rectified linear unit function (ReLU), which is defined as $f(x) = max(0, x)$. For training the neural network, the root-mean-square error (Eq. 1) was used as the loss function and the Adaptive Moment Estimation (Adam) [19] as the optimizer in the search for the best parameters. Such a strategy adapts the learning rate for each parameter with small adjustments for frequent parameters and large adjustments for infrequent parameters.

$$L_{RMSE} = \sqrt{\Sigma_{i=1}^{n} \frac{(\hat{y} - y)^2}{n}} \tag{1}$$

4 Experimental Evaluation

This section presents the set of experiments performed to evaluate our proposal. First, we present our experimental setup, describing the datasets and their preprocessing steps performed, the evaluation methodology and metrics used, and the baselines considered in our comparison. Next, we discuss the experimental results. Finally, we present an experiment with different combinations of information and algorithms to evaluate each component's impact on the final result of the proposed stacking method.

4.1 Setup

4.1.1 Datasets

First, we select two classic datasets in the literature with user ratings on Amazon products. These two datasets contain data about the user's interaction with a product (i.e., rating and timestamp) and the item (i.e., color and size).

- **Amazon Fashion.** It is a dataset of user ratings on items in the fashion section of Amazon; and
- **Amazon Clothing.** It is a dataset of user reviews of items in the clothing section (e.g., clothing, shoes, and jewelry). We prepared an initial random sampling of 100,000 users to use the Amazon Clothing dataset due to the large number of interactions.

Without any special processing, these two datasets have a large portion of users with few interactions. This characteristic is not effective for an assertive

quantitative evaluation of RSs. Thus, for a more diverse evaluation, we perform a filtering process in the two datasets, where we removed users with less than **n** interactions from the datasets. For evaluating two important recommendation scenarios, we used $n = 5$ to evaluate the dataset with users with little and much knowledge and $n = 10$ to evaluate users that contain a reasonable number of interactions (i.e., knowledge for the system). Furthermore, to simplify the application of the methods and evaluation, the user's ratings, which are integer values from 1 to 5, were transformed to 1 if the rating was ≥ 4 and 0, otherwise. Table 1 presents the main statistical measures about the created datasets. It is important to highlight the high sparsity index, a remarkable characteristic of the fashion domain.

Table 1. Dataset statistics.

Dataset	#User	#Item	#Interactions	Sparsity
Amazon Fashion (n = 5)	2,056	7,433	13,452	99.91%
Amazon Fashion (n = 10)	128	1,382	1,654	99.06%
Amazon Clothing (n = 5)	7,528	51,029	70,172	99.98%
Amazon Clothing (n = 10)	2,152	28,338	36,011	99.94%

4.1.2 Evaluation Methodology

To measure the quality of the methods in the recommendation task, we used a leave-one-out split, which is commonly applied by the literature of RSs [14,24]. In this data splitting technique, we insert the last interaction of each user (in temporal order) into the test set and use the remaining data as training. In addition, due to the high time cost of classifying all items in a base, we adopt the strategy [21] of generating a random sample of 99 items that are not interacted by the user for each existing interaction.

The performance of each item on the recommended list is evaluated in this paper by three distinct metrics [5]:

- Hit Ratio (HR): represents the number of items recommended correctly, based on the set of items consumed by the target user (test).
- Mean Reciprocal Rank (MRR): is a statistical measure to evaluate the user's "effort" to find an item relevant to her/him. The closer to the top of a ranking the correct items are for a user, the closer the MMR is to 1, and consequently, the better the solution.
- Normalized Discounted Cumulative Gain (NDCG): is a statistical measure to evaluate the quality of a ranking. The NDCG value is high when the items of higher estimated relevance appear higher up the actual ranking when compared to the items of lower estimated relevance.

We compare our proposal with several algorithms commonly used in the fashion scenario. These algorithms consist of generic and specific strategies for the fashion recommendation scenario:

1. Random: simple model, to define the lower bound of the problem, recommends items randomly.
2. Popular: traditional baseline model of recommender systems, this method ranks the items according to the most popular. We defined Popularity in this work as the number of different users who clicked on the item.
3. SVD (Singular Value Decomposition): SVD is a matrix factorization method used in RSs for collaborative filtering.
4. SVD++: This method minimizes an objective function, which includes an overall bias and two others for users and items, using Stochastic Gradient Descent (SGD) and regularization with L2.
5. BilinearNet: it is a method of embeddings learning with neural networks, which does not use any layers and learns the representations of users and items.
6. NeuMF [14]: is a neural network-based method, which merges generalized matrix factorization using neural networks with multilayer perceptron layers.
7. LightGCN [13]: is a method that learns user and item embeddings by propagating them linearly in the graph of user interaction with the item and uses the weighted sum of the learned embeddings in all layers as the final embedding.

For the selection of the best hyper-parameters of the methods, a search for hyper-parameters' values was performed using an execution over the training from the *leave-one-out*. More specifically, we applied the same data splitting (i.e., *leave-one-out*) in training set to generate a new training and validation set. We executed the methods for a set of hand-picked hyper-parameters in the new training set and validation set. Then we selected the hyper-parameters' set with the highest MRR to the final evaluation in the test set for each method.

We have adopted a recent strategy to evaluate the generated lists [14], which selects the top 10 items to evaluate all metrics. Thus HR measures whether the test item is present in the top 10 list, and NCDG accounts for the hit position by assigning higher scores to hits in the first ranks, while MRR assigns higher scores to the first hits with a smoother (linear) score decrease. We computed the scores for the three metrics for each user, and the average was stored. The entire evaluation process was run five times, with the random sampling of items being different for each of the five runs. All this enables a later statistical evaluation of the methods' gains. This statistical evaluation was done using the Student's t-test with $p = 0.05$. The evaluation methodology is presented in Fig. 2.

4.2 Results

We divided the presentation of the results into two parts. First, we compared the results achieved by our proposal with the results obtained by the other strategies, following the methodology presented in the previous section. Then, we compared different combinations of the information and strategies considered in our proposal to evaluate each component's impact on the final result of the stacking proposal.

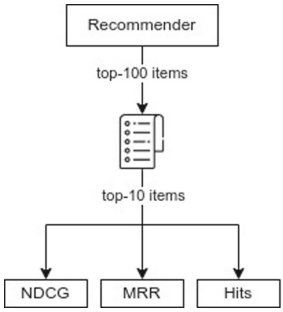

Fig. 2. Evaluation of the list of recommended items.

4.2.1 Comparison to Baselines

Table 2 presents the results of the methods on the databases according to the evaluation policy previously described. Evaluating the baselines previously selected for this work, we can notice that the non-personalized algorithms tend to behave better when we apply less aggressive filtering on the collections data ($n = 5$). Such approaches keep many users with little information (i.e., associated ratings). Thus, techniques that exploit global user interest may be favored in this scenario. However, in the other scenarios, we see the superiority of traditional methods and recently applied strategies in the fashion retail scenario.

In general, it is also possible to observe that our proposed method presents a significantly better result when compared to all baselines, considering all evaluated metrics and datasets. In some scenarios, such as the Amazon Fashion dataset ($n = 10$), our proposal achieves gains of 80% for MRR, 70% for NDCG, and 108% for Hits. These results validate our primary hypothesis that combining multiple pieces of information is effective for the fashion domain. Moreover, even in highly sparse scenarios (with less aggressive filtering), our strategy can also present the best results due to its ability to compose multiple information for user preference prediction.

4.2.2 Evaluating Stacking Components

This subsection presents an experiment in which we vary the combinations over inputs to the proposed stacking method. The main goal of this experiment is to identify the explanations for the results obtained through the method and what are the primary impact factors for the final result. There will always be one of the methods in the input sets, LightGCN or Popular. Table 3 shows the results obtained from this experiment, where we highlight the use of context through the symbol "*" in front of the word MLP (MLP*).

We can see the impact of including contextual information about users on the learning of our stacking model with this analysis. Although there are some cases where the combination of methods without context is competitive with the best combinations, overall, the best values for each metric are obtained using contextual information. This occurs because this information can guide our neural

Table 2. Performance of the different strategies on four datasets. The best results are highlighted in bold. The stacking method (MLP) achieves statistically significant improvements, outperforming all baselines. The symbol ▲ denotes significant gains and • denotes statistical ties by applying Student's t-test with a p value $= \mathbf{0.05}$ over the best baseline.

	MRR	NDCG	Hits
Amazon Fashion (n = 5)			
Random	0.03079	0.01036	0.10438
Popular	0.24759	0.04396	0.32160
SVD	0.23278	0.03781	0.27986
SVD++	0.04975	0.01436	0.12996
BilinearNet	0.23848	0.03713	0.27160
NeuMF	0.24200	0.03942	0.30068
LightGCN	0.25016	0.03966	0.28385
MLP* (LightGCN + Popular)	**0.30309**▲	**0.06172**▲	**0.52578**▲
Amazon Fashion (n = 10)			
Random	0.02915	0.01025	0.09219
Popular	0.06004	0.01181	0.09531
SVD	0.07039	0.01319	0.11094
SVD++	0.08332	0.02058	0.15625
BilinearNet	0.06967	0.01056	0.07969
NeuMF	0.08320	0.01513	0.12187
LightGCN	0.06568	0.01206	0.09688
MLP* (LightGCN + Popular)	**0.15010**▲	**0.03511**▲	**0.32500**▲
Amazon Clothing (n = 5)			
Random	0.02976	0.01016	0.10181
Popular	0.08635	0.02169	0.17545
SVD	0.05184	0.01231	0.10670
SVD++	0.07245	0.01915	0.16400
BilinearNet	0.09200	0.01599	0.12375
NeuMF	0.09753	0.01838	0.14859
LightGCN	0.09131	0.01731	0.13116
MLP* (LightGCN + Popular)	**0.13566**▲	**0.03294**▲	**0.26637**▲
Amazon Clothing (n = 10)			
Random	0.03062	0.01033	0.10390
Popular	0.06673	0.01475	0.11803
SVD	0.05201	0.00990	0.08327
SVD++	0.08378	0.02174	**0.18374** •
BilinearNet	0.07173	0.01186	0.09247
NeuMF	0.07917	0.01586	0.13429
LightGCN	0.07110	0.01282	0.09684
MLP* (LightGCN + Popular)	**0.10998**▲	**0.02353**▲	**0.18494**•

Table 3. Results of the proposed factorial experiment. The asterisk "*" indicates the use of context. The largest values are highlighted in **bold**.

	MRR	NDCG	Hits
Amazon Fashion (n = 5)			
MLP* (Popular + LightGCN)	0.30309	**0.06172**	**0.52578**
MLP* (Popular)	0.09041	0.01810	0.17247
MLP* (LightGCN)	0.07032	0.01716	0.15691
MLP (Popular)	0.24572	0.04366	0.32179
MLP (LightGCN)	0.05444	0.01093	0.08317
MLP (Popular + LightGCN)	**0.30633**	0.05973	0.49368
Amazon Fashion (n = 10)			
MLP* (Popular + LightGCN)	0.15010	0.03511	0.32500
MLP* (Popular)	**0.31656**	**0.09285**	0.77656
MLP* (LightGCN)	0.06457	0.01280	0.10156
MLP (Popular)	0.25637	0.08096	0.72969
MLP (LightGCN)	0.02594	0.00547	0.05000
MLP (Popular + LightGCN)	0.24800	0.08222	**0.80625**
Amazon Cloth (n = 5)			
MLP* (Popular + LightGCN)	0.13566	0.03294	0.26637
MLP* (Popular)	0.07004	0.01872	0.14926
MLP* (LightGCN)	**0.19476**	0.03778	0.28536
MLP (Popular)	0.19086	**0.04515**	**0.38982**
MLP (LightGCN)	0.10115	0.02085	0.15444
MLP (Popular + LightGCN)	0.06994	0.01887	0.14880
Amazon Cloth (n = 10)			
MLP* (Popular + LightGCN)	0.10998	0.02353	0.18494
MLP* (Popular)	**0.34585**	**0.08145**	**0.64870**
MLP* (LightGCN)	0.17003	0.02881	0.22007
MLP (Popular)	0.33231	0.07267	0.54610
MLP (LightGCN)	0.04105	0.00744	0.05604
MLP (Popular + LightGCN)	0.09538	0.02007	0.15409

network to more assertive decisions about the preferences of these users. Something interesting to mention is also the applicability of popular strategies that can always be used with other recommendation algorithms. Furthermore, by combining non-personalized and personalized strategies with contextual information, our proposal stands out regardless of the application scenario. We intend in the future to evaluate how other strategies and information can be incorporated into our proposal and analyze their impact in the fashion domain.

5 Conclusion and Future Work

This paper presents a fashion recommendation proposal that combines different contextual information of users, items, and their interactions with other recommendation models through a stacking process. For dealing with distinct information (i.e., raw contextual information and information derived from modeling processes with distinct recommendations), we proposed using models based on neural networks since these models can combine attributes at multiple levels. From this generic proposal, we instantiated a first solution that uses an MLP with only two recommendation models: (i) non-customized model named Most Popular; and (ii) a personalized method based on LightGCN [13]. We evaluated our proposal on two Amazon data collections, comparing it with seven state-of-the-art proposals in fashion recommendation. Overall, our method showed significant gains of up to 80% MRR, 70% NDCG, and 108% Hits over the other models.

We intend to investigate and expand our analysis on future work by including other baselines and other stacking proposals. We also aim to apply other techniques for tabular data that are simple and efficient for aggregating information from RS scores with context.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
2. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 217–253. Springer, Boston, MA (2011). https://doi.org/10.1007/978-0-387-85820-3_7
3. Akshita, S., Smita, A.: Recommender system: review. Int. J. Comput. Appl. **71**(24), 38–42 (2013)
4. Bao, X., Bergman, L., Thompson, R.: Stacking recommendation engines with additional meta-features. In: Proceedings of the Third ACM Conference on Recommender Systems, pp. 109–116 (2009)
5. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. Knowl.-Based Syst. **46**, 109–132 (2013)
6. Brovman, Y.M., et al.: Optimizing similar item recommendations in a semi-structured marketplace to maximize conversion. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 199–202 (2016)
7. Cheng, H.-T., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 7–10 (2016)
8. Dong, M., Zeng, X., Koehl, L., Zhang, J.: An interactive knowledge-based recommender system for fashion product design in the big data environment. Inf. Sci. **540**, 469–488 (2020)
9. Farfetch (2021)

10. Frejlichowski, D., Czapiewski, P., Hofman, R.: Finding similar clothes based on semantic description for the purpose of fashion recommender system. In: Nguyen, N.T., Trawiński, B., Fujita, H., Hong, T.-P. (eds.) ACIIDS 2016. LNCS (LNAI), vol. 9621, pp. 13–22. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49381-6_2
11. Haruna, K., et al.: Context-aware recommender system: a review of recent developmental process and future research direction. Appl. Sci. **7**(12), 1211 (2017)
12. He, R., McAuley, J.: Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering. In: proceedings of the 25th International Conference on World Wide Web, pp. 507–517 (2016)
13. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: simplifying and powering graph convolution network for recommendation (2020)
14. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.-S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182 (2017)
15. Hu, Y., Yi, X., Davis, L.S.: Collaborative fashion recommendation: a functional tensor factorization approach. In: Proceedings of the 23rd ACM International Conference on Multimedia, pp. 129–138 (2015)
16. Jannach, D., Jugovac, M.: Measuring the business value of recommender systems. ACM Trans. Manag. Inf. Syst. (TMIS) **10**(4), 1–23 (2019)
17. Kang, W.-C., Fang, C., Wang, Z., McAuley, J.: Visually-aware fashion recommendation and design with generative image models. In: 2017 IEEE International Conference on Data Mining (ICDM), pp. 207–216. IEEE (2017)
18. Khusro, S., Ali, Z., Ullah, I.: Recommender systems: issues, challenges, and research opportunities. In: Kim, K., Joukov, N. (eds.) Information Science and Applications (ICISA) 2016. LNEE, vol. 376, pp. 1179–1189. Springer, Singapore (2016). https://doi.org/10.1007/978-981-10-0557-2_112
19. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
20. Konstan, J.A., Riedl, J.: Recommender systems: from algorithms to user experience. User Model. User-Adap. Inter. **22**(1), 101–123 (2012)
21. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434 (2008)
22. Lee, D., Hosanagar, K.: Impact of recommender systems on sales volume and diversity (2014)
23. Nguyen, H.T., et al.: Learning to rank for personalised fashion recommender systems via implicit feedback. In: Prasath, R., O'Reilly, P., Kathirvalavakumar, T. (eds.) MIKE 2014. LNCS (LNAI), vol. 8891, pp. 51–61. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13817-6_6
24. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)
25. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 1–35. Springer, Boston, MA (2011). https://doi.org/10.1007/978-0-387-85820-3_1
26. Taud, H., Mas, J.F.: Multilayer perceptron (MLP). In: Camacho Olmedo, M.T., Paegelow, M., Mas, J.-F., Escobar, F. (eds.) Geomatic Approaches for Modeling Land Change Scenarios. LNGC, pp. 451–455. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-60801-3_27

27. Wakita, Y., Oku, K., Huang, H.-H., Kawagoe, K.: A fashion-brand recommender system using brand association rules and features. In: 2015 IIAI 4th International Congress on Advanced Applied Informatics, pp. 719–720. IEEE (2015)
28. Wang, L., Zeng, X., Koehl, L., Chen, Y.: Intelligent fashion recommender system: fuzzy logic in personalized garment design. IEEE Trans. Hum.-Mach. Syst. **45**(1), 95–109 (2014)
29. Wu, J., et al.: Self-supervised graph learning for recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 726–735 (2021)
30. Yin, R., Li, K., Lu, J., Zhang, G.: Enhancing fashion recommendation with visual compatibility relationship. In: The World Wide Web Conference, pp. 3434–3440 (2019)
31. Zeng, X., Koehl, L., Wang, L., Chen, Y.: An intelligent recommender system for personalized fashion design. In: 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), pp. 760–765. IEEE (2013)