Cloth Recommender System Based on Item Matching

Yan Lei¹, Long Chen¹ and Ziyu Guan¹,a
¹Northwest university, Xi’an, China

Corresponding author: ziyuguan@nwu.edu.cn

Abstract. Recommender systems have become an essential part in our life. Most social Websites use recommender systems to enhance user experience. In online shopping Websites such as Amazon, Clothing is one of the most popular domains, therefore a recommender is of great significance. Previous recommender systems were often focused on retrieving items based on user preference, i.e. similarity to the previous items purchased by the user. However, in Clothing domain, the matching relationships between candidate items and the previously purchased is also important for recommendation. For example, a user may want to buy a new jean rather than suit pants if he/she has just purchased a shirt. This kind of matching relationships also frequently occurs in other life contexts. In this paper, we aim to recommend new clothes that can better match the clothes purchased by a user. This new recommendation strategy would work better in the Clothing domain and complement the current recommendation literature. Experiment results show that our method can lead to better recommendation performance in the Clothing domain.

1. Introduction

Today's society is a network explosion world. Various network applications are endlessly emerging. In these network applications, recommendation systems are a key component. In the online shopping context, recommendation systems build a user preference model based on user purchase records or rating records, then recommend suitable products for target users according to this preference model. The research of recommendation systems not only personalizes user's experience, but also brings a great potential value and potential customers to many companies.

As a specific application of the recommendation system, clothing recommendation is important. Because the clothing is a popular domain in online shopping. Previous recommender systems often focus on retrieving items based on user preference. For example, when we want to purchase the jean, the recommender system will suggest a series of pants based on our preference. This suggestion may bring inconvenience to the user.

Many researchers believe that the common ground between purchasing the previous cloth and purchasing the new cloth is to meet the user preference or requirement. But this is different from our actual situation. When we buy clothes, we will carefully consider the quality or price of clothes. One of the key but overlooked aspect is that we also consider whether the item can match the purchased. If we have purchased one cloth A and this cloth cannot match the previously purchased, we will not wear it. Especially, If we purchase a new cloth which can match the previously purchased better, the cloth A will be forgotten. Therefore, matching relationship between the candidate and the purchased is also important for recommendation systems. The matching is related to the cloth color, texture, shape, catalog, description and so on.
In this paper, we aim to recommend new clothes that can match clothes purchased better. Of course, we think this cloth also needs to meet the target user preference or requirement. When users buy the cloth, they will not buy the cloth hated or disliked. So, in our paper, we also consider the user preference. We take a series of experiments to test our design for recommendation systems. And final results show that our method can lead to a better recommendation performance in the clothing recommendation domain.

In summary, the contribution of this paper can be summarized as following:

(1) We find that matching strength between new items and previous items purchased can affect users' purchases and this is always ignored by current recommender systems.

(2) We fuse user preference and cloth matching into a new model.

(3) We realize and evaluate our model on real datasets.

In section 2, we formally define our primary problem. In section 3, we discuss recommender system related works. In the section 4, we demonstrate our models and our algorithms for the problem mentioned above. In the section 5, we compare our method with other baselines on Amazon dataset. In the section 6, we summarize our paper and provide the future research direction.

2. Problem Description
We first give the notations in our work. Items (clothes) and users are denoted as $I= \{I_1, I_2, \ldots, I_n\}$ and $U= \{U_1, U_2, \ldots, U_m\}$ respectively, where $n$ is number of items, $m$ is the number of the users. Context information include two parts: non-visual content $I^N$ and visual context $I^V$. Non-visual context involves some clothes attributes, such as brand, text, suitable age, category. Visual context are visual features of clothes such as shape, line, outline. The definition of our problem is given as following.

Problem: given a user record $U_i$. Item attributes $I^N$ and image $I^V$ are employed as context information. The target is to predicting the score of item $I_i$ on user $U_i$.

3. Relate Work
Recommendation is a long-term research problem which has attracted much attention. The collaborative filtering (CF) [1] style algorithms are most popular. Further, methods of matrix decomposition are slowly emerging [2][3]. For example, A.T. Hofmann et al proposed a matrix factorization SVD [4]. This Matrix factorization approach is proposed to capture the implicit feature in documents. It specializes in mining the statistical information of documents but ignoring the time-varying user preferences which is a key factor for recommendation. To fill this gap [5], takes dynamics temporal into consideration and fuses time attribute into the matrix factorization of SVD. Since the user preferences are changing over the time. This time SVD++ approach achieves a good performance in the condition of a large scale of training data. However, it is a data-hungry approach. When trained on a small dataset, the performance will decline. Besides, it cannot well capture the user preferences because this model is poor at detecting strong associations among a small set of closely related items. We use a variant time SVD++ model to capture user preference.

Clothes recommendation systems get more and more attention since the Vartak M proposes a combination based recommendation system [6]. This paper recommends user one cloth combination according user's occasion description. But this paper just provides a good idea and a simply realize method. Further, some researchers propose scenario-oriented recommendation [7] [8] [9]. These methods all consider the cloth combination effect, but they ignore the cloth matching strength between the candidate and the user's previous item purchased. Our work exactly considers this relationship and measures the candidates' matching score.

4. Method
The key idea of our method is to fuse matching strength and user preference into a variant time SVD++ model. The final prediction relies on two important parts: user preference and matching strength. We design two corresponding models to capture them. In the following, we first describe the user preference model which is design for user interests mining. Then a cluster method is employed to
compute the matching strength between new items and purchased ones. Last, we design a prediction function to combine above two scores.

4.1. The Preference Model of the User

When choosing a piece of clothing, different persons have different interests. These interests range from colors to styles. For instance, some customers like black and some others prefer white. The user preference is a key factor to affect the purchasing behavior. Hence, we propose a preference model to mining user interests. Most matrix factorization (MF) approaches are used to discover the user preferences. But preferences change over the time, e.g. a fashion style in last year maybe not popular in this year. These MF methods do not take this factor into account except the time SVD++. Time SVD++ fuses the time into the model to better capture the tendency of user preference. However, one fatal flaw of time SVD++ is that the ability measuring item relationship is weak. Because this model uses an indirect way (user's single vector) to measure the relationship of two items and just only gets a weak association. Unfortunately, in the clothing recommendation, a weak clothing association will cause a bad recommend.

On the contrary, neighborhood approach evaluates the preference of a user for an item based on ratings of similar items by the same user. Using the neighborhood approach can get a strong association between two items. But the neighborhood approach cannot capture user preference well. Because this method ignores the rating matrix's implied information. After considering the two methods pro and con, in this paper, we fuse this neighborhood into time SVD++. This guarantees our method can capture the strong association between the clothes and user's feature.

Our model as following:

\[ r_u(t) = \mu + b_i(t) + b_u(t) + \gamma q^T p_u(t) + (1 - \gamma) \sum_{j \in N(i)} \text{sim}(i, j) r_{ij} \]  

(1)

We add the neighbor effect to detect strong associations among a small set of closely related clothes. The \( r_u(t) \) represents the predicting score of cloth \( I_i \) for user \( u \) at the time \( t \). The \( \mu \) represents the average rating of the user \( u \), and the \( b_i(t) \) represents the average rating of \( I_i \) at the time \( t \). The \( b_u(t) \) represents the average rating of user \( u \) at the time \( t \). The back part represents the Matrix factorize. The \( Pu(t) \) represents the implied characteristics of user \( u \) and the \( qi \) represents the implied characteristics of the item. The \( N(i) \) denotes the target neighbor set of item \( I_i \). This equation uses Matrix factorized to get the implied characteristics of user \( u \) or item \( I_i \) based on the time. So it can capture user implied characteristics and item's implied characteristics. This equation combines the advantage of the time SVD++ and the neighborhood model, so this model can mine the user's cloth preference better.

In practical, we can find that people purchased some clothes on the same time. If the purchase date is closer, the clothes are more relevant. So we take the purchased time into consideration. The equation as following:

\[ \text{sim}(i, j) = \frac{\sum_{v \in \text{vec}} (r_{ui} - r_i^v)(r_{uj} - r_j^v)z(v_i, v)}{\sqrt{\sum_{v \in \text{vec}} (r_{ui} - r_i^v)z(v_i, v)} \sqrt{\sum_{v \in \text{vec}} (r_{uj} - r_j^v)z(v_j, v)}} \]  

(2)

\[ z(v_i, v_j) = e^{-|p_u - c_i|} \]  

(3)

We add the exponential decay \( z(v_i, v_j) \) to control the time effect. We can use equation (1) to predict the user preference and get score about the preference.
4.2. The Matching Strength

In daily life, in clothing domain, the matching relationship between candidate items and the previously purchased is also important for recommendation. Further, this kind of matching relationship also frequently occurs in other life contexts. Bad matching will greatly reduce the user purchase intention. In this paper, we aim to recommend new clothes that can match the purchased better. Further, we use the matching strength between the new cloth and the previous cloth combination purchased as the recommend standard. When without considering the calculated consumption, using a loop compute can get the whole matching score, called the naive method in this paper. However, in practice, it is necessary to reduce calculate amount when considering the actual matching scale will reach \( M \times 2N \). The \( M \) represents candidates amount and the \( N \) represents the amount of clothes purchased. In this paper, we propose the cluster method to reduce computation. In the next, we will describe the two kinds of methods in detail.

**Naive Method:** naive method gets matching strength score by computing cloth combination similar strength. Because a good cloth combination always has large similar parts. These similar parts usually include two aspects. The one aspect is the cloth's no-visual context information such as the brand, the texture, the suitable crowd. The other aspect is the visual features \([10] [11] [12]\) include texture, outline, line and so on. So, in order to get an accurate matching strength score, we must consider these two aspects.

In this paper, the naive method uses the cosine \([13]\) to measure the similar strength among clothes. After considering these two aspects, the matching strength can be inferred as following:

\[
M(i, j) = \partial \text{sim}(I^N_i, I^N_j) + (1 - \partial) \text{sim}(I^V_i, I^V_j)
\]

\[
\text{sim}(i, j) = \frac{I_i \cdot I_j}{|I_i| \cdot |I_j|}
\]

We set parameter \( \partial \) to control the weight difference of no-visual feature and visual feature. When the visual feature is more important than the no-visual feature, the \( \partial \) will be smaller. Of course, when the no-visual feature is more important than the visual feature, the \( \partial \) will be greater. In a word, this method is very simple. When the matching scale is small, the method can quickly get the matching strength score. But this will be invalid when the matching scale is huge.

**Improved method:** in order to reduce calculated consumption, we adapt cluster method to classify clothes purchased into several big sets. This is inspired by our real world, e.g.: we always cluster clothes purchased into different combinations. When we consider new clothes, we need to compare the matching score between this cloth and each combination. If the matching scores are not good, we will not consider to buy this cloth. Therefore, we only need to compute matching strength between candidate and these several big sets after clustering. The scale will be reduced to \( M \times N \) by using this improved method. Our improved method relies on two parts. The first part, we cluster clothes purchased into several big sets. The second part, we compute matching strength between the new cloth and each cluster. We will describe these two parts in detail.

We cluster these items by the hierarchical clustering AGES. We use \( C \) to represent the cluster in which the element can match other. All elements in the same cluster have a similar taste, or be used into the same occasion. We also use a feature vector to describe the profile of the cluster. The profile can be described as following:

\[
P(C_i) = [f_1, f_2, \ldots, f_m, f_{n+m+1}]
\]

We use \( n+m+1 \) dimensions to describe the cluster profile, the \( n \) dimension represents cluster's no-visual feature; the \( m \) dimension represents cluster visual feature; and last \( 1 \) dimension represents cluster last update time.

We use a matrix \( M_a \) to record the suitability (matching strength) among cluster, the matrix element \( M_a(i, j) \) represents the suitability between cluster \( C_i \) and \( C_j \). We set parameter \( \text{min}_{d_{\text{dis}}} \) as the minimum
suitability threshold in cluster. When the suitability is larger than the mindis, we think corresponding clusters can match each other well. When the suitability is less than the \( \text{min}_{\text{dis}} \), we think corresponding clusters cannot match each other well. The algorithm is as algorithm 1 as follow Table 1. Firstly, we initialize clusters and the suitability matrix, and select the maximum suitability which needs to meet the requirement of \( \text{min}_{\text{dis}} \). Then we merge and update the two corresponding clusters, and repeat the process until without the suitability. During merging two clusters into one cluster, we need to update the profile of this cluster. When we are updating cluster, we select maximum features per dimension in two clusters. This cluster's profile can describe margin level features of the cluster. If one candidate can match this cluster, the suitability between them will ranges over \((d,1)\), and \(d\) represents suitability of candidate and the cluster. The updating profile of clusters as following:

\[
U(C_i, C_j) = \sum_{f_i \in p(C_i)} \sum_{f_j \in p(C_j)} \max(f_i, f_j)
\]

(7)

This improved method has a small computation, and it will quickly find the candidate with higher score.

| Table 1. Matching Algorithm |
|-----------------------------|
| **Item matching cluster Algorithm 1** |
| **Input:** item set \( S \), the item no-visual context feature \( I^v \), Item visual feature \( I^c \) |
| 1. For each \( j \) in \([1, m]\) do |
| 2. Init-cluster(\( C_i \)) |
| 3. End for |
| 4. For each \( i \) in \([1, m]\) do |
| 5. \( \text{Ma}(i, j) = M(C_i, C_j) \) Reference Formula (6) |
| 6. End for |
| 7. While(True): |
| 8. \( C_o, C_p, s = \text{Findmaxsuit}(\text{Ma}) \) |
| 9. If \( s < \text{min}_{\text{dis}} \): |
| 10. Break |
| 11. \( C_o = \text{Merging}(C_o, C_p) \) |
| 12. Renew-number () |
| 13. Update(\text{Ma}) |
| 14. End for |
| **Output:** \( C = C_1, C_2, \ldots, C_k \) |

### 4.3. The Final Composition
This improved method has a small computation, and it will quickly find the candidate with higher score.

\[
r^*_{ui} = r_{ui} \* \frac{\sum_{j} M(C_j, I_i)^* |c_j|}{\sum_{j} \sum_{k} M(C_k, I_i)^* |c_j|}
\]

(8)

\( r^*_{ui} \) denotes the final score of candidates \( I_i \). \( r_{ui} \) denotes the preference score of \( I_i \). \( N_{I_i} \) represents the cluster set which can match with the \( I_i \) harmoniously. In this paper, harmonious matching when the matching strength beyond the misdis. The right part of equation represents the harmonious matching set proportion based items purchased. This proportion can reflect this candidate matching strength. This combination means that we not only consider user preference, but also consider the matching.

### 5. Experiment
The experiments are designed to prove the effectiveness of our method. And we conduct comparisons of our method with other baseline method on Amazon dataset.
5.1. Datasets
Amazon dataset: the dataset is sourced from Amazon.com. The dataset includes reviews (ratings, text, helpfulness, votes), product metadata (descriptions, category, price, brand, asin, title) and product visual feature. This visual feature has employed a pre-trained convolution neural network, and obtain visual features. We use the visual feature to conduct experiments. In our experiments, we just use one catalog Clothing, Shoes and Jewellery. The statics information about the Amazon datasets as the table (2).

| Dataset   | users  | Items  | Total         |
|-----------|--------|--------|---------------|
| reviews   | 199,748| 331,173| 5,748,920     |
| metadata  | -      | 1,503,384| 1,503,384   |
| Visual features | -  | 1,503,384| 3.2G         |

5.2. Preprocessing
Most user items in Amazon dataset just have few purchase records which has insufficient information for clothes recommendation. Therefore, we remove these users' records. We choose 500 users and corresponding items' metadata information to conduct the experiment. After preprocessing, we get the user record, including (user-ID, user-name, item-ID, rating, time), the item information record includes (item-ID, item-title, price, asin, brand, description, catalog, visual feature).

5.3. Baselines
We compare our model MPS with two baselines: SVD++, time SVD++.
SVD++: this baseline first uses matrix decomposition to process the user rating matrix, then obtains two singular vectors, and finally uses these two singular vectors to describe the potential characteristics of users and products.
time SVD++: This model considers the user's dynamic preference. This model takes the user's purchase time as part of the user rating, then use matrix decomposition to process the user rating matrix.

5.4. Metric
In our experiment, we use RMSE metric and RMSE D-value to evaluate our model.
RMSE: the quality of the results is measured by their root mean squared error.

\[
RMSE = \sqrt{\frac{\sum_{(u,i) \in Testset} (r_{ui} - \hat{r}_{ui})^2}{|Testset|}}
\]

This measure puts more emphasis on large errors compared with the alternative of mean absolute error. Nonetheless, there is evidences that small improvements in RMSE terms can have a significant impact on the quality of the top few presented recommendations.
RMSE D-values: the RMSE difference value.

5.5. Performance
We conduct some experiments in this dataset. First, we conduct our method in different user's feature f and bins b. We set \(\bar{c}=0.5, \gamma=0.5, \min_d=0.5\). Then we show our method MPS training performance in the validation set. We run the dataset for twenty times repeatedly and get the average value of RMSE. The result as the figure 1.
In the figure 1, we intuitively discover that features affect the time the global minimum is found. When the f equals 20 and b equals 20, the global minimum shows at the iteration number equals 20. The other's global minimum shows at the iteration number nearly equals 10. More features lead to
more redundant information, and increasing the time overhead of matrix decomposition. So when the $f$ equals 10, we find the global minimum later. Then, the user's feature also affects the model shape. When the feature equal to 10, the model as a whole is linear. And when the feature equal to 20, the model as a whole presents a curve with a slower rate of decline. When the user feature increasing, the step size of each updating will become smaller during the parameter training process of the model. User feature increasing will directly increase the parameter training amount. Finally, users' bins affect the model training speed. As our bias increasing, the average squared error of the model increases rapidly after the model reaching a minimum.

![Figure 1](image1.png)

**Figure 1.** The model performance on the different features and bins in validation set

Secondly, we compare with baseline methods. We set $\hat{c}=0.5$, $\gamma=0.5$, $\min_{dis}=0.5$. And we show the RMSE difference value in the different feature and different bins, comparing with our method. The result is shown in the figure 2. The figure 2 describes the RMSE D-value.

From Figure 2, we can see that our overall model performs better than other methods. In the Amazon dataset, the association between the cloth and cloth is more strong than the movie dataset. Matrix decomposition models maybe poor at detecting strong associations among a small set of closely related items, such as SVD++, time SVD++. Our method combines the neighbour effect, so we can capture the associations and have a good performance. Our method MPS considers the cloth matching, so the performance is better than cft SVD++ (not consider cloth matching). Then, the SVD++ performs better than the time SVD++. Users' preference in clothing are not as personal as movies and music. The preference of clothing is largely influenced by the public aesthetics, so the change in clothing preferences usually needs a long time span. So the SVD++ has a good performance than the time SVD++.

From our experiment, we can find that this new strategy would work better in the clothing domain and complement the current recommendation literature.

![Figure 2](image2.png)

**Figure 2.** The comparison performance based on our Method MPS in the different feature and bins
6. Conclusion

Now, we will summary this paper. First of all, one of the key but overlooked aspect is the matching strength between the candidate item and the previously purchased. This matching relationship frequently occurs in our daily life. In this paper, we recommend new cloth that can match the previous clothes purchased by a user. We fuse user preference and matching strength into one new model. We conduct and evaluate our model, and experiments proved that our method is more effective than other methods. However, besides the matching with the cloth purchased and the user preference, the cloth recommend is affected by many other factors, such as regional. In the further, we will further consider more factors to produce a more accurate clothing recommend.

Acknowledgments

Authors wishing to acknowledge assistance or encouragement from colleagues, special work by technical staff or financial support from organizations should do so in an unnumbered Acknowledgments section immediately following the last numbered section of the paper.

References

[1] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry 1992. Using collaborative filtering to weave an information tapestry ACM 35,12 (1992), 61–70
[2] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003 Latent dirichlet allocation. J Machine Learning Research Archive 3 (2003), 993–1022.
[3] Andrew Fano and Scott W Kurth 2003. Personal choice point: helping users visualize what it means to buy a BMW. In International Conference on Intelligent User Interfaces. 46–52.
[4] Thomas Hofmann. 2013. Latent semantic models for collaborative filtering ACM.
[5] Yehuda Koren. 2010. Collaborative Filtering with Temporal Dynamics Commun ACM.
[6] Manasi Vartak and Samuel Madden. 2013. CHIC: a combination-based recommendation system. SIGMOD.
[7] Edward Shen, Henry Lieberman, and Francis Lam. 2007. What am I gonna wear: scenario-oriented recommendation. In International Conference on Intelligent User Interfaces
[8] Bo Wang, Jichang Guo, and Yan Zhang. 2015. An Analysis and Application of Fast Nonnegative Orthogonal Matching Pursuit for Image Categorization in Deep Networks. Mathematical Problems in Engineering
[9] Domonkos Tikk 2007 Major components of the gravity recommendation system SigKDD Explorations.
[10] Ruining He and Julian McAuley. 2016. Ups and Downs: Modelling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In International Conference on World Wide Web. 507–517.
[11] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel 2015. Image-Based Recommendations on Styles and Substitutes. SIGIR
[12] Sida Gu, Xiaqiang Liu, Lizhi Cai, and Jie Shen. 2017. Fashion Coordinates Recommendation Based on User Behavior and Visual Clothing Style. In Proceedings of the 3rd International Conference on Communication.
[13] David C Anastasiu and George Karypis. 2015. PL2AP: Fast Parallel Cosine Similarity Search. In Proceedings of the 5th Workshop on Irregular Applications: Architectures and Algorithms (IA3 ’15). ACM, New York