Fuzzy Systems and Data Mining VI
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA200706

ClothNet: A Neural Network Based Recommender System

Xing Hao, Han Zhike[1], Shen Yichen
Zhejiang University City College, Hangzhou, Zhejiang, 310015, China

Abstract. The traditional collaborative filtering recommendation systems have many deficiencies, which make them incompetent in the domain of clothing recommendation; we proposed a new ClothNet model based on CNN, RNN, collaborative filtering and the characteristics of the fashion industry. The accuracy and generalization performance of this model are improved compared with traditional systems. The visual information integrated into the ClothNet model enables the recommendation system to alleviate the cold start problem, and new clothes can be added to the recommendation list faster through the visual information. The addition of temporal information enables ClothNet sharply capturing the impact of seasonal and time changes on user preferences. However, because RNN and CNN have the disadvantage of requiring a large amount of data, combining RNN and CNN will make the model more difficult to converge, so we have adopted the LearningToRank training mode and obtained good results.

Keywords. Convolutional Neural Network, Recurrent neural network, Recommend System, LearningToRank, Collaborative filtering

1. Introduction

1.1. Research background

The Internet has gradually become an essential part of people's life which produces uncountable numbers of information. However, there is a problem - information explosion. Tiktok and Kwai are fledging, one of the reasons is that the recommendation system enables users to receive information they want in a closed feed stream. In the shopping software, recommendation systems are vital component. Through recommendation, long tail goods can be mined to exploit users' purchasing power.

In information retrieval, in order to solve the information overload problem, the recommendation system can learn our habits, reduce searching time and improve work efficiency, and at the same time, in the rapid development environment of network and e-commerce, in order to sell more products at a lower cost, recommendation system, as a means of precise marketing, has become increasingly popular.

The entities in the recommendation system fall into two categories: users (users) and recommended items (items). According to how to use these two types of information, recommendation systems can be divided into two types, personalized recommendation system and non-personalized recommendation system.

[1] Corresponding Author, Han Zhike, Zhejiang University City College, Hangzhou, Zhejiang, 310015, China; E-mail: hanzk@zucc.edu.cn.

Non-personalized recommendation system only considers the information of the item dimension and explored the connection among items, but does not include users' preferences. The most representative traditional methods are association rules and time-sensitive sorting. Since the user's information is not taken into account, there is a lack of targeted recommendations to the user.

The development of the Internet enables us utilizing information more effectively. From this data, we can extract the relationship between users and items, and even the relationship among users, which makes the emergence of personalized recommendation systems possible. CUDA is an accelerator for the development of deep learning, allowing the calculation speed of deep learning surging. In this article, we will introduce some optimization methods for a deep learning-based recommendation system.

Ranking recommendation algorithms can be roughly divided into three categories:

The first category is listwise approach. Listwise approach solves the ranking problem through direct ranking. During the training process, it takes a sorted list of objects (for example, a sorted list of documents in IR) as an example, and trains the ranking function by minimizing the list loss function defined on the list of prediction results and the real list.

The second category is pointwise approach, which looks at a single document at a time in each iteration. They basically take a single document and use it to train a classifier / regressor to predict the correlation between the current query and the real situation.

The third category is the pairwise approach. In the list method, sorting is transformed into sequence classification or sequence regression. The method of pairing is the pairwise sorting loss. For the two pairs of points a and b, if a is sorted before b, we think that the loss is 1, and vice versa.

The first kind of point-pair method is the traditional DEEP FM[1], FM[2], FFM[3],etc.

The second type of paired methods are represented by BPR[4], VBPR[5], etc.

The third type of list method is represented by SoftRank[6], ListNet[7], etc.

The traditional recommendation system and the deep learning-based recommendation algorithm mentioned in the previous section uses a single feature (mainly user, item purchase or rating matrix or a single content-based recommendation), The number of recommendation systems based on multiple content is relatively small.

Besides, the traditional recommendation systems don't exploit the information about the order of purchase time. It only records what items the user purchased, the order of purchasing has been ignored. We will use the recurrent neural network to solve this problem.

1.2. The framework of this article

The main goal of this article is to build a scalable, interpretable, personalized recommendation system with temporal information, which will greatly improve the performance. Our recommendation system can be closer to the changing hobbies of users and communities. Especially in a rapidly mutating field, the relationship between user preferences and time is more encrypted and inseparable; secondly, due to "the long tail" problem and new products are being constantly launched, so we need a cold start solution, we can't rely too much on explicit or implicit user feedback. We can also tap the user's preference for the product from a visual perspective to solve the cold start problem and make more precise recommendations. Our system can extract visual information and

understand the changes in user preferences and fashion preferences in the time dimension. In other words, how fashion is evolving.

At the same time, exploiting visual information can effectively solve the cold start and correctly handle items not co-occur in dataset. This paper proposes a new model--ClothNet, which can effectively extract the corresponding visual information and effectively solve the cold start and non-co-occurrence. Based on the deep neural network model of CNN and RNN, use Pytorch to build and train the model. The loss function is designed using a LearningToRank (sequence learning loss) model [7]. The dataset is the Amazon dataset of women clothing from 2011 to 2017 [8][9][10].

The deep neural network model based on CNN and RNN is the core content of the system. We will introduce the model from the dataset, data preprocessing, model design, training methods and parameters, result comparison and analysis.

2. Related works

Cao[7] et al. have proposed a new approach to LearningToRank, referred to as the listwise approach. In particular, they designed a new and powerful loss function in their work.

Smirnova et al.[11] pointed out that the previous RNN modeling approaches summarize the user state by only considering the sequence of items that the user has interacted with in the past, without taking into account other essential types of context information such as the associated types of user-item interactions, the time gaps between events and the time of day for each interaction. In order to solve this problem, they introduced this context information on the basis of traditional RNN and proposed a new Contextual Recurrent Neural Networks (CRNNs) model.

Rendel [2] proposed the FM (Factorization machines) model in 2010, the purpose is to solve the feature combination problem under sparse data. In traditional machine learning, features are encoded using One-hot, so the feature space is very sparse. In addition, we tend to combine features to create more new features. The FM model can effectively reduce the dimensionality of features. It has the following characteristics:

The FM model can be trained when the feature space is very sparse.

The FM model is of linear time complexity.

The FM model is a general model, and its training data feature value can be any real number.

Rendle et al. [4] also investigated the most common scenario with implicit feedback (e.g. clicks, purchases), and provided ideas on how to utilize implicit feedback information, they also presented a generic optimization criterion and learning algorithm for personalized ranking. He et al. [5] proposed a scalable factorization model to incorporate visual signals into predictors of people's opinions, which can be applied to a selection of large, real-world datasets. They made use of visual features extracted from product images using (pre-trained) deep networks, on top of which they learned an additional layer that uncovers the visual dimensions that best explain the variation in people's feedback. This not only leads to significantly more accurate personalized ranking methods, but also helps to solve cold start issues, and qualitatively to analyze the visual dimensions that influence people's opinions. Weston et al. [12], also proposed powerful factorization models which give us great inspiration. These models are used by us as baselines to provide a comparison of the results of our ClothNet model.

Chatfield et al. [13] proposed a new architecture names CNN-F, where F stands for Fast, and this structure has very good training efficiency. It plays a key role as a visual information extractor in the ClothNet model.

In recent years, many interesting research works have emerged in clothing recommendation. Shen et al. [14] proposed the Scenario-Oriented Recommendation. This algorithm requires users to provide special attributes of products, and analyzes the user's purpose to obtain more accurate recommendation results. Tu et al. [15] introduced hierarchical fashion multimedia mining model and developed a refined contour extraction method to build a color tone analysis model. Zhang et al. [16] proposed a neural network for clothing recommendation based on details such as the user's travel destination's weather, scenery and other characteristics, and achieved good results. Liu et al. [17] improved the Collaborative Filtering recommendation algorithm and proposed the Advanced User-based Collaborative Filtering recommendation algorithm. This algorithm introduces a user-item linked list, which can reduce the space complexity of the algorithm. De Barros Costa et al. [18] introduced an approach that recommends items based on fashion and users' body type. Packer et al. [19] proposed an approach that learns users' visual preferences and predicts based on this. Yu et al. [20] pointed out that the traditional clothing recommendation systems did not consider Aesthetic features, and introduced a special neural network for extracting Aesthetic features.

3. Model Design

3.1. Datasets

The Dataset in this study is based on the USCD's AMAZON user shopping and rating dataset [8][9][10]. After cleaning, the fashion category (women's wear) is selected, and the data volume is 100K +. And the main pictures corresponding to the item in the dataset are collected by a web crawler and used as the visual feature.

We use user comment history as implicit feedback, and we filter users who purchase less than five items. The results after processing are shown in Table 1.

Table 1 Dataset introduction (after processing)

Number of users	Number of users	Implicit feedback	Time span
99748	331173	854211	2014.3-2014.7

3.2. Model Architecture

We use a convolutional neural network to extract visual features, a recurrent neural network [21] to obtain time-series related information, and training by LearningToRank (sorted learning loss) method. At the same time, for each time step output of the recurrent neural network, we processed the L2 loss function with the pooling output of the standard picture after CNN.

3.2.1. Problem Definition

We use implicit feedback information, such as purchase history, click history, and ratings. In implicit feedback, there are non-negative feedback and sorted feedback; Our optimization goal is to sort the purchased items so that the items purchased by the user have higher ranks than the items that weren't purchased [4].

Figure 1 shows the architecture of our model. First, we use customer purchase record data to train FM, and compress user preferences to obtain a user matrix. Then we input the image corresponding to the product into the CNN-F module, and get the prediction result through CNN-F and RNN. And compare the output result of CNN and RNN with the result of FM, get the loss value, we use the loss value for back propagation to train the neural network parameters.

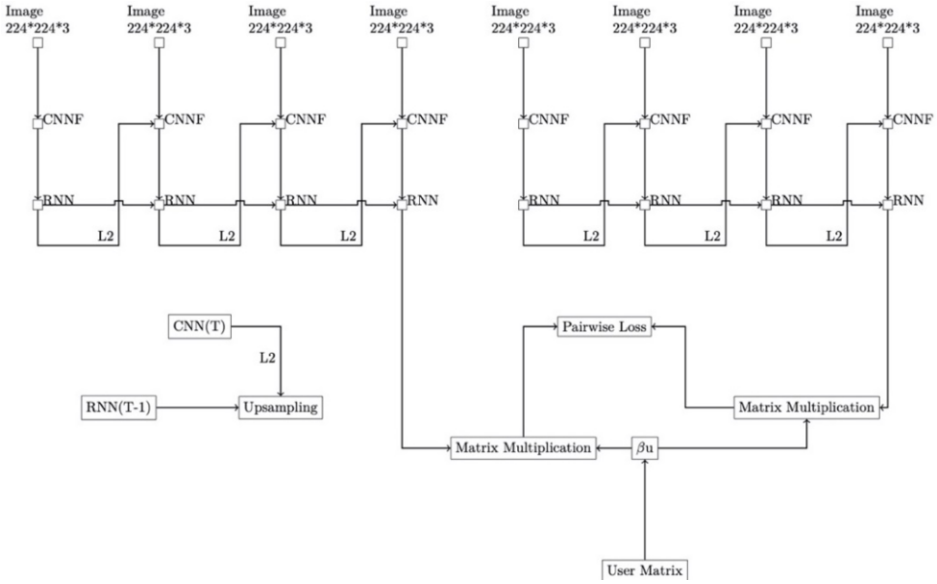

Figure 1. Structure of ClothNet

3.2.2. Preference predictor

We use a factorization machine (FM) to make the final score prediction, which can be regarded as a linear regression model based on matrix factorization initialization [22]. A most basic model for predicting users' commodities can be expressed according to Eqs. (1).

$$x_{u,i} = \alpha + \beta_u + \beta_i \gamma_u^T \gamma_i \tag{1}$$

In the past, the FM models with vision mostly used pre-trained convolutional neural network models (multiplexing backend for other computer vision applications such as target detection, segmentation and other task training models as backend to extract visual information), and use an embedding matrix to map high-dimensional visual information to low-dimensional. [5]

$$x_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u^T \gamma_i + \boldsymbol{\theta}_u^T (\boldsymbol{E} \boldsymbol{f}_i) \tag{2}$$

Our model is mainly used to solve the problem that the previous models directly using the pre-training model, instead we use the convolutional neural network model trained for the task end2end. In this way, we can directly extract visual features from the image and apply them to our model. At the same time, we exclude the item bias term β_i

and latent user-item preference γ_i compared to the above two models. The improved model is in Eqs. (3).

$$x_{u,i} = \alpha + \beta_u + \boldsymbol{\theta}_u^T \boldsymbol{\phi}(\boldsymbol{X_i}) \tag{3}$$

3.2.3. Visual feature extractor

Our convolutional neural network structure uses [13] as backend, which has very good training efficiency, which is one of the reasons why we choose this structure. The Backend has 8 layers of trainable parameters, 5 layers of convolution layers, and finally three layers of fully connected layers. The input image size is 224 * 224. We use ReLU[23] as activation function. Dropout[24] layer is added to prevent overfitting. BatchNorm[25] layer is used to normalize the input layer by adjusting and scaling the output of the convolutional layer. Table 2 shows the structure of the model.

Table 2 The Structure of CNN-F

Conv1	Conv2	Conv3	Conv4	Conv5	Fc1	Fc2	Fc3
64x11x11	256x5x5	256x3x3	256x3x3	256x3x3	4096	4096	K
st. 4,pad 0	st.1,pad 2	st. 1, pad 1	st. 1, pad 1	st. 1, pad 1	Drop 0.5	Drop 0.5	

3.2.4. Deconvolution (upsampling) layer

Due to our small amount of data, and in order to reduce the pressure of the recurrent neural network training, reduce the number of parameters, and accelerate the convergence of the model, we have performed global maximum pooling on the input data at the channel level. However, since we then have a reconstruction loss with the output of the convolutional neural network, the structures of the inputs are required to be the same, so we can use deconvolution or upsampling to interpolate the data. The traditional upsampling methods include bilinear interpolation, monolinear interpolation, Gaussian interpolation, etc., which we will not introduce here. Let's talk about the principle of deconvolution. Deconvolution is also called transposed convolution. We use the transposition of convolution to restore the dimensionality of the data after dimensionality reduction. Of course, such an operation certainly cannot fully restore the state before convolution, because convolution is not a bijective, but an irreversible dimensionality reduction operation, which is similar to hash functions, the same output may correspond to several inputs. But we can combine it with the convolution operation to transfer the gradient back to the recurrent neural network, so that its parameters can converge.

In practice, we are dealing with a training set with noise, finite dimensions, finite time series, and discrete sampling. However, by expressing the problem as a solution to the Toeplitz matrix and using Levinson recursion, we can estimate the filter with the smallest mean square error relatively quickly. We can also directly perform deconvolution in the frequency domain to obtain similar results. This trick is similar to logistic regression.

3.2.5. Temporal information extractor

In this paper, a variant of the recurrent neural network, Long Short-Term Memory Network (LSTM [26]), is used as the model structure for extracting time series information. Figure 2 shows the structure and propagation of LSTM. As a variant of

recurrent neural network, LSTM allows the previous temporal information to flow to the next step or even later steps through the memory gate and the forgetting, so that the current time step can get the information of the previous step. It is beneficial to the implementation of cold start. At the same time, LSTM provides a two-way flow mode, which allows time steps to not only propagate backwards, but also forwards, which solves the problem of one-way timing and allows the model to extract more information. Eqs. (4)-(8) shows is the principle of LSTM forward propagation:

$$f_t = \sigma_g(W_f x_t + U_f c_{t-1} + b_f) \tag{4}$$

$$i_t = \sigma_g(W_i x_t + U_i c_{t-1} + b_i) \tag{5}$$

$$o_t = \sigma_g(W_o x_t + U_o c_{t-1} + b_o) \tag{6}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + b_c) \tag{7}$$

$$h_t = o_t \circ \sigma_h(c_t) \tag{8}$$

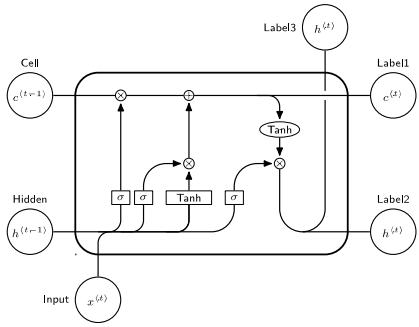

Figure 2. Schematic diagram of LSTM cell unit structure

4. Model loss and training details

4.1. Model loss design

In order to better combine time series information and visual information, we use L2 reconstruction loss [27] to enable time series network to learn time series information. We use pairwise ranking loss to let the model learn the recommendation ability similar to the FM.

4.1.1. L2 reconstruction loss

We adopted the reconstruction loss commonly used in GAN (Generative Adversarial Networks)[28], and made the pointwise mean square error between the output of the convolutional neural network network and the output of the recurrent neural network in the previous time step as our training loss. At the same time, we designed θ as the

attenuation hyperparameter. Since our time series input has the property of padding, we processed the sequence loss function with a long padding sequence, that is, the actual purchase sequence is too short, so that its contribution in the training process is smaller. The calculation formula of MSE is as follows:

$$MSE(\hat{\theta}) = E(\hat{\theta} - \theta)^2 \tag{9}$$

4.1.2. Loss for sorting pairs

In the sequence method, the loss for the sorted pair is transformed into the classification of the sequence or the regression of the sequence. The so-called pair is sorted in pairs. For example, (a, b) indicates that a is in front of b. We don't care how much the ranking pair a (for items observed by the user) is higher than the score for b (for items not observed by the user). As long as the score a is higher than the b score, we truncate the loss 1. Conversely, truncate the loss to 0. During the previous experiment, the model did not converge and eventually collapsed. The main reason for this situation is the large number of parameters of the recurrent neural network, so it requires more training data, and it is more difficult to converge to the best advantage. When we incorporate the convolutional neural network into the RNN, the gradient will be scattered, the model may collapse. The range of pairwise loss is shown below:

$$\begin{cases} 1 \; if \; a > b \\ 0 \; if \; a \leq b \end{cases} \tag{10}$$

4.2. Hyperparameters and training details

4.2.1. Dataset processing

We discarded users who purchased less than 5 items, and correspondingly deleted items that had no purchase history. At the same time, we specifically divided the cold start dataset for the cold start problem-commodities purchased less than 5 times were specifically used as the cold start dataset to test the cold start performance of our model.

4.2.2. Training hyperparameter details

The model image input is 224 * 224 * 3, the length of the padding sequence is 10, the weight attenuation coefficient θ is 0.1, the batch-size is 32 (2 * 16), and 50 epochs are trained. The test and training sets are randomly divided according to the user's 3/7 ratio. We use bilinear interpolation to upsample the data.

5. Comparison of model results

5.1. Comparison of model results

Our model compares random results (AUC 0.5), WARP[29], BPR-MF, FM and VBPR. Use AUC as our result evaluation index.

5.1.1. Evaluation index analysis

We use AUC as the result evaluation index, AUC, Area Under Curve, which means the area under the ROC curve. The AUC indicator measures the quality of our results for ranking. The formal representation of the AUC curve is shown in Eqs. (11).

$$AUC = \frac{1}{|\mathcal{U}|}\sum_{\mathrm{u}\in\mathrm{U}}\frac{1}{|\mathcal{D}_{\mathrm{u}}|}\sum_{(i,j)\in\mathcal{D}_{\mathrm{u}}}\xi(x_{u,i} > x_{u,j}) \tag{11}$$

$\mathcal{D}_{\mathrm{u}}$ represents the complete set of users, and ξ() is a Boolean indicator function. If the observed i score is greater than we think of j for the observed item, then we count 1 and vice versa. Accumulate the technology and then get the proportion of correct ordering. This indicator is also the training loss of the traditional BPR model [10].

5.1.2. Detailed explanation of the comparison model

Stochastic model: We use random probability to rank the observed and unobserved items, so the AUC ratio is exactly 0.5. This indicator represents the result of a random recommendation of the user ’s product, which is the lower limit of the model.

The top K optimized weights approximate matrix decomposition to the ranking loss (WARP) [12]: In order to face the growing dataset and labeling tasks, how to use small-scale datasets for training has become a problem. In the training mode, it aimed at a new ranking method for different score differences. It provides a new online learning mode. Hinge loss is used to cut off the score between different gaps. In order to achieve better optimization results.

Bayesian sorted factorization machine (BPR-MF): It is currently the best implicit user feedback recommendation system based on personal sequences. Use a simple FM and learn the observed priors according to Bayes' theorem as a posterior that can be used in the recommendation system. In this method, we only consider the observed sequence as a sequence of interest to the user. This potential pattern discovery can take advantage of a large amount of implicit feedback in a system. Therefore, the problem of insufficient data amount of the recommendation system is solved.

FM (Factorization Machines) [2]: traditional factorization algorithms. This method is similar to a linear regression model, and uses the second-order term of the vector inner product between items and characters to express the mutual relationship between features. It can be optimized using least squares or stochastic gradient descent.

Bayesian factorization machine with vision (VBPR) [5]: It is currently the best Bayesian sorting algorithm with visual information. It uses pre-trained CNN models to provide visual features. And add the visual information bias item to express the user's preference for visual information, so as to get a visual second-order item similar to the FM, which is our recommendation model more accurate.

5.1.3. Comparison of model results

Table 3 shows the model results, the evaluation index is AUC, and the Dataset is the Amazon Women's Clothing Dataset. Cold start means that our test set contains only goods purchased by less than five people. As we can see, our ClothNet model outperforms many models.

Table 3 Comparison of model results

	RAND	WARP	BPR-MF	FM	VBPR	ClothNet
All	0.5	0.6192	0.6543	0.6678	0.7081	0.7475
Cold Start	0.5	0.3822	0.5196	0.6682	0.6885	0.7268

6. Conclusion

The model we proposed mainly uses convolutional neural networks to extract the visual features of items, and uses LSTM to model the user's purchase dat. Finally, the loss function of pairwise is designed for training based on the LearningToRank method.

At the same time, there are fewer restrictions on the input dimensions based on the RNN. At the same time, we use the FM to decompose the user's implicit feedback (scoring matrix) and use the commodity vector as the initialization of our user matrix.

As for training, the RNN also adds L2 reconstruction loss to allow the model to obtain gradients at every time step, similar to the training method of relay supervision [30], and the relaxation term enables the model with a shorter sequence to converge with long sequences. In the selection of LearningToRank (sequence learning loss) loss functions, we again relaxed the score (the score ranking is only provided by the FM during initialization), and the observed and unobserved samples were truncated so as not to concern about the specific score regression differences, which makes the model easier to train.

For the dataset, we used Amazon women's clothing data as the training set and test set, divided by a ratio of seven to three, filtering users who purchased less than five products, which is convincing.

In terms of results, compared with some existing models, we have improved the accuracy of items by a maximum of 3.7% in the cold start of items (items that are purchased less) (see comparison of experimental results), which proves that our model has a certain ability on cold start.

In terms of future prospects, although our ClothNet model incorporates temporal information, the results when predicting temporal information are not very good, and sometimes even lower than traditional models. The reason is that the amount of training data required by the recurrent neural network is too large, and although we have a large dataset, the timing sequence generated is still very small for the items it contains, so the model is difficult to converge.

Reference

[1] Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. DeepFM: a factorization-machine based neural network for CTR prediction. arXiv preprint arXiv:1703.04247.2017 Mar;arXiv:1703.04247.

[2] Rendle, S.Factorization machines. In 2010 IEEE International Conference on Data Mining;2010 Dec 14-17;Sydney, Australia:IEEE;p. 995-1000.

[3] Juan, Y., Zhuang, Y., Chin, W. S., & Lin, C. J. Field-aware factorization machines for CTR prediction. In Proceedings of the 10th ACM Conference on Recommender Systems;2016 Sept 15-19; Boston,MA.ACM; c2016;p. 43-50.

[4] Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618.2012.

[5] He, R., & McAuley, J. VBPR: visual bayesian personalized ranking from implicit feedback. In Thirtieth AAAI conference on artificial intelligence.2016 Feb 12-17;Phoenix, Arizona:AAAI;c2016;p.144-150.

[6] Taylor, M., Guiver, J., Robertson, S., & Minka, T. Softrank: optimizing non-smooth rank metrics. In Proceedings of the 2008 International Conference on Web Search and Data Mining .2008 Feb 11-12;Palo Alto,CA.ACM;c2008;p. 77-86.
[7] Cao, Z., Qin, T., Liu, T. Y., Tsai, M. F., & Li, H. Learning to rank: from pairwise approach to listwise approach. In Proceedings of the 24th international conference on Machine learning.2007 Jun 20-24;Corvallis,Oregon.ACM;c2007;p. 129-136.
[8] 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.2016 Apr 11-15;Montreal,Canada.ACM;c2016;p. 507-517.
[9] McAuley, Julian, et al. Image-based recommendations on styles and substitutes.In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2015 Aug 9-13;Santiago,Chile.ACM;c2015;p.43-52.
[10] Ni, J., Li, J., & McAuley, J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).2019 Nov 3-7;Hong Kong,China.Association for Computational Linguistics;c2019;p. 188-197.
[11] Smirnova, E., & Vasile, F. Contextual sequence modeling for recommendation with recurrent neural networks. In Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems.2017 Aug 27;Como,Italy.ACM;c2017;p. 2-9.
[12] Weston, J., Bengio, S., & Usunier, N. Wsabie: Scaling up to large vocabulary image annotation. In Twenty-Second International Joint Conference on Artificial Intelligence.2011 Jul 16-22;Barcelona, Catalonia, Spain.IJCAI/AAAI;c2011;p.2764-2770.
[13] Chatfield, K., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets.British Machine Vision Conference, 2014 Sept 1-5,Nottingham, UK.BMVA Press;c2014;
[14] Shen, E., Lieberman, H., & Lam, F. What am I gonna wear? Scenario-oriented recommendation. In Proceedings of the 12th international conference on Intelligent user interfaces.2007 Jan 28-31;Honolulu,Hawaii.ACM;c2007;p.365-368.
[15] Tu, Q., & Dong, L. An intelligent personalized fashion recommendation system. In 2010 International Conference on Communications, Circuits and Systems (ICCCAS) .2010 Jul 28-30;Chengdu,China.IEEE;c2010;p. 479-485.
[16] Zhang, X., Jia, J., Gao, K., Zhang, Y., Zhang, D., Li, J., & Tian, Q. Trip outfits advisor: Location-oriented clothing recommendation. IEEE Transactions on Multimedia. 2017 Apr 24;IEEE;19(11), p.2533-2544.
[17] Liu, Y., Nie, J., Xu, L., Chen, Y., & Xu, B. Clothing recommendation system based on advanced user-based collaborative filtering algorithm. In International Conference on Signal and Information Processing, Networking and Computers.2017 Dec 19;Springer, Singapore;473,p. 436-443.
[18] de Barros Costa, E., Rocha, H. J. B., Silva, E. T., Lima, N. C., & Cavalcanti, J. (2017, April). Understanding and personalising clothing recommendation for women. In World Conference on Information Systems and Technologies. 2017 Apr 11-13;Springer, Cham;c2017,569,p.841-850.
[19] Packer, C., McAuley, J., & Ramisa, A.Visually-aware personalized recommendation using interpretable image representations. arXiv preprint arXiv:1806.09820.
[20] Yu, W., Zhang, H., He, X., Chen, X., Xiong, L., & Qin, Z. Aesthetic-based clothing recommendation. In Proceedings of the 2018 World Wide Web Conference.2019 Apr 23-27;Lyon,France.ACM;c2018;p. 649-658.
[21] Pineda, F. J. Generalization of back-propagation to recurrent neural networks. Physical review letters,1987 Nov;59(19), 2229.
[22] Koren, Y., & Bell, R. Advances in collaborative filtering. In Recommender systems handbook . 2015.Boston,MA:Springer.p. 77-118.
[23] Glorot, X., Bordes, A., & Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the fourteenth international conference on artificial intelligence and statistics. 2011 Apr 11-13;Fort Lauderdale, USA.JMLR.org;c2011;p. 315-323.
[24] Witten, I. H., & Frank, E. Data mining: practical machine learning tools and techniques with Java implementations. 2000 Jan.San Francisco,CA:Morgan Kaufmann.
[25] Ioffe, S., & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning.2015 Jul 6-11;Lille,France;JMLR.org;c2015;p.448-456.
[26] Hochreiter, S., & Schmidhuber, J. Long short-term memory. Neural computation,1997 Dec;9(8), 1735-1780.
[27] Mathieu, M., Couprie, C., & LeCun, Y. Deep multi-scale video prediction beyond mean square error. arXiv preprint arXiv:1511.05440.2015 Nov.

[28] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y.. Generative adversarial nets. In Advances in neural information processing systems.2014 Dec.Montreal Canada.ACM.p. 2672-2680.

[29] Weston, J., Bengio, S., & Usunier, N. Large scale image annotation: learning to rank with joint word-image embeddings. Machine learning.2010.Boston,MA:Springer.81(1), p.21-35.

[30] Wei, S. E., Ramakrishna, V., Kanade, T., & Sheikh, Y. (2016). Convolutional pose machines. In Proceedings of the IEEE conference on Conference on Computer Vision and Pattern Recognition.2016 Jun 27-30;Las Vegas,NV;IEEE Computer Society;c2016;p.4724-4732.