Postprint of Clothing Compatibility Prediction Model Based on Hypergraph Representation

Authors: Li Jian, Li Zhuo, Ma Tianxiang, Liang Jifeng

Date: 2022-04-07T15:01:56+00:00

Abstract

To address the limitation that existing clothing compatibility models predominantly focus on exploring compatibility between paired items, we propose a clothing compatibility prediction model based on hypergraph representation. The model first constructs a clothing hypergraph based on the different categories of fashion items in existing datasets and their matching relationships, where each node represents a clothing item and each hyperedge represents an outfit composed of multiple clothing items. To better infer clothing compatibility from the hypergraph, the model converts the hypergraph into a conventional graph and utilizes graph neural networks to simulate complex interactions between nodes. Finally, an attention mechanism is introduced to calculate clothing compatibility scores, thereby enhancing the model' s predictive capability. Experimental results demonstrate that on two fashion outfitting tasks—the clothing fill-in-the-blank task and clothing compatibility prediction—the model achieves accuracies of 77.29% and 96.23%, respectively, representing significant improvements over other baseline models.

Full Text

Preamble

Vol. 39 No. 8
Application Research of Computers
ChinaXiv Partner Journal

Outfit Compatibility Prediction Model Based on Hypergraph Representation

Li Jian[1],[3]†, Li Zhuo[1],[2],[3], Ma Tianxiang[4], Liang Jifeng[4]

[1] School of Microelectronics, Tianjin University, Tianjin 300072, China
[2] Peng Cheng Laboratory, Shenzhen, Guangdong 200093, China

[3] Tianjin Key Laboratory of Imaging & Perception Microelectronics Technology, Tianjin 300072, China
[4] State Grid Hebei Electric Power Co., Ltd. Electric Power Research Institute, Shijiazhuang 050021, China

Abstract: Existing outfit compatibility models primarily focus on pairwise item compatibility. To address this limitation, this paper proposes an outfit compatibility prediction model based on hypergraph representation. The model first constructs a fashion hypergraph from existing datasets, where each node represents a clothing item and each hyperedge represents an outfit composed of multiple items. To better infer outfit compatibility from the hypergraph, the model converts hypergraphs into traditional graphs and utilizes graph neural networks to simulate complex interactions between nodes. Finally, an attention mechanism is introduced to calculate outfit compatibility scores, enhancing the model' s predictive capability. Experimental results demonstrate that the model achieves accuracies of 77.29% and 96.23% on the outfit fill-in-the-blank task and outfit compatibility prediction task respectively, showing significant improvement over baseline models.

Keywords: outfit compatibility; hypergraph representation; graph neural network; attention mechanism

0 Introduction

In recent years, with the rapid development of the fashion industry, clothing has played an increasingly important role in people' s daily lives. A well-coordinated outfit can enhance one's charm and fully express personality. According to statistics, during the 2021 Double Eleven shopping festival, Alibaba' s e-commerce platform Tmall recorded a total transaction volume of 540.3 billion yuan, an increase of 42.1 billion yuan compared to 498.2 billion yuan in the previous year. This indicates that this potential market is expected to create substantial wealth, and clothing research has attracted growing attention due to this vast market. Consequently, research related to clothing has emerged, including personalized fashion design, outfit composition, item recommendation, and fashion trend prediction, with particular emphasis on outfit matching.

However, outfit matching is a complex task that depends on multiple subjective factors such as fashion style, cultural background, and trend dynamics. All these factors may vary from person to person and even change over time, making it difficult for everyone to coordinate appropriate and decent outfits. Therefore, analyzing massive amounts of clothing data to establish a suitable outfit matching method holds significant social and economic importance.

Numerous efforts have been dedicated to solving the outfit matching problem. For instance, Viet et al. used SiameseNet to map individual items from an image space to a style space, then measured the distance between items to predict pairwise clothing compatibility. McAuley et al. proposed using Low-rank Mahalanobis Transformation to map items into a style space for calculating similarity. Later, Han et al. represented an outfit as a sequence with a specific order and utilized bidirectional LSTMs to predict the next item given a set of clothing items, as well as to compute the compatibility score of the assembled outfit.

These methods primarily employ two representation approaches: pairwise representation and sequential representation. Pairwise representation cannot reflect the complex relationships among multiple clothing items, while sequential representation imposes a fixed order that does not exist in actual outfits. More importantly, relationships among items in an outfit are not ordered, as each item relates not only to its immediate predecessor or successor in a sequence. Both pairwise and sequential representation models consider only compatibility between item pairs, making the final outfit compatibility prediction depend solely on pairwise comparisons. However, outfit compatibility depends not only on pairwise item characteristics but also on the influence of other items within the same outfit.

Therefore, the key to solving outfit compatibility lies in properly representing relationships among multiple fashion items rather than focusing solely on pairwise items. Recent studies have addressed complex relationships in outfit matching through graph neural networks. For example, Cui et al. represented outfits as graphs where each node denotes a category and each edge represents different relationships between items, then introduced an attention mechanism to output outfit compatibility scores. Subsequently, Cucurull et al. proposed treating clothing compatibility as an edge prediction problem using graph autoencoders, improving prediction performance by incorporating contextual information. Although various graph-based models have achieved successful results on outfit matching problems, traditional graph edges connect only two nodes, inherently establishing pairwise relationships and failing to fundamentally solve the problem of predicting overall outfit compatibility.

To address these issues, this paper proposes using hypergraphs to reflect the complex and high-order relationships among multiple items in an outfit. A hypergraph is a generalized concept capable of representing complex networks. In traditional graphs, the number of nodes connected by an edge is strictly defined as 2, whereas in hypergraphs, each edge can connect more than two nodes, enabling each hyperedge to represent a complete outfit. To better predict outfit compatibility, this paper proposes a novel model OCPCE to simulate interactions between outfits and fashion items. The framework is shown in Figure 1. The model first constructs a clothing hypergraph from the dataset (leftmost in Figure 1), where each hypernode represents a different clothing item and each hyperedge represents an outfit composed of multiple items (edges with the same style represent hyperedges). It then randomly selects a hyperedge for compatibility prediction. To better represent complex relationships between hyperedges and nodes, the model connects nodes within each hyperedge pairwise to form a simple graph, facilitating better propagation of node interactions. Subsequently, it aggregates neighbor information through the message passing mechanism of graph convolutional neural networks to iteratively update node state representations. Finally, when calculating clothing compatibility, unlike existing work that assumes all items contribute equally to outfit compatibility, this model introduces an attention mechanism to model the different impacts of individual items on outfit compatibility, thereby enhancing predictive capability.

1 Methodology

1.1 Problem Definition

Outfit compatibility must consider the coherence of all fashion items combined in an outfit, not just pairwise compatibility. The research objective is to model compatibility relationships between outfits and fashion items to predict overall outfit compatibility. Given an outfit collection $\mathcal{O} = \{o_1, o_2, ..., o_m\}$, where o_i represents the i-th outfit, we randomly select an outfit composed of multiple items from the collection. By modeling relationships between outfits and fashion items, we compute an overall compatibility score for the outfit and predict whether it is compatible.

1.2 Feature Extraction and Hypergraph Construction

a) Visual Feature Extraction

Fashion item images contain rich information such as color, pattern, and stripes, which greatly assist in predicting outfit compatibility. This paper utilizes convolutional neural networks to extract visual information from fashion item images. Compared to traditional feature extraction methods like SIFT, SURF, and PCA, convolutional neural networks have proven to be advanced models for image feature extraction. We select the pre-trained Inception-V3 deep neural network provided by Google for visual feature extraction. Fashion item images are input into the Inception-V3 network, and the output of its linear layer is used as the visual feature for each item. The visual feature dimension for each clothing item is 2048.

b) Text Feature Extraction

Text information for fashion items primarily comes from their titles, which mostly consist of words or short phrases. Therefore, this paper employs word embedding models to extract text features, a method proven effective in text feature extraction. First, a vocabulary is built based on words from fashion item titles in the dataset. Since titles lack standardized formatting, many meaningless words appear, such as 'a', 'an', 'de', etc. Therefore, words with fewer than three characters are filtered out to ensure vocabulary validity. After statistical processing, a vocabulary of 2757 dimensions is obtained.

c) Hypergraph Construction

This paper constructs a fashion hypergraph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ based on category information and collocation relationships between fashion items in existing datasets to simulate complex relationships between outfits and items. In the hypergraph, each node represents the category of different fashion items, and each hyperedge (edges with the same style) represents collocation relationships among multiple categories. However, not any nodes can form a hyperedge. Only nodes belonging to categories of fashion items appearing in outfits from the dataset can constitute a hyperedge. For example, nodes representing categories such as tops, bottoms, shoes, bags, and accessories can form a hyperedge. By inputting each clothing item into its corresponding node in the hypergraph, each hyperedge represents a complete outfit.

1.3 Model Design

This section first introduces notation used in the model. For each fashion item i, its visual feature is v_i, text feature is t_i, category is c_i, corresponding hypernode in the hypergraph is u_i, and the state representation of hypernode u_i in the model is h_i.

a) Node Initialization

The model' s input consists of visual features v_i and text features t_i of each fashion item, which are used to initialize the feature representation of their corresponding nodes. For each item, its visual and text features are first mapped to a space of size d. Since each item belongs to a different category, a different linear mapping matrix is required for each category when mapping to the style space. The visual and text features in the style space are then concatenated as the feature representation for each item. Finally, this feature representation is initialized as the initial feature representation of the hypernode corresponding to each fashion item in the hypergraph, as shown below:

$$h_i^0 = \tanh(W_i^c[W_i^v v_i, W_i^t t_i] + b_i)$$

b) Hyperedge-to-Graph Conversion

In the proposed OCPCE model, each hyperedge in the hypergraph represents a complete outfit. To better simulate complex and high-order relationships between outfits and clothing items, the model converts each hyperedge into a traditional graph. Currently, there are two methods for hyperedge-to-graph conversion: clique expansion, which connects all vertices in a hyperedge (e.g., a hyperedge with three vertices becomes a simple graph with three edges connecting all pairs), and star expansion, which adds a new node to each hyperedge and connects all original nodes to this new node. The second approach introduces new nodes that may propagate invalid information, causing erroneous information propagation. Therefore, this paper adopts clique expansion to convert hyperedges into simple graphs.

c) Node Interaction

Following the GGNN method, this paper utilizes graph neural networks to model node interactions on traditional graphs. Node interaction refers to each node aggregating state information from its neighbors and its own state to update its representation. After each interaction, the node state is updated as:

$$h_i^{k+1} = f\left(\sum_{u\in\mathcal{N}(i)} A_{iu}W_1 h_u^k + W_2 h_i^k + b\right)$$

where h_i^{k+1} represents the state of node i after the $(k+1)$-th interaction, W_1 and W_2 are trainable weight matrices for information propagation, and A is the adjacency matrix representing connectivity between nodes, where $A_{iu} = 1$ if nodes i and u are connected.

After aggregating neighbor state information, each node updates its final representation through Gated Recurrent Units (GRU):

$$h_i^{k+1} = \mathrm{GRU}(h_i^k, f_i^{k+1})$$

where h_i^{k+1} represents the final representation of node i after $(k+1)$ propagation steps.

d) Compatibility Score Calculation
To evaluate whether multiple clothing items form a highly compatible outfit, the OCPCE model computes the compatibility score for each outfit based on the final state representations of nodes contained in its hyperedge. Unlike existing work that simply aggregates pairwise compatibility scores, this paper argues that different fashion items have varying importance to overall outfit compatibility. Therefore, an attention mechanism is proposed to distinguish the importance of each item in an outfit. The attention mechanism is calculated as:

$$\hat{y}_u = \sigma(W_1 h_u^k + b_1)$$

$$\hat{x}_u = \delta(W_2 h_u^k + b_2)$$

where W_1 and W_2 are trainable weight matrices, σ and δ are LeakyReLU and Sigmoid activation functions respectively. $\hat{y}_u$ represents the importance of different items' impact on outfit compatibility, and $\hat{x}_u$ represents compatibility scores of individual items in the outfit.

Thus, the compatibility score for outfit o is:

$$s_o = \frac{1}{m}\sum_{u\in o}\hat{y}_u \cdot \hat{x}_u$$

where $|m|$ indicates outfit o consists of m fashion items.

1.4 Objective Function

To better predict outfit compatibility, this paper adopts the Bayesian Personalized Ranking (BPR) algorithm for this task. BPR assumes that positive sample outfits have higher compatibility scores than negative samples. The specific objective function is:

$$\mathcal{L} = -\sum_{(o,o^-)\in\mathcal{D}} \ln(\eta(s_o - s_{o^-})) + \lambda\|\Theta\|_2^2$$

where $\mathcal{D}$ is the compatibility modeling dataset, each pair (o, o^-) represents an existing outfit o (positive sample) and a non-existing outfit o^- (randomly generated negative sample), η is the Sigmoid function, Θ represents all trainable model parameters, and λ denotes regularization to prevent overfitting.

2 Experiments

2.1 Dataset

The existing Polyvore dataset originates from the popular fashion website Polyvore.com, which allows members to create fashion outfits using different clothing items or like and save others' creations. The dataset contains 164,379 individual items forming 21,899 different outfits. Graph partitioning techniques are used to split the dataset into training, validation, and test sets, with 17,316 outfits for training, 1,497 for validation, and 3,076 for testing. The split ensures no overlap between sets, meaning items appearing in the test set do not appear in the training set. Each item in the dataset contains rich information such as images, text descriptions, and categories (e.g., jeans, shirts, shoes, etc.).

If an outfit in the original dataset contains more than 8 items, it indicates duplicate items; if fewer than 3 items, the outfit is incomplete. Therefore, this paper generates a new dataset Polyvore-N by removing outfits with more than 8 items or fewer than 3 items to maintain outfit uniqueness and completeness.

2.2 Experimental Settings

All experiments select hyperparameters using the validation set, and performance comparisons are conducted on the test set. Stochastic Gradient Descent is adopted to optimize the objective function, proven effective in neural network optimization. Additionally, a grid search strategy adjusts model hyperparameters: batch size is searched in $\{8, 12, 16, 20, 24\}$, regularization rate and learning rate are fine-tuned in $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$, and propagation layers k are searched in $\{0, 1, 2, 3\}$ to achieve optimal performance. The Adam optimizer is used to optimize the entire prediction model and update parameters. All experiments are conducted on a server with a Quadro M4000 GPU. Training stops when the objective function converges or reaches the maximum number of iterations.

2.3 Comparison Methods

- **Random**: A model based on random guessing.

- **SiameseNet**: SiameseNet feeds pairs of fashion items into a Siamese network, maps them to a style space, and compares their distances. Overall outfit compatibility is computed by averaging pairwise compatibility scores.
- **Bi-LSTM**: Bi-LSTM employs bidirectional LSTM to mine sequential relationships between fashion items in an outfit for compatibility modeling and score calculation.
- **VCP**: VCP introduces graph autoencoders to compute compatibility scores between two fashion items based on their visual features and contextual information.
- **GGNN**: GGNN utilizes graph neural networks to model relationships between outfits and clothing items for compatibility score calculation.

2.4 Outfit Fill-in-the-Blank Task

The Outfit Fill-in-the-Blank (FITB) task is a standard benchmark widely used in fashion compatibility research. Given a well-coordinated outfit, one item is randomly replaced with a blank, while three other items are arbitrarily selected from the dataset as distractors to form a candidate set along with the ground-truth item. The ground-truth item is assumed to be more compatible with the original outfit than other candidates. The task aims to identify the correct item from the candidate set to fill the blank. Performance is evaluated by accuracy in selecting the correct answer from four candidates.

Table 1 compares the proposed model with alternative models on the FITB task. Key conclusions include: (1) Random performs poorly, indicating random guessing insufficiently reflects overall outfit compatibility; (2) Bi-LSTM outperforms Random, likely because it learns latent compatibility knowledge better by representing outfits as sequences and capturing high-order relationships beyond pairwise comparisons; (3) VCP, which also averages pairwise compatibility, outperforms Bi-LSTM, with improvements attributed to contextual information that better reflects item relationships, proving contextual information's effectiveness; (4) Graph-based GGNN achieves better performance, demonstrating that graph structures can effectively infer compatibility information, validating that graph representation better models fashion item interactions than sequential or pairwise representations; (5) The proposed model achieves the best performance, benefiting from hypergraph representation that effectively simulates complex and high-order relationships between outfits and items, with self-attention further capturing latent compatibility knowledge.

Figure 2 visualizes example outfits randomly selected to demonstrate the model' s FITB task performance.

Table 1: Experiment Results of Different Models in Outfit Fill-in-the-Blank Task

Model	Accuracy (FITB)
Random	24.92%
Bi-LSTM	46.24%
VCP	58.28%
GGNN	74.19%
OCPCE	77.29%

For more intuitive analysis of OCPCE versus NGNN and Bi-LSTM, Figure 3 visualizes specific sample cases from the dataset. Boxes indicate correct answers selected by models, while unboxed items indicate incorrect selections. In Example 1, all models correctly inferred the outfit needed shoes. In Example 2, OCPCE and NGNN selected correctly, while Bi-LSTM chose incorrectly due to its sequential nature—the adjacent top position heavily influenced the blank item selection, leading to pants being chosen. In Example 3, only OCPCE selected the correct fashion item. Although NGNN inferred the missing category, its selection was not the most compatible item. Bi-LSTM again chose incorrectly due to its sequential representation. OCPCE' s correct selections across all examples demonstrate that hypergraph representation effectively simulates interactions between outfits and items.

2.5 Outfit Compatibility Prediction Task

The Outfit Compatibility Prediction (CP) task aims to generate a compatibility score for a given outfit, where scores closer to 1 indicate higher compatibility and scores near 0 indicate incompatibility. This reflects real-world scenarios where users want to coordinate outfits and determine item compatibility. For evaluation, a compatible outfit set is constructed from the dataset (all outfits are compatible). An incompatible outfit set is generated by randomly selecting items from compatible outfits. By scoring both sets and comparing results, model performance is assessed using the AUC (Area Under Curve) metric from ROC curves.

Table 2 shows experimental results on the CP task. Similar to FITB, the proposed model achieves the best performance, demonstrating that hypergraphs effectively reveal high-order relationships among fashion items, enhancing predictive performance. Graph-based GGNN also shows strong performance, confirming graph structures' effectiveness. VCP with contextual information outperforms Bi-LSTM and Random but remains less competitive, indicating that modeling only pairwise compatibility or representing outfits as sequences cannot effectively predict outfit compatibility. These observations support the FITB task analysis.

Table 2: Experiment Results of Different Models in Outfit Compatibility Prediction Task

Model	AUC(CP)
Random	50.12%
Bi-LSTM	77.11%
VCP	90.13%
GGNN	94.77%
OCPCE	96.23%

2.6 Impact of Different Components

To verify each component' s contribution, an ablation study disables individual modules and compares performance. Table 3 shows results where OCPCE(-h) denotes the variant without the hypergraph module, OCPCE(-w) without attention, and OCPCE(-h-w) without both modules. Observations include: (1) The full model outperforms all ablated variants, proving the importance of both hypergraph and attention modules; (2) Disabling the hypergraph module degrades performance, confirming that hypergraphs effectively model high-order relationships between outfits and items; (3) The model without attention performs second-best, validating the necessity of the attention mechanism.

Table 3: The Effect of Different Components on Model Performance

Model	Accuracy (FITB)	AUC(CP)
OCPCE(-h-w)	75.28%	94.89%
OCPCE(-h)	75.94%	95.63%
OCPCE(-w)	76.07%	95.84%
OCPCE	77.29%	96.23%

2.8 Impact of Different Modalities

To comprehensively validate the model' s effectiveness, experiments are conducted under different modality combinations: OCPCE(VI) for visual modality only, OCPCE(TE) for text modality only, and OCPCE(VI+TE) for combined modalities. Table 4 shows performance comparisons. Findings indicate: (1) Multi-modal models outperform single-modal models, showing both text and visual modalities improve compatibility modeling; (2) Visual-only models outperform text-only models, suggesting fashion item modeling relies more on visual information (color, pattern) than text information (material, category).

Table 4: Effects of Different Modality on Model Performance

Model	Accuracy (FITB)	AUC(CP)
OCPCE(TE)	75.78%	95.28%
OCPCE(VI)	76.34%	95.63%
OCPCE(TE+VI)	77.29%	96.23%

Model	Accuracy (FITB)	AUC(CP)

3 Conclusion

To better predict outfit compatibility, this paper proposes using hypergraphs to represent relationships between outfits and items, as hyperedges can connect multiple nodes to represent a complete outfit. To better infer compatibility from hypergraphs, hyperedges are converted to traditional graphs to capture complex relationships among items. Experiments on real datasets across different fashion matching tasks demonstrate the proposed model' s effectiveness in learning fashion compatibility. However, since each user has unique aesthetic and style preferences, future research could incorporate personal preferences into outfit matching technology by quantifying user body type, skin tone, and other information to achieve personalized recommendations.

References

[1] Kang W C, Fang C, Wang Z, et al. Visually-aware fashion recommendation and design with generative image models[C]//Proc of IEEE International Conference on Data Mining. Piscataway, NJ: IEEE Press, 2017: 207-216.

[2] Feng Z, Yu Z, Yang Y, et al. Interpretable partitioned embedding for customized multi-item fashion outfit composition[C]//Proc of ACM on International Conference on Multimedia Retrieval. Piscataway, NJ: ACM Press, 2018: 143-151.

[3] Shih Y S, Chang K Y, Lin H T, et al. Compatibility family learning for item recommendation and generation[C]//Proc of the AAAI Conference on Artificial Intelligence. Piscataway, NJ: IEEE Press, 2018, 32(1).

[4] Al-Halah Z, Stiefelhagen R, Grauman K. Fashion forward: Forecasting visual style in fashion[C]//Proc of the IEEE International Conference on Computer Vision. Piscataway, NJ: IEEE Press, 2017: 388-397.

[5] Liu Rui, Peng Dunlu. A fashion compatibility learning model guided by clothing style features[J/OL]. Journal of Chinese Computer Systems: 1-6[2021-12-23]. http://kns.cnki.net/kcms/detail/21.1106.TP.20210622.1546.008.html.

[6] Yang Yiran, Wu Qiaoying. Research progress of intelligent clothing matching recommendation[J]. Journal of Zhejiang Sci-Tech University: Natural Science, 2021, 45(01): 1-12.

[7] Veit A, Kovacs B, Bell S, et al. Learning visual clothing style with heterogeneous dyadic co-occurrences[C]//Proc of International Conference on Computer Vision. Piscataway, NJ: IEEE Press, 2015: 4642-4650.

[8] McAuley J, Targett C, Shi Q, et al. Image-based recommendations on styles and substitutes[C]//Proc of the 38th International ACM SIGIR Conference on

Research and Development in Information Retrieval. Piscataway, NJ: ACM Press, 2015: 43-52.

[9] Han X, Wu Z, Jiang Y G, et al. Learning fashion compatibility with bidirectional lstms[C]//Proc of the 25th ACM International Conference on Multimedia. Piscataway, NJ: ACM Press, 2017: 1078-1086.

[10] Hong Richang, Li Lei, Cai Junjie, et al. Coherent semantic-visual indexing for large-scale image retrieval in the cloud[J]. IEEE Trans on Image Processing, 2017, 26(9): 4128-4138.

[11] Hong Richang, Yang Yang, Wang Meng, et al. Learning visual semantic relationships for efficient visual retrieval[J]. IEEE Trans on Big Data, 2015, 1(4): 152-161.

[12] Chen Long, Zhang Hanwang, Xiao Jun, et al. Counterfactual critic multi-agent training for scene graph generation[C]//Proc of IEEE International Conference on Computer Vision. Piscataway, NJ: IEEE Press, 2019.

[13] Cui Zeyu, Li Zekun, Wu Shu, et al. Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks[C]//Proc of World Wide Web Conference. San Francisco: ACM Press, 2019.

[14] Cucurull G, Taslakian P, Vazquez D. Context-aware visual compatibility prediction[C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2019: 12617-12626.

[15] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2016: 2818-2826.

[16] Ji Rongrong, Xie Xing, Yao Hongxun, et al. Mining city landmarks from blogs by graph modeling[C]//Proc of the 17th ACM International Conference on Multimedia. Piscataway, NJ: ACM Press, 2009: 105-114.

[17] Gao Yue, Wang Meng, Zha Zhengjun, et al. Visual-textual joint relevance learning for tag-based social image search[J]. IEEE Trans on Image Processing, 2012, 22(1): 363-376.

[18] Li Y, Tarlow D, Brockschmidt M, et al. Gated graph sequence neural networks[J]. IEEE Trans on Signal Processing, 2020, 68(1): 6303-6318.

[19] Maas A L, Hannun A Y, Ng A Y. Rectifier nonlinearities improve neural network acoustic models[C]//Proc of the 30th International Conference on Machine Learning. Piscataway, NJ: IEEE Press, 2013: 1-5.

[20] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback[C]//Proc of the 25th Conference on Uncertainty in Artificial Intelligence. Piscataway, NJ: ACM Press, 2013.

[21] Zhang Tong. Solving large scale linear prediction problems using stochastic gradient descent algorithms[C]//Proc of the 21st International Conference on

Machine Learning. Piscataway, NJ: IEEE Press, 2004: 116.