DEIM Forum 2020 A4-3

Fashion Dressing Match and Recommendation by a Graph Neural Network

Guan Fengming † Taro Tezuka ‡

† Graduate School of Library, Information and Media Studies, University of Tsukuba 1-2-12 Kasuga, Tsukuba, Ibaraki 305-0821 Japan

‡ Faculty of Library, Information and Media Science, University of Tsukuba 1-2 Kasuga, Tsukuba, Ibaraki 305-8550 Japan

E-mail: † guanfengming33@gmail.com, ‡ tezuka@slis.tsukuba.ac.jp

Abstract Nowadays, fashion plays an increasingly significant part of our society due to its capacity for displaying personality. However, the majority of existing algorithms in fashion recommendation, based on analyzing similarities, tend to recommend customers items that are very similar to those they have bought before.

This research aims to resolve fashion dressing match problems, which means finding a way to select new items to match with given fashion items to form a compatible outfit.

we will first use a graph to represent the data set. The nodes mean items and edges represent the interaction between nodes. Using the features that extracted from the image of the items to initialize the state of each node and model the node interactions by graph neural networks (GNN) and use the link prediction to estimate whether two items could match.

There are two tasks in fashion recommendation to measure the quality of the method: outfit compatibility prediction and fill-in-the-blank. Our experiments in these tasks show that our approach has advantages in results.

Keyword Fashion, Graph, Graph Neural Networks

1. Introduction

Fashion plays an important part in people’s daily life. In general, fashion is important because it is a reflection of every culture in the world. It was a way to distinguish different social groups and a way to be differenciated according to your status. According to Trendex’s study, the sales of woman’s dressing in United States keeps growing since 2011, representing the huge market for the fashion industry. Customers’ demands of fashion recommendation are also rising with the rapid development of fashion market. What's more, they often need advice on fashion dressing match, which suggests an item that fits well with an existing set. So far, the most existing algorithm[1], based on recognition techniques, tends to recommend coustomers items that similar to those they have bought before.

Different from clothing recognition, fashion dressing match is much more complicated and sophisticated because of the subjective. The key to do recommendation of matching problem is to calculate the score of the compatibility. In this paper, we propose to use graph neural network to make compatibility predictions. We use graph structure to demonstrate the relationship of products, the nodes represent items and each edge connect pairs of items that appear in the same outfit. The feature of nodes will be the visual feature extracted from the image of items by GoogleNet InceptionV3[2], then we use a graph attention network model to gather the information form their neighbour nodes to learn link-prediction problem.

We are inspired by Cui et al’s[3] experiment, which put forward to use graph to represent fashion data set and using NGNN (node-wise graph neural network). Our model is based on the encoder-decoder architecture. An encoder using GCN to encodes nodes to a vector.

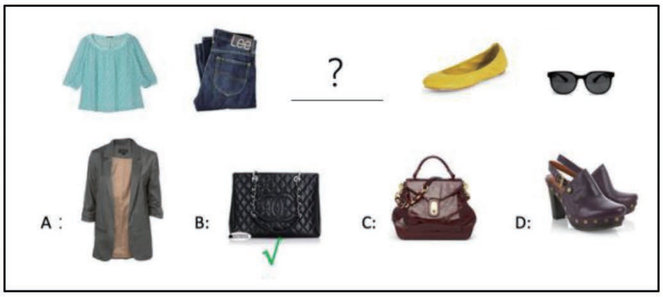
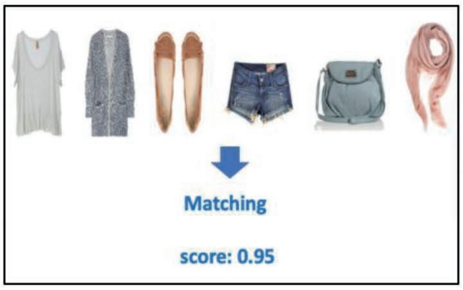
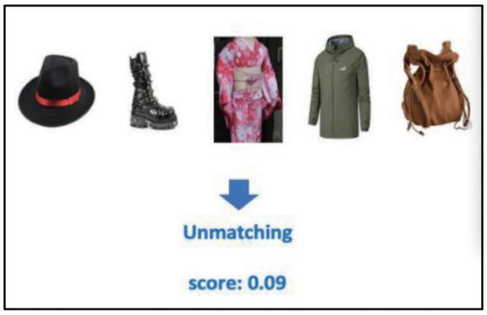


Figure 1: Fill-in-the-blank



(a)



(b)

Figure 2: Compatibility prediction. The score (between 0 and 1) shown in (a) (b) represents whether this outfits are matching or not.

We conduct experiments on two tasks proposed before: (1) Fill-in-the-blank: choosing an item from several items that fits a given set of components of an outfit; (2) Compatibility prediction: predicting the compatibility of an outfit and estimate a score for the match. Our experiments result on the Polyvore data set show that our approach has advantages in results.

2. Releated work

2.1Fashion recommendation

Fashion recomendation is a very important application of computer vision[4]. Traditionally, the majority of research in this domain focus on Fashion Recognition and Retrieval[5] and also some task based on fashion image attribute learning[6]. Specifically, there are studies of estimating the compatibility of fashion outfits. Veit et al.[7] proposed to learn clothing matches based on the Amazon co-purchase data set, and Iwata et al.[8] put forward a topic model to recommend "Tops" for "Bottoms". McAuley et al.[9] used Low-rank Mahalanobis Transformation to map pairs of items into a latent space and measured the distance. Li et al.[10] utilized an Recurrent Neural Network (RNN) to extracted multi-modal information from fashion items. Cui et al.[3] started to represent outfits as graph and using node-wise graph nerual network(NGNN) to realize the embedding part, however the NGNN was based on GGNN, and it will be less effective when dealing with bigger graph. In our experiment, we use GCN in the encoder and represent the items instead of categories as nodes. Besides, we connected each couple of nodes that appear in the same outfit with an edge.

2.2Graph Neural Networks

Graph Neural Networks (GNNs) were introduced by Gori et al.[11] and Scarselli et al.[12] as a generalization of recursive neural networks that can directly deal with a more general class of graphs. The original GNNs consist of an iterative process, which propagates the node states until equilibrium.

Li et al.[13] first proposed the Gated Graph Neural Network (GGNN), which using Gated Recurrent Units for updating. N.Kipf et al.[14] provides Graph Convolutional Network (GCN) motivated by a localized first-order approximation of spectral graph convolutions. Hamilton et al.[15] come up with GraphSAGE, which computes node representations in an inductive manner. This technique operates by sampling a fixed-size neighborhood of each node, and then performing a specific aggregator over it. The graph attention network was created by Veličković et al.[16], which using attention mechanism to decide the weight of different neighbours of the node.

3. Data preprocessing

3.1Dataset

Polyvore is a very popular fashion website (www.polyvore.com), where users upload and comment outfit data. Xintong Han's group collected data from the internet into a dataset. This dataset of fashion outfits contains rich and comprehensive information about the outfits. It includes information of all the items and outfits like images, catagories, titles, number of people who like and view the outfit and so on. The whole dataset includes 21,889 outfits, and it has been split into 17,316 for training, 1,497 for validation and 3,076 for testing.

3.2Feature extract

As same as Cui's experiment, we apply GoogleNet InceptionV3 model to extract the visual features from the images of all the items, which has been proved to be advanced in representing the image. The visual feature of each item is a 2048-dimensional vector and are normalized before embedding.

4. Approach

Our model is based on the graph auto-encoder framework defined by Kipf et al[17]. In the framework, an incomplete graph will be input to the encoder part. We will then use a Graph Convolutional Network (GCN) to make embedding for each node. After that, the new state vector of nodes will include their neighbours' feature. Then, the new vectors of nodes are used by the decoder part to predict the missing edges in the graph.

The dataset can be represented as an undirected graph as $G = (V, E)$, with V denoting the set of nodes and E representing the edges in network G. This graph has N nodes $i \in V$ and edges $(i, j) \in E$ denote the relationship of node pairs. The state of nodes is represented by H = $\{h_1, h_2, \dots, h_N\}$, and $h_i \in \mathbb{R}^F$, H is an matrix of $\mathbb{R}^{N\times F}$, We use $h_i^T \in \mathbb{R}^{\tilde{F}}$ to denote GCN's outputs that also contain $\tilde{F}$ node features. N_i represent all the neighbors of the node i.

The essence of GCN is the weighted summation of the features of neighboring nodes. This computation process is shown in equation (1), where Node j is a neighbor of node i, and N_i represent all the neighbors of the node i. w is the weight matrix.

$$h_i^{t+1} = ReLU(\textstyle\sum_{j\in N_i} \frac{1}{|N_i|} h_j^t w_1^t + h_i^t w_2^t) \qquad (1)$$

After the approach of encoder, we could get the final state of every node i which is h_i^T. Then we use the decoder to computes the posibility that two nodes could be linked. The method used in decoder is known as metric learning[18], which has been use in many other fields[19]. Usually metric learning is defined as learning a function d(·, ·): $\mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}_0^+$ that represents the distance between two N-dimensional vectors. Therefore, our decoder function was first proposed from other metric learning approaches[20]. The equation (2) shows the process of decoder.

$$p = \sigma(|h_i^T - h_j^T|\widetilde{w} + b) \qquad (2)$$

The decoder outputs p denotes the probability that pairs of nodes i and j are connected by an edge. Here | * | is absolute value, and and b are parameters. The output p should be in the interval [0; 1].

The whole model is trained to predict compatibility among the items. At the beginning we have an edge set E, before we start training we randomly delete some edges that should be appeared before and using those edge to construct a subset. The set of edges removed is denoted by E^+, and the set of left one named $\tilde{E}$. As to the part of edges that shouldn't be there at the beginning, we represent them as a set E^- Training the encoder-decoder model to predict the edges in Etrain = (E^+, E^-)that contain both positive and negative edges. The output of the model should be a score of every edge and the ground truth value of edges in $E^+$is 1, and for edges in E^+ is 0.

5. Experiment

5.1 Fill in the blank (FITB).

The Polyvore dataset contains a total of 164379 items that form 21899 different outfits. The maximum number of items per outfit is 8, and the average number of items per outfit is 6:5. The fill-in-the-blank question aims at choosing the most suitable item from a given set to match an outfit. The FITB task totally contains 3076 questions.

FITB task can be framed as a link-prediction (edge prediction) problem. Using S (include N items s) to denote the set of items in the outfit and C (include M items c) to symbolize the set of items could possibly be chosen. Our model could generate the probability scores of edges between item pairs as $p_{ij} = (s_i, c_j)$ for all i = 0,1...N-1 and j = 0,1...M-1. We could finally get the score of each c_j as $\sum_{j=0} p_{ij}$, and chose the item with highest score as figure 3.

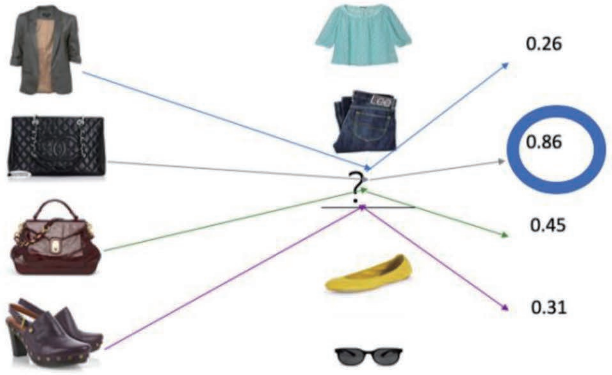
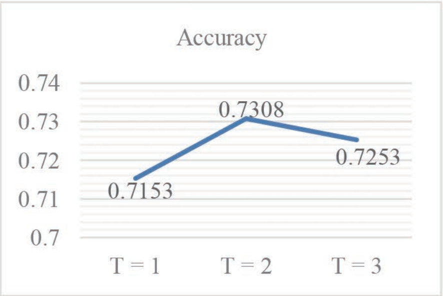


Figure3: finish the fill-in-the-blank task by choosing the item that gets the best score

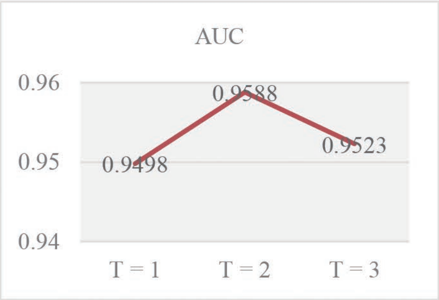
5.2 Compatibility task

The compatibility task has 3076 appropriate outfits and 4000 improper outfits that have been chosen randomly. This task is similar to the filling-in-the-blank task. However it is based on the situation that someone may want to get fashion recommendation by judging the score(from 0 to 1)of the outfits chosen by themselves. In this task we still use link prediction. The compatibility score of the outfit is the average of all possible score of every combination of two items. The performance of task is evaluated using the AUC of the ROC curve[21].

5.3 Parameter Setting



(a)



(b)

Figure 4: The accuracy and auc result of different setting of layer

In the experiment we test different number of graph convolutional layers and find that when we use 2 layers we could get the best performance in both the accuracy and the auc.

We use the Relu function for activation and RMSProp for optimization. For the hyper-parameters of this experiment, we set the training process as 10 epochs (early stop), the learning rate as 0.001, batch size as 16 and T (graph convolutional layers) as 2.

5.4 Experimental Result

method	Accuracy (1 task)	AUC (2 task)
Random	24.07%	0.5104
SiameseCNNs	48.09%	0.7087
LMT	50.91%	0.6782
Bi-LSTM	67.01%	0.8427
GCN	73.08%	0.9588

Table 1: Comparison of different models

For the two tasks, we compared the results of our method with lots of releated works as Table 1.

6. Conclusion

In this experiment, we use GCN with a link-prediction method to predict the score of the outfits. Instead of focusing on finding the similarity of different items to meet people's needs, we tend to focus on the matching problem and get a good result. In future research, we will keep using graph neural networks and also adjust the parameters to get better performance.

7. Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP16K00228, JP16H02904.

References

[1] Liu Z, Luo P, Qiu S, et al. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 1096-1104.

[2] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2818–2826

[3] Cui Z, Li Z, Wu S, et al. Dressing as a Whole: Outfit Compatibility Learning Based on Node-wise Graph Neural Networks[C]//The World Wide Web Conference. ACM, 2019: 307-317.

[4] K. Chen, K. Chen, P. Cong, W. H. Hsu, and J. Luo, "Who are the devils wearing prada in new york city?" in Proceedings of the 23rd ACM international conference on Multimedia. ACM, 2015, pp. 177–180.

[5] Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan, "Street-to-shop:Cross-scenario clothing retrieval via parts alignment and auxiliary set,"in CVPR, 2012, pp. 3330–3337.

[6] Q. Chen, J. Huang, R. S. Feris, L. M. Brown, J. Dong, and S. Yan,"Deep domain adaptation for describing people based on fine-grained

[7] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. J. Belongie,"Learning visual clothing style with heterogeneous dyadic co-occurrences,"ICCV, 2015. [Online]. Available: http://arxiv.org/abs/1509.07473

[8] T. Iwata, S. Watanabe, and H. Sawada, "Fashion coordinates recommender system using photographs from fashion magazines," in IJCAI, 2011.

[9] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel.2015. Image-based recommendations on styles and substitutes. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 43–52.

[10] Yuncheng Li, Liangliang Cao, Jiang Zhu, and Jiebo Luo. 2017. Mining fashion outfit composition using an end-to-end deep learning approach on set data. IEEE Transactions on Multimedia 19, 8 (2017), 1946–1955.

[11] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains.In IEEE International Joint Conference on Neural Networks, pp. 729734, 2005.

[12] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. IEEE Transactions on Neural Networks, 20(1):61–80, 2009.

[13] Li Y, Tarlow D, Brockschmidt M, et al. Gated graph sequence neural networks[J]. arXiv preprint arXiv:1511.05493, 2015.

[14] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

[15] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. Neural Information Processing Systems (NIPS), 2017.

[16] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017.

[17] T. N. Kipf and M. Welling. Variational graph auto-encoders.In NIPS Workshop on Bayesian Deep Learning, 2016. 2, 3

[18] A. Bellet, A. Habrard, and M. Sebban. A survey on metric learning for feature vectors and structured data. arXiv preprint arXiv:1306.6709, 2013. 4

[19] Xing E P, Jordan M I, Russell S J, et al. Distance metric learning with application to clustering with side-information[C]//Advances in neural information processing systems. 2003: 521-528

[20] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In ICML Deep Learning Workshop, 2015.

[21] Bradley A P. The use of the area under the ROC curve in the evaluation of machine learning algorithms[J]. Pattern recognition, 1997, 30(7): 1145-1159.