

A Product Similarity Method Based on Deep Confidence Network

Hong Liao¹, Zhuchao Yu^{1,*}, Yaxin Cao¹, Mengjin Du¹ and Chengcheng Sun²

¹School of Business Administration, Northeastern University, No. 195 Chuangxin Road, Hunnan District, Shenyang, 110169, China

²Institute of Software, Northeastern University, No. 195 Chuangxin Road, Hunnan District, Shenyang, 110169, China

*Corresponding author

Keywords: Deep confidence network, Product similarity, Deep learning, Product recommendation network, Shortest path.

Abstract. To improve product recommendation network, this paper mainly proposes a product similarity calculation algorithm based on deep confidence network. A high-dimensional product is firstly constructed and then input into the DBN model to obtain low-dimensional product feature data. Founded on the low-dimensional product feature data, the similarity between products can be calculated by the cosine formula. Through the data experiment, it is found that as the output dimension of the low-dimensional product feature matrix decreases, the similarity of the product similarity matrix also decreases, which means that the information extracted from the original input matrix is refined, and the effective information of a product is increasing, which means that the information extracted from the original input matrix is refined, and the effective information of a product is increasing.

Introduction

The rapid development of new technologies has resulted in information overload, information trek and other problems under the Internet environment. Recommendation algorithms and recommendation systems are effective means to address such problems. It has become an important research topic in the fields of electronic commerce and intelligent information processing to provide suitable product recommendation technology for Internet consumers.

Recommendation algorithms are essentially machine learning, while deep learning is a kind of machine learning. The concept of deep learning originated in the study of artificial neural network, and show the strong concentration of learning essential characteristics of data sets from a few sample capacity [1]. In recent years, deep learning has become a craze of artificial intelligence, which brings new opportunities for the research of recommendation system [2].

Deep learning provides a more accurate and faster method for the recommendation algorithm [3]. This paper will calculate the similarity between products based on the deep confidence network model in deep learning for recommending more accurate products to users.

Related Work

The product recommendation network was aimed to build relationships between products within the e-commerce website and presented them to customers in the form of visual links, thereby providing customers with more choices.

Most of the previous studies combined the input and output networks without differentiation to form a comprehensive factor (such as PageRank). Thus they are unable to separate the demand impact of the two networks [4]. Carmi et al. studied that the direction of product links is important as it directs consumers' attention and generates traffic to a product [5]. LIN et al. examine product recommendation networks in e-commerce to investigate how the demand of a product is influenced by product network attributes in terms of network diversity and network stability [6].

In the application of enterprise-level recommendation system, the music platform Spotify uses the circular neural network to predict the user's behavior, analyzes the song style by deep learning, and

makes recommendation according to the music style [7]. Netflix and Google successfully applied deep neural network RNN to their video recommendation system [8]. Microsoft applied deep learning to applications and news recommendations, and achieved good results. With the improvement of big data processing capacity, deep learning has been applied more widely.

The DBN (Deep Belief Nets) network to be used in this paper is a machine learning model emerging in recent years, which was proposed by Geoffrey Hinton in 2006. It combines unsupervised learning process with supervised learning process and has made great achievements in the field of image processing, but its application in the field of information extraction still needs to be developed.

Method

The product similarity calculation algorithm based on the deep confidence network first calculates the shortest path between the product node pairs, converts the original product recommendation network into a product relationship matrix, and converts it into a high-dimensional product relationship matrix. This high-dimensional product relationship matrix is then used as an input matrix into the DBN model to obtain low-dimensional product feature data, and the cosine formula is used to calculate the similarity between products based on the low-dimensional product feature data.

Input Matrix

In the product recommendation network, the relationship between different products is not only the presence or absence, but also the difference in distance. The length of the shortest path of a pair of product nodes represents the closeness of the product. To more accurately express feature weight, the original dataset is formed by the shortest paths between product nodes. The product relationship matrix of a product is converted from a $1 \times N$ matrix to a $6 \times N$ matrix, where N is the total number of products. Suppose that the shortest path of a product node and the remaining product nodes ranges from 0 to 5, then the corresponding position in the output matrix is 1, and the rest is 0.

DBN Model

DBN model is a generative model. By training the weights between neurons, the entire neural network can generate training data according to the maximum probability. The DBN model adopted in this paper is composed of two layers of RBM, and the output layer has 50 output units.

Restricted Boltzmann Machines (RBM) is the component of the DBN model. The RBM network has several parameters, the weight matrix $W_{n \times m}$ between the visible layer and the hidden layer, the offset of the visible node $b = (b_1, b_2, \dots, b_m)$, and the offset of the hidden node $c = (c_1, c_2, \dots, c_n)$. These parameters determine what kind of n -dimensional sample the RBM network encodes from a m -dimensional sample.

DBN is a neural network composed of multi-layer RBM, which can be regarded as both a generating model and a discriminant model. The training process can be described as using unsupervised greedy layer by layer method to pre-train and obtain weights.

The structure of the deep confidence network is shown in Figure 1, where, $P(v, h^1, h^2, \dots, h^l) = P(v|h^1)P(h^1|h^2) \dots P(h^{l-2}|h^{l-1})P(h^{l-1}|h^l)$.

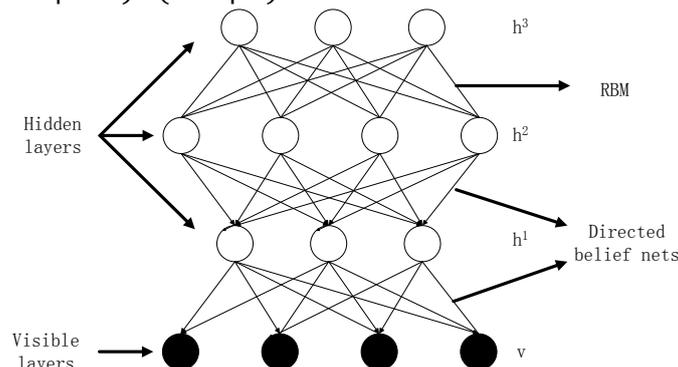


Figure 1. Structure of Directed Belief Nets

For the output data, the size of the top-level output matrix of the network also becomes $M*L$. To calculate the similarity between products. It can be converted into a $1*L$ matrix by a formula.

$$P_j = \sum_{i=0}^{M-1} P_{output_i,j} * (i+1) \quad (1)$$

Calculate Product Similarity Matrix

Assume P_i, P_j are the product feature vectors after the model training, then the similarity of product i, j is:

$$\cos(\theta) = \frac{P_i * P_j}{|P_i| |P_j|} \quad (2)$$

Through the above formula, the similarity between all products can be calculated to obtain the product similarity matrix.

The specific flow of the algorithm is as follows:

The input is the product relationship matrix relation and the unit number of top-level network in DBN model is L , and the final Output is a similarity matrix between products.

The algorithm steps are follows:

- (i) Create a $M*N$ empty matrix for a product, convert it into a $M*N$ 0,1 matrix;
- (ii) Input the 0,1 matrix of a product into the two-layer DBN model, perform layer-by-layer training, adjust parameters, and finally obtain L output feature values;
- (iii) Convert the $M*L$ matrix into a $1*L$ matrix according to;
- (iv) Repeat 1-3 processes for each product to obtain low-dimensional features of each product;
- (v) Calculate the similarity matrix between products by using the cosine formula.

A product similarity matrix A_L is obtained by calculating the similarity of all products. In order to evaluate the dimension reduction results, the product similarity matrix A_{170} calculated according to the product feature matrix with an output dimension of 170 is defined, and the similarity of the product similarity matrix calculated by the product feature matrix of different output dimensions L and A_{170} is defined as:

$$\text{similarity} = \frac{1 - (A_L - A_{170})^2}{A_{170}^2} \quad (3)$$

Data Experiment

Original Dataset

The e-commerce platform will recommend a number of products for users in a list, named as "Customers who bought this item also bought" (co-purchase). Meanwhile, there are visual links between focus products and recommended products, through which users can reach the recommended product pages and get detailed information about products. The original dataset can be collected by the following steps: the webpage of a certain product is treated as an initial layer, and products in the co-purchase column are treated as the second layer which can be accessed through hyperlinks; after recording relationships of products in these two layers, products in the co-purchase column under each page of products in the second layer are treated as the third layer.

Experimental Result

Comparing the product similarity matrix calculated by the product feature matrix with the output dimensions of 10, 15, 20, 30, 40, 50, 70, 100, 130, 150 and the product feature matrix with the output

dimension of 170. The similarities between product similarity matrices are shown in Figure 2.

As the output dimension of the product feature matrix decreases, the similarity of the product similarity matrix also decreases, which means that the information extracted from the original input matrix is refined, and effective information of a product is increasing. While considering the simplicity of calculation, in order to retain the effective information to the greatest extent, in the actual dimension reduction process, the product feature matrix with output dimension 50 is used as the basis for calculating the similarity matrix of products, and based on this, the product recommendation network is improved.

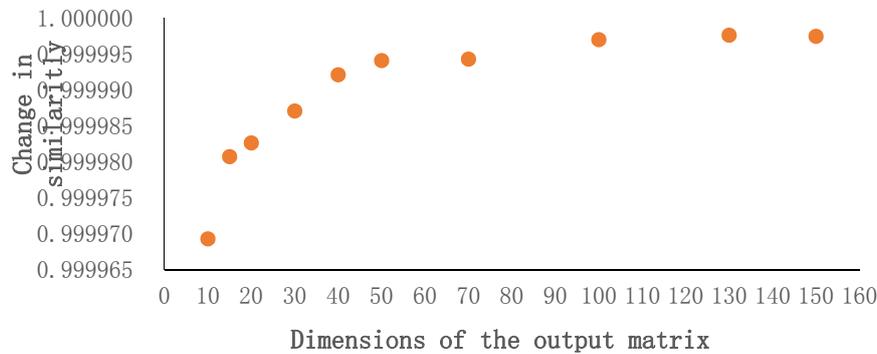


Figure 2. The similarities between the product similarity matrices

Conclusion

This paper mainly proposes a product similarity calculation algorithm based on deep confidence network, providing an improvement method of product recommendation network. Through data experiments, it is found that as the output dimension of the low-dimensional product feature matrix decreases, the similarity of the product similarity matrix also decreases, which means that the information extracted from the original input matrix is refined, and the effective information of a product is increasing.

Acknowledgements

Our thanks to the National Natural Science Foundation of China under projects No. 71171043, and the National Training Programs of Innovation and Entrepreneurship for Undergraduates in Northeastern University, China.

References

- [1] Hecht-Nielsen R. Theory of the backpropagation neural network [J]. *Neural Networks*, 1988, 1(Supplement-1): 445-448.
- [2] Oestreicher-Singer, G.; Sundararajan, A. Recommendation networks and the long tail of electronic commerce [J]. *MIS Quarterly*, 2012, 36: 65-83.
- [3] Wehrmann J, et al. Adult Content Detection in Videos with Convolutional and Recurrent Neural Networks [J]. *Neurocomputing*, 2017, 227: 432-438.
- [4] Oestreicher-Singer, G; Sundararajan A. The Visible Hand? Demand Effects of Recommendation Networks in Electronic Markets [J]. *Manage. Sci.*, 2012, 58: 1963-1981.
- [5] Oestreicher-Singer, G., Sivan, L. The network value of products [J]. *Journal of Marketing*, 2013, 77(3): 1-14.
- [6] Lin Z. J., Khim-Yong G.; Cheng-Suang H. The Demand Effects of Product Recommendation Networks: An Empirical Analysis of Network Diversity and structural stability [J]. *MIS Quarterly*, 2017, 41: 397-426.

- [7] Peng Y, Zhu W, Zhao Y, et al. Cross-media analysis and reasoning: advances and directions [J]. *Frontiers of Information Technology & Electronic Engineering*, 2017, 18(1): 44-57.
- [8] Liu Z H, Lai Y C, Ye N, et al. Connective distribution and attack tolerance of general networks with both preferential and random attachments [J]. *Physics Letters A*, 2002, 303(5-6): 337-344.
- [9] Hecht-Nielsen R. Theory of the backpropagation neural network [J]. *Neural Networks*, 1988, 1(Supplement-1): 445-448.