

DSSMFM: Combining user and item feature interactions for recommendation systems

Weishan Zeng*

Beijing University of Posts and Telecommunications, Beijing, China

Keywords: DSSM, FM, Feature extraction, User attributes vector, Item attributes vector.

Abstract. Effort has been done to optimize machine learning algorithms by applying relevant knowledges in data fields in recommendation systems. Ways are explored to discover the relationship of features independently, making the model more effective and robust. A new model, DSSMFM is proposed in this paper which combines user and item features interactions to improve the performance of recommendation systems. In this model, data are divided into user features and item features represented by one-hot vectors. The pre-training for the model is proceeded through FM, and implicit vectors are obtained for both user and item features. The implicit vectors are used as the input of DSSM, and the training of the DSSM part of the model will maximize the cosine distances of the user attributes vectors and the item attributes vectors. According to the experimental results on dataset of ICME 2019 Short Video Understanding and Recommendation Challenge, the model shows improvements on some results of the baselines.

1 Introduction

Undoubtedly, feature play an essential role in many recommendation systems, and a good feature brings about significant benefits to the model and algorithm itself. However, it is rare to acquire the best results by using original features directly. The feature engineering aims to get better training data through relevant knowledge in the data field, but there are obvious shortcomings in traditional feature engineering inevitably. One, in the huge real scenario of the industry, features are in quantity and complicated, which makes manual feature extraction infeasible. Two, manual feature extraction substitutes into prior factors, which, to some extent, effaces the relationship between potential features. Three, each new data set needs to be rebuilt. Different from the traditional machine learning that is a process composed of multiple independent modules which contain complicated feature engineering, some deep learning models are about "end-to-end". According to the end-to-end network, it makes the model can learn independently without human intervention. So, in the field of recommendation systems, it is of great significant to learn complicated and selective feature interaction [1] by using DNN. The deep neural network is good at high-order feature

* Corresponding author: wzsender@qq.com

presentation, but it is weak in interpretability and cross-learning of low-order features, which limits the performance of systems. Therefore, it is hoped to find a way that does not need sundry manual feature engineering, and meanwhile, combines vector interactive representation of low-order feature that are applicable to large and scattered data and high-order feature presentation and powerful interpretability. According to the research, DSSM [2] is to map query and doc into the space of common dimension; a model with corresponding representation meaning can be obtained through DNN structure and maximizing cosine similarity between query and doc vectors. FM [3] algorithm refers to a machine learning algorithm based on matrix decomposition, which aims to explore feature portfolio in large-scale sparse data. In addition, FM explores cross-correlation of low-order features automatically.

In summary, in this paper, we propose a neural network-based model, DSSMFM-Deep Structured Semantic Model & Factorization Machine, and learned feature interaction through a distinct vectorization way. The potential relationships between user and item were expressed via cosine distance constraints.

2 Preliminaries

2.1 Embedding layer

In the recommendation systems, the data are usually sparse, with large characteristic dimension, and the spatial - temporal relationship among each other is weak. Therefore, it is difficult to directly apply dnn in data sets with such features. In other words, it is necessary to complete the switch from vectors characterized by high dimension and sparsity to those by low-dimensional and denseness; the latter ones tend to update constantly, and the relationship among features will be enhanced. For example, the input data is [userid = uid02, authorid = aid01, ..., itemid = iid02], then it will be represented as a high-dimensional sparse vector by One-Hot coding [4]:

$$\underbrace{[0, 1, 0, \dots, 0]}_{\text{Userid}} \underbrace{[1, 0, 0, \dots, 0]}_{\text{Authorid}} \dots \underbrace{[0, 1, \dots, 0]}_{\text{Itemid}}$$

The output form of the embedded layer is as follows:

$$e = [e_1, e_2, \dots, e_m]$$

The above m represents the number of characteristic fields, and $e_i \in \mathbb{R}^D$ is the embeddings of a field.

2.2 Features split

Briefly speaking, the DSSM model describes the relationship between query and doc. We used it to find the relationships between user and item, and distinguished features of data, with acquired user features and item features as input to DSSM. It aims to describe the mapping relationships between user vector and item vector. As for user attributes vector and item attributes vector after training, similar feature vectors will gather together into clusters in multi-dimensional space, and the formed user cluster and item cluster become more closely related under the constraint of maximizing cosine distance, such as figure 2a and 2b.

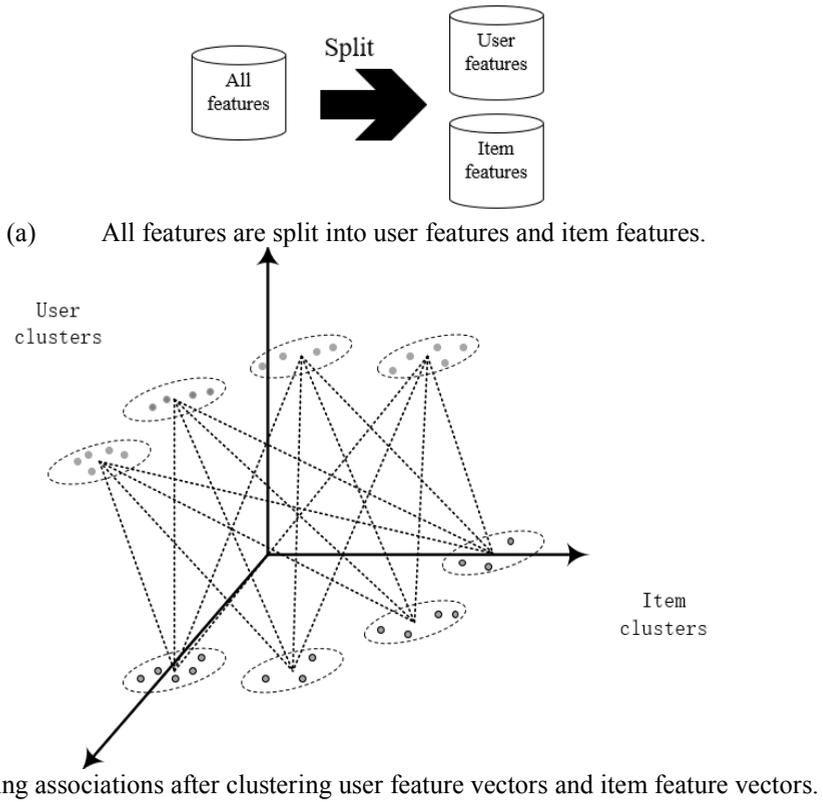


Fig. 2. Features splitting and mapping.

3 DSSMFM Model

3.1 The FM component

The two parts of the model were mainly used. The first part is Factorization Machine that performs the implicit vector of features as inner product to realize the cross of the feature.

$$\hat{y}(x) := w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j \quad (1)$$

$w_0 \in \mathbb{R}, w_i \in \mathbb{R}, \mathbf{V} \in \mathbb{R}^n \times k, \langle \mathbf{v}_i, \mathbf{v}_j \rangle := \sum_{f=1}^k v_{i,f} \cdot v_{j,f}$, Where k is the vector dimension and N is the number of features, w_0 is the global offset, w_i is the weight of feature i , $\widehat{w}_{i,j} := \langle \mathbf{v}_i, \mathbf{v}_j \rangle$ is the weight of correlation degree between feature i and feature j .

In FM, the user and item features were used as training data input of FM, acquiring implicit vectors corresponding to user and item features based on the consideration of global features. Each feature was given a bias weight and k -dimension vector, and the feature interaction was modeled as inner product of its vector $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$. In the training of One-Hot coding data input, FM learned good structured data representation in potential space, which is conducive to building model. In addition, the spatial position relation of each feature was granted by implicit vector of each feature obtained by FM pre-training.

3.2 The DSSM component

The second part is Deep Structured Semantic Models. In this part, on each side is a DNN, with one on the user side and the other on the item side. DNN on both sides of DSSM loaded the implicit vector of each feature obtained from FM pre-training in the first part. Besides loading the feature vector belonging to user side, DNN on user side also loaded overall feature vector. DNN on item side performed training in the same way as DNN on user side.

$$l_1 = w_1x \tag{2}$$

$$l_i = f(w_i l_{i-1} + b_i) \tag{3}$$

$$y = f(w_n l_{n-1} + b_n) \tag{4}$$

$$R(Q, D) = \text{cosine}(y_Q, y_D) = \frac{y_Q^T y_D}{\|y_Q\| \|y_D\|} \tag{5}$$

x is the input vector, y is the output vector, l_i denotes hidden layer i , w_i is the parameter matrix of layer i , b_i is the offset i . y_Q, y_D denotes user side output vector and item side output vector. $R(Q, D)$ is the relevance score.

3.3 The combination component

The architecture of the whole model is shown in Figure 3. First, the data were encoded with One-Hot code, input to FM for pre-training, acquiring each feature implicit vector. Then, both sides of DSSM loaded user feature + overall feature and item feature + overall feature respectively to undergo maximized cosine distance training, and to draw the conclusions.

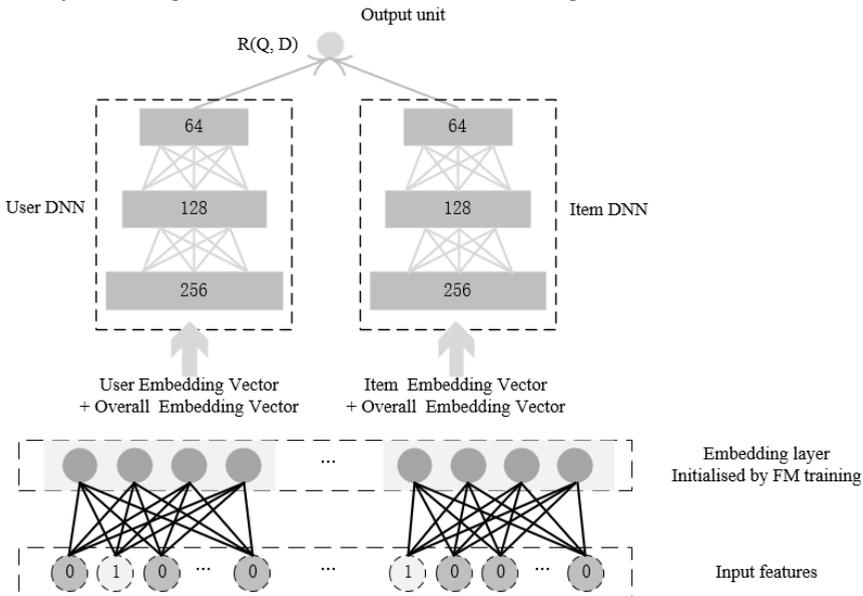


Fig. 3. The architecture of DSSMFEM.

4 Experiments

4.1 Experiment setup

Data. We use the dataset of ICME 2019 Short Video Understanding and Recommendation Challenge to evaluate our model, which is a short video Click dataset, including short video information and user viewing feedback. The characteristic of this dataset is summarized in Table 1. The dataset consists of 15536829 instances with different lines of user viewing feedback information, of which 1 / 3 are positive labels, and the field attributes include user ID, user's city, video ID, author ID, video city, source of watching video, music ID, user device ID, video release time, video duration, and whether the viewing is completed. User attributes features are composed of user ID, user's city, source of watching video, and user device ID. item attributes features are composed of video ID, author ID, video city, music ID, video release time, and video duration. The number of user attributes features is 137978, and the number of item attributes features is 4024934. Filter out the missing values, and the proportion of training data, verification data and test data are 7:2:1.

Table 1. Statistics of the evaluation dataset. M indicates million and K indicates thousand.

Dataset	Instances	Fields	User features (sparse)	Item features (sparse)
copy	15.53M	10	137K	4M

Models. We use the following model for comparison:

LR: Logistic Regression-A linear model with simple implementation and fast training speed.

FM: Factorization Machine-A non-linear model based on matrix decomposition, which is suitable for many sparse data and the exploration of cross relationship between multi-dimensional features.

DNN: Deep Neural Networks-A model for the composition of multi-layer fully connected layers.

FNN [5]: Factorization-machine supported Neural Networks-A DNN model basing on FM.

DSSM: Deep Structured Semantic Model-A semantic model based on depth network by maximizing cosine distance training.

WIDE&DEEP: Wide & Deep Model [6]-A model to explore low and high order features, memory ability of integrated linear model and generalization ability of DNN model.

DEEPPFM: DeepFm [7]-A model based on FM and DNN cooperative training.

DCN: Deep & Cross Networks [8]-A model of learning the interaction of high-order features.

XDEEPPFM: xDeepFm [9]-A compression interaction model of learning explicit and implicit high-order feature interaction.

Metric. To evaluate the performance of each CTR model, we used the MSE and the area under ROC curve (AUC). The AUC metric is widely used measure for evaluating the CTR performance.

4.2 Performance comparison

Each feature value is treated as word, and the user word list and item word list are formed to prepare for the subsequent DSSM training. It is sent to FM (xlearn) for pre-training, and all feature hidden vectors are obtained. The dimension of feature hidden vector is 64. DSSM loads hidden vectors of features, and inputs user attributes features + overall features and

item attributes features + overall features on both sides for training. In this experiment, the learning rate of deep neural network is 1×10^{-6} , the drop rate is 0.1, the batch size is 10000, and the regularize ratio is 0.001. Experimental data are show in Table 2.

Table 2. Overall performance of different models on the dataset of ICME 2019 short video understanding and recommendation challenge.

Model name	AUC	Depth	Model name	AUC	Depth
LR	68.68%	-	WIDE&DEEP	71.50%	DNN:3
FM	69.47%	-	DEEPPFM	72.34%	DNN:3
DNN	72.50%	DNN:3	DCN	72.86%	Cross:3, DNN:3
FNN	74.03%	DNN:3	XDEEPPFM	71.91%	CIN:2, DNN:3
DSSM	71.29%	DNN:3	DSSMFM	74.62%	DNN:3

5 Summary

In this paper, it proposed a neural network-based model DSSMFM. DSSMFM had two special points: One, it reduced the manual feature engineering process and was capable of interpretation. Two, it was able to learn updated constantly high-order features interaction via the representation of low-order features interaction effectively; it learned the mapping relationship between user and item by maximizing the cosine distance between user attributes and item attributes feature vector. The experimental results showed that performance of our model was better than the traditional model. In the future, more work should be made on describing feature representation on both sides, adjusting network structure, strengthening the learning ability of the model in one feature, and trying to achieve better performance through collaborative training.

6 Acknowledgement

The author would like to thank the teachers and students for their advice, which is very meaningful for the revision of this paper. This work is supported by NSFC 61773037.

References

1. Covington P, Adams J, Sargin E. Deep neural networks for youtube recommendations[C]//Proceedings of the 10th ACM conference on recommender systems. ACM, 2016: 191-198.
2. Huang P S, He X, Gao J, et al. Learning deep structured semantic models for web search using clickthrough data[C]//Proceedings of the 22nd ACM international conference on Information & Knowledge Management. ACM, 2013: 2333-2338.
3. Rendle S. Factorization machines[C]//2010 IEEE International Conference on Data Mining. IEEE, 2010: 995-1000.
4. Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.
5. Zhang W, Du T, Wang J. Deep Learning over Multi-field Categorical Data[J]. 2016.
6. Cheng H T, Koc L, Harmsen J, et al. Wide & Deep Learning for Recommender Systems[J]. 2016.

7. Guo H, Tang R, Ye Y, et al. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction [J]. 2017.
8. Wang R, Fu B, Fu G, et al. Deep & cross network for ad click predictions[C]//Proceedings of the ADKDD'17. ACM, 2017: 12.
9. Lian J, Zhou X, Zhang F, et al. xdeepfm: Combining explicit and implicit feature interactions for recommender systems[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2018: 1754-1763.
10. Elkahky A M, Song Y, He X. A multi-view deep learning approach for cross domain user modeling in recommendation systems[C]//Proceedings of the 24th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2015: 278-288.
11. Han X, Leung T, Jia Y, et al. Matchnet: Unifying feature and metric learning for patch-based matching[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3279-3286.