

# Diversity Regularized Interests Modeling for Recommender Systems

Junmei Hao<sup>1</sup>, Jingcheng Shi<sup>1\*</sup>, Qing Da<sup>1</sup>, Anxiang Zeng<sup>2</sup> and Yujie Dun<sup>3</sup> and Xueming Qian<sup>3</sup> and Qianying Lin<sup>1</sup>

<sup>1</sup>Alibaba Group

<sup>2</sup>Xi'an Jiaotong University

{junmei.hjm@alibaba-inc.com, jingcheng.sjc, daqing.dq, qianying.lqy}@alibaba-inc.com,  
renzhong@taobao.com, {dunjy, qianxm}@mail.xjtu.edu.cn

## Abstract

With the rapid development of E-commerce and the increase in the quantity of items, users are presented with more items hence their interests broaden. It is increasingly difficult to model user intentions with traditional methods, which model the user's preference for an item by combining a single user vector and an item vector. Recently, some methods are proposed to generate multiple user interest vectors and achieve better performance compared to traditional methods. However, empirical studies demonstrate that **vectors generated from these multi-interests methods are sometimes homogeneous**, which may lead to sub-optimal performance. In this paper, we propose a novel method of **Diversity Regularized Interests Modeling (DRIM)** for Recommender Systems. We apply a capsule network in a multi-interest extractor to generate multiple user interest vectors. Each interest of the user should have a certain degree of distinction, thus we introduce **three strategies as the diversity regularized separator** to separate multiple user interest vectors. Experimental results on public and industrial data sets demonstrate the ability of the model to capture different interests of a user and the superior performance of the proposed approach.

## 1 Introduction

With the rapid development of E-commerce over years, recommender systems play an increasingly important role in the E-commerce platform. In general, the recommender systems consist of two stages, the matching stage and the ranking stage. The matching stage is mainly matching users with relevant items, quickly retrieving a fraction of items that users are potentially interested in from the massive inventory, and then handing them to the ranking stage. The ranking stage assigns a score to each item according to a desired objective function. Intuitively, the main purpose of the two stages is to learn user and item representations to support efficient retrieval

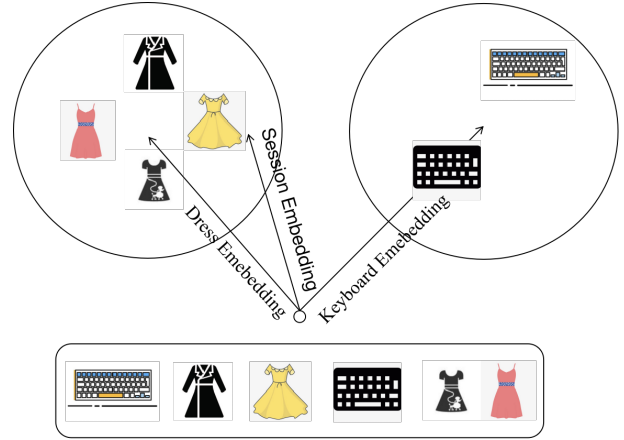


Figure 1: A single vector can not express multiple interests.

and ranking of items for users. Some recent works leverage the algorithms to learn how to represent the user interest vector. Collaborative filtering-based methods [Sarwar *et al.*, 2001][Koren *et al.*, 2009] extract user interests from historical behaviors, which are the preferred for new recommender systems in most cases but may suffer from sparsity problem during computation. Thus, deep learning-based methods are introduced in recommender systems to model user interests with low-dimensional embedding vectors. For example, the deep neural network proposed for YouTube video recommendation (YouTube DNN) [Covington *et al.*, 2016] generates one fixed-length vector for each user from the historical behaviors of users. However, using one vector to represent the user assumes that the user only has a single preferred interest within a session. As can be seen in Figure 1, if a user clicks dresses multiple times and keyboards few times within a session, the learned user vector is likely very close to dresses. During the matching stage, the nearest neighbor algorithm matches items related to dresses. However, keyboards are also of interest to a user. Multiple vectors that represent different interests of users are thus necessary.

As user interests broaden, user modeling also needs to be more expressive. Deep Interest Network (DIN) [Zhou *et al.*, 2017] applies local activation unit on user historical behaviors to adaptively capture the diversity of user interests. How-

\*Contact Author

ever, since the candidate size for the matching stage is on the scale of billions, recalculating user representation for each item is computationally infeasible. Multi-interest modeling in the matching stage requires greater model expressiveness. Greater model expressiveness entails more parameters, which introduces huge costs for computation, storage, and model optimization. MIND [Li *et al.*, 2019] applies the capsule network to generate multiple user vectors not only to reduce additional cost but also to improve the performance. Moreover, ComiRec [Cen *et al.*, 2020] modifies MIND for considering the order information. However, neither MIND nor ComiRec considers the diversity constraints of multiple user interests. In extreme cases, vectors generated from the MIND and ComiRec are the same.

In this paper, we propose a novel diversity regularized interest model for recommender systems, called DRIM. We firstly generate multiple interest vectors through a capsule network in our model. To prevent the generated interest vectors from converging to the same one, different interest vectors should be regularized. Thus, we introduce three strategies as the diversity regularized separators to discriminate user multiple interests. The diversity of multiple user vectors is controllable in our model through the three diversity regularized strategies. Each interest of the user has a certain degree of distinction, regularizing the diversity of user multiple interests in this model. These user vectors are in the matching stage for retrieving relevant items from billion-scale items.

To summarize, the main contributions of this work are as follows:

- We propose a comprehensive framework that generates multiple interests for a user and integrates the diversity-regularizing mechanism of multi-interest components.
- We introduce three different diversity regularized separators to model user multiple interests, improving both complexity and accuracy.
- Our framework achieves state-of-the-art performance on two real-world challenging datasets for the recommender systems.

**Organization:** The rest of the paper is organized as follows. Section 2 summarizes related work of our model. Section 3 introduces our proposed framework in detail. In Section 4, we conduct extensive experiments and case studies. Finally, we conclude in Section 5.

## 2 Background and Related Work

In this section, we introduce the related work about traditional models and deep learning methods for recommender systems, as well as capsule networks we used in this paper.

### 2.1 Traditional Model for Recommendation

Collaborative filtering methods [Sarwar *et al.*, 2001][Koren *et al.*, 2009] are main traditional methods used in recommender systems. Collaborative filtering methods make recommendations based on user-item similarity. Rendle *et al.* [Rendle *et al.*, 2010] combine matrix factorization with personalized Markov chain to model both the long-term intents of

users and the sequence effects. Following this work, Liang *et al.* [Liang *et al.*, 2016] propose a co-factor model, combining matrix factorization with item embedding to improve the performance of standard matrix factorization and to model the sequence pattern. Factorization Machines (FMs) [Rendle, 2010] model all interactions between variables using factorized parameters and thus can resolve sparsity problems in recommender systems.

### 2.2 Deep Learning for Recommendation

Due to the significant improvement in performance compared to traditional models, deep learning has been integrated into many industry-scale recommender systems. Neural Collaborative Filtering (NCF) [He *et al.*, 2017] uses a multi-layer neural network to model the interaction between users and items. Neural Factorization Machines (NFM) [He and Chua, 2017] fully combines the second-order linear feature extracted by FM and the higher-order nonlinear feature extracted by the neural network. Furthermore, the low-order and high-order combination features can be extracted at the same time by DeepFM [Guo *et al.*, 2017]. Deep & Cross Network (DCN)[Wang *et al.*, 2017] has higher computational efficiency and can extract higher-order crossover features.

### 2.3 Capsule Network

The concept of "capsules" is first proposed by Hinton *et al.* [Hinton *et al.*, 2011] in 2011. They consider a capsule as a group of neurons whose activity vectors represent the instantiation parameters of a specific type of entity such as an object or an object part. The length of the output vector of a capsule represents the probability that the entity represented by the capsule is in the current input. Next, the dynamic routing method [Sabour *et al.*, 2017] is introduced to learn the weights on the connections between capsules. Afterwards, Hinton *et al.* [Hinton *et al.*, 2018] propose expectation-maximization algorithm to overcome several deficiencies. Stacked Capsule Autoencoders (SCAE) [Kosiorek *et al.*, 2019] uses geometric relationships between parts to reason about objects. The capsule network has been applied in recommender systems recently. MIND [Li *et al.*, 2019] utilizes dynamic routing mechanisms in recommender systems to capture multiple interests of users in the matching stage. Following this, ComiRec [Cen *et al.*, 2020] modifies MIND for considering the order information to apply the capsule network in the sequential recommendation. The main difference with existing methods is that our new diversity regularized interest model adapts MIND by adding a diversity separator, where each interest of the user has a certain degree of distinction to better model multi-interests.

## 3 Model

In this section, we formulate the problem and introduce the diversity regularized interests model in detail.

### 3.1 Problem Formulation

The recommender system is a two-stage system, with two stages, matching and ranking. The purpose of our work is to generate user vectors for the matching stage to model multi-interests. We have a set of users  $u \in U$ , a set of items  $i \in I$ ,

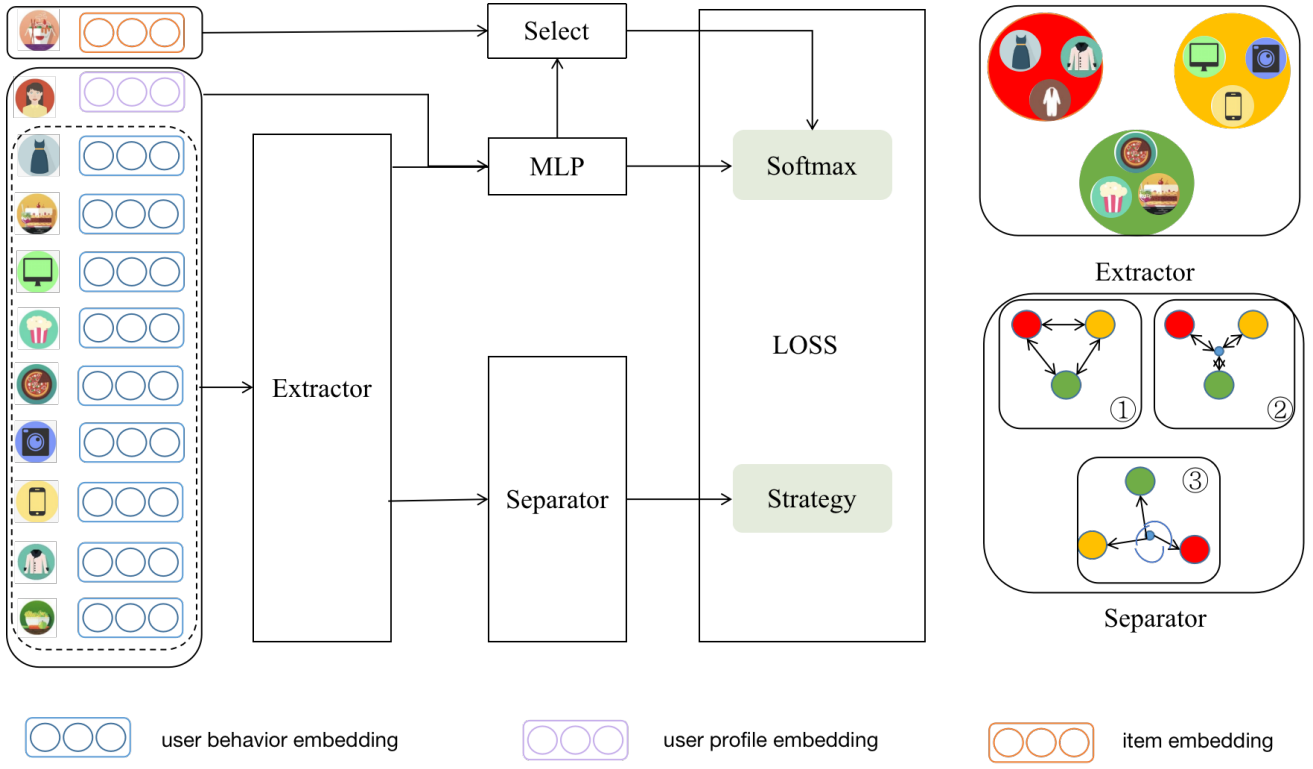


Figure 2: The Diversity Regularized Interests Model framework. **The extractor takes user behaviors as input and outputs user multiple vectors.** Id features from the input layer are transformed into embeddings. User behavior embeddings are fed into the multi-interest extractor layer, which produces **interest capsules**. The red, yellow, green circles represent the different interests respectively. Then the diversity separator (① or ② or ③, one of the three) applies strategies on interest capsules for regularizing user vectors to some extent diverse. ① represents the max entropy loss, ② represents the mean square loss, and ③ represents the diverse loss. By concatenating interest capsules with user profile embedding and transforming the concatenated capsules by multilayer perceptron, user representation vectors are obtained. **An extra select function is introduced to guide the training process.** Furthermore, strategy and softmax loss are jointly applied to the training process.

and a sequence of user historical behaviors  $h_u^i$  for each user. Our model can generate multiple vectors representing user interests  $V_u$ , where  $V_u = (v_u^1, v_u^2, \dots, v_u^K) \in R^{d \times K}$  given the user historical behaviors  $h_u^i$  as input. Notations are summarized in Table 1.

Table 1: Notations

### 3.2 Incorporate with profile

As shown in Figure 2, the input of our model is user profile and user history behaviors. The history behaviors consist of a list of item IDs. The item IDs are transformed into item embeddings through an embedding layer. To enrich the feature space, user profile(user id, gender, etc) are also fed into the embedding layer. A multi-interest extractor module and a diversity separator module receive the embedding of user history behaviors and generate multiple diverse interests for each user.

In this paper, **we apply the clustering process to aggregate the user’s historical behaviors into several clusters.** A cluster of items represents a user’s particular interest. Here we not only design the multi-interest extraction layer to generate multiple user interest vectors but also design diversity separators to regularize the diversity of multi-interests.

Notation	Description
$u$	a user
$i$	an item
$U$	the set of users
$I$	the set of items
$h_u^i$	user historical behaviors
$d$	the dimension of user/item embeddings
$K$	the number of interest embeddings
$V_u$	the matrix of interest embeddings of user $u$
$c_{ij}$	the routing logit
$S$	the bilinear mapping matrix
$b_{ij}$	the coupling coefficients

### 3.3 Multi-Interest Extractor

Our model designs the multi-interest extractor based on the dynamic routing method, for representation learning in the capsule network.

#### Dynamic Routing.

We utilize a dynamic routing method to capture multiple interests for users. We consider a two-layer capsule structure, which includes the history layer and interest layer respectively. The history layer represents the user history behavior and the interest layer represents the user interests. We use the dynamic routing method from CapsNet. We adopt dynamic routing for computing vector inputs and outputs of capsules. Let  $h_i \in R^d$  represent the capsule  $i$  of the history layer and  $v_u^j \in R^d$  represent the interest capsule  $j$ . We calculate the capsule  $j$  of the interest layer based on the history layer. The routing logit  $c_{ij}$  between history capsule  $i$  and interest capsule  $j$  is calculated by

$$c_{ij} = v_j^T S h_i, \quad (1)$$

where  $S \in R^{d \times d}$  denotes the bilinear mapping matrix parameter shared across each pair of history and interest capsules.

The  $b_{ij}$  are the coupling coefficients between history capsules  $i$  and interest capsules  $j$ . For particular history capsule  $i$ , the coupling coefficients between all the interest capsules and it sum to 1. It is calculated by applying softmax on routing logits as

$$b_{ij} = \frac{\exp(c_{ij})}{\sum_t \exp(c_{it})}, \quad (2)$$

As the coupling coefficients are calculated, the candidate vector for interest capsule  $j$  is calculated as

$$z_j = \sum_i b_{ij} S h_i, \quad (3)$$

Then, the embedding of interest capsule  $j$  can be obtained with a non-linear "function" as

$$v_j = \text{squash}(z_j) = \frac{\|z_j\|^2}{1 + \|z_j\|^2} \frac{z_j}{\|z_j\|}, \quad (4)$$

#### Argmax operator to select one particular interest for target item in training

After obtaining the interest embedding through a multi-interest extraction layer based on user history behaviors, we adopt an argmax operator to choose a corresponding user interest embedding vector for a target item  $i$ , since a particular target item belongs to one interest in common sense:

$$v = V_u[:, \text{argmax}_{1 \leq j \leq K} (v_j^T e_i)], \quad (5)$$

where  $e_i$  denotes the embedding of the target item  $i$  and  $K$  is the number of interest embeddings.

### 3.4 Diversity Regularized Separators

To obtain diverse interests of users extracted from the history capsules, we propose a separator layer to make the interest capsule distinct. The main idea of the diversity separator layer is to introduce three loss functions to increase the distance among the interest clusters and to regularize the diversity.

#### Max Entropy Loss

According to the law of entropy growth and SCAE [Kosiorek et al., 2019], with the degree of confusion increasing, the entropy value also increases. We apply to maximize between-interest capsules entropy to regularize the diversity of the interest capsules vector.

$$\text{loss}_{\text{entropy}} = - \sum_{k=1}^K H(v_k) \quad (6)$$

#### Mean Square Loss

To increase the diversity of multiple interests of a user  $u$ , we make the distance between each interest vector and the mean vector much farther. That means the mean square of the error between each specific user interest vector and the mean vector of a user  $u$  becomes much larger. To do that, the distance of the interest differentiation is obvious as shown in 2, which is more helpful to regularize multi-interest vectors diversity.

$$\text{loss}_{\text{mean}} = - \sum_{k=1}^K (v_k - \text{mean}(v_k))^2 \quad (7)$$

#### Diverse Loss

Inspired by Yu et al. [Yu et al., 2011], we utilize the sum of a pairwise difference to measure the total diversity. This is a metric to measure the effectiveness of multiple interests separation.

Thus, for a pair of vectors  $v_i$  and  $v_j$ , we measure their diversity using the angle between them

$$\text{loss}_{\text{div}} = 1 - \frac{v_i^T v_j}{\|v_i\| \cdot \|v_j\|} \quad (8)$$

### 3.5 Joint Training

A joint training framework is proposed here by optimizing two losses together: a softmax loss for the matching objective and one of the three diversity separator losses for interest capsule differentiation. Those two representations are fused for the final prediction.

#### Softmax Loss

After obtaining the user particular interest embedding vector  $v_u$  and the target item embedding  $e_i$ , we can compute the probability of the user  $u$  interacting with the target item  $i$  as

$$P(i|u) = \frac{\exp(v_u^T e_i)}{\sum_{k \in I} \exp(v_u^T e_k)}, \quad (9)$$

The objective function of our model is to minimize the following negative log-likelihood

$$\text{loss}_{\text{softmax}} = \sum_{u \in U} \sum_{i \in I} -\log P(i|u). \quad (10)$$

#### Separator Loss

Separator losses are introduced in 3.4. The three loss functions are independent.

Joint loss training enables the network to simultaneously train for user vector accuracy and interest vector differentiation:

$$loss_{DRIM-entropy} = loss_{softmax} + \lambda loss_{entropy} \quad (11)$$

$$loss_{DRIM-mean} = loss_{softmax} + \lambda loss_{mean} \quad (12)$$

$$loss_{DRIM-div} = loss_{softmax} + \lambda loss_{div} \quad (13)$$

We call them DRIM-entropy, DRIM-mean, DRIM-div for short. These loss functions balance the accuracy and diversity of the recommendation by a controllable factor  $\lambda \geq 0$ .

### 3.6 Serving

At serving time, the user’s behavior sequence and user profile are fed into the model, producing multiple representation vectors for each user. Then, these representation vectors are used to retrieve top  $N$  items by an approximate nearest neighbor approach. These items constitute the final set of candidate items for the matching stage of recommender systems. Please note that, when a user has new actions, it will alter his/her behavior sequence as well as the corresponding user representation vectors, thus our model enables real-time inference for the recommendation matching stage.

## 4 Experimental and Evaluation

In this section, we evaluate the performance between our methods and existing methods on several datasets. The statistics of the two datasets are shown in Table 2.

### 4.1 Datasets

We use two datasets for evaluating performance, Amazon Books<sup>1</sup> and OURS respectively. Amazon Books is one of the most widely-used public datasets for e-commerce recommendations. OURS is a dataset generated from the log of one international E-commerce App, containing historical behaviors of randomly sampled 1100000 of users in 2 weeks. For Amazon Books, we only keep items that have been reviewed at least 20 times and users who have reviewed at least 20 items. For OURS, we only keep items that have been reviewed at least 10 times and users who have reviewed at least 10 items.

Since the main task of the matching stage is the next item prediction problem, we choose it to evaluate the methods’ performance. The user behavior sequence is sorted by time. We hold the first 80% user sorted behavior sequence as the training set and the rest user behavior sequence as the test set. In Amazon Books, each training sample is truncated at length 10. And in OURS, each training sample is truncated at length 50.

<sup>1</sup><http://jmcauley.ucsd.edu/data/amazon/>

### 4.2 Evaluation Metrics

We use the following metric to evaluate the performance of our proposed model. Hit rate (HR) measures the percentage that recommended items contain at least one correct item interacted by the user, which is a commonly used evaluation criterion.

$$HR@N = \frac{\sum_{u \in U} I(N)}{|U|} \quad (14)$$

where  $U$  denotes the number of users in the test set and  $I(N)$  denotes the indicator function meaning whether the target item occurs in top  $N$  or not.

Table 2: Statistics of datasets

Datasets	# users	# items	# interactions
Amazon Books	173,901	163,328	4,910,406
OURS	1,100,000	531,830	18,221,613

### 4.3 Parameter Configuration

The number of dimensions  $d$  for user and item embeddings is set to 36. We sample 5 negative samples for sampled softmax loss. We use Adam optimizer [Kingma and Ba, 2015] with learning rate  $lr = 0.001$  for optimization.

### 4.4 Comparing Methods

We compare our method with the following baselines. We compare our proposed models, DRIM-entropy, DRIM-mean, DRIM-div with state-of-the-art models. In our experimental setting, models should give the prediction for the users of test sets.

- **MostPopular** is a traditional recommendation method that only recommends items to a user according to the popularity.
- **YouTube DNN [Covington *et al.*, 2016]** is one of the most successful deep learning model used for industrial recommender systems.
- **MIND [Li *et al.*, 2019]** is related to our model. It designs a multi-interest extractor layer based on a capsule network, for clustering past behaviors and extracting interests.
- **ComiRec [Cen *et al.*, 2020]** is a recent state-of-the-art model. This model integrates the multi-interest components and controllable aggregation module in unified recommender systems.

### 4.5 Experimental Results

For multiple interest models, each user representation vector independently retrieves top- $N$  candidate items. Thus, our model retrieves a total  $K * N$  items for each user. We sort the items by the inner product of the item embedding and the corresponding user interest representation vector. After sorting, top- $N$  items from these  $K * N$  items are viewed as the final candidate items of the models.

Table 3: Performance comparison on the two datasets

Dataset	Metric	Most Popular	YoutubeDNN	MIND	COMIREC	DRIM-entropy	DRIM-mean	DRIM-div
Amazon Books	HR@50	0.0155	0.4749	0.5796	0.7892	<b>0.8062</b>	0.6463	0.7073
	HR@100	0.0219	0.5848	0.7136	0.8785	<b>0.8957</b>	0.7275	0.8298
OURS	HR@50	0.0136	0.2240	0.3436	0.3542	0.3540	0.3672	<b>0.4145</b>
	HR@100	0.0173	0.4030	0.4235	0.4573	0.4351	0.4630	<b>0.5663</b>

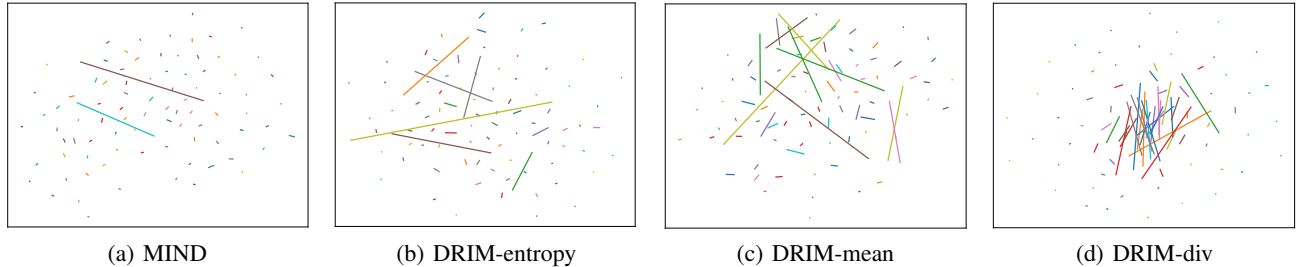


Figure 3: The dimensionality reduction figures of Amazon Books on our proposed DRIM and MIND.

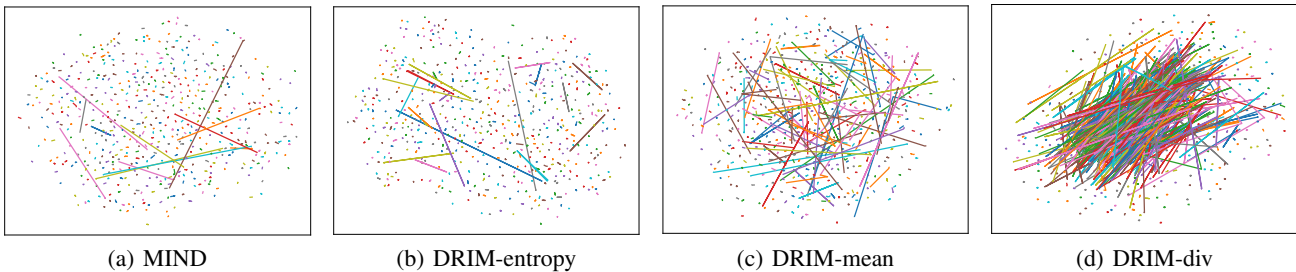


Figure 4: The dimensionality reduction figures of OURS on our proposed DRIM and MIND.

Table 3 summarizes the performance of our model and all baselines on two datasets in terms of HR@ $N$  ( $N=50,100$ ). Our model surpasses all of the baselines by a wide margin on both datasets. The non-personalized method, MostPopular, is beaten by other methods, revealing the power of the personalized feature for improving the matching stage of recommender systems. It can be observed that methods employing multiple user representation vectors perform better than employing single user representation vector. Therefore, multi-interest modeling is effective for modelling user’s diverse interests as well as boosting recommendation accuracy. Moreover, we can observe that ComiRec-DR outperforms MIND due to the difference of the dynamic routing method which can make user representation vectors distinguishable.

#### 4.6 Model Visualization

We have a closer look at some trained user interest vectors. For demonstration purpose only, **we set the number of interests to 2 in this part**. We first apply t-Distributed Stochastic Neighbor Embedding (t-SNE) [Husnain *et al.*, 2019] to reduce user interest vector dimension into 2 so each user has 2 points mapped from interest vectors. We connect the 2 points of the same user by lines and plot the points and lines in the

Figure 3 and 4.

Then, we connect the points mapped from the interest vectors of a user using line segments. The greater the segment length, the better differentiation between interest vectors. We can see that most line segments in Figure 3(a) and 4(a) are short which means the embeddings from MIND are not well separated. In stark contrast, there are many more longer lines in (b), (c), (d) in Figure 3 and 4 which means the embeddings generated by DRIM are better separated.

## 5 Conclusion

In this paper, we propose a novel method of Diversity Regularized Interests Modeling for recommender systems, namely DRIM, to explore users’ diverse interests and regularize the diversity of the multiple interests for the matching stage in e-commerce recommendation. Specifically, we design a multi-interest extractor layer with a variant of dynamic routing to extract users’ diverse interests and a diversity regularized separator layer with three regularization strategies to regularize the diversity of interests. Empirical study indicates that DRIM achieves superior performance on public benchmarks.



## References

- [Cen *et al.*, 2020] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. Controllable multi-interest framework for recommendation. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 2942–2951. ACM, 2020.
- [Covington *et al.*, 2016] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, pages 191–198. ACM, 2016.
- [Guo *et al.*, 2017] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: A factorization-machine based neural network for CTR prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1725–1731. ijcai.org, 2017.
- [He and Chua, 2017] Xiangnan He and Tat-Seng Chua. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 355–364. ACM, 2017.
- [He *et al.*, 2017] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 173–182. ACM, 2017.
- [Hinton *et al.*, 2011] Geoffrey E. Hinton, Alex Krizhevsky, and Sida D. Wang. Transforming auto-encoders. In *Artificial Neural Networks and Machine Learning - ICANN 2011 - 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I*, volume 6791 of *Lecture Notes in Computer Science*, pages 44–51. Springer, 2011.
- [Hinton *et al.*, 2018] Geoffrey E. Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with EM routing. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.
- [Husnain *et al.*, 2019] Mujtaba Husnain, Malik Muhammad Saad Missen, Shahzad Mumtaz, Muhammad Muzamil Luqman, Mickaël Coustaty, and Jean-Marc Ogier. Visualization of high-dimensional data by pairwise fusion matrices using t-sne. *Symmetry*, 11(1):107, 2019.
- [Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [Koren *et al.*, 2009] Yehuda Koren, Robert M. Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- [Kosiorrek *et al.*, 2019] Adam R. Kosiorrek, Sara Sabour, Yee Whye Teh, and Geoffrey E. Hinton. Stacked capsule autoencoders. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15486–15496, 2019.
- [Li *et al.*, 2019] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. Multi-interest network with dynamic routing for recommendation at tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 2615–2623. ACM, 2019.
- [Liang *et al.*, 2016] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M. Blei. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, pages 59–66. ACM, 2016.
- [Rendle *et al.*, 2010] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 811–820. ACM, 2010.
- [Rendle, 2010] Steffen Rendle. Factorization machines. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, pages 995–1000. IEEE Computer Society, 2010.
- [Sabour *et al.*, 2017] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 3856–3866, 2017.
- [Sarwar *et al.*, 2001] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, pages 285–295. ACM, 2001.
- [Wang *et al.*, 2017] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17, Halifax, NS, Canada, August 13 - 17, 2017*, pages 12:1–12:7. ACM, 2017.
- [Yu *et al.*, 2011] Yang Yu, Yu-Feng Li, and Zhi-Hua Zhou. Diversity regularized machine. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 1603–1608. IJCAI/AAAI, 2011.
- [Zhou *et al.*, 2017] Guorui Zhou, Chengru Song, Xiaoqiang Zhu, Xiao Ma, Yanghui Yan, Xingya Dai, Han Zhu, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. *CoRR*, abs/1706.06978, 2017.