Joint Training Capsule Network for Cold Start Recommendation

Tingting Liang
Hangzhou Dianzi University
Hangzhou, China
liangtt@hdu.edu.cn

Congying Xia
University of Illinois at Chicago
Chicago, US
cxia8@uic.edu

Yuyu Yin
Hangzhou Dianzi University
Hangzhou, China
yinyuyu@hdu.edu.cn

Philip S. Yu
University of Illinois at Chicago
Chicago, US
psyu@uic.edu

ABSTRACT

This paper proposes a novel neural network, joint training capsule network (JTCN), for the cold start recommendation task. We propose to mimic the high-level user preference other than the raw interaction history based on the side information for the fresh users. Specifically, an attentive capsule layer is proposed to aggregate high-level user preference from the low-level interaction history via a dynamic routing-by-agreement mechanism. Moreover, JTCN jointly trains the loss for mimicking the user preference and the softmax loss for the recommendation together in an end-to-end manner. Experiments on two publicly available datasets demonstrate the effectiveness of the proposed model. JTCN improves other state-of-the-art methods at least 7.07% for CiteULike and 16.85% for Amazon in terms of Recall@100 in cold start recommendation.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

Recommender systems, Cold start, User preference estimation

ACM Reference Format:
Tingting Liang, Congying Xia, Yuyu Yin, and Philip S. Yu. 2020. Joint Training Capsule Network for Cold Start Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401243

1 INTRODUCTION

The effectiveness of current recommender systems highly relies on the interactions between users and items. These systems usually do not perform well when new users or new items arrive. This challenge is widely known as cold start recommendation [7]. To alleviate this problem, models have been proposed to leverage side information such as user attributes [1] or user social network data [7, 12] to generate recommendations for new users. These models can be grouped into three categories based on how they use the side information: similarity-based models [11] which calculate similarities between items based on the side information; matrix factorization methods with regularization [3] that regularize the latent features based on auxiliary relations; matrix factorization methods with feature mapping [2] which learn a mapping between the side information and latent features. According to [6], these cold start models can be viewed within a simple unified linear framework which learns a mapping between the side information and the interaction history.

Recently, deep learning models (DNNs) have emerged to tackle the cold start problem by providing a larger model capacity. Volkovs et al. [13] regard cold start recommendation as a data missing problem and modifies the learning procedure by applying dropout to input mini-batches. It highly depends on the generalization ability of the dropout technique to generalize the model from warm start to cold start. [6] is the first work that proposes to solve the cold start problem in the Zero-shot Learning (ZSL) perspective. It leverages a low-rank auto-encoder to reconstruct interaction history from the user attributes. However, it is a two-step method which firstly learns the reconstruction and then solves the recommendation for the cold start users or items in the second step. A two-step method might suffer from the error propagation problem.

To avoid the aforementioned problems and fully understand the content in the side information, we propose an end-to-end joint training capsule network (JTCN) for cold start recommendation. In JTCN, a user is represented explicitly in two folds: the high-level user preference and the content contained in the side information. The high-level user preference is aggregated from the low-level interaction history through the attentive capsule layer with a dynamic routing-by-agreement mechanism [10, 15].

A mimic loss is proposed to mimic the high-level user preference for cold start users or items from the side information. We argue that it is more explainable to mimic the high-level user preference than the low-level interaction history. It would be natural to infer user preference from the side information other than non-existent interaction history. Another softmax loss is used to train the regular recommendation process. Our goal is to not only mimic the high-level user preference for the cold start users or items, but also effectively do recommendations for them. We propose to achieve our goals by jointly training these two losses together in an end-to-end manner. In summary, the contributions of this paper are:

- **Joint Training**: A joint training framework is proposed for the cold start recommendation by training the mimic loss for the cold start and the softmax loss for the recommendation together.
- **Capsule Network**: An attentive capsule layer is proposed to aggregate high-level user preference from the low-level interaction history via a dynamic routing-by-agreement mechanism.
- **Demonstrated Effectiveness**: Experiments on two real-world datasets show that our proposed model outperforms baselines consistently for the cold start recommendation task.

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401243

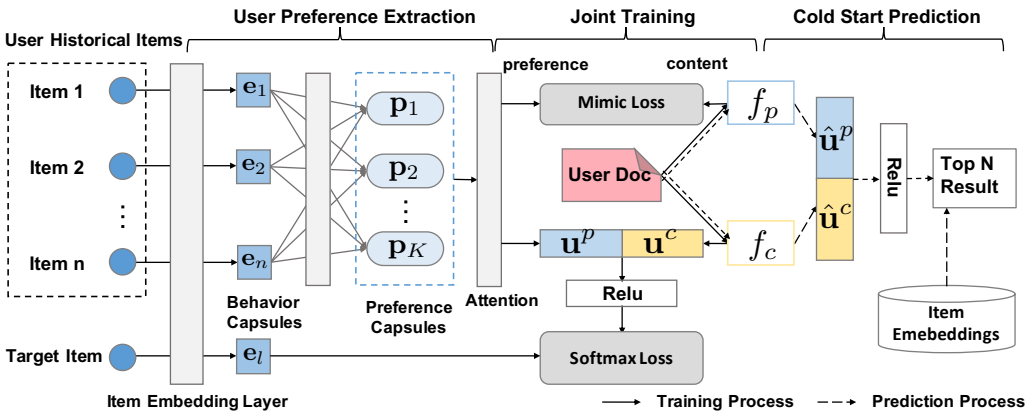


Figure 1: The architecture of JTCN. Id features of items are transformed into embeddings through the embedding layer. The embeddings of historical items are fed into the attentive multi-preference extraction layer, which consists of one multi-preference extraction layer and one attention layer, to obtain the user preference representation. The content representation produced by network f_c is fused with the preference embedding to form the softmax loss. The output of network f_p is used to approximate user preference through a mimic loss. When making predictions for a new user, f_c and f_p is able to generate a comprehensive representation for the user based on the side information, including both preference and content information.

2 PROPOSED MODEL

2.1 Problem Statement

We consider a recommender system with a user set $\mathcal{U} = \{u_1, \ldots, u_N\}$ and an item set $\mathcal{V} = \{v_1, \ldots, v_M\}$, where N is the number of users and M is the number of items. The user-item feedback can be represented by a matrix $R \in \mathbb{R}^{N \times M}$, where $R_{ij} = 1$ if user i gives a positive feedback on item j, and $R_{ij} = 0$ otherwise. Let $\mathcal{U}(j) = \{i \in \mathcal{U} | R_{ij} \neq 0\}$ be the set of users that had shown preference to item j, and $\mathcal{V}(i) = \{j \in \mathcal{V} | R_{ij} \neq 0\}$ be the set of items that user i gave positive feedback. This paper focuses on the cold start scenario in which no preference clue is available, namely, $\mathcal{V}(i) = \emptyset$ or $\mathcal{U}(j) = \emptyset$ for a given user i or given item j. Our objective is to generate personalized recommendation results for each fresh user or item based on its corresponding side information.

2.2 User Multi-Preference Extraction

The framework of our proposed model is illustrated in Figure 1. The framework here is mainly for cold start users, and the framework for cold start items can be modeled in the same manner. As shown in Figure 1, the input of JTCN consists of user documents which contain the side information (labeled as User Doc in the figure), historical items, and the target item. The former two can be respectively used for extracting user properties and preferences. The target item is the one that we use to make a prediction for the user during the training process. The usage of user documents will be discussed in Section 2.3. This section focuses on the extraction of high-level user preferences.

2.2.1 Embedding Layer. The user preference extraction part starts with the item embedding layer which embeds the id features of items into low-dimensional dense vectors. For the target item, the embedding is denoted as $\mathbf{e}_l \in \mathbb{R}^d$. For the historical items of user i (*i.e.,* $\mathcal{V}(i)$), corresponding item embeddings are gathered to form the set of user preference embeddings $\mathrm{E}_i = \{\mathbf{e}_j, j \in \mathcal{V}(i)\}$.

2.2.2 Attentive Capsule Layer for Multi-preference Extraction. An important task of our JTCN is to learn a network for mimicking high-level preference representations for cold start users from their side information. Therefore, it is crucial to construct a representative user preference embedding during the training process. Representing user preference by a simple combination (*e.g.,* averaging, concatenation) of vectors $\mathbf{e}_j \in \mathrm{E}_i$ is not conducive to extracting diverse interests of users. Inspired by [5], we propose to apply the recently proposed dynamic routing in capsule network [10] to capture multiple preferences for users. Considering that not all the preference capsules contribute equally to aggregate the high-level user preference representation. We further propose to adopt the attention mechanism to discriminate the informative capsules.

We consider two layers of capsules, which we name as behavior capsules and preference capsules, to represent the user behavior (historical items) and multiple user preferences respectively. Dynamic routing is adopted to compute the vectors of preference capsules based on the vectors of behavior capsules in an iterative way. In each step, given the embedding $\mathbf{e}_j \in \mathbb{R}^d$ of behavior capsule j and vector $\mathbf{p}_k \in \mathbb{R}^d$ of preference capsule k, the routing logit is calculated by

$$b_{jk} = \mathbf{p}_k^T \mathbf{S} \mathbf{e}_j, \tag{1}$$

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

The coupling coefficients between behavior capsule j and all the preference capsules sum to 1 and are determined by performing the "routing softmax" on logits as:

$$c_{jk} = \frac{\exp(b_{jk})}{\sum_t \exp(b_{jt})}. \tag{2}$$

With the coupling coefficients calculated, the candidate vector for preference capsule k is computed by the weighted sum of all behavior capsules:

$$\mathbf{z}_k = \sum_j c_{jk} \mathbf{S} \mathbf{e}_j. \tag{3}$$

The embedding of preference capsule k is obtained by a non-linear "squash" function as:

$$\mathbf{p}_k = squash(\mathbf{z}_k) = \frac{||\mathbf{z}_k||^2}{1 + ||\mathbf{z}_k||^2} \frac{\mathbf{z}_k}{||\mathbf{z}_k||}. \tag{4}$$

Suppose we have K preference capsules, which means there are K distinct preferences of users extracted from the historical items on which the users gave positive feedback. We apply an attention layer to emphasize the informative capsules. There exist several effective ways to calculate the attention score and this paper adopts the multi-layer perceptron (MLP) as

$$a_k = \mathbf{h}^T \text{ReLU}(\mathbf{W}_a \mathbf{p}_k + \mathbf{b}_a), \quad \alpha_k = \frac{\exp(a_k)}{\sum_{k=1}^{K} \exp(a_k)}, \tag{5}$$

where $\mathbf{W}_a \in \mathbb{R}^{d \times d_a}$, $\mathbf{b}_a \in \mathbb{R}^{d_a}$, and $\mathbf{h} \in \mathbb{R}^{d_a}$ are the attention layer parameters. The final attentive weight is normalized by the softmax function.

With the attentive weights assigned to the preference capsules, the high-level user preference can be formed as the weighted sum:

$$\mathbf{u}^p = \sum_{k=1}^{K} \alpha_k \mathbf{p}_k. \tag{6}$$

2.3 Joint Training

A joint training framework is proposed here by optimizing two losses together: a softmax loss for recommendation and a mimic loss for generating user preferences for cold start users without interaction history. The user representation in JTCN is represented in two folds, the user preferences and the content. Two MLP networks, namely f_c and f_p, are used to map the user document into one content space and one preference space for the cold start users. Those two representations are fused together for the the final prediction.

2.3.1 Softmax Loss. The output of f_c denoted by $\mathbf{u}^c$ is fused together with the high-level user preference embedding defined by (6) to form the user embedding. The representation of user i can be generated by

$$\mathbf{u}_i = \text{ReLU}(\mathbf{W}_u[f_c(\text{X}_i), \mathbf{u}_i^p] + \mathbf{b}_u), \tag{7}$$

where X_i denotes the input user document, $\mathbf{W}_u \in \mathbb{R}^{d \times 2d}$ and $\mathbf{b}_u \in \mathbb{R}^d$ are the parameters. $[\cdot, \cdot]$ denotes the concatenation operation and $\text{ReLU}(\cdot)$ is the Rectified Linear Unit. With the user vector $\mathbf{u}_i$ and the target item embedding $\mathbf{e}_l$, the probability of the user interacting with the target item can be predicted by

$$\Pr(\mathbf{e}_l|\mathbf{u}_i) = \frac{\exp(\mathbf{u}_i^{\text{T}} \mathbf{e}_l)}{\sum_{j \in \mathcal{V}(i)} \exp(\mathbf{u}_i^{\text{T}} \mathbf{e}_j)}. \tag{8}$$

We use the *softmax loss* as the objective function to minimize for the recommendation training:

$$\mathcal{L}_{softmax} = -\sum_{(i,l) \in \mathcal{D}} \log \Pr(\mathbf{e}_l|\mathbf{u}_i), \tag{9}$$

where $\mathcal{D}$ is the collection of training data containing user-item interactions.

2.3.2 Mimic Loss. In order to learn preference information from the document of a new user, we propose to use the output of f_p to approximate the high-level user preference representation defined by (6). We define the *mimic loss* as the mean square difference as:

$$\mathcal{L}_{mimic} = \frac{1}{|\mathcal{D}|} \sum_{(i,l) \in \mathcal{D}} \sum_{d} (\mathbf{u}_i^p - f_p(\text{X}_i))^2. \tag{10}$$

Jointly training the following combination of softmax loss and mimic loss enables the network to better imitate the high-level preference and capture content from the side information of new users, which greatly improve the cold start recommendation performance:

$$\mathcal{L}_{joint} = \mathcal{L}_{softmax} + \mathcal{L}_{mimic}. \tag{11}$$

2.4 Cold Start Prediction

Once training is completed, as shown in the right side of Figure 1, we fix the model and make a forward pass through f_c and f_p to get the representation for a new user based on its side information as:

$$\mathbf{u}_{new} = \text{ReLU}(\mathbf{W}_u[\hat{\mathbf{u}}^c, \hat{\mathbf{u}}^p] + \mathbf{b}_u), \tag{12}$$

where $\hat{\mathbf{u}}^c = f_c(\text{X}_{new})$ and $\hat{\mathbf{u}}^p = f_p(\text{X}_{new})$. At last, the preference score of the new user on item j is decided by the inner product of the corresponding embeddings:

$$\hat{r}_{new,j} = \mathbf{u}_{new}^{\text{T}} \mathbf{e}_j. \tag{13}$$

3 EXPERIMENTS

3.1 Datasets

We choose two public datasets for evaluating cold start recommendation performance. 1) CiteULike[1] with 5,551 users, 16,980 articles, and 204,986 implicit user-article feedbacks. CiteULike contains article content formation in the form of title and abstract. We use a vocabulary of the top 8,000 words selected by tf-idf [14]. 2) Amazon Movies and TV[2] [8]. We convert the explicit feedbacks with rating 5 to implicit feedbacks. We filter the user and items with interactions less than 10 and finally get 14,850 users, 23,232 items, and 548,296 interactions. The vocabulary size of words selected by tf-idf from item titles and descriptions is 10,000.

Since only item side information is available, we recommend users for the cold start items. For both datasets, we randomly select 20% of items as the cold start items which will be recommended users at test time. We use Recall and NDCG as evaluation metrics.

3.2 Baselines

We compare the proposed JTCN with several representative recommendation models including three content-based methods **KNN** [11], **FM** [9], and **VBPR** [3], two deep learning methods **LLAE** [6] and **DropoutNet** [13]. KNN uses content information to compute the cosine similarity between items. DropoutNet uses WMF [4] as the pre-trained model for input preference. Except for KNN and LLAE which do not have the parameter of latent factor, we set the number of latent factors $d = 256$ for all methods. The other hyperparameters of all the compared methods are tuned to find an optimal result. For JTCN, the number of preference capsules is set $K = 5$, the dimension of attention layer is set $d_a = 128$, and the *Adam* optimizer with the learning rate of 0.0005 is adopted. For all the methods except KNN, we use the early stopping strategy with a patience of 10.

3.3 Results

3.3.1 Model Comparison. The experimental results of our JTCN as well as baselines on two datasets are reported in Table 1 in terms of Recall@100 and NDCG@100 (with $d = 256$). The best results are listed in bold, and the second best results are marked with star (*). Clearly, JTCN remarkably outperforms baseline models on both datasets. KNN shows poor performance in the cold start scenario, which led by the rough estimation of content-based similarity without any historical interaction. The improvement obtained by FM compared with KNN indicates the advantage of feature interaction. VBPR performs slightly better as it is proposed to alleviate the cold

[1] http://www.citeulike.org.
[2] http://jmcauley.ucsd.edu/data/amazon/

Table 1: Performance Comparison on two datasets in terms of Recall@100 and NDCG@100.

Methods	CiteULike		Amazon	
	Recall	NDCG	Recall	NDCG
KNN	0.2981	0.3453	0.0564	0.2358
FM	0.5100	0.4583	0.0924	0.2260
VBPR	0.5426	0.4825	0.0891	0.2215
LLAE	0.5816	0.5286*	0.1264*	0.2439
DropoutNet	0.6011*	0.5226	0.1013	0.2815*
JTCN	**0.6436**	**0.5432**	**0.1477**	**0.3364**
Improve	7.07%	2.76%	16.85%	19.50%

start problem by using both latent factors and content factors that are extracted from auxiliary information [3]. It can be easily observed that deep learning based methods, LLAE and DropoutNet, which are dedicated to cold start problem, perform better than the traditional content-based baselines. However, all the baselines only reconstruct or extract content factors from the input side information in the test stage. JTCN outperforms all baselines improving Recall@100 by 7.07% and 16.85% on two datasets over the best baseline. This indicates that combining the content information with preference information generated based on the raw input of new users or items can effectively improve the performance of cold start recommendation.

In addition, DropoutNet has a need for the pre-trained model to generate preference input for the main DNN, which may limit its generalization on different datasets. In contrast, the proposed JTCN doesn't need such a pre-trained model to handle the input by learning directly from the input raw features.

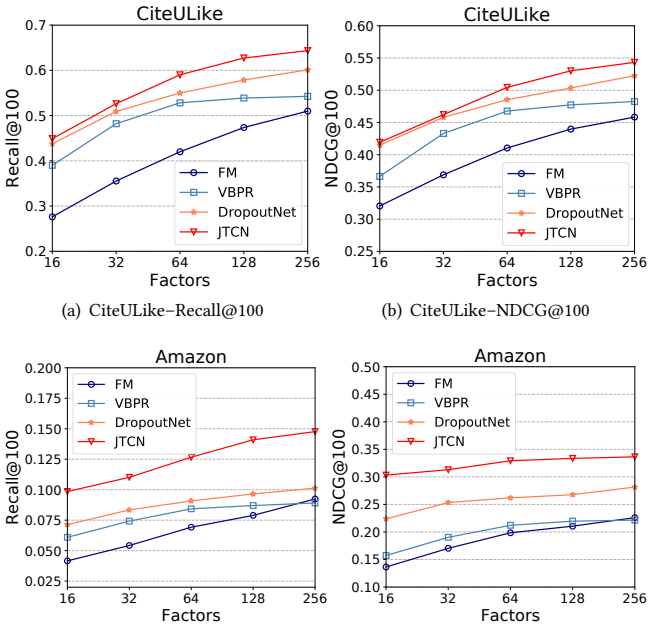
3.3.2 Impact of Latent Factors. To analyze the importance of latent factors, we compare the performance of FM, VBPR, and DropoutNet with the proposed JTCN with respect to the number of latent factors. As Figure 2 shows, JTCN consistently outperforms the baselines. With the increase of the number of latent factors, the performance improvement compared with the best baseline method generally increases. It may be because the combination of content and preference representations is more informative, which requires a relatively larger hidden dimension to incorporate.

4 CONCLUSION

In this paper, a novel neural network model, namely joint training capsule network (JTCN) is first introduced to harness the advantages of capsule model for extracting high-level user preference in the cold start recommendation task. JTCN optimizes the mimic loss and softmax loss together in an end-to-end manner: the mimic loss is used to mimic the preference for cold start users or items; the softmax loss is trained for recommendation. An attentive capsule layer is proposed to aggregate high-level preference from the low-level interaction history via a dynamic routing-by-agreement mechanism. Experiments on two real-world datasets show that our JTCN consistently outperforms baselines.

ACKNOWLEDGMENTS

This work is supported in part by NSFC under grant 61872119, NSF under grants III-1526499, III-1763325, III-1909323, CNS-1930941, and Key Research & Development Plan of Zhejiang Province under grant 2019C03134.



(a) CiteULike–Recall@100 (b) CiteULike–NDCG@100

(c) Amazon–Recall@100 (d) Amazon–NDCG@100

Figure 2: Performance of Recall@100 and NDCG@100 *w.r.t* the number of predictive factors on the two datasets.

REFERENCES

[1] Ignacio Fernández-Tobías, Matthias Braunhofer, Mehdi Elahi, Francesco Ricci, and Iván Cantador. 2016. Alleviating the new user problem in collaborative filtering by exploiting personality information. *User Modeling and User-Adapted Interaction* 26, 2-3 (2016), 221–255.

[2] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. 2010. Learning attribute-to-feature mappings for cold-start recommendations. In *ICDM*. 176–185.

[3] Ruining He and Julian McAuley. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *AAAI*. 144–150.

[4] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM*. 263–272.

[5] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *CIKM*. 2615–2623.

[6] Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, Yang Yang, and Zi Huang. 2019. From zero-shot learning to cold-start recommendation. In *AAAI*, Vol. 33. 4189–4196.

[7] Jovian Lin, Kazunari Sugiyama, Min-Yen Kan, and Tat-Seng Chua. 2013. Addressing cold-start in app recommendation: latent user models constructed from twitter followers. In *SIGIR*. 283–292.

[8] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *EMNLP*. 188–197.

[9] Steffen Rendle. 2010. Factorization machines. In *ICDM*. 995–1000.

[10] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *NIPS 2017*. 3856–3866.

[11] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*. 285–295.

[12] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, Lexing Xie, and Darius Braziunas. 2017. Low-rank linear cold-start recommendation from social data. In *AAAI*. 1502–1508.

[13] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. Dropoutnet: Addressing cold start in recommender systems. In *NIPS*. 4957–4966.

[14] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *KDD*. 448–456.

[15] Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and S Yu Philip. 2018. Zero-shot User Intent Detection via Capsule Neural Networks. In *EMNLP 2018*. 3090–3099.