Improving Cold-start Item Advertisement For Small Businesses

Yang Shi
Rakuten, Inc.
San Mateo, USA
yang.shi@rakuten.com

Young-joo Chung
Rakuten, Inc.
San Mateo, USA
youngjoo.chung@rakuten.com

ABSTRACT

In this paper, we study cold-start item advertisements for small businesses on a real-world E-commerce website. From analysis, we found that the existing cold-start Recommender Systems (RSs) are not helpful for small businesses with a few sales history. Training samples in RS models can be extremely biased towards popular items or shops with sufficient sales history, and can decrease advertising performance for small shops with few or zero sales history. We propose two solutions to improve advertising performance for small shops: *negative sampling* and *Meta-shop*. Negative sampling focuses on changing the data distribution and Meta-shop focuses on building novel meta-learning models. By including sales information in the training of both methods, we are able to learn better cold-start item representations from small shops while keeping the same or better overall recommendation performance. We conducted experiments on a real-world E-commerce dataset and demonstrated that the proposed methods outperformed a production baseline. Specifically, we achieved up to 19.6% relative improvement of Recall@10k using Meta-shop compared to the traditional cold-start RS model.

CCS CONCEPTS

• **Information systems → Recommender systems**; *Similarity measures*.

KEYWORDS

cold-start recommendation, recommender system, negative sampling, meta-learning

ACM Reference Format:
Yang Shi and Young-joo Chung. 2021. Improving Cold-start Item Advertisement For Small Businesses. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR eCom'21).* ACM, Virtual Event, Montreal, Canada, 7 pages.

1 INTRODUCTION

Recommender systems are the core component of E-commerce business. E-commerce companies such as Amazon, eBay, and Rakuten not only need to provide personalized recommendations to customers but also need to provide potential customers to merchants for specific items through advertisements. In both scenarios, if we have enough purchase history of the users or items, we can build recommender systems through collaborative filtering (CF) [15, 21]. However, in the *cold-start* setting where users or items do not have sufficient past transactions, this solution is not applicable. Content-based models have been proposed to solve this problem [14, 24]. Instead of using purchase history, these models used content features (i.e. side information) such as user profile and item information. Hybrid models that combine content-based models and collaborative filtering have been popular as well. The main idea is to map the side information and the feedback to separate low-dimensional representations, combine the representations, and use them for predicting the final interaction. In this paper, we focus on improving a hybrid recommender system for cold-start item advertisements.

CF-based and hybrid recommender systems for cold-start items suffer from the long-tail issue. To build recommender systems, we need to sample training data from purchase history. If we randomly sample training data, the model will be performant only for popular items because its sales are dominant in the purchase history [17, 22]. As a result, the item advertising performance for large or popular businesses with many and/or popular items would be satisfactory, the performance for small businesses with less-selling items or new businesses without previous sales history would be unsatisfactory.

To improve the advertising performance of cold-start items for all shops, we propose two solutions: *negative sampling* and *Meta-shop*. We tested various negative sampling methods that change the data distribution by increasing the occurrence of the items from small shops in training samples. Meta-shop adopts an optimization method from meta-learning literature and builds per-shop recommendations. Regarding recommendation for each shop as one task, Meta-shop learns local (shop-specific) parameters which will be aggregated to update global parameters. As a result, the Meta-shop model can quickly adjust its parameters to unseen items or shops. To the best of our knowledge, this is the first work that creates cold-start item recommender systems that consider shop sales and build per-shop recommendations. Our contributions are as follows:

- We study a novel problem in a real-world E-commerce site, namely improving advertising performance for shops with few or zero sales history.
- We propose two solutions for the aforementioned problem: Negative sampling and Meta-shop. Meta-shop is a novel meta-learning-based recommender that can quickly adapt itself to unseen items or shops.
- Experimental result on a real-world E-commerce dataset demonstrated that both methods outperformed an existing production baseline. Particularly, we achieved up to 19.6% relative improvement of Recall@10k using Meta-shop compared to the traditional cold-start RS model.

SIGIR eCom'21, July 15, 2021, Virtual Event, Montreal, Canada

2 RELATED WORK

2.1 Hybrid recommender systems for cold-start problems

Various hybrid methods have been proposed to tackle the cold start problem. Collaborative topic regression (CTR) [33] and its variants [13, 34] have been one of popular model structures. Based on the availability of feedback information, CTR interpolates between content-based representations generated from side information by Latent Dirichlet Allocation (LDA) [4] and feedback-based ones generated from interaction matrices by weighted matrix factorization (WMF).

Recent hybrid methods utilize neural networks to learn representations from side information. To solve the cold-start problem in the music recommendation system, DeepMusic [28] created multiple embeddings based on contents (e.g. artist biographies and audio spectrogram) using neural networks like convolutional neural networks. Several researchers [6, 9, 19, 20, 38] utilized various autoencoders to learn representation of side information. For example, Dong *et al.* [9] combined additional stacked denoising autoencoder [31] and matrix factorization to integrate side information.

Attention mechanisms were also introduced to improve the recommendation performance. Attentional collaborate&content models (ACCM) [27] used attention mechanisms to adjust the importance of source information. For cold start items whose feedback information is missing, ACCM pays more attention to the item's side information to make predictions.

Some works focused on integrating side information and the preference representations generated by feedback information. Bianchi *et al.* utilize feedbacks from multi-brand retailers and align item embeddings across shops using translation models. Dropoutnet [32] takes preference representation and side information as inputs for items. For a cold start item, the values of preference representations are set to be zero because past interaction is missing; CB2CF [3] learns a mapping function from side information to preference representation. Both Dropoutnet and CB2CF need to first learn preference representations by applying a matrix factorization algorithm to the interaction matrix.

2.2 Recommender systems for long-tail/sample-bias problems

Researchers proposed to use auxiliary information and domain adaptation to improve non-popular item feature learning [7, 12, 36]. However, the information may not be available. Our proposed methods do not require additional information, so it can be easily applied to existing datasets.

More recently, meta-learning is thriving as a new method to train a model that can quickly adapt to new tasks with only a few or zero training samples [11, 30]. It has been applied to different areas such as image classification [1, 5] and language models [2, 35]. Manasi *et al.* and Lee *et al.* [18, 29] applied meta-learning to recommendation and treated each user's recommendation as separated tasks. Each task predicts whether a user likes an item or not. Scenario-specific sequential meta learner [10] learned user's behaviors from different scenarios such as "what to take when traveling" and "how to dress up yourself on a party". Here, tasks are based on scenarios, When a new scenario comes, the model can quickly adjust and recommend accurately. Pan *et al.* [25] proposed a two-step model to improve click-through rate (CTR) predictions for cold-start advertisements (ads) by first train a traditional classification model using warm-start ads and then add a meta-learning module to fine-tune cold-start item feature embeddings. Besides using meta-learning to learn user and item features, several works also focused on using meta-learning to optimize model structures. MetaSelector [23] combined different recommender systems with meta-learning trainable weights. Sequential meta-Learning method [39] transferred the old model's parameters to the new model when new data comes, without retraining the model again using both old data and new data. Our proposed Meta-shop does limit its focus on user or item-level feature learning. By grouping users and items per shop and training in a meta-learning fashion, we learn better user and item representations which can help cold-start item recommendations, especially for small shops.

3 LIMITATIONS OF EXISTING COLD-START RECOMMENDER SYSTEMS

In this section, we describe our existing cold-start item recommender system and demonstrate the limitations of this solution in terms of sample selection bias and performance differences in shops with different amounts of sales.

The baseline model [26] is shown in Figure 1. We designed this model based on the general hybrid recommender systems like Neural Collaborative filtering (NCF) [16]. Experimental results demonstrated that this model can solve the cold-start problem efficiently. Item inputs are concatenated learnable one-hot embeddings from side information such as price and title. User inputs are summations of all item representations of which they purchased in the training period. The score is computed using Euclidean distance and loss is computed using contrastive loss [8] using positive and negative samples. Positive samples consist of users who purchased a target item and negative samples consist of users who didn't purchase the item. We use positive/negative users instead of items to learn effective representations for item recommendation and to distinguish users for a target item.

We also adopt a rule-based model called LV2: For each query item, we return the most frequent buyers who purchased items from the same level 2 genre as the query item. Note that genre (category) taxonomy has five levels from broad genres (Lv1, e.g. shoes) to specific genres (Lv5, e.g. running shoes).

We trained the baseline model using a total ~30M purchase history with ~3M users and ~1.3M items from Ichiba[1] purchase data in genre A. We evaluated the Recall@1M per test *shop*. We evaluated per shop recall instead of per item, since we want to satisfy every shop with our model. The distribution of the number of shops with different Recall@1M is shown in Figure 2. The proportion of shops with recall 0.8+ is 20.7% using LV2 method and 62.5% using the baseline model. Clearly the baseline model is better than the LV2 model. *However, we noticed a huge difference in the number of shops with recall below and above 0.8 in the result of the baseline model.* We further analyzed the input features for shops with recall 0.8+ and 0.8-, and found that *the number of sales per shop is uneven. Specifically,*

[1] https://www.rakuten.co.jp/

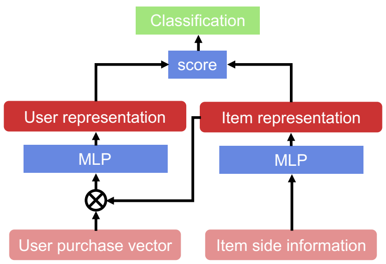

Figure 1: Baseline model for cold-start item recommendation. ⊗ operation is a summation of all item representations of which the user purchased.

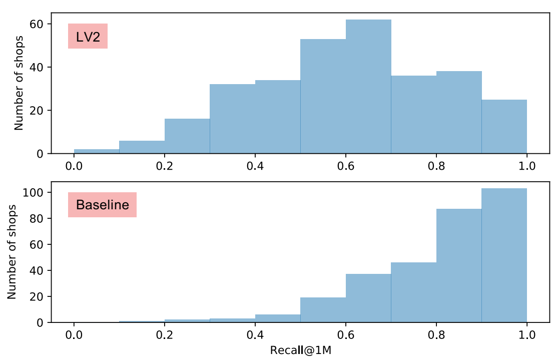

Figure 2: Distribution of test recalls per shop.

bad-performed shops have fewer sales. For example, 59.9% shops with recall 0.8+ have more than 10k sales in training data, while more than 52% with recall 0.8- shops have less than 5k sales. The details are shown in Figure 3. We also computed the number of sales over different proportions of shops in Figure 4. 5% of shops accounted for over 80% of training samples. From this analysis, we concluded that **the existing cold-start recommendation models may not be helpful for small shops/businesses which have fewer sales history because of Sample Selection Bias (SSB)[37].**

4 PROPOSED METHODS

We provide two solutions to solve the SSB problem for small businesses: change the data distribution and change the model. In Section 4.1, we will discuss changing the data distribution using negative sampling (NS). In Section 4.2, we will focus on Meta-shop, a new model based on meta-learning.

4.1 Negative sampling

We change the data distribution by increasing the occurrence of the items from small shops in training samples. As described in Section 3, the baseline model learns representation using positive and negative samples. During sampling, *we select negative users from all users who purchased from small shops.* In this way, more item representations from small shops will be learned during the

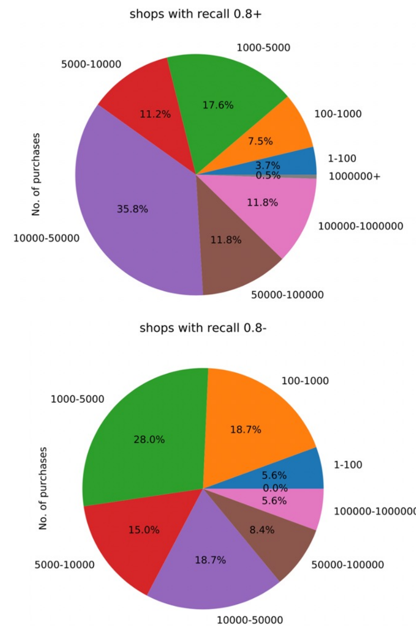

Figure 3: Distribution of the number of sales per test shop in training data. Upper: test shops with Recall@1M 0.8+, Lower: test shops with Recall@1M 0.8-. Overall, lower section shops have fewer sales.

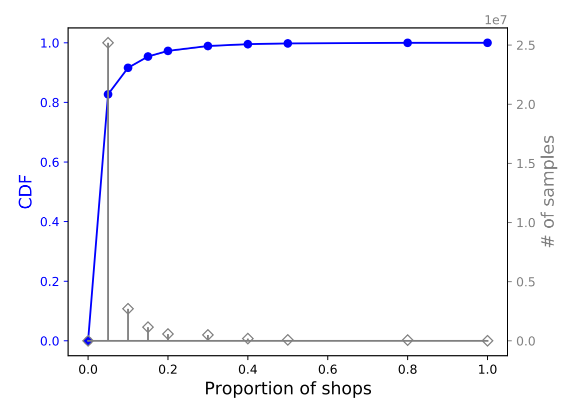

Figure 4: Number of training samples over proportion of shops from Genre A.

training. The final loss is:

$$\mathcal{L} = \sum_{(i,u^+)} \text{loss}(dist(i,u^+)) + \sum_{(i,u^-)} \text{loss}(dist(i,u^-)) \quad (1)$$

where (i,u^+) are positive samples (u^+ purchased i) and (i,u^-) are negative samples (u^- did not purchase i). We will compare the performance changes by the new NS in Section 5.4. In the original NS, the negative users come from all users who purchased items from a different level 3 genre.

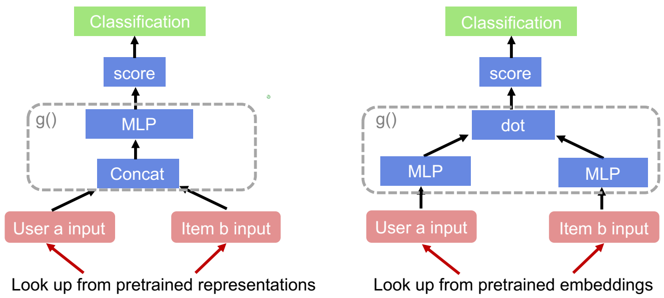

Figure 5: Meta-shop models: Left is M1 which utilizes concatenation of user and item inputs. Right is M2 which have separated sub-models for users and items.

4.2 Meta-shop

In this section, we propose a novel meta-learning framework to solve cold-start item recommendations for small businesses: Meta-shop. In the Meta-shop model, we define per shop recommendations as separated tasks. If there are total S shops, we will have S tasks.

4.2.1 Model. Model inputs are user feature f_u and item feature f_i. They can be trainable one-hot embeddings or pre-trained representations. For simplicity, we use the pre-trained representations from the baseline model. The model parameters are $g()$. We define loss as $\mathcal{L}(f_u, f_i; g()) = loss(y, \hat{y})$, where y is the purchase label (0 for not purchase, 1 for purchase) and $\hat{y} = g(f_u, f_i)$ is the predicted label. We design two different $g()$ as shown in Figure 5. In **M1**, user and item inputs are concatenated before feeding into a multi-layer perceptron (MLP) network. In **M2**, user and item inputs are separately fed into two different MLPs, and at the last layer, the two hidden features are combined by a dot product.

4.2.2 Optimization. We adopt the MAML algorithm [11] and show optimization steps in Algorithm 1. For each shop, we randomly select fixed number of samples as a support data set and use the rest of samples as a query data set. We evaluate query performance based on a support data set. The main idea is to recursively combine knowledge from all tasks to guide each task and then summarize all tasks to learn a generalized model that can be quickly updated to learn any tasks.

Algorithm 1: Meta-shop Training

```
Result: g()
initialization of g();
while not converge do
    sample batch of shops P;
    for shop p in P do
        set g^p() = g();
        local update: g^p ← g^p − α∇_{g^p} L(support^p; g^p()) ;
    end
    global update g(): g ← g − β Σ_{p∈P} ∇_g L(query^p; g^p());
end
```

5 EXPERIMENTS

In this section, we detail experiments conducted to demonstrate effectiveness of our proposed methods. We verify its effectiveness with following questions: (Q1) how good is the performance for all cold-start item recommendations? (Q2) is there any change in the per-shop recall distribution compared to the baseline model (Figure 2)? (Q3) how much improvement was gained for small businesses' cold item advertising? The first question considers item-level performance, the second and third consider shop-level performance.

5.1 Dataset

We extracted sales history from Rakuten Ichiba and pre-processed the data. We used the sales data from January 2020 to September 2020 for training, and used October 2020 as test data. For NS experiments, we kept users with at least 4 purchases for training. We kept cold-start test items with at least 6 purchases in test set. For Meta-shop experiment, we kept shops with at least 13 purchases in both training and test time, as the model training requires support and query sets for each shop. In each shop, 10 purchases were randomly selected to be the support set, and the rest went into the query set. Also, we classified shops into cold shops and warm shops. Cold shops are shops never shown in training period, while warm shops exist in the training. Cold-start test items can come from both cold and warm shops. Note that cold-start items did not appeared in training period. But because during the test time, we also performed a random support set update, each cold item may not be completely cold. The statistics of dataset is listed in Table 1.

Table 1: Dataset statistics

NS	User	Item	Cold shop	Warm shop	Purchases
Train	2,944,124	1,290,889	N/A	N/A	36,740,388
Test	2,944,124	913	N/A	N/A	26,655
Meta-shop	**User**	**Item**	**Cold shop**	**Warm shop**	**Purchases**
All	9,435,537	1,488,582			42,893,287
Train	9,095,846	1,409,483	0	7,035	40,118,920
Test	9,387,803	894	9	354	23,325

5.2 Evaluation metrics

We computed Recall for all items and all shops. Assume M is the number of items, N is the number of shops, r_i is the i-th item recall value, shop j has M_j items and rs_j is the j-th shop recall value: $R_{item} = (\sum_{i \in \{N\}} r_i)/M$, $R_{shop} = (\sum_j rs_j)/N$, where $rs_j = (\sum_{i \in \{M_j\}} r_i)/M_j$.

5.3 Negative sampling

We define small/large shops based number of sales N (i.e. if shop A has less than N sales during training, A is a small shop). We set N to be 46, which is the median of the sales for all shops in training. For each positive item-user purchase pair, we draw one item-user negative pair. The ratio of negative samples versus positive samples is 1:1. We tested three negative sampling methods:

- N0: The original NS; the negative user comes from all users who purchased items from a different level 3 genre.

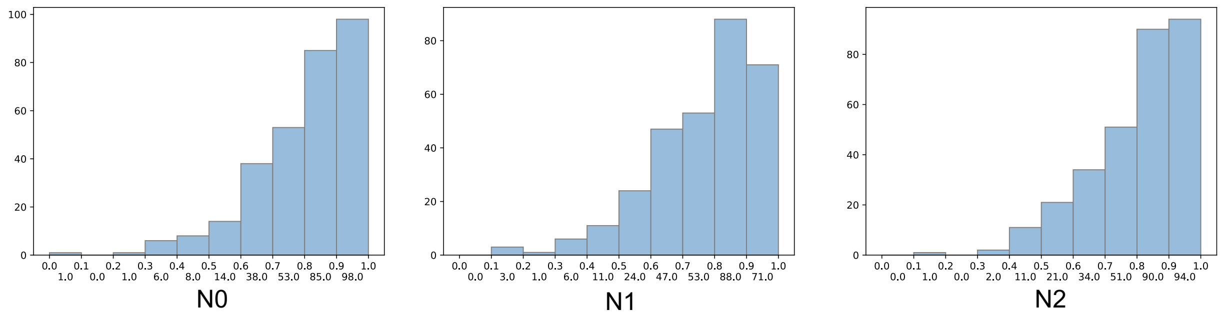

Figure 6: Distribution of test shops with different Recall@1M. x-axis is the recall value, y-axis is the number of shops, bottom integers are counts of each bar.

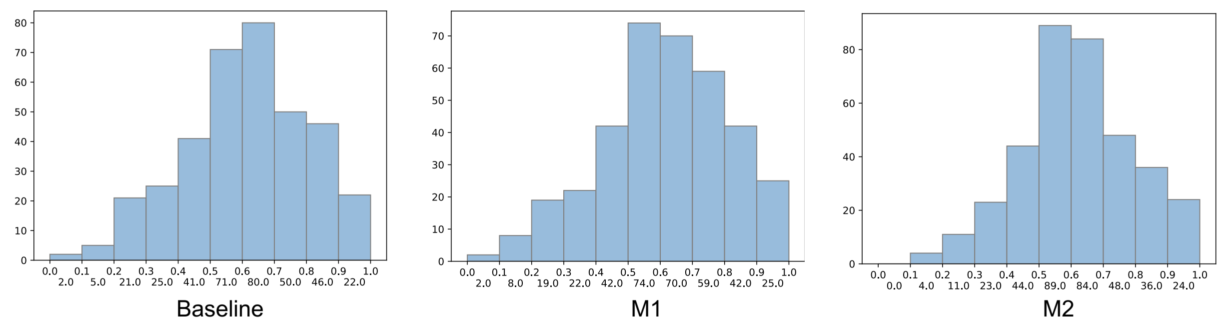

Figure 7: Distribution of test shops with different Recall@1M.

- N1: The negative user is selected either from all users who purchased from small shops or users who purchased from different level 3 genres with 0.5 probability each.
- N2: For items in large shops, we use N0, for items in small shops, we use N1.

Q1: In Table 2, we show item-level average recall using different negative sampling methods. N0 and N2 have the highest Recall@1Million. **Q2:** At shop-level, N2 keeps an average recall for all test shops similar to the recall of N0, while achieving a much smaller variance. In Figure 6, we show the recall distribution of test shops. We can see that the slop gets smaller from N0 to N1, which indicates the recall distributes more evenly among shops. Also, we observe that N1 has the largest portion of shops (7.9%) which have a mean recall between 0.5 and 0.6, compared to N0 (4.6%) and N2 (6.9%). However, N1 also has the least portion of shops with a mean recall above 0.9 in Figure 6. **Q3:** To check the performance for small businesses, we compute statistics of shops with improved and decreased performances in Table 4. When comparing N1 to N0, shops with improved performances have a small average number of sales, and shops with decreased performances are relatively large shops with a larger number of sales. This implies that N1 improve recommendation accuracy for small shops but will hurt the performance for large shops. On the other hand, N2 is a combination of N0 and N1, so it maintains a good performance balance between small shops and large shops.

In summary, we conclude that N2 is the best choice to reduce sample selection bias for small shops while maintaining overall performance.

Table 2: Recall@1M using different negative sampling

NS	Item-level	Shop-level	
	Mean	Mean	Variance
N0	**0.838**	**0.807**	0.166
N1	0.807	0.769	0.174
N2	0.835	0.804	**0.159**

Table 3: Average number of sales per test shop with better/worse Recall@1M compared to N0

NS	Better	Worse
N1	47	105
N2	109	100

Table 4: Median number of sales per test warm-shops with better/worse Recall@1M compared to baseline model

NS	Better	Worse
M1	13,351	15,326
M2	11,023	22,631

Table 5: Recall for all cold items from cold shops and warm shops

Cold Shop			
Model	@100,000	@500,000	@1,000,000
LV2	0.074	0.223	0.331
Baseline	0.125	0.329	0.521
M1	**0.150**	**0.361**	0.523
M2	0.108	0.326	**0.530**
Warm Shop			
LV2	0.206	0.380	0.482
Baseline	0.288	0.497	0.654
M1	**0.295**	**0.509**	**0.659**
M2	0.212	0.4687	0.6444
Overall			
LV2	0.203	0.376	0.478
Baseline	0.283	0.492	0.650
M1	**0.291**	**0.505**	**0.655**
M2	0.209	0.465	0.641

5.4 Meta-shop

As we discussed in Section 4.2, we have two models: M1 and M2. We compare them with the baseline model from Section 3. We used mean square error as the loss function for all meta-shop training. We performed hyper-parameter tuning and set $\alpha = 5e^{-6}$, $\beta = 5e^{-5}$.

Q1: We summarize recall of all cold-start items in Table 6. Overall, M1 performs the best. It has a 2.5% relative improvement for Recall@50k compared to the baseline. **Q3:** Moreover, there are huge performance improvements of cold-start items from cold shops when using Meta-shop method. The relative improvements are 19.6%, 9.8%, 1.8% for Recall@10k, 50k, 1M.

Q2: We plot Recall@1M distribution of test shops in Figure 7 and list the mean and variance of Recall@1M per shop in Table 6. Both M1 and M2 hold larger portion of shops with Recall@1M higher than 0.5 and have higher mean values compared to the baseline. M2 is the best model in terms of the variance in recall of all test shops.

In summary, both Meta-shop methods outperform the baseline in terms of item-level and shop-level recall. Between M1 and M2, M1 is better for improving item-level recall, while M2 is better for shop-level recall. However, we have to mention that M1 cannot store user/item features separately and have to be fed with them for computing the score. It is less convenient for inference compared to M2, which can pre-store all embeddings of users and items and compute the score using the dot product. We see similar concerns in previous work [10].

Table 6: Left: portion of shops with Recall@1M above certain value. Right: Recall@1M mean and variance for all test shops

Model	0.5+	0.6+	0.7+	0.8+	Mean	Variance
LV2	0.363	0.218	0.140	0.066	0.429	0.222
Baseline	0.741	0.545	0.325	0.187	0.602	0.197
M1	0.744	0.540	0.347	0.185	0.609	0.202
M2	0.774	0.529	0.298	0.165	**0.615**	**0.177**

6 CONCLUSION AND FUTURE WORK

In this paper, we proposed two solutions to improve advertising performance for cold-start items from small shops that have insufficient sales history. Negative sampling changes data distribution by increasing the occurrence of the items from small shops in training samples. This helped the model learning better representation of items from small shops. Meta-shop utilized a meta-learning strategy to learn parameters that can be adapted quickly to unseen items or shops. The experimental results showed that both negative sampling and Meta-shop were effective to increase recommendation performance of small shops. Both methods increased the percentage of shops of Recall@1M over 0.5 without sacrificing overall recall. Particularly, Meta-shop improved relative Recall@10K by 19.6%.

For future work, we are planning to introduce the shop's sales history as a guide of Meta-shop. With this, the model can decide which information it should pay more attention to: item side information or sales information. We are also planning to apply our methods to different datasets to verify whether it can be generalized to other E-commerce datasets.

REFERENCES

[1] Antreas Antoniou, Harrison Edwards, and Amos Storkey. 2019. How to train your MAML. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HJGven05Y7

[2] Trapit Bansal, Rishikesh Jha, and Andrew McCallum. 2020. Learning to few-shot learn across diverse natural language classification tasks. Proceedings of the 28th International Conference on Computational Linguistics, 5108–5123.

[3] Oren Barkan, Noam Koenigstein, Eylon Yogev, and Ori Katz. 2019. CB2CF: a neural multiview content-to-collaborative filtering model for completely cold item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys)*. https://doi.org/10.1145/3298689.3347038

[4] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research (JMLR)* (2003).

[5] Qi Cai, Yingwei Pan, Ting Yao, Chenggang Yan, and Tao Mei. 2018. Memory Matching Networks for One-Shot Image Recognition. CVPR, 4080–4088.

[6] Minmin Chen, Zhixiang Xu, Kilian Q. Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*.

[7] Zhihong Chen, Rong Xiao, Chenliang Li, Gangfeng Ye, Haochuan Sun, and Hongbo Deng. 2020. ESAM: Discriminative Domain Adaptation with Non-Displayed Items to Improve Long-Tail Performance. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 579–588. https://doi.org/10.1145/3397271.3401043

[8] S. Chopra, R. Hadsell, and Y. LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. 539–546 vol. 1. https://doi.org/10.1109/CVPR.2005.202

[9] Xin Dong, Lei Yu, Zhonghuo Wu, Yuxia Sun, Lingfeng Yuan, and Fangxi Zhang. 2017. A hybrid collaborative filtering model with deep structure for recommender systems. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

[10] Zhengxiao Du, Xiaowei Wang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Sequential Scenario-Specific Meta Learner for Online Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2895–2904. https://doi.org/10.1145/3292500.3330726

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (Sydney, NSW, Australia) *(ICML'17)*. JMLR.org, 1126–1135.

[12] Chen Gao, Xiangning Chen, Fuli Feng, Kai Zhao, Xiangnan He, Yong Li, and Depeng Jin. 2019. Cross-Domain Recommendation Without Sharing User-Relevant Data. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 491–502. https://doi.org/10.1145/3308558.3313538

[13] Prem Gopalan, Jake M. Hofman, and David M. Blei. 2015. Scalable recommendation with hierarchical Poisson factorization. In *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence (UAI)*.

[14] Casper Hansen, Christian Hansen, Jakob Grue Simonsen, Stephen Alstrup, and Christina Lioma. 2020. Content-Aware Neural Hashing for Cold-Start Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 971–980. https://doi.org/10.1145/3397271.3401060

[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web* (Perth, Australia) *(WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. https://doi.org/10.1145/3038912.3052569

[16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*.

[17] Adit Krishnan, Ashish Sharma, Aravind Sankar, and Hari Sundaram. 2018. An Adversarial Approach to Improve Long-Tail Performance in Neural Collaborative Filtering. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 1491–1494. https://doi.org/10.1145/3269206.3269264

[18] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1073–1082.

[19] Sheng Li, Jaya Kawale, and Yun Fu. 2015. Deep Collaborative Filtering via Marginalized Denoising Auto-encoder. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM)*. https://doi.org/10.1145/2806416.2806527

[20] Tianyu Li, Yukun Ma, Jiu Xu, Bjorn Stenger, Chen Liu, and Yu Hirate. 2018. Deep Heterogeneous Autoencoders for Collaborative Filtering. In *Proceedings of the 18th IEEE International Conference Data Mining (ICDM)*.

[21] G. Linden, B. Smith, and J. York. 2003. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing* 7, 1 (2003), 76–80. https://doi.org/10.1109/MIC.2003.1167344

[22] Yudan Liu, Kaikai Ge, Xu Zhang, and Leyu Lin. 2019. Real-Time Attention Based Look-Alike Model for Recommender System. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2765–2773. https://doi.org/10.1145/3292500.3330707

[23] Mi Luo, Fei Chen, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Jiashi Feng, and Zhenguo Li. 2020. MetaSelector: Meta-Learning for Recommendation with User-Level Adaptive Model Selection. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 2507–2513. https://doi.org/10.1145/3366423.3379999

[24] Raymond J. Mooney and Loriene Roy. 2000. Content-Based Book Recommending Using Learning for Text Categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries* (San Antonio, Texas, USA) *(DL '00)*. Association for Computing Machinery, New York, NY, USA, 195–204. https://doi.org/10.1145/336597.336662

[25] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm Up Cold-Start Advertisements: Improving CTR Predictions via Learning to Learn ID Embeddings *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 695–704. https://doi.org/10.1145/3331184.3331268

[26] Ramin Raziperchikolaei, Guannan Liang, and Young joo Chung. 2021. Shared Neural Item Representations for Completely Cold Start Problem (To be appeared). In *Proceedings of the 15th ACM Conference on Recommender Systems (Recsys)*.

[27] Shaoyun Shi, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Attention-based Adaptive Model to Unify Warm and Cold Starts Recommendation. In *Proceedings of the 27th ACM International on Conference on Information and Knowledge Management (CIKM)*. https://doi.org/10.1145/3269206.3271710

[28] Aäron van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS)*.

[29] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. 2017. A Meta-Learning Perspective on Cold-Start Recommendations for Items. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc.

[30] Ricardo Vilalta and Youssef Drissi. 2002. A Perspective View and Survey of Meta-Learning. Artificial Intelligence Review, 77–95.

[31] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre antoine Manzagol. 2010. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research (JMLR)* (2010).

[32] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. DropoutNet: Addressing Cold Start in Recommender Systems. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*.

[33] Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

[34] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative Deep Learning for Recommender Systems. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. https://doi.org/10.1145/2783258.2783273

[35] Wenpeng Yin. 2020. Meta-learning for Few-shot Natural Language Processing: A Survey. Arxiv.

[36] Bowen Yuan, Jui-Yang Hsia, Meng-Yuan Yang, Hong Zhu, Chih-Yao Chang, Zhenhua Dong, and Chih-Jen Lin. 2019. Improving Ad Click Prediction by Considering Non-Displayed Events. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 329–338. https://doi.org/10.1145/3357384.3358058

[37] Bowen Yuan, Jui-Yang Hsia, Meng-Yuan Yang, Hong Zhu, Chih-Yao Chang, Zhen hua Dong, and Chih-Jen Lin. 2019. Improving Ad Click Prediction by Considering Non-displayed Events. In *CIKM*. 329–338.

[38] Shuai Zhang, Lina Yao, and Xiwei Xu. 2017. AutoSVD++: An Efficient Hybrid Collaborative Filtering Model via Contractive Auto-encoders. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. https://doi.org/10.1145/3077136.3080689

[39] Yang Zhang, Fuli Feng, Chenxu Wang, Xiangnan He, Meng Wang, Yan Li, and Yongdong Zhang. 2020. How to Retrain Recommender System? A Sequential Meta-Learning Method. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1479–1488. https://doi.org/10.1145/3397271.3401167