Hybrid Meta-Learning for Cold-Start Recommendation

Tianze Hu
University of Arizona
USA

ABSTRACT

By absorbing knowledge from past experiences, the objective of few-shot meta-learning is to obtain a prior that can enable fast adaptation to new tasks. The existing model-agnostic meta-learning (MAML) algorithm provides a good initialization via a double-gradient decent for few-shot learning, where the tasks are drawn from a fixed distribution. However, when dealing with various tasks from multiple distributions such as recommendation tasks, meta-learning with a single prior might fail in cold-start recommendation due to its insufficient capability for adaptation. To solve this problem, we consider to learn multiple proper priors to extend meta learning into cold-start recommendation. In this paper, we propose a hybrid meta-learning (HML) method, which is able to modulate its meta-learned prior parameters by task based cluster method, allowing more efficient fast adaptation. In addition, to provide multi-initialization for meta learning, our HML method is learned with the proposed hard task cluster (HTC) scheme. The results demonstrate the effectiveness of our model in modulating the meta-learned prior and outperform state-of-the-art methods.

KEYWORDS

Meta-learning, Recommendation, MAML

ACM Reference Format:
Tianze Hu. 2018. Hybrid Meta-Learning for Cold-Start Recommendation. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

How to build a personalized recommender system with a high performance for each user is an interesting topic [26] in recommender systems. In cold-start situation, we have no information or limited information about new users or items [36], which indicates that a few rows or columns in the rating matrix contain no or very few valid elements. So the latent factor model in this case is likely to be ineffective. Many prior works attempt to address this problem by utilizing the side information [11]. Some other works intend to find the similarity among the items based on features, by which the system can recommend the proper goods to new users [9]. Correspondingly, proper users can be recommended to new items in a similar way [37].

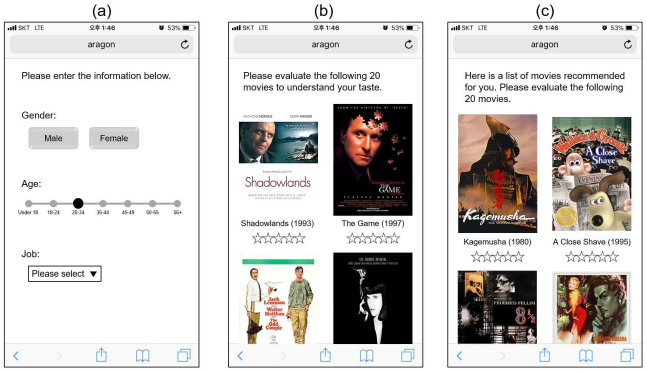

Figure 1: Simplified example of evidence candidates.

Figure 1 shows an simplified example. In the first page (a), systems collect the information under privacy issues. And in second page (b), Netflix initially presents popular movies and television programs to new users, and the user chooses the videos that he/she likes among the candidates. Afterward, the system recommends some programs based on the videos selected by the user, which is shown in third page (c). Recently, deep learning methods are also applied to make recommendations in this scenario for improving performance [8, 20].

Meta-learning algorithm learns to efficiently solve new tasks by extracting prior experiences. Due to its efficiency of few-shot learning, it draws much attention in cold start recommendation. Of particular interest are optimization-based approaches, such as classic model-agnostic meta-learning (MAML) [13]. MAML aims to find a prior for fast adaptation to new tasks after a few gradient updates, which has recently been applied in recommender systems. As the first try, Vartak et al. [40] propose a MAML based meta-learning model for binary-item classification and recommendation, although the application is very limited. Then, Fei et al. [7] design a general federate MAML based framework that is able to cope with various meta-learning algorithms. By applying MAML in estimating the preference of cold-start users, Lee et al. [26] propose the MELU method. Meta-learning [1, 4, 33, 35] have attempted to tackle this problem by seeking an initialization of model parameters that a small number of gradient updates will lead to superior performance on a new task. With the aim of learning faster on a new task, meta-learning makes it possible by leveraging large quantities of past experiences collected across a distrition of tasks to learn a common structure [35]. This kind of fast and flexible learning is challenging, since the agent must integrate its prior experience with a small

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

amount of new information, and meanwhile prevent overfitting to the new data. Furthermore, the form of prior experience and new data will depend on the specific task. Consequently, the mechanism for meta-learning should be general to the task and the form of the corresponding computation for the widest applicability. MAML adopts a naive way to train the model for a sequence of tasks, where in average speaking the initial parameters are largely good for new tasks [13].

All these works assume that tasks follow a fixed distribution and one good initalization is enough for quickly adapting to all new tasks after a few gradient updates. However, Vuorio et al. [41] identify and demonstrate the limitation of having to rely on a single initialization in a family of widely used model-agnostic meta-learners.

While most of the existing model-agnostic meta-learners rely on a single initialization, different tasks sampled from a complex task distributions can require substantially different parameters, making it difficult to find a single initialization that is close to all target parameters [41].

In order to solve this problem, we consider to adaptively learn multiple proper priors by extending the existing MAML. The challenge lies in associating each task with one of the meta-learners requires additional task identity information, which is often not available or could be ambiguous.

To overcome this issue, we aim to develop a task-cluster based meta-learner that is able to effectively cluster tasks and acquire prior parameters. In this paper, we propose a Hybrid-Initialization meta-learning (HML) framework which is able to modulate its meta-learned prior parameters by task based cluster method.

Contribution: The primary contribution of this work is a hybrid meta-learning framework with multiple priors for different categories of tasks, by forming task based cluster that could execute fast adaptation in few-shot learning.

The second main contribution of this paper is proposed a hard task cluster (HTC) scheme which combine an effective meta-training curriculum and cluster method.

Experiment result shows that our hybrid meta-learning algorithm outperforms baselines and demonstrate the effectiveness of HML in wide-range distribution recommendation tasks.

The paper is organized as follows. In Section 2, we discuss related works in curriculum learning and meta-learning. In Section 3, we propose our method. Finally, we conduct experiments on cold-start recommendation in Section 4, and demonstrate the effectiveness of the proposed method.

2 HYBRID META LEARNING FOR COLD-START RECOMMENDATION

2.1 Framework of HML

The core idea of HML framework is to utilize multiple priors for different categories of tasks to overcome the limitation of MAML in complex tasks.

The framework contains two phase, i.e., Meta-Training Phase and Meta-Test Phase.

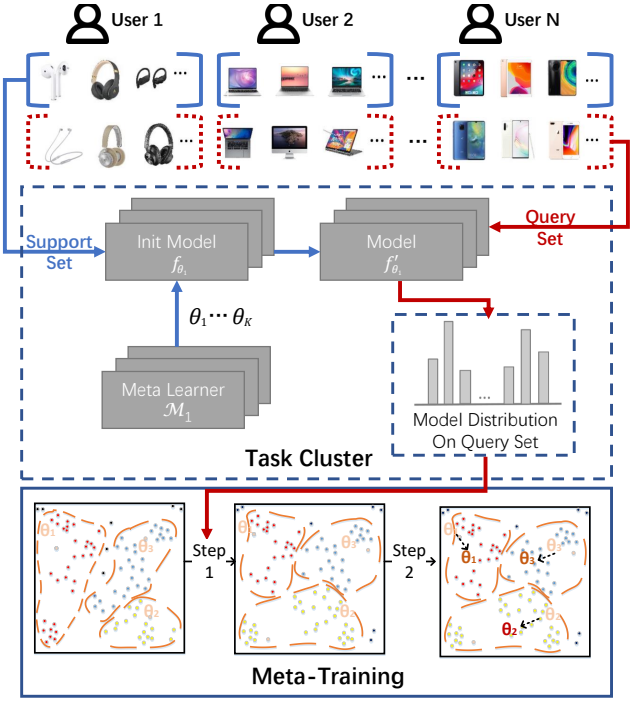

Figure 2: Framework of HML, where multiple initializations can be learned using HTC and meta update.

2.1.1 Meta-Training Phase. Meta-training phase proceeds by alternating between two steps:

Update step Calculate the new meta-learner based on the tasks in each cluster.

Assignment step Assign each task to the cluster which is intuitively the "nearest" meta-learner.

As shown in Figure 3, the step 1 is update step, the step 2 is assignment step, The meta-training phase has converged when the assignments no longer change.

2.1.2 Meta-Test Phase. meta-test phase also proceeds by alternating between two steps:

Assignment step Assign each test task to the cluster, and assign the meta-learner of the cluster to it.

Evaluate step Based on the meta-learner, we further update the parameter via the gradient decent method.

Based on the core idea of our general HML framework, we propose a specific Curriculum Learning-Based Solution in the folllowing sections.

2.2 Curriculum Learning-Based Solution for our General HML Framework

Based on the general framework, we propose HTC scheme for assignment step in both meta-training and meta-test phases. The update step and evaluate step in each phase are mentioned below.

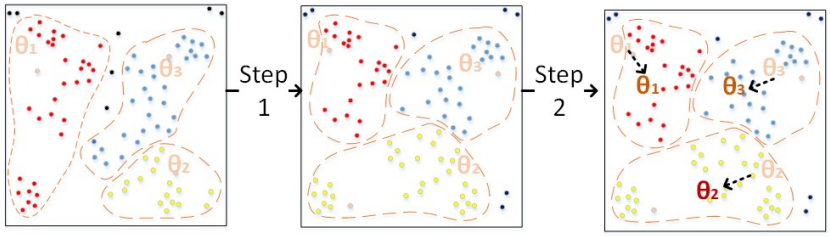

Figure 3: Illustration example of Multi-Initialization updating.

Table 1: Differences of task partitioning in MAML and HML

Method		MAML	HML
train	meta-train	Support	Support
		Query	Query
	HTC	None	Support
			Query
test	HTC	None	Resample Sup_s and Sup_q from Support
			Sup_s
			Sup_q
	meta-test	Support	Support
		Query	Query

2.2.1 Meta-Training Phase. For each cluster, this phase aims to learn a meta-learner from multiple episodes. In each episode, meta-training has a two-stage optimization. Stage-1 is called base-learning, where the cross-entropy loss is used to optimize the parameters of the base-learner. Given an episode T, the base-learner f_θ is learned from episode training data $T^{(tr)}$ and its corresponding loss $\mathcal{L}_{\mathcal{T}^{tr}}(f_\theta)$,

$$\min_\theta \mathcal{L}_{\mathcal{T}^{tr}}(f_{\theta'}) = \mathcal{L}_{\mathcal{T}^{tr}}\left(f_{\theta-\alpha\nabla_\theta \mathcal{L}_{\mathcal{T}^{tr}}(f_\theta)}\right) \tag{1}$$

After optimizing this loss, the baselearner has parameters$\tilde{\theta}_{\mathcal{T}}$. Stage-2 contains a feed-forward test on episode test datapoints. The test loss is used to optimize the parameters of the meta-learner.

Then, the meta-learner is updated using test loss $\mathcal{L}_{\mathcal{T}^{te}}(f_\theta)$. After meta-training on all episodes, the meta-learner is optimized by test losses,

$$\theta \leftarrow \theta - \beta\nabla_\theta \mathcal{L}_{\mathcal{T}^{te}}(f_{\theta'}) \tag{2}$$

Therefore, the number of meta-learner updates equals to the number of episodes.

2.2.2 Meta-Test Phase. In this phase, after tasks are clustered by HTC, and given the meta-learn as initialization in cluster which the test task belongs to. Given $\mathcal{T}_{unseen}$, the meta-learner$\tilde{\theta}_{\mathcal{T}}$ teaches the base-learner $\theta_{\mathcal{T}_{unseen}}$ to adapt to the objective of $\mathcal{T}_{unseen}$ by some means. Then, the test result on $\mathcal{T}^{te}_{unseen}$ is used to evaluate the meta-learning approach. If there are multiple unseen tasks, the average result on $\mathcal{T}^{te}_{unseen}$ will be the final evaluation.

2.2.3 Hard Task Cluster (HTC) scheme. In this section, we introduce a method to assign tasks into different clusters in both meta-training and meta-test phase.

Algorithm 1 Hybrid Meta-Learning (HML)

Require: $p(\mathcal{T})$: distribution of tasks from multiple domains
Require: α, β: step size hyperparameters
Randomly initialize θ_k for recommendation model f_{θ_k}
while not converge **do**
  Sample batch of tasks T from all $p(\mathcal{T})$
  for all task T_i in T **do**
    Find the cluster labelled by θ^*: $\theta^* = Cluster\left(T_i, f_{\theta_k}\right)$
    Compute the inner gradient $\nabla_{\theta^*}\mathcal{L}_{T^{(i)}_{sup}}(f_{\theta^*})$
    Realize adaptation: $(\theta^*)' = \theta^* - \alpha\nabla_{\theta^*}\mathcal{L}_{T^{(i)}_{sup}}(f_{\theta^*})$
  end for
  for all meta parameter θ_k **do**
    Aggregate all query sets labelled by θ_k: $T_{que,k}$
    Update using: $\theta_k \leftarrow \theta_k - \beta\sum\nabla_{\theta_k}\mathcal{L}_{T_{que,k}}\left(f_{\theta'_k}\right)$
  end for
end while

HTC is essentially a simple but effective scheme to identify tasks in each of different domains. Specifically, HTC module takes each task T_i as input to learn the most suitable initialization θ^*. Denote by $T^{(i)}_{sup}$ the support set and $T^{(i)}_{que}$ the query set in T_i. TThe support set $T^{(i)}_{sup}$ of task T_i is input into every recommendation model $f_{\theta_k}, k = 1, 2, \cdots, K$, which is optimized by minimizing the loss of $f_{\theta_k}\left(T^{(i)}_{sup}\right)$. After fast adaptation using one or a few gradient updates, the model f_{θ_k} becomes $f_{\theta'_k}$ to fit query set $T^{(i)}_{que}$. The query set $T^{(i)}_{que}$ of task T_i is input into every recommendation model $f_{\theta'_k}$ for evaluation. We calculate the accuracy from every recommendation model $f_{\theta'_k}\left(T^{(i)}_{que}\right)$, where the accuracy is measured by the following MSE loss

$$\mathcal{L}_{T^{(i)}_{que}}\left(f_{\theta'_k}\left(T^{(i)}_{que}\right)\right) = \sum\left(y_{i,j} - \hat{y}_{i,j}\right)^2, \tag{3}$$

where $y_{i,j}$ is the preference of user i for item j, and $\widehat{y}_{ij}$ is preference predicted by recommendation model $f_{\theta'_k}\left(T^{(i)}_{que}\right)$.

The task T_i is assigned to the cluster represented by initialization θ^* that generates the highest accuracy.

In summary, the procedure is represented as $\theta^* = Cluster\left(T_i, f_{\theta_k}\right)$.

As shown in Table 1, similar to train phase, tasks in test phase alse clustered by HTC. The only difference is that HTC's inputs

Table 2: Recommendation results, where NDCG@10 is used.

Type	Method	MovieLens			Bookcrossing		
		MAE	RMSE	NDCG	MAE	RMSE	NDCG
Recommendation of existing items for existing users	WD	0.7206	0.9107	0.4577	0.9976	1.3904	0.4195
	DeepFM	0.7244	0.9152	0.4592	0.9894	1.3824	0.4177
	MeLU	0.7137	0.9082	0.4654	0.9807	1.3624	0.4288
	HML	**0.6968**	**0.8573**	**0.4959**	**0.9622**	**1.1353**	**0.4032**
Recommendation of existing items for new users	WD	0.9385	1.1520	0.4252	1.3863	1.7623	0.3996
	DeepFM	0.9530	1.1666	0.4217	1.3824	1.7673	0.3916
	MeLU	0.9080	1.0767	0.4357	1.4663	1.6352	0.4011
	HML	**0.8433**	**1.0036**	**0.4245**	**1.2567**	**1.4735**	**0.4045**
Recommendation of new items for existing users	WD	0.9515	1.1720	0.3796	1.5440	1.9438	0.3878
	DeepFM	0.9497	1.1723	0.3726	1.5464	1.9438	0.3807
	MeLU	0.9275	1.1006	0.3878	1.5303	1.7273	0.3923
	HML	**0.8643**	**1.0008**	**0.3834**	**1.4348**	**1.5906**	**0.3952**

of test phase are Sup_s and Sup_q, which are resample from support data.

3 EXPERIMENT

3.1 Datasets

We evaluate the performance of our model on two recommendation benchmark datasets: MovieLens and Bookcrossing. We evaluate the performance on two benchmark datasets:

- **MovieLens** [21] consists of 1 million ratings of 6040 users and 3706 movies. Each rating is an integer in the range of 1 to 5. The ratings are highly sparse, where no ratings occupy 95.8% in MovieLens. The side information for users contains the user's age, gender, occupation and zipcode while the side information for items contains the category of movie genre and release date.
- **Bookcrossing** [43] contains 1149780 books from 278858 users, where each rating is an integer from 0 to 10 and no ratings occupy 99.9%. Some attributes of books and users are also provided and being utilized as the side information.

3.2 Baselines

In order to evaluate the performance, we consider Wide & Deep (WD) [8] ,DeepFM [20] and MeLU [26] as baselines in our experiments.

3.3 Evaluation metric

The root mean squared error (RMSE), mean absolute error (MAE) and NDCG@k are used as evaluation metrics, where RMSE and MAE are defined as

$$RMSE = \sqrt{\frac{1}{|T_q|}\sum_{y_{i,j}\in T_q}\left(y_{i,j}-\hat{y}_{i,j}\right)^2}, \tag{4}$$

$$MAE = \frac{\sum_{y_{i,j}\in T_q}\left|y_{i,j}-\hat{y}_{i,j}\right|}{|T_q|}, \tag{5}$$

where T_q is the query set of test tasks and $|T_q|$ is the total number of ratings in T_q.

NDCG@k are defined as

$$NDCG@K = Z_K\sum_{i=1}^{K}\frac{2^{r_i}-1}{\log_2(i+1)},$$

where Z_K is the normalizer to ensure that the perfect ranking has a value of 1; r_i is the graded relevance of item at position i.

3.4 Experiment Setup

MovieLens and Bookcrossing are commonly used for evaluating the performance of recommendation. Both datasets provide basic user and item information, such as user's age and publication year, and the datasets have explicit feedback information. Similar to [27], we divided the items and users into two groups (existing/new) to evaluate the performance under item-cold-start and user-cold-start condition.

3.5 Performance Comparison

We consider **regular recommendation (Recommendation of existing items for existing users)** with existing items and users, **new-user recommendation (Recommendation of existing items for new users)** with existing items, and **new-item recommendation (Recommendation of new items for existing users)** with existing users, and compare our HML with baselines. Table 2 shows RMSE, MAE and NDCG@K, which demonstrates that HML achieves the best performance for all cases.

4 CONCLUSION

By extending model-agnostic meta-learning (MAML) to seek multiple proper priors for complex tasks, a hybrid meta-learning (HML) algorithm is proposed, which is able to modulate its meta-learned prior parameters by task based cluster method, allowing more efficient fast adaptation. In addition, we introduce the hard task cluster (HTC) scheme as an effective learning curriculum task based cluster method for HML. Experimental results demonstrate that our method achieves a superiority over the state-of-the-art methods on cold-start recommendation.

REFERENCES

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. 2016. Learning to learn by gradient descent by gradient descent. In *In Neural Information Processing Systems (NIPS)*.

[4] Y. Bengio, S. Bengio, and J. Cloutier. 2002. Learning a synaptic learning rule. In *Ijcnn-91-seattle International Joint Conference on Neural Networks*.

[7] Fei Chen, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. 2018. Federated meta-learning for recommendation. *arXiv preprint arXiv:1802.07876* (2018).

[8] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. ACM, 7–10.

[9] Zhiyong Cheng, Xiaojun Chang, Lei Zhu, Rose Catherine Kanjirathinkal, and Mohan S. Kankanhalli. 2019. MMALFM: Explainable Recommendation by Leveraging Reviews and Images. *ACM Trans. Inf. Syst.* 37, 2 (2019), 16:1–16:28. https://doi.org/10.1145/3291060

[11] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative Memory Network for Recommendation Systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. 515–524. https://doi.org/10.1145/3209978.3209991

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *34th ICML* (2017).

[20] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *26th IJCAI*. 1725–1731.

[21] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[26] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation. In *25th ACM SIGKDD 2019, Anchorage, AK, USA, August 4-8, 2019*. 1073–1082. https://doi.org/10.1145/3292500.3330859

[27] 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*. 1073–1082.

[33] Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *In International Conference on Learning Representations (ICLR)*.

[35] Jurgen Schmidhuber. 1987. Evolutionary principles in self-referential learning. *On learning how to learn: The meta-meta-... hook.) Diploma thesis, Institut f. Informatik, Tech. Univ. Munich* (1987).

[36] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, Lexing Xie, and Darius Braziunas. 2017. Low-Rank Linear Cold-Start Recommendation from Social Data. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. 1502–1508. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14828

[37] Suvash Sedhain, Scott Sanner, Darius Braziunas, Lexing Xie, and Jordan Christensen. 2014. Social collaborative filtering for cold-start recommendations. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 345–348.

[40] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. 2017. A Meta-Learning Perspective on Cold-Start Recommendations for Items. In *NeurIPS, 4-9 December 2017, Long Beach, CA, USA*. 6904–6914. http://papers.nips.cc/paper/7266-a-meta-learning-perspective-on-cold-start-recommendations-for-items

[41] Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. 2019. Multimodal Model-Agnostic Meta-Learning via Task-Aware Modulation. In *NeurIPS*. 1–12.

[43] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*. ACM, 22–32.