Studies on Information Augmentation for Cold-Start Recommendation

李, 玉

Studies on Information Augmentation for Cold-Start Recommendation

Yu Li

Department of Economic Engineering

Graduate School of Economics, Kyushu University

January 2024

Abstract

Deep learning-based recommendation systems have achieved tremendous success. However, training a deep neural network from scratch requires a large amount of labeled data. This issue needs to be addressed in recommendation systems, as there will always be new users with little or no interaction history, as well as new items that are rarely or never rated. Meta-learning methods are proposed and have been widely employed in tasks without densely labeled data. Recently, deep meta-learning-based recommendation systems have been proposed to fine-tune deep neural networks with a set of items rated by the user, referred to as a support set. However, how to construct an effective support set is less explored for meta-learning-based recommendation. Specifically, an effective support set should be small, so the user is not bothered, yet informative enough for the recommender to quickly capture the user's preferences. Therefore, in this thesis, we focus on enhancing the informativeness of the support set, thereby significantly improving recommendation effectiveness without increasing the size of the support set. To enhance the informativeness of the support set, we employ two methods: (i) actively selecting informative items to construct the support set (Chapters 3 and 4), and (ii) increasing the informativeness of items in the support set (Chapter 5).

In Chapter 3, we propose Information Gain-Driven Cold-Start Recommendation (InfoRec), which actively queries informative items to estimate user preferences. Specifically, InfoRec quantifies item informativeness by reducing uncertainty about user preferences for items unseen by the user. Theoretically, we model the active support set construction as a submodular optimization problem. Based on this modeling and submodular theory, we prove that greedily selecting the item with the largest information gain would achieve a sufficiently good solution for the optimization problem. In the experiments, we compare the cold-start recommendation performance of our method with two groups of baseline methods: (i) traditional methods widely used for recommendation, and (ii) meta-learning-based methods designed to solve the cold-start problem using meta-learning algorithms. These methods are tested on two standard public recommendation datasets, namely MovieLens-1M and BookCrossing. Experiments demonstrate that InfoRec outperforms baseline methods on two benchmark datasets in different cold-start scenarios.

Dealing with millions of items makes it impractical to iterate through all of them to calculate their information gain and select the most informative item each time a recommendation is generated. Therefore, in Chapter 4, we formulate the sequential

recommendation task as a Markov decision process (MDP). Specifically, we model the recommendation process as a multi-step interaction process. In other words, the recommendation model takes feedback from the last time step as input and outputs a new item, as an action, to query the user's preference. Human feedback is then considered the observation of the recommender at the next recommendation time step. Based on the formulated MDP, we develop an imitation learning framework in which a deep neural network mimics the optimal item selection policy during the training phase and performs informative item sampling with a single feedforward process. The optimal item selection policy is derived from InfoRec, which has been theoretically and empirically proven to be near-optimal for cold-start recommendation. Although InfoRec requires iterating over all items during each recommendation step, this is only necessary during the training phase. As a result, it does not impact real-time performance during the test phase when interacting with humans. In the experiments, we adhere to the experimental setting of the previous work, and we empirically prove that our approach can quickly and effectively learn the preferences of new users and items without iterating through all items at each recommendation step.

In addition to actively selecting informative items, we also augment the item with information from the internet. Meta-learning based recommendation methods often have embedding process and use categorical features, e.g., user/item id, to model the user/item representation. Therefore, the performance of these methods is limited to the information contained in the user and item features. Unfortunately, the information pertaining to categorical features is insufficient, as the model would lack awareness of the distinctive characteristics associated with each category. In Chapter 5, we propose LLM-MetaRec to enhance cold-start recommendation quality using two types of prior knowledge: recommendation specific prior knowledge learned from meta-learning, and general prior knowledge from the LLMs. Our approach involves encoding both item and user information using the LLMs, allowing the downstream recommender neural network to leverage LLMs embeddings and effectively incorporate general prior knowledge. Furthermore, we train the recommendation neural network with meta-learning to enable the recommender to acquire recommendation-specific prior knowledge from previous tasks. In experiments, we present evidence of LLM-MetaRec's superiority over baseline techniques under different cold-start scenarios.

Acknowledgments

First and foremost, I wish to express my deepest gratitude to my supervisor, Professor Tetsuya Furukawa, for his support and guidance throughout my doctoral studies. I vividly recall the challenges of my first year in the pursuit of a PhD, a period shaped by the global pandemic that prevented me from entering Japan. Despite the distance, my supervisor helped me explore research direction by maintaining weekly online discussions. He is such a knowledgeable, patient, and lovely person to work with. Thanks to Professor Tetsuya Furukawa, these three and a half years become one of the most memorable experiences in my life, and being his student is such a great privilege.

I especially thank Qiming Zou and Dr. Ke Lu, whose deep insights and enthusiasm inspired me to pursue PhD research in this field, despite all the difficulties that I had to overcome. Qiming Zou has been a fantastic partner, providing insightful discussions that enriched my research work. Dr. Lu deserves special thanks for hosting me at AI-GOOGLING company, and providing me with the fantastic and enriching opportunity to conduct novel research. I also appreciate the financial support provided by the China Scholarship Council (Grant Number 202108050161).

Last but certainly not least, the support from my family and friends has always been the pillar of my PhD. I will dedicate this thesis to my parents and my grandparents, for everything that I have accomplished.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

The digital economy, propelled by cutting-edge technologies such as the Internet, big data, cloud computing, and artificial intelligence, is undergoing rapid development [29, 37, 59]. It is increasingly intertwining with various stages of economic and social progress and is pivotal in restructuring global resource components, reshaping the worldwide economic framework, and altering the global competitive landscape [28,35,41]. In the digital economy, where vast amounts of data are generated and processed, Recommendation Systems (RSs) have become indispensable for various fields such as e-commerce, streaming services, social media, etc [12,23,27,31,47,52]. They not only facilitate personalized content delivery but also drive customer engagement and satisfaction. By recommending relevant products, services, or content, businesses can increase user retention, drive sales, and create a more personalized and efficient digital experience [61]. Moreover, recommendation systems assist businesses in making data-driven decisions, improving operational efficiency, and staying competitive in the rapidly evolving digital landscape [3].

The evolution of data-driven machine learning algorithms [5, 6, 11, 22, 60, 69, 75, 77], has driven substantial advancements in both academic and industrial research. These advancements have significantly enhanced the performance of recommendation systems, improving aspects such as accuracy, diversity, and interpretability [15, 40, 78]. Particularly, due to expressive representation learning abilities to discover hidden dependencies from sufficient data, deep learning based methods have been introduced in contemporary recommendation models [15, 78]. By leveraging numerous training instances with diverse data structures, e.g., graphs [40], recommendation models with deep neural architectures are typically designed to effectively capture nonlinear and nontrivial user/item relationships. Deep learning-based models are typically trained from scratch using a labeled dataset [32]. In the recommendation scenario, the labeled dataset consists of item-rating pairs, where the rating is provided by users. This labeled dataset is usually referred to as "interactions," since the recommender queries the rating of an item, and the user responds with a rating for the item. The challenge lies in the fact that the performance of deep learning-based recommendation systems heavily depends on the availability of a substantial number of interactions. However, for new users/items, the number of interactions is limited, which is commonly referred to as the cold-start problem [46]. Consequently, the recommender is required to quickly capture (i) the preferences of the new user for items on the online platform; and (ii) the preferences of previous users for the new item. In a word, data insufficiency issue bottleneck deep learning based recommendation models.

Meta-learning algorithms are proposed for few-shot learning tasks, wherein a deep neural network is trained to solve multiple tasks, each of which contains only a few labeled data points [24]. Here, these tasks are sampled from the same domain, e.g., a task is recommending movies to a user. The key idea of meta-learning is to acquire prior knowledge, referred to as meta-knowledge, from previous tasks, and then fine-tune this

prior knowledge with a small set of labeled examples, referred to as the support set, from a new task. Since the learned prior knowledge accurately describes the domain features, the algorithms require a few labeled data points for fine-tuning on a new task from the same domain as the previous tasks. Several works introduce meta-learning, especially the well-known Model-Agnostic Meta-Learning (MAML) [13], into recommendation systems to alleviate the cold-start problem [9, 33]. These works learn prior knowledge from previous recommendation tasks and utilize the prior knowledge to facilitate the learning of a new user preference by observing a few interactions, i.e., users' support set. An efficient support set can quickly identify the preference of a new user and thus should be carefully selected.

The construction of an effective support set is less explored in meta-learning-based recommendation. Previous meta-learning methods either adopt all the interaction history or randomly sample items for the support set. The support set may contain uninformative items, leading to two problems: (i) Recommendation systems may suggest highly irrelevant items to new users, resulting in a poor user experience during the support set construction phase; and (ii) a low-quality support set may fail to capture the personality of new users, jeopardizing long-term recommendation quality. Therefore, a strategy for constructing an effective support set is essential. Specifically, the meta-learning-based recommender should be fine-tuned with a small support set, ensuring that the user is not overwhelmed, while still being informative enough for the recommender to quickly capture the user's preferences.

1.2 Contributions of This Thesis

1.2.1 Information Gain based Dynamic Support Set Construction

In practice, not all items are equally informative. If the recommendation model is very confident about the value of currently unobserved items, then acquiring them would be redundant. On the other hand, there is no point to acquire uninformative items that do not help the model prediction. We propose Information gain driven cold-start Recommendation (InfoRec), which actively acquires informative items for estimating the user preference. Specifically, our InfoRec quantifies item informativeness with the reduction of uncertainty about the user preference for other items. Experiments demonstrate that (i) InfoRec outperforms baseline methods when dealing with new users/items; (ii) InfoRec consistently improves its performance along with the increasing of the support set size; (iii) InfoRec improves the long-term performance in a time-sensitive cold-start recommendation task.

1.2.2 Support Set Construction by Imitation Learning

In a recommendation system, there could be millions of items. It is not practical to iterate through all items to calculate their information gain and select the most informative item each time a recommendation is generated. Therefore, we formulate the sequential recommendation task as a Markov decision process (MDP). Based on the formulated MDP, we develop an imitation learning framework in which a deep neural network mimics the optimal item selection policy during the training phase and performs informative item sampling with a single feedforward process. Experiments demonstrate that our method achieves comparable performance without evaluating the information gain of each item during the test phase.

1.2.3 Integrating Prior Knowledge from Large Language Models

In addition to actively selecting informative items, we also enhance the item with information from the internet. Specifically, the large language model is pre-trained with internet data, containing a wealth of information about the human world. By embedding the item and user descriptions using both the large language model and the deep neural network trained for the recommendation task, the downstream recommender, which takes the concatenated embeddings as input, can learn to develop efficient recommendation strategies based on general knowledge from the language model and specific knowledge from the recommendation task. Experiments demonstrate that, without additional training, general prior knowledge from pre-trained LLMs improves the performance of off-the-shelf recommender in cold-start recommendation tasks.

1.3 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 2, we start with a discussion of the research works related to ours and introduce the preliminary. Chapter 3 discusses our work for information gain based dynamic support set construction. Chapter 4 describes our algorithm for support set construction by imitation learning. In Chapter 5, we develop an approach which combines meta-learning and LLMs together in order to improve the prediction accuracy. Chapter 6 gives the conclusion and the future directions. Fig.1.1 illustrates the organization of the thesis.

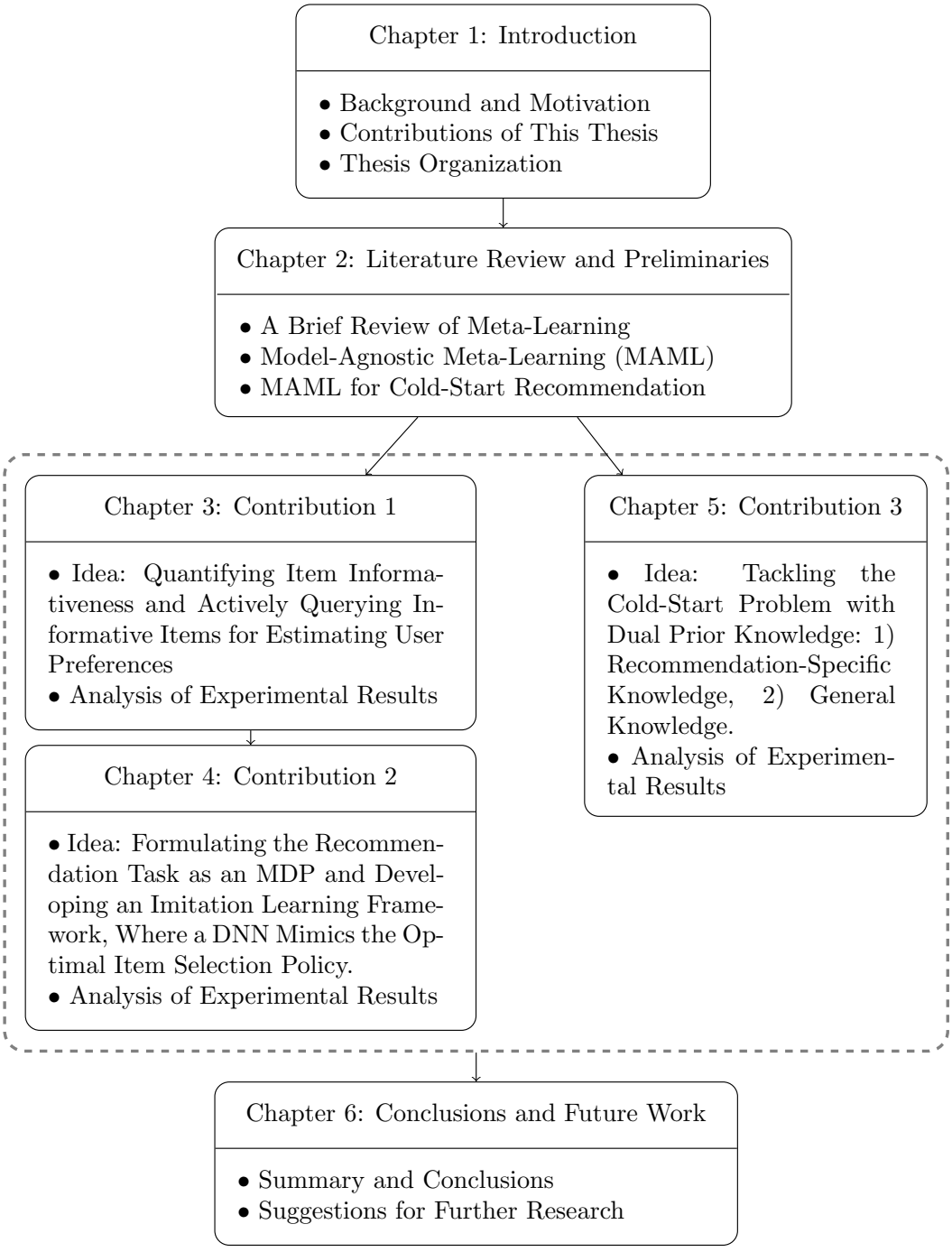


Figure 1.1: The diagram of the thesis organization.

Chapter 2

Literature Review and Preliminaries

This chapter introduces related work of research topics in this thesis. It starts with the introduction of Meta-learning and Model-Agnostic Meta-learning model. After that, it reviews the development of Meta-learning in recommendation system and the efforts to alleviate the cold-start problem. For the related work, this chapter discusses the shortcomings of current methods.

2.1 A Brief Review of Meta-Learning

Traditional Machine Learning (ML) and Deep Learning (DL) models train a specific task from scratch through: (a) the training phase in which a model is initiated randomly and then updated, and (b) the test phase in which the model is evaluated. While ML and DL have achieved remarkable success in a wide range of applications, they are notorious for requiring a huge number of samples to generalize [1]. In many real-world scenarios, gathering amounts of data is costly, time-consuming, and may even be impractical due to physical system constraints [13].

Humans possess the extraordinary capability of learning a new concept even after

minimal observation. For instance, consider the task described in Fig. 2.1, where the objective is to identify the artist to whom a given test painting on the right belongs. In this scenario, the training data consists of paintings from two different artists, each with only two samples. Despite this limited training data, we can still make accurate predictions regarding the attribution of the paintings in the test dataset. Humans have an extraordinary ability to acquire new concepts with minimal training data and adapt rapidly to new tasks. To do so, they build upon their prior experience, reusing concepts and abstractions they have built up over time to adapt from only small amounts of new data efficiently. In other words, humans often rely on their prior knowledge rather than starting from scratch when solving new tasks.

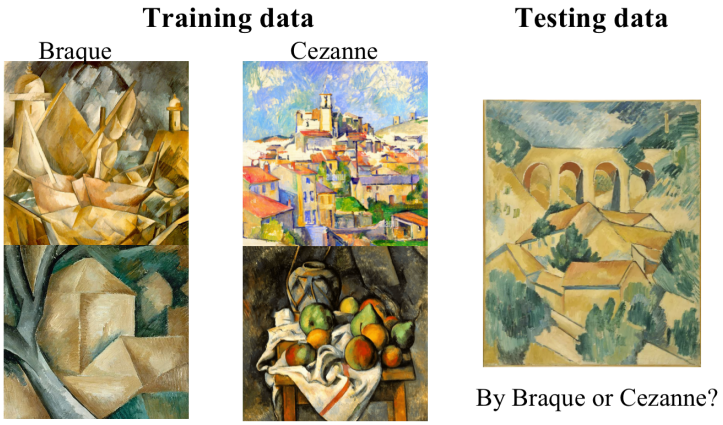


Figure 2.1: An example of accurate predictions with minimal training data.

Motivated by humans' intelligence, meta-learning was introduced with the goal of formulating prior knowledge as a common initialization across tasks, which is then used to quickly adapt to new tasks from small amounts of task-specific data [54]. Driven by the goal to scale AI to a real-world setting and approaches to the human

intelligence paradigm, meta-learning has recently regained significant interest in few-shot problems using deep learning. Currently, a number of meta-learning methods have been proposed. These can generally be grouped into distinct subtypes, including Metric-based meta-learning, Model-based meta-learning, and Optimization-based meta-learning. Optimization-based methods, which rely on stochastic gradient descent, have recently attracted significant attention. Therefore, in this work, we focus on optimization-based methods, and in particular the Model-Agnostic Meta-Learning (MAML) [13].

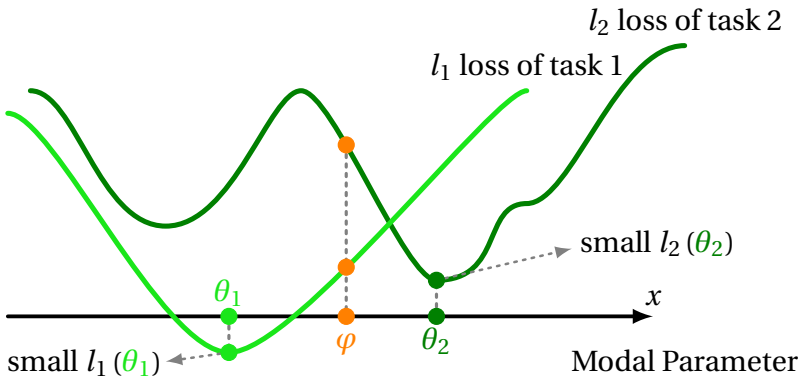


Figure 2.2: An illustration of rapid convergence to local optimum using meta parameter φ.

Intuitively, the MAML algorithm posits a meta parameter φ, enabling the model to quickly approach a local optimum when faced with new tasks. Fig. 2.2 illustrates the existence of a meta parameter φ that enables each task model to approach a local optimum rapidly. The horizontal axis represents the model parameter, simplified to a single parameter, while the vertical axis denotes the loss. The meta parameter φ, positioned in the center, is used to initialize the model parameter for tasks 1 and 2. After training, the models for both tasks quickly converge toward a local optimum. In

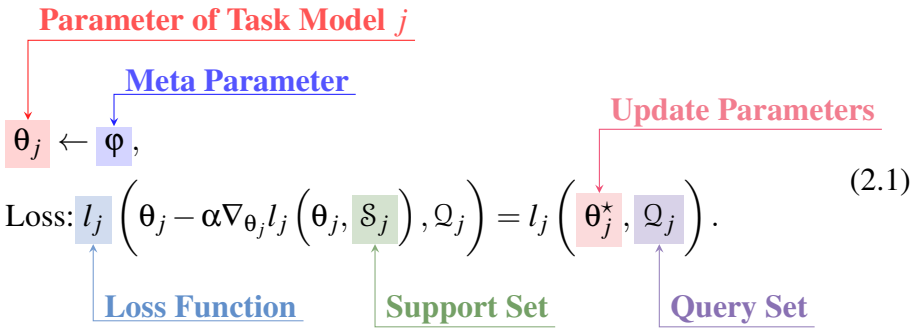
other words, the model parameters for tasks 1 and 2 rapidly reach θ_1 and θ_2, where they achieve lower loss values. In the next section, we will introduce in detail how to obtain the meta parameter.

2.2 Meta-Learning Task Construction for Recommendation

MAML aims to learn prior knowledge, i.e., meta parameter by incorporating additional meta training data $\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$ from previous n recommendation tasks in recommendation systems. Note that the previous recommendation task refers to the historical recommendation for non-cold users with non-cold items and interaction history is a sequence of ratings for the recommended items. The prior knowledge facilitates the learning of a new user preference by observing a small number of user-item interactions, i.e., the meta testing data [13]. MAML includes two stages, namely, meta-training and meta-testing.

Meta-testing: Firstly, in the meta-testing phase, assuming we already have good initial meta parameter φ. Then, when a new task arises, assigning the parameter φ of the meta-network to the network specific to task $\mathcal{T}_j$ to obtain θ_j. Next, using the training dataset of the new task $\mathcal{T}_j$, denoted as $\mathcal{S}_j$, to fine-tune or adapt θ_j to $\theta_j^\star$. Finally, we test the fine-tuned $\theta_j^\star$ on the test dataset of the new task $\mathcal{T}_j$. Note that, to differentiate between these two forms of training and testing sets, the latter training and testing sets are termed "Support set" and "Query set," respectively. More formally, meta-testing

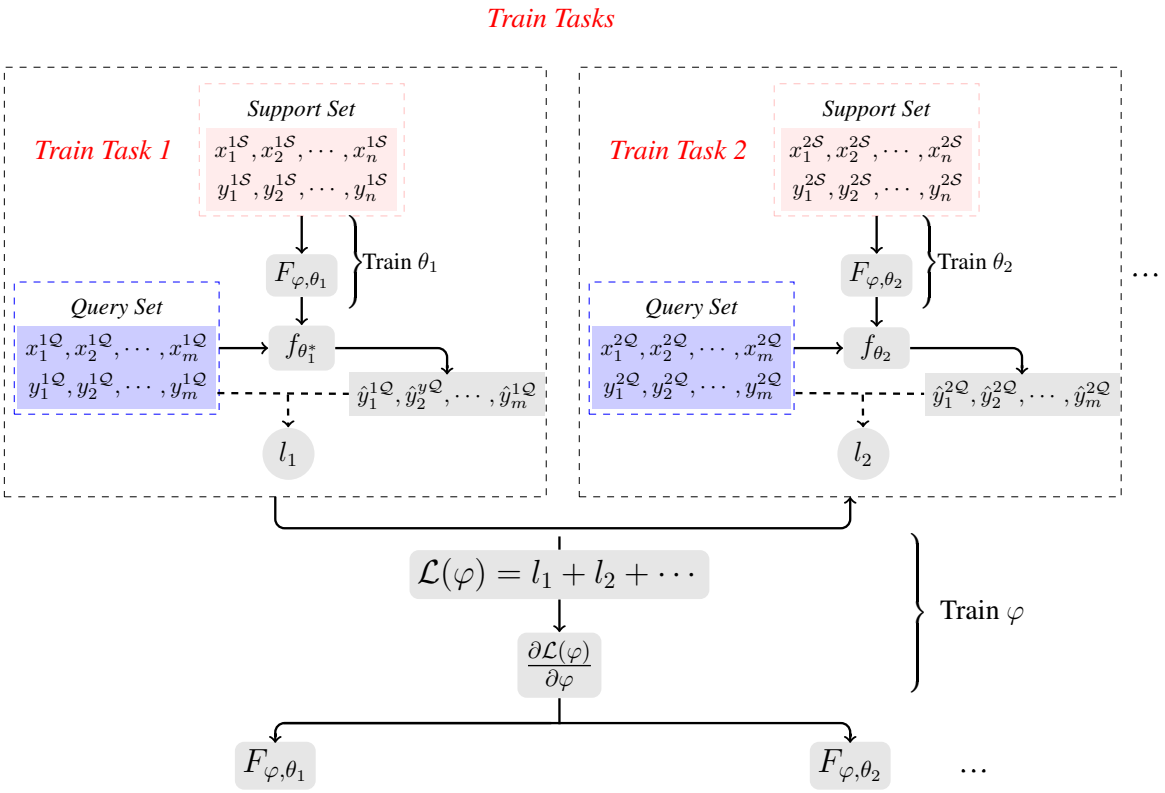
for new task $\mathcal{T}_j$ can be formalized as



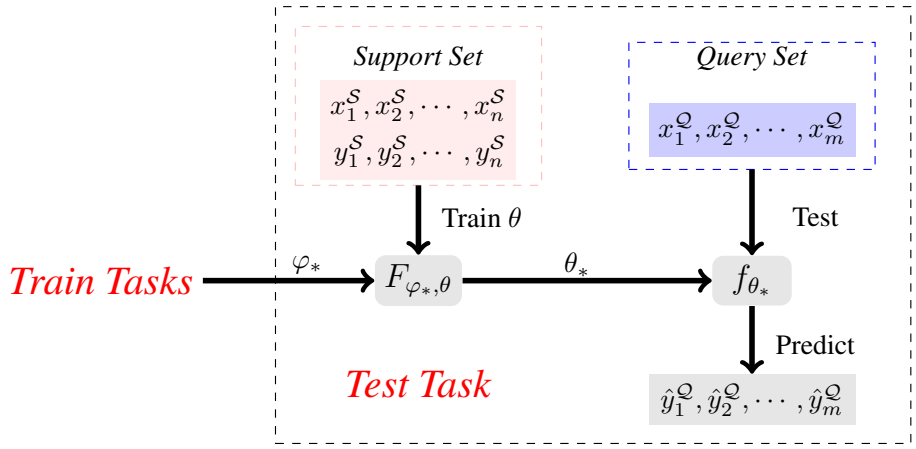
After updating the support set for task $\mathcal{T}_j$, the task-specific parameter θ_j^* will be employed to generate personalized recommendations for task $\mathcal{T}_j$. The meta-testing phase resembles deep learning, involving the training on a dataset followed by testing on separate testing data. The key distinction lies in the initial parameter: meta-testing employs meta parameter, whereas deep learning relies on random initialization. Now, let's explore how to obtain the meta parameter in the following section.

Meta-training: The training procedure of MAML follows a simple machine learning principle: test and train conditions must match [65]. Therefore, each additional meta training data $\mathcal{D}_i$ is split into training set $\mathcal{S}_i$ and testing set $\mathcal{Q}_i$, i.e., $\mathcal{D}_i = \langle \mathcal{S}_i, \mathcal{Q}_i \rangle$. Let $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ denote inputs and outputs, where $\mathcal{X}$ and $\mathcal{Y}$ are the sets of inputs and outputs, respectively. The datasets take the form $\mathcal{S}_i = \left\{ \left(x_k^{i\mathcal{S}}, y_k^{i\mathcal{S}} \right) \right\}_{k=1}^{K}$, and similarly for $\mathcal{Q}_i$. Meta-training can be formalized as bi-level optimization. More formally, Let us denote the recommendation model as f_φ with initial meta parameter φ. During meta-training, MAML first samples several tasks $\mathcal{T}_i \sim p(\mathcal{T})$, where $p(\mathcal{T})$ is the distribution over tasks. Then, the recommendation model will update its parameter through inner-loop (or local) update with support set $\mathcal{S}_i$ and outer-loop (or global) update with the query set $\mathcal{Q}_i$.

Inner-loop update process: First, we have a randomly initialized meta parameter



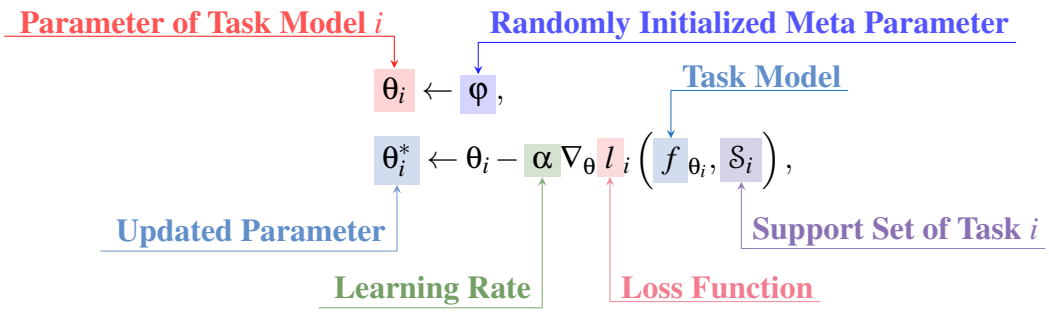
(a) The training process of MAML



(b) The testing process of MAML

Figure 2.3: An illustration that showcases the training and testing process of MAML.

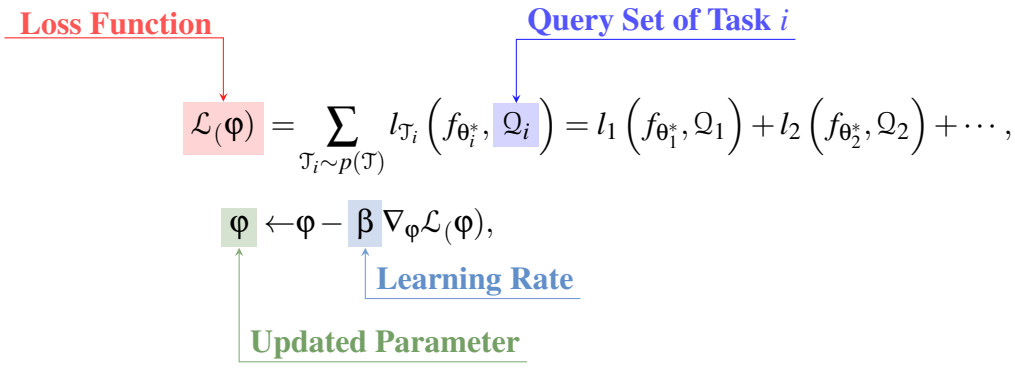
φ. Assigning the parameter φ of the meta-network to the network specific to task $\mathcal{T}_i$ to obtain θ_i. MAML locally updates parameter θ_i to θ_i^* by gradient $\nabla_\theta l_{\mathcal{T}_i}(f_\theta, \mathcal{S}_i)$ for each tasks $\mathcal{T}_i$, which means each task $\mathcal{T}_i$ has an optimal parameters θ_i^*. The parameter update with gradient descent on the support task $\mathcal{S}_i$ can be expressed as follows:



(2.2)

where α is a local update learning rate, θ_i is the model parameter towards task $\mathcal{T}_i$ and $l_{\mathcal{T}_i}\left(f_{\theta_i}, \mathcal{S}_i\right)$ denotes the loss on the support set $\mathcal{S}_i$ for task $\mathcal{T}_i$.

Outer-loop update process: After inner-loop update, MAML globally updates meta parameter φ based on the sum of gradients $\nabla_\varphi \sum_{\mathcal{T}_i \sim p(\mathcal{T})} l_{\mathcal{T}_i}\left(f_{\theta_i}, \mathcal{Q}_i\right)$ as Fig. 2.2 (a). The update equation for the meta parameter φ can be expressed as follows:



(2.3)

where β is a global update learning rate, and $l_{\mathcal{T}_i}\left(f_{\theta_i}, \mathcal{Q}_i\right)$ denotes the loss on the query set $\mathcal{Q}_i$ for task $\mathcal{T}_i$.

A meta-learner model trains in an "Episodic" mode. In each episode, a meta-learner model takes several $\mathcal{S}_i$ to learn task specific parameter θ_i, then applies on the

corresponding $\mathcal{Q}_i$ to make predictions and learn meta parameter φ. The meta-learner updates based on the prediction error over episodes as shown in Fig. 2.4. Over a series of episodes, the meta-learner learns to learn from the small dataset. This phase is called meta-learning.

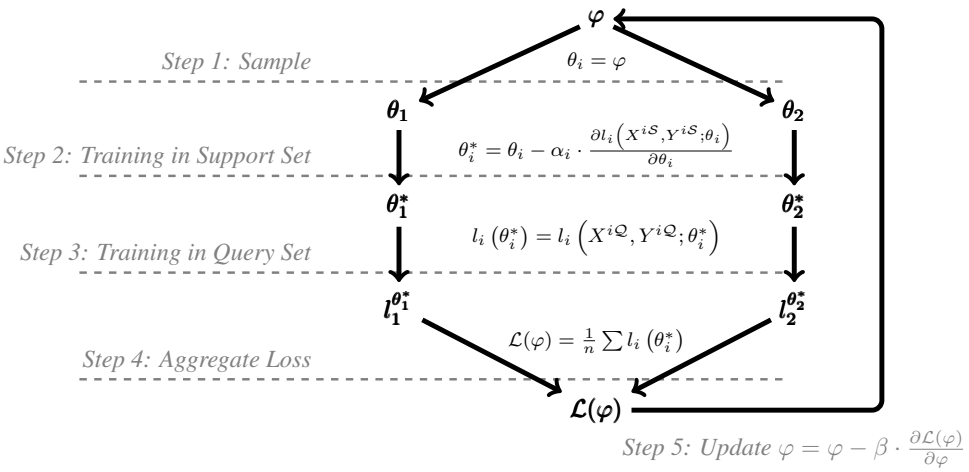


Figure 2.4: A simple flowchart illustrating the training process of MAML.

In summary, the meta parameter in MAML serves as a starting point for the optimization process when adapting to new tasks. They capture the prior information learned during the meta-training phase about how the model should adapt its weights quickly and effectively for new tasks. When confronted with the necessity of adapting the model to a novel task, the learned meta parameter are employed as the initial configuration of the model's weights. Subsequently, the model is subjected to a fine-tuning process on the new task using a small amount of task-specific data. This fine-tuning process should be much quicker and require fewer iterations than training from random initialization.

2.3 Cold-Start Recommendation

Augmenting item/user attribution with auxiliary information is a popular approach for cold-start recommendation [14, 16, 20, 38, 50, 76]. Another common method involves learning user/item latent representations that capture user-item relationships [14, 62]. Additionally, some studies have designed model training mechanisms specifically for the cold-start problem [66, 82]. For example, the dropout strategy has been employed to simulate the missing data scenario in the cold-start setting [66, 82].

Graphs, as structured data, have been widely utilized to address the cold-start problem, incorporating various types such as knowledge graphs [67, 70–72], heterogeneous information networks [58, 73, 80], and social networks [34, 39, 55]. For instance, social networks are employed in collaborative filtering methods [55].

Cross-domain methods transfer knowledge from other domains to facilitate learning in the target domain [4, 25, 36, 79]. For instance, CATN [79] learns review-based associations across domains. However, a drawback of this method is that there may be no users with interaction histories in multiple domains.

However, the above methods are mostly designed to utilize side information, which is not always available. In this work, we propose to enhance cold-start recommendation performance using only user-item interactions.

2.4 Meta-Learning for Cold-Start Recommendation

Recommendation systems play a pivotal role in contemporary digital environments, facilitating users' access to personalized content, products, and services. The challenge of making relevant recommendations becomes particularly pronounced in the cold-start scenario, where a recommendation system has limited or no historical data

about a user or item. Meta-learning, which incorporates additional data to extract task-related information, aims to train a model that can rapidly adapt to a new task with a few samples [13]. Several researchers cast the cold-start task as a meta-learning task and utilized the meta-learning technique to alleviate the cold-start recommendation problem. For example, Vartak et al. propose two distinct deep neural network architectures based on a meta-learning strategy to condition the recommendation model on the task [63]. s^2 Meta applied a feedforward neural network as the recommendation model and updated it via meta-optimization approaches [10]. MeLU proposes a user preference estimator, which integrates user and item attribute information for user preference estimation based on MAML algorithm and learns global parameter initialization of preference estimator as prior knowledge [33]. Similarly, MAMO also utilizes user and item attribute information and uses two memory matrices that can store task-specific memories, which are used to guide the model fast predicting user preference, and feature-specific memories, which are used to guide the model with personalized parameter initialization [9]. However, all these meta-learning based methods did not pay attention to how the support set is constructed when adapting to new tasks, which could greatly influence user engagement with the recommendation system.

Chapter 3

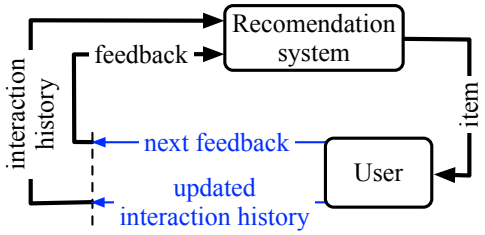
Information Gain based Dynamic Support Set Construction for Cold-Start Recommendation

3.1 Overview

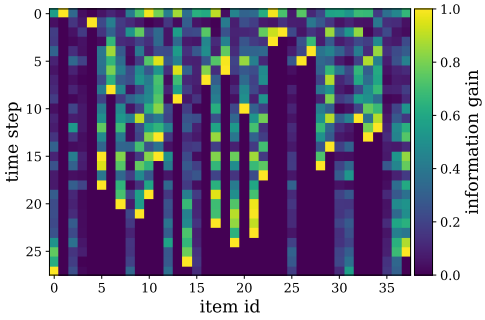
Inspired by few-shot learning, several works introduce meta-learning, especially the well-known Model-Agnostic Meta-Learning (MAML) [13], into recommendation systems to alleviate the cold-start problem. The main idea of MAML is to learn prior knowledge from previous recommendation tasks and utilize the prior knowledge to facilitate the learning of a new user preference by observing a small number of interactions, i.e., the user's support set. During training, the interaction history of a non-cold user consists of a sequence of ratings for non-cold items, which is then randomly divided into a support set and a query set. During test phase, the interaction history of a user, either cold or non-cold, is a sequence of ratings for items that are either cold or

non-cold. The support set can be randomly sampled or collected by another module. MAML is a prediction model that does not assume any specific characteristics of the support set. However, an efficient support set can quickly identify the preference of a new user and thus should be carefully selected.

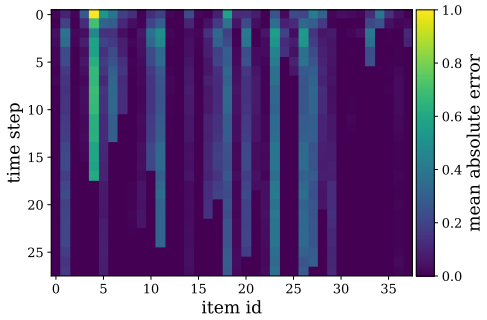
This Chapter proposes Information gain driven cold-start Recommendation (InfoRec) to mitigate the cold-start problem by selecting informative items as the support set for meta-learning. In practice, not all items are equally informative. If the recommendation model is very confident about the value of currently unobserved items, then acquiring them would be redundant. On the other hand, there is no point to acquire uninformative items that do not help the model prediction. This leads us to investigate the question: how to acquire informative items? This Chapter is motivated by the insight that the rating of an informative item can help the recommendation model quickly identify the user's preferences for other items and improve the model prediction performance. As a motivating example, suppose that a user who likes the movie *Harry Potter* tends to like *Pirates of the Caribbean*, and vice versa. As long as we know the user's preference of *Harry Potter*, we can get a relatively certain rating prediction for *Pirates of the Caribbean*, which means that knowing the user preference of *Harry Potter* significantly reduces the uncertainty about user preference for *Pirates of the Caribbean*. Therefore, we determine and acquire items with high information gain as a support set so that it can quickly reduce the uncertainty of the user preference. Compared to the random selection method, our method can quickly decrease the mean absolute error between the true value and prediction of each item as shown in Fig. 3.1. Concretely, InfoRec consists of two key components: (1) modeling and approximating the information gain of each item, and (2) constructing the support set by sampling items with high information gain and adapting a user preference estimator on the support set.



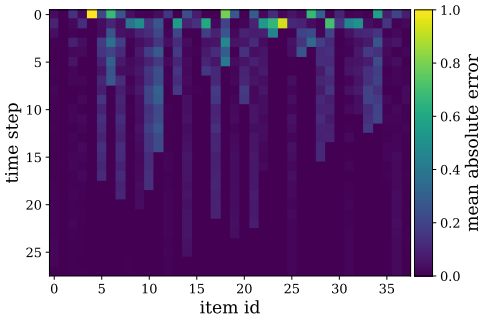
(a) The diagram depicts the interaction between a user and a recommendation system. At each time step, the system asks the user for their preference on an item, and the user provides feedback, such as ratings. A time step represents a specific moment of interaction between the user and the recommendation system.



(b) The colormap represents the normalized item information gain at each time step. Items with high information gain are located in the yellow area while items with low information gain are in the purple area.



(c) The colormap represents the normalized mean absolute error with random selection method at each time step.



(d) The colormap represents the normalized mean absolute error with our method at each time step.

Figure 3.1: A running example of our method.

The contribution of this Chapter is to show the effectiveness of active support set construction for meta-learning-based recommendation and propose a method InfoRec, which constructs the support set via dynamically selecting items with high information gain for improving recommendation performance. The previous method focused on selecting items for rating and used the (item, rating) pairs as the inputs of traditional recommendation techniques, such as matrix factorization [17] and decision-tree-based recommendation algorithm [56]. We focus on selecting a support set used for fine-tuning a neural network recommendation method, which is specific to meta-learning. Our work is motivated by the understanding that the cold-start performance of meta-learning based method is heavily influenced by the interplay between the meta-learner and the support set. By focusing on this relationship and optimizing the meta-learner and support set, we improve the cold-start recommendation performance. In experiments, we demonstrate that InfoRec outperforms baseline methods on two benchmark datasets in both non-cold-start and cold-start scenarios.

3.2 Related Works

Previous studies have demonstrated the importance of incorporating active learning techniques to identify informative items [81]. To enhance the accuracy of recommendations, the recommendation system can utilize active learning techniques to identify informative data and gain a better understanding of user preferences. This approach is especially beneficial in cold-start scenarios, where there is limited user-item interaction.

Several active learning methods have been proposed for recommendation systems, including popularity-based approaches, which select frequently rated items [17], and coverage-based approaches, which aim to increase recommendation diversity by se-

lecting items that have a large Euclidean distance between their feature vectors [56]. However, these approaches tend to select the same set of items for different users, which fails to capture the personal preference [81]. Other methods for determining user preferences involve gathering information through user rating of uncertain items, with uncertainty typically measured using entropy [49] and variance [44]. However, selecting an item based only on the uncertainty of this item could result in reducing uncertainty in a local manner. For example, an item targeting a specific or specialized group of users will have high uncertainty. Picking this item will reduce the uncertainty of only a few other items which are related to this item.

In order to reduce uncertainty in a global manner, influence-based strategies select items that minimize the rating prediction uncertainty for all the items [2, 19, 51]. Specifically, these methods propose to select items that are influential. The influence of an item is measured by the changes in the predicted ratings of other unrated items, knowing the rating of this item. In this work, we suppose that the cold-start recommendation performance is determined by the interplay between the support set and the predictor, e.g., MAML. Based on this motivation, we adapt influence-based method support selection to meta-learning which is capable of capturing the features of a specific user with a handful of interactions.

3.3 Preliminaries and Problem Definition

In this section, we first list the notations used in this Chapter. Then we portray the MAML and its limitations for recommendation systems, as we combine our support set selection method with a MAML based recommendation method. Finally, this section introduces some of the basic concepts of information theory.

Table 3.1: Summary of Notation

Notations	Descriptions
$\oplus$	vector concatenation
u_i	user i
v_j	item j
$\mathcal{U}$	a set of users
$\mathcal{V}$	a set of items
v^*	item with the highest information gain
c_u	user attribute
c_v	item attribute
e_u	user embedding
e_v	item embedding
$[e_u \oplus e_v]$	the concatenation of the user embedding and the item embedding
r_u^v	rating of item given by user
$\hat{r}_u^v$	rating prediction of item given by user
$\mathcal{I}_u$	interaction sequence for user
$\mathcal{T}_i$	i-th recommendation task
$\mathcal{S}_i$	support set of a task $\mathcal{T}_i$
$\mathcal{Q}_i$	query set of a task $\mathcal{T}_i$
f_θ	recommendation model/function parameterized with meta-parameters θ
θ	meta-knowledge obtained with the recommendation model
F_{θ_r}	fully connected neural network parameterized with θ_r
θ'	fine-tuned parameters of the recommendation model
θ_i	task-specific parameters of a personalized model for $\mathcal{T}_i$
α	local update learning rate in the optimization-based meta-learning
β	global update learning rate in the optimization-based meta-learning
$\mathcal{L}(f_\theta, *)$	loss function of the recommendation model f_θ over a given dataset
$H(X)$	entropy of a discrete random variable X
$IG(X;Y)$	the information gain of X obtained from a random variable Y
$D_{KL}(p\|q)$	KL divergence between two probability distributions $p(x)$ and $q(x)$

3.3.1 Basic Concepts in Information Theory

In this part, we will give a brief introduction to basic information theoretic concepts that we will apply in this Chapter.

Entropy The concept of entropy measures the uncertainty of a discrete random variable [57]. More precisely, entropy $H(X)$ of a discrete random variable X with probability distribution $p(x)$ is defined as follows:

$$H(X) = -\sum_{x_i \in X} p(x_i) \log p(x_i), \tag{3.1}$$

where x_i denotes a specific value of random variable X. Note that the larger the entropy, the more unpredictable the variable is.

Conditional entropy The conditional entropy of X given another discrete random variable Y is:

$$H(X \mid Y) = -\sum_{y_j \in Y} p(y_j) \sum_{x_i \in X} p(x_i \mid y_j) \log \Big(p(x_i \mid y_j) \Big), \tag{3.2}$$

where $p(y_i)$ is the prior probability of y_i, while $p(x_i \mid y_j)$ is the conditional probability of x_i given y_j. It shows the uncertainty of X given Y.

Information gain Information gain or mutual information between X and Y is used to measure the amount of information shared by X and Y together. More precisely, information gain $I(X;Y)$ with joint probability $p(x_i, y_j)$ of x_i and y_j is defined as

$$I(X;Y) = H(X) - H(X \mid Y) = \sum_{x_i \in X} \sum_{y_j \in Y} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)\, p(y_j)}, \tag{3.3}$$

The higher the information gain is, the more entropy is removed and variable X is more predictable. In recommendation systems, to reduce the uncertainty of user preference,

we select informative items, i.e., items with high information gain by maximizing information gain.

Kullback–Leibler divergence The Kullback–Leibler divergence (KL divergence) of $q(X)$ from $p(X)$, denoted $D_{KL}(p(X)\|q(X))$, is a measure of the information lost when $q(X)$ is used to approximate $p(X)$. $D_{KL}(p(X)\|q(X))$ is defined in Equation 3.4.

$$D_{KL}(p(X)\|q(X)) = \sum_{x_i \in X} p(x_i) \log \frac{p(x_i)}{q(x_i)} \tag{3.4}$$

Where $p(X)$ and $q(X)$ are two probability distributions of a discrete random variable X. Note that D_{KL} is also called the 'KL distance', but it is not strictly a distance: in general $D_{KL}(p\|q) \neq D_{KL}(q\|p)$.

3.3.2 Problem Formulation

This work considers learning the preferences of different users as different tasks. Assume that we have a set of users and items, denoted by $\mathcal{U} = \left\{u_1, u_2, u_3, \cdots, u_{|\mathcal{U}|}\right\}$ and $\mathcal{V} = \left\{v_1, v_2, v_3, \cdots, v_{|\mathcal{V}|}\right\}$, respectively. Given user u with attribute c_u and his/her interaction $\mathcal{I}_u = \left\{\left(v_j, r_u^{v_j}\right) \mid j = 1,2,3...m\right\}$, where each item v is associated with attributes c_v and corresponding rating r_u^v. During the meta-training phase, the interaction is randomly divided into support set $\mathcal{S}_u$ and query set $\mathcal{Q}_u$. During the meta-testing phase, our method actively constructs the support set and considers the rest of the items in user interaction as the query set.

3.4 Our Method: InfoRec

In this section, we describe the details of InfoRec. Section 3.4.1 presents a recommendation model. In Section 3.4.2, we model and approximate the information gain of

each item. In Section 3.4.3, we construct the support set by sampling items with high information gain and adapting neural network in the support set to identify the user's preference. The overview of InfoRec is illustrated in Fig.3.2.

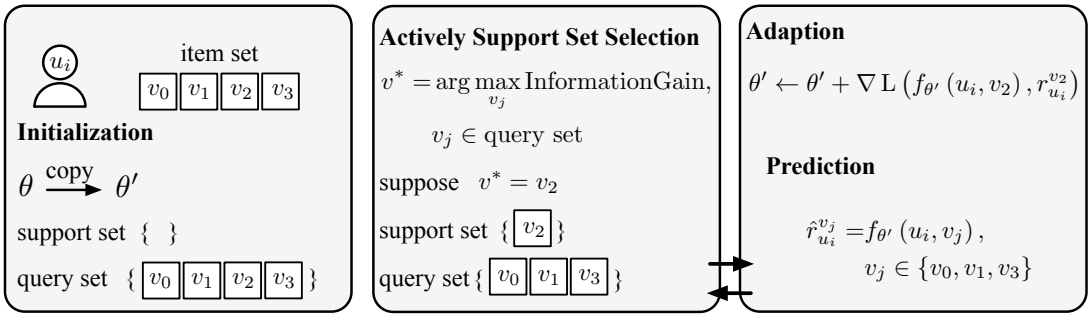


Figure 3.2: Overview of InfoRec.

3.4.1 Architecture of Our Method InfoRec

The recommendation model f_θ includes an embedding module that encodes user information and item information, and a recommendation module which takes an embedding as an input and outputs a rating prediction.

Embedding module The embedding module learns embeddings of users and items. As attributes of users and items have different categorical values, we provide two different strategies to learn the user and item embeddings. Taking item v for example, when item attributes have a single categorical value, such as rate, each attribute is represented with a randomly initialized embedding vector. Then we concatenate them together to obtain the initial item embedding. Suppose m item attributes $c_v^1, c_v^2, ..., c_v^m$, the item embedding procedure is defined as: $e_v^1 = \left[E_1 c_v^1 \oplus E_1 c_v^1 \oplus \ldots, \oplus E_m c_v^m\right]$, where $[\cdot \oplus \cdot]$ is the concatenation operation, and E_m represents the embedding matrix. For

attributes with multiple categories, such as genre, each attribute is first converted to multi-hot representation and then transformed into a low dimension representation by a single layer feed-forward network as: $e_v^2 = (W\boldsymbol{e}_i + b)$, where W denotes weight matrix, b denotes bias vector, and $\boldsymbol{e}_i$ is the multi-hot representation of c_v^i. The final item embedding e_v is concatenation of $[e_v^1 \oplus e_v^2]$. Notice that user embeddings e_u are obtained by following a similar embedding process.

Recommendation module Given the user embedding e_u, and a list of item embedding e_v, the user and item embeddings are concatenated and then mapped by two hidden layers to a low dimension representation. The last layer of recommendation module can compute the preference rating $\hat{r}_u^v$ for each item by: $\hat{r}_u^v = F_{\theta_r}\left([e_u \oplus e_v]\right)$, where F denotes the fully connected neural network with the parameters θ_r.

3.4.2 Item Information Gain

In information theory, the information gain after observing a random variable is the average level of uncertainty reduction for target random variables. From the information gain perspective, this work considers the user preference for an item as a random variable and the user preference of other items as target variables. Therefore, recommending an informative item can be formalized as maximizing the expected uncertainty reduction, i.e., information gain.

Specifically, we consider random variable $f_\theta(u, v')$ as a prediction of user rating for item v' and random variable r_u^v as an observed rating of item v given by a user u. The entropy of random variable $f_\theta(u, v')$ given the fact that r_u^v is known can be written as $H\left(f_\theta(u, v') \mid r_u^v\right)$. Since we aim to find the item which can help the recommendation model to predict user preferences for all other items, we evaluate the average

information gain of an item over all unrated items, i.e.,

$$I(f_\theta(u,v');r_u^v) = \mathbb{E}_{p(v')}\left[H\left(f_\theta(u,v')\right) - H\left(f_\theta(u,v') \mid r_u^v\right)\right], \tag{3.5}$$

where $\mathbb{E}_{p(v')}(\cdot)$ is the expectation over the distribution of item v', $u \in \mathcal{U}, v' \neq v$ and $v, v' \in \mathcal{V}$. Calculating information gain exactly is difficult since the entropy estimation of high dimensional variables could be difficult [43]. Besides, the neural network tends to be overconfident in its prediction, i.e., only one class has a high probability, even for unseen inputs, which leads to the low quality probability distribution [18]. Fortunately, the information gain can be converted to KL divergence. The information gain of a random variable X obtained from an observation of a random variable Y is defined as:

$$\begin{aligned} I(X;Y) &= H(X) - H(X \mid Y) \\ &= H(X) + H(Y) - H(X,Y) \\ &= \sum_x p(x) \log \frac{1}{p(x)} + \sum_y p(y) \log \frac{1}{p(y)} - \sum_{x,y} p(x,y) \log \frac{1}{p(x,y)} \\ &= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \\ &= D_{KL}(p(X,Y)||p(X)p(Y)). \end{aligned} \tag{3.6}$$

Therefore, the information gain can also be calculated with the KL divergence between the joint distribution of X and Y and the product of the marginal distribution

of X and Y. Then Eq. (3.5) can be rewritten as:

$$\begin{aligned} I(f_\theta(u,v');r_u^v) &= D_{\mathrm{KL}}\left[p(r_u^v, f_\theta(u,v'))\|p(r_u^v)p\left(f_\theta(u,v')\right)\right] \\ &= \mathbb{E}_{p(v')}\log\frac{p(f_\theta(u,v')\mid r_u^v)}{p\left(f_\theta(u,v')\right)} \\ &= \mathbb{E}_{p(v')}\left[\log p(f_\theta(u,v')\mid r_u^v) - \log p(f_\theta(u,v'))\right] \\ &\propto \mathbb{E}_{p(v')}||f_{\theta'}(u,v') - f_\theta(u,v')||_2^2 \end{aligned} \tag{3.7}$$

where θ is fine-tuned on the tuple $\left[v, r_u^v\right]$ and we derive θ'. Notation $\propto$ represents the proportional relationship. As introduced before, the uncertainty estimation of neural networks tends to be deterministic. Thus, in the last two lines of the Eq. (3.7), we use the predicted outputs as an alternative way to approximate probability distribution over outputs. Now, we can use Eq. (3.7) to help us calculate the information gain of items rather than calculate the entropy directly.

3.4.3 Cold-Start Recommendation via Meta-Learning with Actively Selected Support Set

In this part, we construct an informative support set based on the information gain defined in Section 3.4.2 for new users and fine-tune the prediction model on the selected support set.

In standard meta-learning, there are two stages, i.e., meta-training and meta-testing. Recommendation model f_θ is trained in meta-training to quickly adapt to new users with a few interactions. The meta-training stage is the same as in MAML in which the support set is randomly sampled. In this part, we assume that we have a fully-trained model f_θ via meta-training and focus on the meta-testing stage.

During meta-testing, to adapt to a specific user, model parameters θ are updated

by a few gradient steps on the user's support set, deriving θ'. However, for a new user u, the initial support set $\mathcal{S}_u$ is empty. Therefore, we dynamically select support items based on the item information gain proposed in Section 3.4.2 and adapt f_θ in the support set to identify new user's preferences. In our method, the model adaption is along with the support set construction process as following steps.

1) Initialization. Initialize $\mathcal{S}_u = \emptyset$ and $\theta' = \theta$.

2) Estimate item information gain. For each item $v \notin \mathcal{S}_u$, we estimate its information gain over other items $v' \notin \mathcal{S}_u$ with Eq. (3.7). Note that Eq. (3.7) requires r_u^v which is not available at this stage. Therefore, we replace r_u^v with $\hat{r}_u^v = f_{\theta'}(u, v)$ which is an estimation of r_u^v.

3) Sample support set item. We assign a probability $q(v)$ to each item v according to their estimation of information gain,

$$q(v) \propto I\left(\hat{r}_u^v; f_{\theta'}(u, v')\right),\ v' \in \mathcal{V},\ v' \neq v. \tag{3.8}$$

Intuitively, items with high information gain would be more likely to be recommended to new users, i.e., $v \sim q(v)$. The user will assign a rating to the item, i.e., we receive r_u^v.

4) Fine-tune θ' on the current support set. After adding labeled pair $\left(v, r_u^v\right)$ into support set $\mathcal{S}_u$, we perform adaption on $f_{\theta'}$ by calculating the gradient of following loss function.

$$\min_{\theta'} \quad \mathcal{L} = -\sum_{v \in \mathcal{S}_u} \left[r_u^v \log\left(\hat{r}_u^v\right) + (1 - r_u^v) \log\left(1 - \hat{r}_u^v\right) \right] \tag{3.9}$$

$$\text{s.t.} \quad \hat{r}_u^v = f_{\theta'}(u, v). \tag{3.10}$$

We then take one/several gradient step(s) to update θ'. Specifically, in the experiment, we take 5 gradient steps. If the size of the support set equals N, then the support

set construction stage is stopped; otherwise, go back to step 2). The adapted $f_{\theta'}$ is then used for recommendation. The entire training procedure of our proposed method InfoRec is shown in Algorithm 1.

Algorithm 1: InfoRec (meta-testing phase)

Input : learning rate for inner loop α, recommendation model f_θ parameterized by θ, item set $\mathcal{V}$, support set size N

Output: recommendation model $f_{\theta'}$ fine-tuned on the selected support set

Initialize θ' with θ

Support set $\mathcal{S}_u = \emptyset$

while *size of* $\mathcal{S}_u < N$ **do**

  for *item* $v \in \mathcal{V} \setminus \mathcal{S}_u$ **do**

    predict user preference for item v, $\hat{r}_u^v = f_{\theta'}(v)$

    calculate item information gain, $q(v) \propto I\left(\hat{r}_u^v; f_{\theta'}(u, v')\right),\ v' \in \mathcal{V},\ v' \neq v$

  end

  add item v^*, i.e., item having the highest information gain, into $\mathcal{S}_u$, $\mathcal{S}_u = \mathcal{S}_u \cup \{v^*\}$

  Compute adapted parameters using $\mathcal{S}_u$, $\theta' \leftarrow \theta' - \alpha \nabla_{\theta'} \mathcal{L}\left(f_{\theta'}, \mathcal{S}_u\right)$

end

3.4.4 Theoretical Analysis

We intuitively and theoretically analyze the reason that significant performance improvement is expected with our method.

Intuitively, querying the user preference for a new item would introduce new information to the user, which helps the recommendation method makes better recommen-

dation subsequentially. However, different items introduce different amounts of new information which depends on the user and the current interaction history. Our method estimates the information gain of each item conditioned on current interaction history and selects the item with the largest estimated information gain. If the information gain estimation is accurate, the recommendation method would quickly identify the user preferences for other items with less interaction.

Theoretically, we model the active support set construction as a submodular optimization problem and show the reasonability of our method. Specifically, in the following, we assume that the negative prediction loss function is proportional to the information contained in the support set and is a monotone and submodular function. Consequently, the support set construction can be considered as a submodular optimization problem. Then, we give Theorem 1 which shows that, for a submodular optimization problem, greedily selecting the item with the largest information gain would achieve a good enough solution.

We first give a definition of *monotone and submodular function*.

Definition 1 (Submodular function)**.** *Let X be a finite set. Function $f : 2^X \to \mathbb{R}$ is a submodular function if for subset $S \subseteq T \subseteq X$ and $x \in X \backslash T$,*

$$f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T). \tag{3.11}$$

Definition 2 (Monotone set function)**.** *A set function f is monotone if for all $A \subseteq B \subseteq N$ the constraint $f(A) \leq f(B)$ holds.*

The submodularity can be understood intuitively as a diminishing return property. That is, if f measures "benefit", then the marginal benefit of adding x to S is at least as high as the marginal benefit of adding it to a larger set T. In other words, since T is a superset of S and already contains more information, adding x will not help as much.

We make the following assumption.

Assumption 1. *Given the loss function defined in Eq.(3.9), if the utility function* $f(\mathbb{S}) = -\mathcal{L}_{\phi}(\mathbb{S}) \propto H(\mathbb{S})$*, then* f *is monotone and submodular.*

Intuitively, we assume that adding a new item to a larger support set would introduce less new information than adding a new item to a smaller support set. Note that we quantify the information contained in the support set as $H(\mathbb{S})$. This assumption is reasonable since, for a new item, there would be a similar item in the support set when the support set is large. And then, the new item would introduce less new information. Besides, the larger support set provides more information for the recommendation method and thus the recommendation method can achieve higher negative prediction loss.

We greedily select the item which introduces the largest information gain to $H(\mathbb{S})$. In Theorem 1, we show that if the function is monotone and submodular, using a greedy algorithm to optimize the function gives a lower-bound performance guarantee of a factor of $1-1/e$ of the optimum [18], and in practice, these greedy solutions are often within a factor of 0.98 of the optimal [43]. More formally,

Theorem 1. *[57] Let* f *be a monotone, submodular, non-negative function on* X*. The greedy algorithm, which starts with* S *as the empty set and at every step picks an element* x *which maximizes the marginal benefit* $f(S \cup \{x\}) - f(S)$*, provides a set* S *that achieves a* $(1-1/e)$*-approximation of the optimum.*

The above Theorem justifies the reasonability of our method.

3.5 Experiments on Cold-Start Recommendation

In this section, experiments are conducted in order to answer the following questions:

- **Q1**. Does InfoRec improve the cold-start recommendation performance compared to baseline methods?
- **Q2**. Does InfoRec consistently achieve superior performances with varying sizes of support sets?
- **Q3**. Does InfoRec improve long-term recommendation performance in cold-start scenarios?

3.5.1 Experimental Settings

We verify our method in four different non-cold/cold settings: a) recommendation on existing items for existing users; b) recommendation on existing items for cold users; c) recommendation on cold items for existing users; and d) recommendation on cold items for cold users. As a recommendation system, it is essential to perform well in both cold scenarios, i.e., setting a, and non-cold scenarios, i.e., setting b, c, and d. Although the focus of this Chapter is on the cold-start problem, reporting the results on the non-cold scenario can also be useful. The inputs and meta-parameters used in the non-cold scenario are precisely the same as those in the cold-start scenario. The only difference is that the training data contains non-cold users or items.

Dataset description The experiments are carried out on MovieLens-1M[1] dataset and Bookcrossing[2] dataset which contain explicit feedback information as well as the basic user and item attribute information. Table 3.2 gives a summary of the statistics of these three datasets.

Baseline methods To validate the effectiveness of InfoRec, we choose two kinds of baselines: (1) Traditional methods, including PPR [47] and Wide & Deep [6], which

[1]https://grouplens.org/datasets/movielens
[2]http://www2.informatik.uni-freiburg.de/ cziegler/BX

Table 3.2: Basic statistics of the MovieLens-1M, Bookcrossing, and Netflix.

	MovieLens-1M	Bookcrossing	Netflix
Users	6,040	278,858	480,189
Items	3952	271,379	17,770
Interactions	$1,000,209$	$1,149,780$	100,480,507
Density	4.190%	0.0015%	1.178%
User attributes	Gender, Age, Occupation, Zip code	Age, Location	-
Item attributes	Publication year, Rate, Genre,	Publication year, Author, Publisher	Genres, Directors, Actors, Descriptions
Rating scale	$1 \sim 5$	$1 \sim 10$	$1 \sim 5$

are classic and widely used for recommendation; (2) Meta-learning based methods, including s^2 Meta ([10]), MeLU ([33]) and MAMO [9], which are designed for solving the cold-start problem by meta-learning algorithms. We describe the baseline methods as follows.

- **PPR**: constructed joint representation of user/item pairs based on their individual feature and built a predictive feature-based regression model which is optimized by minimizing pairwise user preference.
- **Wide & Deep**: built a wide linear model with feature transformations and a deep neural network with dense embeddings. By jointly training the wide and deep components simultaneously, it can obtain both memorization and generalization.
- **MeLU**: proposed a user preference estimator based on MAML, which integrates both user and item attribute information for user preference estimation, and learned global parameter initialization of preference estimator as prior knowledge. According to the number of local updates, MeLU has two versions, i.e., MeLU-1 and MeLU-5.

- s^2 **Meta**: applied a feedforward neural network as the recommendation model and updated it via meta-optimization approach. Specifically, s^2 Meta introduced an update strategy and early-stop controller which can automatically control the learning process to converge to a good solution.

- **MAMO**: used two memory matrices that can store task-specific memories, which are used to guide the model fast predicting user preference, and feature-specific memories, which are used to guide the model with personalized parameter initialization.

- **InfoRec**: quantified the information gain of an item with the reduction of uncertainty about the user preferences for the rest of items, which are not in the support set, and dynamically selects the item with the highest information gain at each timestep.

Evaluation metrics We evaluate the performance of recommendation methods based on the prediction accuracy and correctness of ranking. Therefore, the performance indicators used in this work are the Mean Absolute Error (MAE), and Normalized Discounted Cumulative Gain (nDCG).

- **MAE:** Given a set of users in the test data U', the MAE is the average of the absolute values of the individual prediction errors on all samples in the test set. Each prediction error is the difference between the true value r_u^v and the predicted value $\hat{r}_u^v$. MAE is calculated by

$$MAE = \frac{1}{|U'|} \sum_{u \in U'} \frac{1}{|Q_u|} \sum_{v \in Q_u} |r_u^v - \hat{r}_v^u|,$$

- **nDCG@N:** nDCG is a measure of ranking quality. We first introduce cumulative gain (CG) and discounted cumulative gain (DCG) which are the basics of nDCG.

CG is the sum of the top-N actual rating values sorted by predicted rating values. DCG improves it by taking the ordering of the items into account. Given a list of top-N recommendations, CG@N and DCG@N are calculated by

$$CG@N = \sum_{n=1}^{N} R_{u,n}, DCG@N = \sum_{n=1}^{N} \frac{2^{R_{u,n}} - 1}{\log_2(i+1)},$$

where $R_{u,n}$ is the rating given by user for item at rank position n. As DCG is not normalized so it may vary for different users, researchers propose the ideal discounted cumulative gain (IDCG) as the normalization constant, which calculates the top-N sorted actual rating values. This is an ideal list and it has the largest DCG value. The nDCG@N is computed as

$$nDCG@N = \frac{1}{|U'|} \sum_{u \in U'} \frac{DCG@N}{IDCG@N}.$$

MAE evaluates the accuracy of rating prediction, where a lower value indicates better model performance. nDCG@N considers the preference ordering performance on observed query sets, where a higher value indicates a better performance. In this work, we utilize the nDCG with $N = 3$ as in MAMO ([9]).

3.5.2 Dataset Preprocessing

We preprocess MovieLens-1M and Bookcrossing datasets as follows:

- **MovieLens-1M**: MovieLens-1M has a more even distribution with an average interaction value. And thus, we followed the methodology outlined in reference [33], where 80% of the users were randomly selected as training users (existing users) and the remaining as test users (cold users). We then performed 5-fold cross-validation on this dataset. According to the order of release time,

we consider the top 80 percent of movies as existing items and the remaining 20 percent as cold items.

- **Bookcrossing**: The original Bookcrossing data has several obviously misleading values. For instance, the original user age ranges from 0 to 237, which does not correspond to the actual situation. Therefore, we restrict the user age range to 5 to 90. We followed the methodology outlined in reference [9], where 90% of the users were randomly selected as training users (existing users) and the remaining as test users (cold users). We then performed 10-fold cross-validation on this dataset. For the item, we regard items with more than 5 ratings as warm items, and the others as cold items.

It is important to note that in order to ensure a fair comparison with the baseline method [33] and [9], we have followed the same dataset preprocessing methodology for different datasets. This has been done to maintain consistency and achieve comparable results.

3.5.3 Result and Analysis

To answer **Q1**, in part **Cold-start performance comparison (Q1)**, we actively construct the support set of the same size as baseline methods. To answer **Q2**, in part **Impact of support set size (Q2)**, we progressively add new items to the support set which is initialized as an empty set and evaluate the change of performance of our method and baseline methods.

Cold-start performance comparison (Q1)

In this experiment, we compare the overall performance of all methods on Movielens-1M and Bookcrossing datasets in both non-cold-start and three cold-start scenarios defined above. To enable a fair comparison, we adopt the same support set size as the

Table 3.3: Comparison of the performance of various recommendation methods.

		MovieLens-1M		Bookcrossing	
Type	Method	MAE	nDCG@3	MAE	nDCG@3
Recommendation on existing items for existing users	PPR	**0.1820**	**0.9831**	3.8092	0.8494
	Wide & Deep	0.9047	0.9117	1.6206	0.9172
	s^2 Meta	0.7567	0.8870	1.5147	0.8781
	MeLU-1	0.7661	0.8904	0.7799	0.9572
	MeLU-5	0.7567	0.8919	0.7955	0.9552
	MAMO	0.8725	0.8866	1.4879	0.8879
	InfoRec	0.6869	0.9174	**0.3715**	**0.9930**
Recommendation on existing items for cold users	PPR	1.0748	0.8468	3.8430	0.8434
	Wide & Deep	1.0694	0.8639	2.0457	0.8515
	s^2 Meta	0.8443	0.8401	1.8738	0.8490
	MeLU-1	0.7884	0.8810	1.8701	0.8527
	MeLU-5	0.7854	0.8812	1.8767	0.8532
	MAMO	0.8967	0.8799	1.6217	0.8565
	InfoRec	**0.7300**	**0.9036**	**1.4499**	**0.9258**
Recommendation on cold items for existing users	PPR	1.2441	0.7632	3.6821	0.8367
	Wide & Deep	1.2655	0.7721	2.2648	0.8437
	s^2 Meta	**0.8860**	**0.8323**	1.9767	0.8463
	MeLU-1	0.9361	0.7990	2.1047	0.8441
	MeLU-5	0.9275	0.8005	2.1236	0.8440
	MAMO	0.9306	0.8315	1.7379	0.8402
	InfoRec	0.8943	0.8094	**1.5290**	**0.9126**
Recommendation on cold items for cold users	PPR	1.2596	0.7634	3.7046	0.8381
	Wide & Deep	1.3114	0.7874	2.3088	0.8405
	s^2 Meta	0.9130	0.8148	2.0921	0.8422
	MeLU-1	0.9299	0.8011	2.1475	0.8410
	MeLU-5	0.9235	0.7752	2.1721	0.8422
	MAMO	**0.8894**	**0.8709**	1.8188	0.8384
	InfoRec	0.8922	0.8127	**1.5555**	**0.9150**

official open-source implementation[1] of MeLU.

The performance of InfoRec depends on two factors: 1) the informativeness of the constructed support set; 2) the performance of the predictor, e.g., MeLU, of predicting user preference with a given support set. Ideally, the two factors benefit each other. The mean results are presented in Table 3.3.

Specifically, Table 3.3 presents the evaluation of the baseline methods for four types of recommendations on MovieLens-1M and Bookcrossing datasets. The average value is shown in each cell. The method with the highest performance is highlighted in bold and the second-best method is denoted by an underline. Note that the recommendation of non cold-start scenario represents the performance on the training dataset. The other three cold-start scenarios indicate the test performance. From the results, we have the following findings:

- In non cold-start scenario, PPR demonstrates superior recommendation performance and shows the best MAE and nDCG@3. The recommendation of non cold-start scenario represents the performance on the training dataset. PPR performs the best on the training dataset, but this can be attributed to overfitting. On the other hand, when tested on the three cold-start scenarios, both the PPR and the Wide & Deep methods performed poorly while InfoRec achieves comparable performance on the training dataset, and consistently surpasses other methods in three cold-start test settings.

- Among baselines, meta-learning based methods, i.e., s^2 Meta, MAMO, InfoRec, and MeLU, outperform the other methods in cold scenarios, which validates the advantage of the meta-learning in alleviating the cold-start issue. The reason is that the cold-start setting matches the training setting of meta-learning. In the

[1] https://github.com/hoyeoplee/MeLU.git

training phase, the predictor is required to achieve high prediction accuracy by fine-tuning the deep neural network with a small support set. Because of the fine-tuning process, the meta-learning method can utilize the prior knowledge learned from the training data set and adapt to new users or items with few interactions.

- In the cold-start setting, combining InfoRec consistently improves the performance of the original meta-learning method, i.e., MeLU. The reason is that InfoRec provides a support set containing sufficient information about the user. And the predictor fine-tuned on the support set gets a personalized model which captures the user preference for a large number of items outside the support set.

Impact of support set size (Q2)

We evaluate our method against other baseline item selection methods using MAE and nDCG@3 metrics, considering various support set sizes. For a fair comparison, we need to combine the different methods with one meta-learning method, e.g., MeLU. Otherwise, we cannot conclude whether the performance gains are from the meta-learning method or the item selection methods.

To show that our method makes meaningful item selection, we compare our method with random item selection. The random selection strategy randomly chooses items as a support set. As shown in Fig. 3.3 and Fig. 3.4, InfoRec significantly outperforms the random selection strategy.

To show the superiority of our method, we compare it with existing item selection, i.e., popularity-based method [17], uncertainty-based method [49], and coverage-based method [56]. In the popularity-based method, the system ranks items based on their popularity and recommends the most popular items to the user. The uncertainty-based method aims to recommend items that have diverse ratings among users, i.e., there are users who both like and dislike the item. Coverage-based methods are aimed at recom-

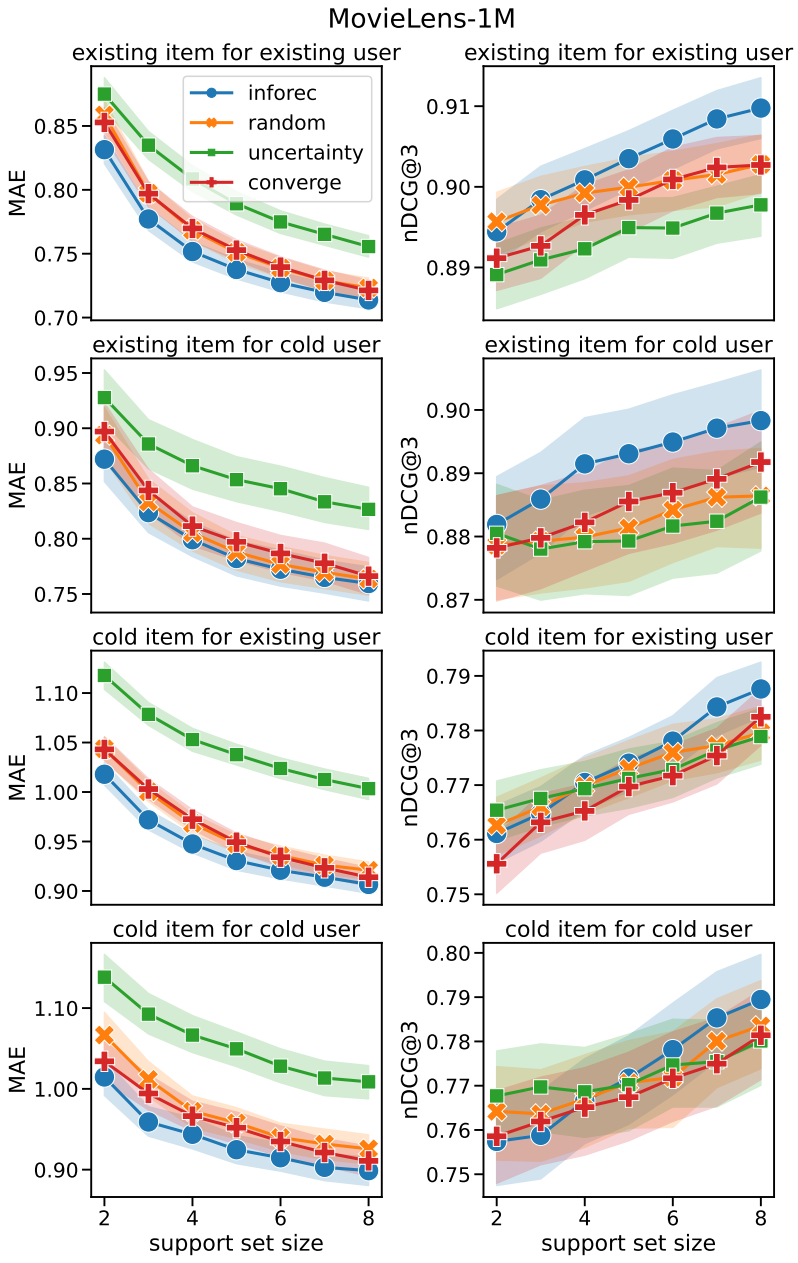


Figure 3.3: MAE and nDCG@3 in MovieLens-1M dataset were achieved after obtaining user preferences for items in the support sets with various sizes.

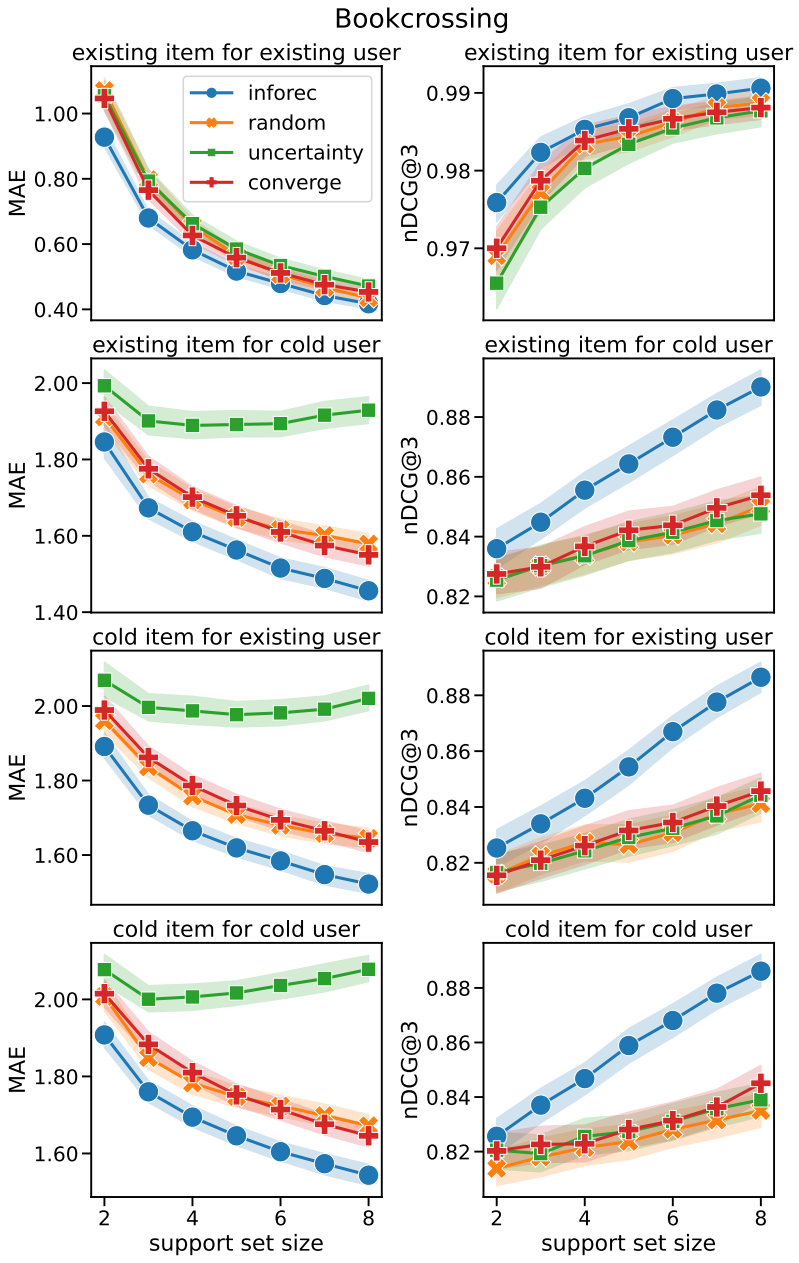


Figure 3.4: MAE and nDCG@3 in Bookcrossing dataset were achieved after obtaining user preferences for items in the support sets with various sizes.

mending a variety of items to users, rather than just the most popular or relevant ones, which works by measuring the diversity of the items recommended to users. As shown in Fig. 3.3 and Fig. 3.4, InfoRec consistently outperforms baseline methods for all datasets in terms of MAE and nDCG@3. Specifically, the uncertainty-based method selects items that have large rating entropy. This method fails when selecting items with fewer ratings and irrelevant to a large number of items. Coverage-based methods select items that increase the diversity of the support set. The diversity of a support set can be quantified by calculating the average Euclidean distance between the feature vectors of the items in the set. However, a support set containing diverse items may not completely capture user preference. Because an item that is very different from many items tends to provide less new information for predicting user preferences. InfoRec achieves superior performance since it selects items that maximize information gain for predicting user preferences of all other items, which helps meta-learning based predictors quickly adapt their model for specific users.

3.6 Experiments on Time-Sensitive Cold-Start Recommendation

3.6.1 Experimental Settings

For time-sensitive cold-start recommendations, a dataset is divided into multiple time periods and each time period contains three months. When evaluating the recommendation performance of a model in a time period, a recurrent module utilizes historical interactions up to $t-1$ period to make predictions for the upcoming period t. And, a meta-learning module leverages a small number of recent interactions from a current period t as a support set. Specifically, to evaluate the performance of a meta-learning

model in a period, the meta-learning model employs five interactions from the first month of the current period t as the support set, while the remaining interactions from the subsequent two months serve as the query set.

Dataset description The experiments are carried out on Netflix[1]. The Netflix dataset is a large collection of movie rating data collected by the online movie rental company Netflix. The dataset contains explicit feedback information as well as a basic user and item attribute information. The detailed information of the Netflix dataset is shown in Table 3.2.

Baseline methods We compared the proposed method with two models: MeLU [33], which have been introduced above, and a time-sensitive meta-learning-based method [45]. As the original paper [45] did not provide a meaningful shorthand name and instead referred to their method as "Ours", we have chosen to give the method a more descriptive name for clarity and brevity: DynamicMeta. DynamicMeta is designed to capture the time-evolving factors from all of a user's historical interactions while also quickly learning time-specific factors based on a small number of current interactions. To achieve this, DynamicMeta uses a module that can quickly adapt to each time period, even when there are only a few interactions available. This module is learned using the MAML framework, which helps it adapt to changes over time.

Evaluation metric To compare with the original results reported by the above baseline methods, we employ the Rooted Mean Square Error (RMSE) as evaluation metrics. RMSE squares the prediction error before summing it up, it tends to castigate the large errors more heavily and it could be calculated as

$$RMSE = \sqrt{\frac{1}{|U'|}\sum_{u \in U'}\frac{1}{|Q_u|}\sum_{v \in Q_u}(r_u^v - \hat{r}_v^u)^2}.$$

[1] www.netflix.com

3.6.2 Dataset Preprocessing

To compare with the state-of-the-art results, we preprocessed the Netflix dataset as the open-source official implementation[1] of DynamicMeta [45]. Specifically, we keep user-item interactions from January 2002 to December 2005. The dataset was split into 16 distinct periods by dividing it over a three-month interval. The users are partitioned into meta-train and meta-test sets using a random allocation with a ratio of approximately 8:2 (700:144). For baselines that use meta-learning modules, the support set is randomly sampled from the interactions of the first month, while the query set consists of interactions from the remaining months during evaluation in a time period. The support set size is fixed to 5 interactions.

3.6.3 Result and Analysis

To answer **Q3**, we provide comprehensive results for each prediction period to illustrate the progression of predictions over time.

Time-sensitive cold start performance comparison

In this section, we construct a support set for time-sensitive cold-start recommendations. For Netflix, the recommendation method queries the user preferences for five items, i.e., constructing a support set, in each time period.

We combine our method with the time-sensitive meta-learning based method [45], DynamicMeta. We found that the official implementations of MeLU and DynamicMeta can achieve better performance with optimized hyperparameter settings. To ensure a fair comparison between our method and the baseline methods, we report the results of both the official and our unofficial implementations, labeled as MeLU^{+} and

[1] https://github.com/ritmininglab/A-Dynamic-Meta-Learning-Model-for-Time-Sensitive-Cold-Start-Recommendations.git

DynamicMeta^{+}. The only difference between our unofficial implementations and the official ones is the use of improved hyperparameter settings to achieve better performance. This approach allows us to compare the effectiveness of the algorithms under the same conditions.

To evaluate the long-term performance of our method, we visualize the RMSE of different methods in different time periods. Note that we only report the RMSE results because of the availability of the published results of this metric. As shown in Fig. 3.5, our method achieves the best performance in all time periods. The reason for the superior performance is that InfoRec chooses items that are relevant to as many items as possible. And, items that are relevant to items in the constructed support set would be recommended to the user in some time period.

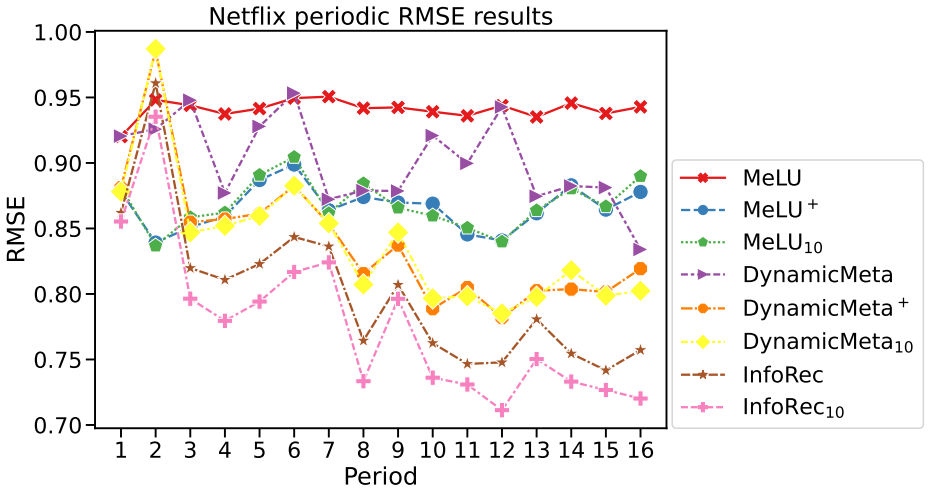


Figure 3.5: Netflix Periodic RMSE Results.

Impact of support set size In this part, we evaluate the long-term performance of our method and the other two meta-learning-based methods, i.e., MeLU and Dynam-

icMeta, with larger support set sizes.

According to Section 3.6.1, the fixed support set size is 5. Ideally, an efficient recommendation method should improve its performance with a larger support set, since the model has more user-specific information. To evaluate the performance of the MeLU, DynamicMeta, and InfoRec methods with larger support set sizes, we compare their performances with a support set size of 10, denoted as MeLU^{+}_{10}, $\text{DynamicMeta}^{+}_{10}$, and InfoRec_{10}. The subscript 10 indicates that these methods are evaluated with a support set size of 10, whereas the methods with a support set size of 5 are denoted without the subscript.

As in the previous part, we evaluate the RMSE of all methods across 16 periods, where each period contains user interactions for a period of three months. We calculate and report the RMSE of all methods in each period. For a period, we fine-tune the models using the support set (with size 5 or 10) sampled from the user interactions in the first month of each period and calculate the RMSE based on the interactions in the following two months.

Fig. 3.5 shows the comparison between MeLU^{+}, DynamicMeta^{+} and InfoRec with their versions that have larger support sets, MeLU^{+}_{10}, $\text{DynamicMeta}^{+}_{10}$, and InfoRec_{10}. The results indicate that adding an additional 5 items to the support set has a negligible impact on the performance of MeLU and DynamicMeta. This could be due to two reasons: Firstly, the models are trained on a fixed support set size of 5, which leads to overfitting of the model to this specific support set size. Secondly, the additional items may not provide any informative features to the model. On the other hand, for InfoRec, the inclusion of new items in the support set leads to further improvement in performance across all time periods. This suggests that InfoRec can benefit from a larger support set for long-term recommendation purposes.

3.7 Summary

In this Chapter, we proposed Information gain driven cold-start Recommendation (InfoRec). Rather than selecting random items as a support set, this Chapter selected informative items that can quickly reduce the uncertainty about the user's preferences. Experiments demonstrated that InfoRec outperforms baseline methods in three types of cold-start settings and a long-term time-sensitive cold-start recommendation setting. We also empirically prove that InfoRec can consistently improve recommendation performance when selecting new items to the support set. However, InfoRec needs to evaluate the information gain from each item in the item set, which is inefficient in a real-time system. A promising direction is to generate a small set of items that have a high probability of including the most informative item, and thus the model selects high-value items with less evaluation.

Chapter 4

Support Set Construction of Meta-Learning for Cold-Start Recommendation by Imitation Learning

4.1 Overview

In the age of the information explosion, recommendation systems have demonstrated their utility in addressing the issue of information overload within the e-commerce domain and have become an integral aspect of our daily experiences [26,65]. Specifically, recommendation systems have the potential to yield mutually beneficial outcomes for both consumers and businesses. From the consumer perspective, these systems facilitate time-efficient, personalized product and service recommendations that are tailored to align with individual preferences and interests. This can enhance consumer satis-

faction and foster brand loyalty, ultimately resulting in increased sales for businesses. Additionally, from the business perspective, recommendation systems can improve sales performance by matching consumers with products or services that they are more likely to purchase, allowing for the identification of consumer trends and preferences that can inform future marketing and product development efforts. Furthermore, these systems can assist businesses in optimizing their pricing strategies by identifying consumer willingness to pay for various products or services.

Recommendation systems can be broadly classified into three main categories: collaborative filter-based, content-based, and hybrid systems. Collaborative filter-based systems employ predictions of user-item ratings based on the ratings of other users with similar preferences, and recommendations are generated based on these predicted ratings. However, such systems rely heavily on a significant amount of user-item interactions and may struggle with new users or items [53]. Content-based systems make recommendations by incorporating additional information such as item descriptions and user profiles into the recommendation process [68]. However, these systems may have the limitation of suggesting the same items to users with similar interests, regardless of their previous ratings. Hybrid systems, which combine both collaborative and content information to improve recommendation quality, are widely used in various applications [6]. However, these systems may not be suitable for recommendations when user-item interaction data are sparse. As such, a critical challenge in the field of recommendation systems is how to efficiently capture users' preferences with a limited amount of user-item interactions, which is commonly referred to as the cold-start problem [74].

In order to quickly and accurately identify the preferences of new users with minimal interactions, several recent approaches have introduced the concept of meta-learning [9,10,13,33], into recommendation systems to address the cold-start problem.

The fundamental idea behind this approach is to leverage prior knowledge acquired from previous recommendation tasks to improve the efficiency of learning a new user's preferences by observing only a few interactions, referred to as the support set. Careful selection of the support set is crucial for providing personalized recommendations. However, previous methods commonly rely on random selection strategies for choosing the support set, which can result in two main issues. Firstly, recommendation systems may suggest irrelevant items to new users, leading to a poor user experience during the support set construction phase [84]. Secondly, a poorly-constructed support set may fail to accurately capture the unique characteristics of new users, thus jeopardizing recommendation quality [42].

This Chapter proposes a meta-learning-based support set selection method for cold-start recommendations, by treating the selection process as a sequence of interactions between users and the recommendation agent and modeling this process using a Markov Decision Process (MDP) framework. By formulating the problem in this way, we leverage Imitation Learning (IL) [21] to learn optimal support set selection strategies. The IL approach involves constructing an expert policy using a training dataset, and then having the recommendation agent imitate the expert's actions. This allows the recommendation agent to actively select an informative support set that enables the recommendation system to quickly and effectively learn the preferences of new users. This Chapter makes several key contributions to the field of recommendation systems. Firstly, we propose the use of Imitation Learning to address the cold-start problem in recommendation systems within the meta-learning framework. Secondly, through extensive experimentation, we demonstrate the effectiveness of our proposed approach in solving the cold-start problem in recommendation systems.

4.2 Preliminary

Imitation Learning (IL) is a methodology for acquiring a behavior policy through the observation and imitation of expert demonstrations, which are typically presented in the form of state-action trajectories. This framework is commonly defined in the context of a Markov Decision Process (MDP) without an explicitly-defined reward function. The objective of IL is for an agent to learn an optimal policy, denoted as $\pi : S \rightarrow A$, by observing and mimicking expert demonstrations, represented as $\{\tau_1, \tau_2, \ldots\}$, where each τ is a sequence of state-action pairs, $\{(s_0, a_0), (s_1, a_1), \ldots, (s_N, a_N)\}$.

A specific approach to IL is Behavioral Cloning (BC) [21], in which the agent policy, denoted as π, is estimated directly from expert demonstrations, denoted as π_{E}. More formally, given demonstrations generated by an expert policy π_{E}, supervised learning is employed to optimize the agent policy π_θ in order to recover the expert policy π_{E} by minimizing the difference between the agent's generated actions and the expert's actions.

$$\pi^* = \arg\min_{\pi_\theta \in \Pi} \mathbb{E}_{s \sim d_{\pi_E}} \left[L\left(\pi_\theta(s), \pi_E(s)\right) \right], \tag{4.1}$$

where Π denotes the set of all possible policies, d_{π_E} is the state distribution encountered by the expert, and loss function L measures the distance between two action vectors in the action space.

4.3 Our Method

We employ the technique of imitation learning to learn the optimal support set selection strategy for recommendation systems. The objective of imitation learning methods is to learn an imitation policy that reproduces the behavior of an expert policy, which is able to select an optimal support set. To accomplish this, we construct expert demon-

strations in Section 4.3.1 and learn the imitation policy in Section 4.3.2.

4.3.1 Expert Datasets Construction of Imitation Learning

In this section, we construct the expert demonstration of a set of pairs of user-item embedding and its corresponding ground truth score, which precisely evaluates the optimality of action.

Definition 3 (Score)**.** *A score quantifies the usefulness of the item for helping the recommendation model identify user preference for all items. We have* $r_i = -1/M \sum \mathcal{L}_\phi(\mathbb{S})$*, where* $\mathcal{L}$ *is the loss function,* ϕ_i *is the encoder fine-tuned on item i,* $\mathbb{Q}$ *is the query set without item i which have M items. Specifically, the loss function is defined as*

$$\mathcal{L}_\phi(\mathbb{S}) = -1/M \sum_{(v,y)\in\mathbb{Q}} |h(\phi(x)) - y|_2^2, \tag{4.2}$$

where $\mathbb{Q} = \mathcal{H} \setminus \mathbb{S}$*, h is the predictor network, and* ϕ *is fine-tuned on support set* $\mathbb{S}$*.*

Intuitively speaking, the score of an item refers to negative mean query loss, which is obtained by supposing the item is a temporary support set while the rest item of the interaction is a query set, updating the encoder ϕ on the temporary support set, and calculating the loss on the query set.

For example, suppose a user with his/her interaction $\{(1,2),(2,1),(3,5)\}$, where the first element is the item and the second is the corresponding rating. In order to get the score of item 1, suppose item 1 is the current temporary support set while item 2 and 3 is the query set. Then update the encoder ϕ on the temporary support set as follows

$$\phi_1 \leftarrow \phi - \alpha \nabla_\phi \mathcal{L}_\phi(\mathbb{S}), \tag{4.3}$$

where $\mathcal{L}$ is the loss function to calculate the difference between the rating and the prediction rating, $\mathbb{S}$ is the temporary support set, i.e. $\mathbb{S} = \{(1,2)\}$. The score r_1 of item 1 is calculated by the following equation

$$r_1 = -\mathcal{L}_{\phi_1}(\mathbb{S}), \tag{4.4}$$

where $\mathbb{Q}$ is the query set, i.e. $\mathbb{Q} = \{(2,1),(3,5)\}$ and M is the length of query set. This is the same for obtaining the score of items 2 and 3.

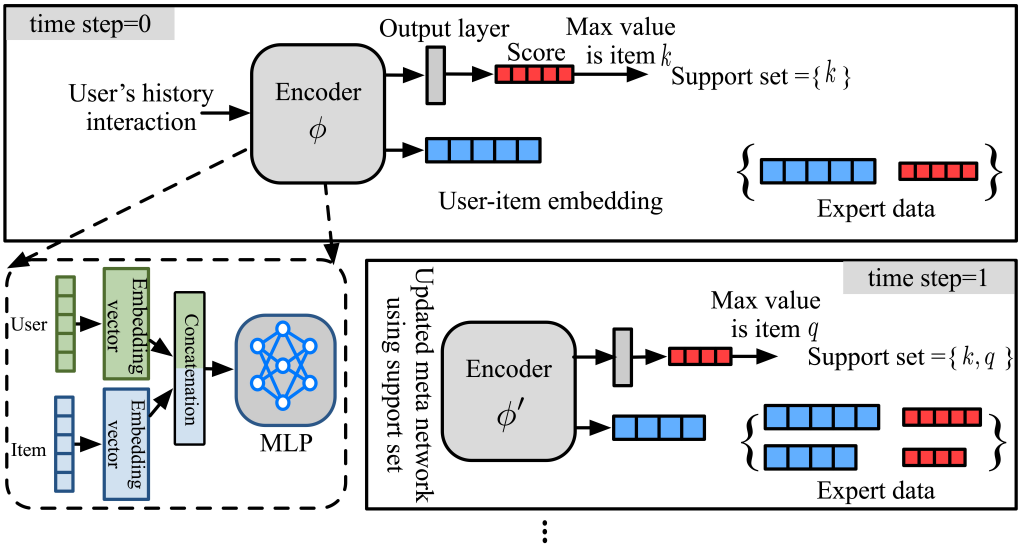


Figure 4.1: Expert datasets construction

We construct the expert dataset $\mathcal{D}_{\text{demo}}$ with the following steps.

- **Step 1: Initialization**. For user u, his/her support set $\mathbb{S} = \emptyset$ and query set $\mathbb{Q} = \mathcal{H} \setminus \mathbb{S}$, where $\setminus$ denotes set difference, at the beginning.
- **Step 2: Calculate ground truth score for an item**. For each item $v_i \in \mathbb{Q}$, we first calculate the user-item embedding with the pair (v_i, u). We fine-tune the

encoder ϕ on the current support set and derive ϕ'. The fine-tuned encoder ϕ' takes the pair (v_i, u) as input and outputs its embedding e_i.

Then, we calculate the score r_i for (v_i, u). We suppose item v_i is the temporary support set $\mathbb{S}^{\text{tem}} = \{(v_i, y_i)\}$, fine-tune ϕ' on $\mathbb{S}^{\text{tem}}$ and derive ϕ^{tem}. With ϕ^{tem}, we calculate the query loss of the rest of the items in $\mathbb{Q}$. We consider the score r_i as the negative mean of the query loss. Intuitively, if item v_i is informative for recognizing the user preference of other items, then the score r_i would be high, otherwise, it will be low.

Finally, we add pair (e_i, r_i) to expert dataset $\mathcal{D}_{\text{demo}}$.

- **Step 3: Update support set**. After calculating all items' scores, we select the item $v^* \in \mathbb{Q}$ with the highest score value and add it to support set $\mathbb{S} = \mathbb{S} \cup \{(v^*, y^*)\}$ while deleting the item from the query set $\mathbb{Q} = \mathbb{Q} \setminus \{(v^*, y^*)\}$. Go back to Step 2 until the size of the support set equals a given hyperparameter N.

4.3.2 Imitation Learning Process

With the expert demonstrations, an imitation policy is trained that can map the user-item embedding to a score, effectively reproducing the expert's policy. This learning problem can be formulated as a supervised learning problem. This approach requires an imitation policy, represented by π_θ, and an objective function, L, that quantifies the similarity between the demonstrated behaviors and the imitation policy.

Imitation policy π_θ is a Multilayer Perceptron (MLP), which takes user-item embedding e as input and outputs prediction score value $\hat{r}$. To calculate the error of the model during the optimization process, we chose $L2$ loss functions which is the squared

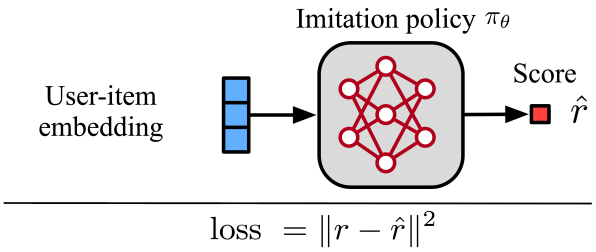


Figure 4.2: Imitation policy learning with the expert demonstrations.

difference between ground truth score r recorded in expert demonstration and prediction score $\hat{r}$, i.e., $L = \|r - \hat{r}\|_2^2$. We find optimal imitation policy π_{θ^*} by minimizing loss L.

4.3.3 Inference Process

With the learned imitation policy π_{θ^*}, we select the support set for a new user.

- **Step 1**: The user's initial support set $\mathbb{S} = \emptyset$.
- **Step 2**: We fine-tune encoder ϕ with support set $\mathbb{S}$ and derive ϕ'. We embed user information and each candidate item with ϕ' and get user-item embedding e.
- **Step 3**: The imitation policy π_{θ^*} takes e as input and predicts the score of each item. We query the rating y of the item v_* with the highest score value and add it to the support set $\mathbb{S} = \mathbb{S} \cup \{(v^*, y^*)\}$ and delete the item from the query set $\mathbb{Q} = \mathbb{Q} \setminus \{(v^*, y^*)\}$. Go back to step 2 until the support set size reaches the given number N.

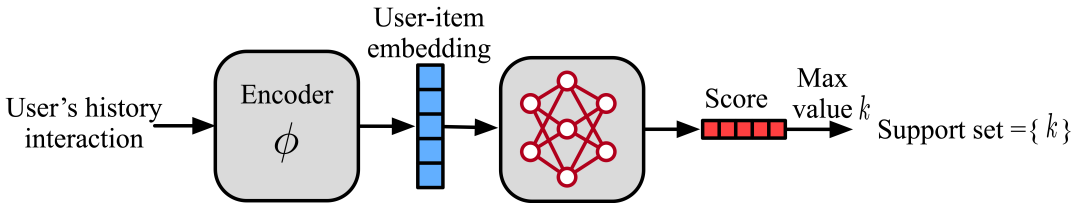


Figure 4.3: Inference process

4.4 Experiments

4.4.1 Experimental Settings

Dataset description. This study conducts experiments MovieLens-1M [83], which is commonly used in the recommendation systems literature. MovieLens-1M contains explicit feedback information as well as user and item attribute information, which provides valuable information for building and evaluating recommendation systems. A summary of this dataset can be found in Table 4.1.

Table 4.1: Summary of the MovieLens-1M dataset.

Dataset	MovieLens-1M
Number of users	6,040
Number of items	3,952
Number of ratings	$1,000,209$
Density	4.19%
User attributes	Gender, Age, Occupation, etc.
Item attributes	Director, Rate, Genre, etc.
Rating Scale	$1 \sim 5, 5$ discrete values

Cold-start scenarios. In order to demonstrate the applicability of our approach in different cold-start scenarios, this study evaluates the recommendation performance in four distinct settings: (a) recommendations for existing users on existing items, (b) recommendations for new users on existing items, c) recommendations for existing users on new items, and d) recommendations for new users on new items. In order to conduct the experiments in these scenarios, this Chapter preprocesses the datasets as follows: this study randomly selects 80 percent of the users as existing users, while the remaining 20 percent as cold users. Additionally, in order to reflect the movie release time, this study considers the top 80 percent of movies as existing items and the remaining 20 percent as cold items, by ordering them by the movie release time.

Baseline methods. In order to establish the validity of our proposed approach, this Chapter employs two types of baselines for comparison: (a) Traditional methods, including PPR [47] and Wide & Deep [6], which are well-established and widely utilized in the field of recommendation systems; (b) Meta-learning based methods, comprising of s^2 Meta [10], MeLU [33], and MAMO [9], which address the cold-start problem through the utilization of meta-learning algorithms.

Evaluation metrics. In this study, the performance of the recommendation approaches is evaluated based on prediction accuracy and ranking correctness. To this end, two performance indicators, Mean Absolute Error (MAE) and Normalized Discounted Cumulative Gain (nDCG), are used. MAE evaluates the accuracy of rating prediction, with lower values indicating better model performance, while nDCG takes into account the preference ordering performance on observed query sets, where higher values indicate better performance. In particular, we use nDCG@3, as proposed in MAMO [9].

4.4.2 Results and Analysis

Table 4.2: Performance comparison of different recommendation methods.

		MovieLens-1M	
Type	Method	MAE	nDCG@3
Recommendation on existing items for existing users	PPR	**0.1820**	**0.9831**
	Wide & Deep	0.9047	0.9117
	s^2 Meta	0.7567	0.8870
	MeLU-1	0.7661	0.8904
	MeLU-5	0.7567	0.8919
	our method	0.6982	0.9060
Recommendation on existing items for cold users	PPR	1.0748	0.8468
	Wide & Deep	1.0694	0.8639
	s^2 Meta	0.8443	0.8401
	MeLU-1	0.7884	0.8810
	MeLU-5	0.7854	0.8812
	our method	**0.7583**	**0.8892**
Recommendation on cold items for existing users	PPR	1.2441	0.7632
	Wide & Deep	1.2655	0.7721
	s^2 Meta	**0.8860**	**0.8323**
	MeLU-1	0.9361	0.7990
	MeLU-5	0.9275	0.8005
	our method	0.9093	0.7944
Recommendation on cold items for cold users	PPR	1.2596	0.7634
	Wide & Deep	1.3114	0.7874
	s^2 Meta	0.9130	**0.8148**
	MeLU-1	0.9299	0.8011
	MeLU-5	0.9235	0.7752
	our method	**0.9042**	0.7942

In order to establish the validity of our proposed method, an analysis of the overall performance of various approaches was conducted, i.e., recommending existing items to cold users, recommending cold items to existing users, and recommending cold items to cold users. Specifically, our experiments are performed on a Linux (Ubuntu 18.04) PC with CPU (Intel(R) i9-9900X, 10 cores, 20 threads, 3.5GHz) and GPU (Nvidia RTX 3090 24G). We employ Python 3.8 (PyTorch framework) as our programming language. The results of the comparison are depicted in Table 4.2, with the highest performance represented in bold font. From the results, we have the following findings: (1) In the scenario of existing items & users, traditional method, PPR displayed superior performance compared to meta-learning based methods on the MovieLens-1M dataset. However, this method exhibited suboptimal performance in the three cold-start scenarios evaluated; (2) In an examination of the baseline methods, it was determined that the meta-learning based methods, specifically s^2 Meta, MAMO, our method, and MeLU, demonstrated superior performance in cold-start scenarios. This observation supports the effectiveness of meta-learning in addressing the issue of the cold-start problem; (3) Evaluation of our proposed method in comparison to MeLU revealed that it consistently exhibited superior performance. This validates the significance of the dynamic support set construction strategy incorporated in our method.

4.5 Summary

In this Chapter, we present an alternative approach for support set selection in recommendation systems by treating the selection process as a sequence of interactions between users and the recommendation agent and modeling it using a Markov Decision Process framework. We leverage Imitation Learning to learn optimal support set selection strategies by constructing an expert policy using a training dataset and having the

recommendation agent imitate the expert's actions. This allows the recommendation agent to actively select an informative support set, which enables the recommendation system to quickly and effectively learn the preferences of new users.

Chapter 5

Integrating Prior Knowledge from Meta-Learning and Large Language Models for Cold-Start Recommendation

5.1 Overview

To tackle the cold-start issue in recommendation systems, researchers have drawn inspiration from few-shot learning and integrated meta-learning, notably Model-Agnostic Meta-Learning (MAML) [13]. MAML's core concept involves acquiring knowledge from past recommendation tasks and leveraging it to expedite the learning of a new user's preferences through minimal interactions. Meta-learning based methods have shown promise in improving cold-start recommendation quality by utilizing prior knowledge from previous tasks. As an illustration, Lee et al. propose a MAML-based rec-

ommendation system which learns a neural network via MAML for quickly adapt its parameter for a new user or a new item with few interaction histories. Moreover, they acquired a universal parameter initialization for the preference estimator as prior knowledge [33]. Dong et al. design two types of memories: i.e., task-specific memory and feature-specific memory which further facilitate the cold-start recommendation [9]. These methods often have embedding process and use categorical features, e.g., user/item id, to model the user/item representation. Therefore, the performance of these methods is limited to the information contained in the user and item features.

Unfortunately, the information pertaining to categorical features is insufficient, as the model would lack awareness of the distinctive characteristics associated with each category. Taking the example of movie attributes, such as the director, only the director's name is provided without additional details. Considering two movies: i) "Spirited Away," directed by Hayao Miyazaki, and ii) "Your Name," directed by Makoto Shinkai. If only the director's name is utilized as input information, the recommendation model would not know the differences in their respective directing styles. Consequently, it becomes necessary to incorporate more descriptive information about these directors, which is widely prevalent on the internet. To enhance the recommendation quality even further, we propose incorporating general knowledge learned from internet data.

Recently, advanced Natural Language Processing (NLP) techniques, such as Large Language Models (LLMs), have demonstrated remarkable achievements in NLP domain because of their formidable ability to model text [7, 8, 64]. Unlike conventional models, which are trained directly on labeled data for particular tasks, LLMs undergo pre-training on extensive volumes of textual data from various sources like articles, books, websites, and publicly accessible written materials [7]. Consequently, LLMs provides a prior knowledge which is general for different down-stream tasks [30, 48]. Moreover, since LLMs learns from a large amount of text data, LLMs have greater

potential in modeling complicated contextual information.

In this Chapter, we propose LLM-MetaRec to enhance cold-start recommendation quality using two types of prior knowledge: recommendation-specific prior knowledge learned from meta-learning, and general prior knowledge from the LLMs. Our approach involves encoding both item and user information using the LLMs, allowing the downstream recommender neural network to leverage LLMs embeddings and effectively incorporate general prior knowledge. Furthermore, we train the recommendation neural network with meta-learning to enable the recommender to acquire recommendation-specific prior knowledge from previous tasks. By combining these two sources of prior knowledge, the recommender can infer user preferences even with limited item ratings. In experiments, we present evidence of LLM-MetaRec's superiority over baseline techniques on two benchmark datasets under three distinct cold-start scenarios, i.e., recommending existing items for cold users, recommending cold items for existing users, and recommending cold items for cold users.

5.2 Our Method: LLM-MetaRec

In this section, we introduce the details of LLM-empowered meta-learning based recommendation method (LLM-MetaRec). Firstly, we present the general user preference estimator. Following this, we elaborate on integrating LLMs into the estimator, aiming to enhance recommendations by leveraging general prior knowledge from the LLMs. Finally, the estimator is trained with meta-learning to attain recommendation-specific prior knowledge.

5.2.1 General User Preference Estimator

The general framework of the user preference estimator, employed in numerous contemporary methods [33], is illustrated in Fig. 5.1. This framework comprises two fundamental components: an embedding module, designed to acquire embeddings of items from their textual information, and a rating prediction module, which is implemented to compute personalized rating scores for item ranking.

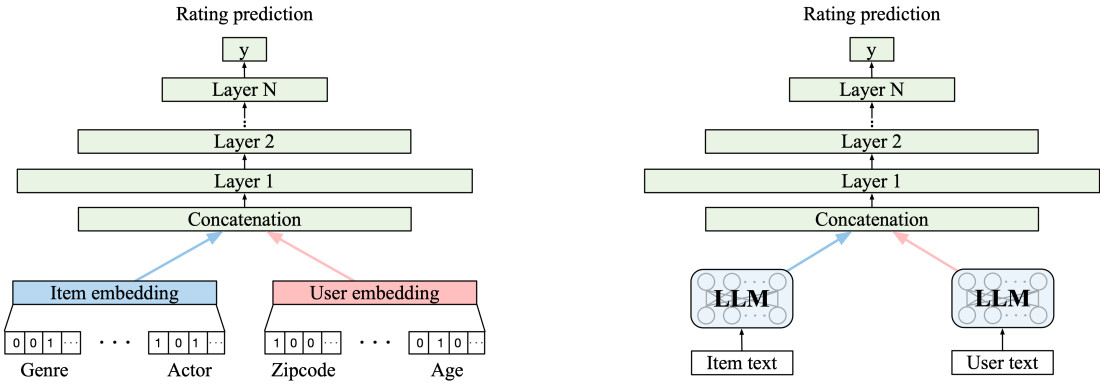


Figure 5.1: A General user preference estimator (left) and LLM-empowered user preference estimator (right).

Embedding module. The embedding module plays a critical role in learning representations of users and items within the context of recommendation systems. Since the attributes associated with users and items often possess distinct categorical values, it is imperative to employ two distinct strategies for learning embeddings of user and item. To illustrate the process for item embedding, let's consider the specific item as an example. When the attribute is categorical, e..g, rating, the corresponding embedding is randomly initialized. Subsequently, these attribute-specific embeddings are concatenated together to generate the initial item embedding. Formally, assuming there are

m attributes for each item, and the procedure for item embedding can be described as follows:

$$e_v^1 = \left[E_1 c_v^1 \oplus E_2 c_v^2 \oplus \ldots, \oplus E_m c_v^m\right], \tag{5.1}$$

where is the symbolic representation of the concatenation operation, and is the symbolic embedding matrix. When the attributes could have multiple possible categories, such as actor, the attributes are initially transformed into a multi-hot representation, where multiple elements are set to '1' to indicate the presence of various categories and '0' otherwise. Following this, a low-dimensional representation of the multi-hot vector is generated using a feed-forward network with a single layer as below:

$$e_v^2 = (W\boldsymbol{e}_i + b), \tag{5.2}$$

where denotes weight matrix, represents the multi-hot representation of and denotes bias vector. Therefore, we concatenate the all the embeddings to get , i.e., . Recommendation module. We concatenate and , and then feed the concatenated embedding to the downstream neural network to generate a lower-dimensional representation. The final layer of the recommendation module is responsible for computing the preference rating , which can be expressed as follows:

$$\hat{r}_u^v = F_{\theta_r}\left([e_u \oplus e_v]\right), \tag{5.3}$$

where the neural network denoted by F is equipped with fully connected components and corresponding parameters .

5.2.2 LLM Empowered User Preference Estimator

In this section, we present the LLM-MetaRec framework, illustrated in Fig. 5.2. The embedding module is instantiated using general prior knowledge from pre-trained large language models (LLMs). This allows us to capture contexts within user attributes, such as age and occupation, and item attributes, including genre, writers, directors, actors, and publication year. To prepare the data for the LLMs input, we start by converting all attributes into a textual format. For example, "The title of the movie is row['title']. This movie is released at row['released']. The genre are row['genre']. The director of this movie is row['director']. The writer of this movie is row['writer']. The actors of this movie are row['actors'], ...". The LLMs takes the user and item textual descriptions and converts them into embeddings. We concatenate the user and item embeddings to form a unified representation. By leveraging this combined embedding, the downstream neural network, which consists of multiple layers, can effectively incorporate prior knowledge from the internet dataset for recommendation purposes. The reason is that the LLM will generate item embeddings that group items with implicit relationships. For instance, if two movies are directed by two directors who share similar styles, these two movies will likely be grouped closely together in the embedding space.

5.2.3 Model Training

Following [33], we employ meta-learning to learn a recommendation specific prior knowledge from a labeled training dataset. With the learned prior knowledge, a model can quickly recognize the preference of a new user with a few interactions with this user. Specifically, we utilize the MAML (Model-Agnostic Meta-Learning) method, a popular meta-learning approach. MAML is designed to learn effective initial neural

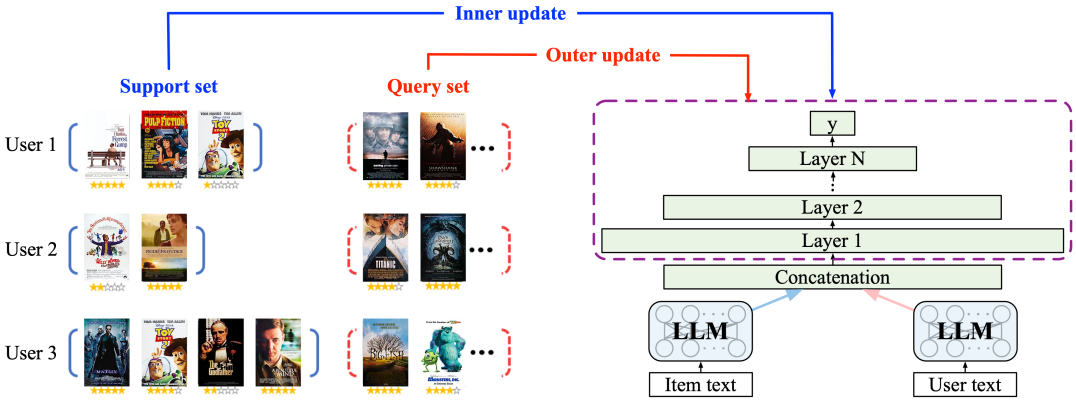


Figure 5.2: The diagram of MAML and LLM based approach for cold-start recommendation.

network weights that can be fine-tuned with only a few labeled data points. With MAML, the recommendation specific prior knowledge is stored in the initial neural network weights. MAML employs an inner- and outer-loop learning strategy. For the training, we divide a labeled training dataset into a support set and a query set. In the inner-loop, the neural network (with its initial weights) is fine-tuned using a small support set containing a few labeled data points. In the outer-loop, the initial weights are updated based on the test loss calculated with the fine-tuned model on a query set. This learning strategy encourages the model to learn effective initial weights, which can then be fine-tuned with a small support set, resulting in minimal loss on the query set—data points without labels during the test phase. Fig. 2 visualizes the learning strategy, which includes an example support set and an example query set. In the context of recommendation tasks, we use an L1-norm loss for prediction since the model is tasked with predicting user ratings (ranging from 1 to 5) for recommended items. The parameters of the LLM are frozen to prevent forgetting prior knowledge

gained from internet datasets when exposed to learning signals from a specific task.

5.3 Experiments

5.3.1 Experimental Settings

The proposed method undergoes validation across four distinct scenarios involving both non-cold and cold settings: a) recommending warm item to warm user; b) recommending warm item to new user; c) recommending new item to warm user; and d) recommending new item to new user.

Dataset description. This Chapter conducts experiments on two distinct datasets: MovieLens-1M and BookCrossing. Both datasets are well-established sources in the field of recommendation systems research, encompassing explicit feedback data, as well as fundamental user and item attribute information.

Baseline methods. To validate the effectiveness of LLM-MetaRec, we have selected two distinct categories of baseline approaches for comparison purposes. (1) Traditional recommendation methods: Specifically, we include the widely recognized and extensively utilized PPR [47] and Wide & Deep [6]. These conventional methods have been extensively employed in the field of recommendation systems and serve as established benchmarks for performance evaluation. (2) Meta-learning based recommendation methods: This category encompasses innovative approaches such as s^2 Meta [10], MeLU [33], and MAMO [9], which leverage meta-learning algorithms to address the intricate cold-start problem.

Evaluation metrics. The evaluation of recommendation methods hinges upon two key performance metrics: prediction accuracy and ranking correctness. To quantify these aspects rigorously, we adopt two metric. Mean Absolute Error (MAE) evaluates

the accuracy of rating prediction, and a lower MAE value indicates that the model's predictions are closer to the actual ratings provided by users, signifying better overall performance in predicting user preferences. Normalized Discounted Cumulative Gain @N (nDCG@N) evaluates the preference ordering performance on observed query sets. In simple terms, it assesses how well the model ranks and prioritizes the recommended items based on user preferences. A higher nDCG@N value indicates better performance in suggesting items that align with users' interests, leading to more relevant and satisfying recommendations. In this Chapter, we utilize the nDCG with N=3 as in MAMO [9].

5.3.2 Dataset Preprocessing

We conduct preprocessing on the MovieLens-1M and BookCrossing datasets in the subsequent manner:

MovieLens-1M: This dataset exhibits a well-balanced distribution of interactions, i.e., the dataset have an average number of interactions. To conduct experiments, we adopted the methodology described in [33]. Specifically, we randomly selected 80% of the users from the dataset to serve as training users, representing existing users with prior interactions. The remaining 20% of users were designated as test users, representing cold users with limited or no interaction history. To ensure rigorous evaluation, A 5-fold cross-validation is performed on the partitioned dataset. Additionally, in order to create a clear demarcation between existing and cold items, we ordered the movies based on their release time and identified the top 80% as existing items, while the bottom 20% were categorized as cold items.

BookCrossing: The presence of evidently misleading values is evident in this dataset. For instance, the user age field exhibited unrealistic values, ranging from 0

to 237, which clearly don't align with real-world scenarios. To address this issue, we took a practical approach and restricted the user age range to a more reasonable span of 5 to 90 years. Following the guidelines presented in [9], we partitioned the dataset for training and testing purposes. Specifically, the warm/cold user split is 9/1, representing cold users who have limited or no previous interactions in the system. This division allowed us to evaluate the performance of our approach effectively. To assess the robustness and generalization capabilities of our method, we employed 10-fold cross-validation on the refined dataset. The 10-fold cross-validation extract 10 subsets from the original dataset and performs training and test on the 10 subsets. We average the results over the results from the 10 subsets. Regarding the items, we categorized them into warm and cold items based on their popularity. Items with five or more ratings were considered warm items, indicating that they had received a substantial level of user engagement. On the other hand, items with fewer than five ratings were categorized as cold items, signifying their limited exposure in the system.

5.3.3 Result and Analysis

In this experiment, we comprehensively evaluate the performance of all methods on two widely used datasets across four distinct recommendation settings. These settings include recommending existing items for existing users, recommending existing items for cold users, recommending cold items for existing users, and recommending cold items for cold users. Specifically, our experiments are performed on a Linux (Ubuntu 18.04) PC with CPU (Intel(R) i9-9900X, 10 cores, 20 threads, 3.5GHz) and GPU (Nvidia RTX 3090 24G). We employ Python 3.8 (PyTorch framework) as our programming language. Table 5.1 shows the test results of of all the baseline methods on the 2 datasets under various cold-start setting. We emphasize the best results

with bold fonts and the second-best results with underlining. This analysis provides a comprehensive comparison of the various approaches, allowing us to gain insights into their effectiveness in addressing different cold-start scenarios. From Table 5.1, we have following observations:

- In the non-cold-start scenario, the method named PPR achieves the best results, attaining top-notch scores in MAE and nDCG@3. Since this is the non-cold start scenario, the result reflects that PPR may overfit to the warm users and items. In contrast, PPR and Wide & Deep methods exhibited poor performance under the other three cold-start scenarios, indicating their unsuitability for cold-start recommendations. On the other hand, LLM-MetaRec method consistently outperforms other baseline methods in all three cold-start settings while achieving comparable performance levels. This indicates that LLM-MetaRec effectively addresses the cold-start problem and provides promising results even in situations with limited interaction data for new users/items. The findings highlight the significance of incorporating meta-learning and leveraging general knowledge from Large Language Models to enhance recommendation quality, particularly in challenging cold-start scenarios.

- Meta-learning based methods, namely Meta, MAMO, LLM-MetaRec, and MeLU, consistently demonstrate superior performance in cold scenarios compared to other approaches. The good performance results in the learning strategy of meta-learning. Specifically, in the cold-start recommendation, the model is allowed to access to a few interactions of a user, and meta-learning force the model to learn the user preference from a small support set during training phase. The alignment between the test setting and training setting significantly improve the performance of meta-learning based methods.

Table 5.1: Comparison of the performance of various recommendation methods.

		MovieLens-1M		BookCrossing	
Type	Method	MAE	nDCG@3	MAE	nDCG@3
Recommendation on existing items for existing users	PPR	**0.1820**	**0.9831**	3.8092	0.8494
	Wide & Deep	0.9047	0.9117	1.6206	0.9172
	s^2 Meta	0.7567	0.8870	1.5147	0.8781
	MeLU-1	0.7661	0.8904	**0.7799**	**0.9572**
	MeLU-5	0.7567	0.8919	0.7955	0.9552
	MAMO	0.8725	0.8866	1.4879	0.8879
	LLM-MetaRec	0.7362	0.8814	1.2535	0.8638
Recommendation on existing items for cold users	PPR	1.0748	0.8468	3.8430	0.8434
	Wide & Deep	1.0694	0.8639	2.0457	0.8515
	s^2 Meta	0.8443	0.8401	1.8738	0.8490
	MeLU-1	0.7884	0.8810	1.8701	0.8527
	MeLU-5	0.7854	0.8812	1.8767	0.8532
	MAMO	0.8967	0.8799	1.6217	0.8565
	LLM-MetaRec	**0.7548**	**0.8823**	**1.2394**	**0.8602**
Recommendation on cold items for existing users	PPR	1.2441	0.7632	3.6821	0.8367
	Wide & Deep	1.2655	0.7721	2.2648	0.8437
	s^2 Meta	**0.8860**	**0.8323**	1.9767	0.8463
	MeLU-1	0.9361	0.7990	2.1047	0.8441
	MeLU-5	0.9275	0.8005	2.1236	0.8440
	MAMO	0.9306	0.8315	1.7379	0.8402
	LLM-MetaRec	0.8901	0.8248	**1.2584**	**0.8571**
Recommendation on cold items for cold users	PPR	1.2596	0.7634	3.7046	0.8381
	Wide & Deep	1.3114	0.7874	2.3088	0.8405
	s^2 Meta	0.9130	0.8148	2.0921	0.8422
	MeLU-1	0.9299	0.8011	2.1475	0.8410
	MeLU-5	0.9235	0.7752	2.1721	0.8422
	MAMO	0.8894	**0.8709**	1.8188	0.8384
	LLM-MetaRec	**0.8832**	0.8311	**1.2703**	**0.8511**

- In the cold-start setting, the combination of LLM-MetaRec consistently enhances the cold-start recommendation performance of the vanilla meta-learning based recommendation method, MeLU. This improvement is attributed to LLM-MetaRec's ability to leverage the general prior knowledge provided by LLMs, which offer a wealth of pre-trained information derived from extensive textual data. By effectively incorporating this knowledge, LLM-MetaRec overcomes the limitations of MeLU and achieves superior results in various cold-start scenarios.

5.4 Summary

In this Chapter, LLM-MetaRec significantly improve the cold-start recommendation performance of a meta-learning based recommendation based method. By leveraging both recommendation-specific and general prior knowledge, we bridge the gap between new users/items and personalized recommendations, thereby improving the overall user experience in various online application domains. We demonstrate the performance of LLM-MetaRec with two datasets, i.e., MoveiLens-1M and Bookcrossing. The experiment results show that LLM-MetaRec outperforms baseline methods in various cold-start settings.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, we alleviate the cold-start problem in the recommendation system. Our approach is built upon meta-learning based methods for cold-start recommendation. Specifically, we focus on enhancing the informativeness of the support set, thereby significantly improving recommendation effectiveness without increasing the size of the support set. To enhance the informativeness of the support set, we employ two methods: i) actively selecting informative items to construct the support set (Chapter 3 and 4), and ii) increasing the informativeness of items in the support set (Chapter 5).

Chapter 3 introduces Information Gain driven Cold-Start Recommendation (InfoRec). InfoRec actively selects items to query users based on their informativeness, measured by the reduction of uncertainty regarding user preferences for unseen items. Theoretically, we prove the optimality the item selection strategy based on submodular theory. Empirically, we prove that InfoRec improves both short-term and long-term cold-start recommendation performance with the limited size of the support set.

Since it is impractical to iterate through millions of items for information gain cal-

culation at each recommendation step, in Chapter 4, we learn a neural item selection policy with imitation learning. Specifically, we formulate the sequential recommendation task as a Markov Decision Process (MDP). Leveraging this formulation, we develop an imitation learning framework where a DNN mimics the optimal item selection policy from Chapter 3, facilitating informative item sampling through a single feedforward process.

Chapter 5 extends our approach beyond the active selection of informative items by incorporating additional information from the internet. We leverage a large language model pre-trained on internet data, offering a wealth of general knowledge. By embedding item and user descriptions using both the language model and the DNN trained for recommendation tasks, our downstream predictor can efficiently develop strategies based on general knowledge from the language model and specific insights from the recommendation task. Therefore, we further improve the cold-start recommendation performance.

6.2 Future Work

We classify future work into two directions: i) recommending items to a new user without interaction history, i.e., zero-shot recommendation; and ii) recommending items with attributes unknown to the LLM.

For the first direction, zero-shot recommendation means that we need to recommend informative items to users who have no interaction with the system before. However, these users provide no information for the recommendation models or the informative item selection policy. Therefore, we will conduct research on constructing a common item set that contains diverse items and is capable of quickly capturing user preferences. The challenge is to ensure that the common item set will help the recom-

mendation models to capture the preferences of users with different features, such as gender, age, and job, even though without interaction data with these users.

In the second direction, as information is generated rapidly, the LLM would always be trained on outdated information. For example, new directors release new movies every day, and the LLM would not have information about these directors. We need an automated mechanism that can find the most relevant information on the internet based on specific items. This information should not only be related to the item but also help us establish connections between these items and previous items, i.e., items with a significant amount of interaction data with users.

Bibliography

[1] A. Adadi. A survey on data-efficient algorithms in big data era. *Journal of Big Data*, 8(1):24, 2021.

[2] M. Aleksandrova, A. Brun, A. Boyer, and O. Chertov. Identifying representative users in matrix factorization-based recommender systems: application to solving the content-less new item cold-start problem. *Journal of Intelligent Information Systems*, 48(2):pp 365–397, 2017.

[3] J. P. Bharadiya. Machine learning and ai in business intelligence: Trends and opportunities. *International Journal of Computer (IJC)*, 48(1):123–134, 2023.

[4] Y. Bi, L. Song, M. Yao, Z. Wu, J. Wang, and J. Xiao. A heterogeneous information network based cross domain insurance recommendation system for cold start users. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 2211–2220, 2020.

[5] D. Billsus and M. J. Pazzani. Learning collaborative information filters. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 46–54, 1998.

[6] H. Cheng, L. Koc, J. Harmsen, and et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 7–10, 2016.

[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. Association for Computational Linguistics, 2019.

[8] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H. Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*, pages 13042–13054, 2019.

[9] M. Dong, F. Yuan, L. Yao, X. Xu, and L. Zhu. MAMO: memory-augmented meta-optimization for cold-start recommendation. In *Proceedings of the Conference on Knowledge Discovery & Data Mining*, pages 688–697, 2020.

[10] Z. Du, X. Wang, H. Yang, J. Zhou, and J. Tang. Sequential scenario-specific meta learner for online recommendation. In *Proceedings of the International Conference on Knowledge Discovery & Data Mining*, pages 2895–2904, 2019.

[11] G. K. Dziugaite and D. M. Roy. Neural network matrix factorization. *CoRR*, abs/1511.06443, 2015.

[12] A. Felfernig, S. P. Erdeniz, C. Uran, S. Reiterer, M. Atas, T. N. T. Tran, P. Azzoni, C. Király, and K. Dolui. An overview of recommender systems in the internet of things. *Journal of Intelligent Information Systems*, 52(2):pp 285–309, 2019.

[13] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1126–1135, 2017.

[14] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *ICDM 2010, The 10th IEEE International Conference on Data Mining*, pages 176–185, 2010.

[15] C. Gao, Y. Zheng, N. Li, Y. Li, Y. Qin, J. Piao, Y. Quan, J. Chang, D. Jin, X. He, and Y. Li. Graph neural networks for recommender systems: Challenges, methods, and directions. *CoRR*, abs/2109.12843, 2021.

[16] L. Gao, H. Yang, J. Wu, C. Zhou, W. Lu, and Y. Hu. Recommendation with multi-source heterogeneous information. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 3378–3384, 2018.

[17] N. Golbandi, Y. Koren, and R. Lempel. Adaptive bootstrapping of recommender systems using decision trees. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining*, pages 595–604, 2011.

[18] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1321–1330, 2017.

[19] M. A. Hameed, R. Sirandas, and O. A. Jadaan. Information gain clustering through prototype-embedded genetic k-mean algorithm (IGCPGKA): a novel heuristic approach for personalisation of cold start problem. In *The Second International Conference on Computational Science*, pages 390–395, 2012.

[20] C. Hansen, C. Hansen, J. G. Simonsen, S. Alstrup, and C. Lioma. Content-aware neural hashing for cold-start recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 971–980, 2020.

[21] F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19, 2016.

[22] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182, 2017.

[23] X. He, H. Zhang, M. Kan, and T. Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 549–558. ACM, 2016.

[24] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey. Meta-learning in neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(9):5149–5169, 2022.

[25] G. Hu, Y. Zhang, and Q. Yang. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 667–676, 2018.

[26] G. Hu, Y. Zhang, and Q. Yang. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 667–676, 2018.

[27] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 263–272. IEEE Computer Society, 2008.

[28] M. Imamov and N. Semenikhina. The impact of the digital revolution on the global economy. *Linguistics and Culture Review*, 5(S4):968–987, 2021.

[29] X. Jiang. Digital economy in the post-pandemic era. *Journal of Chinese Economic and Business Studies*, 18(4):333–339, 2020.

[30] Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751. ACL, 2014.

[31] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[32] Y. LeCun, Y. Bengio, and G. E. Hinton. Deep learning. *Nat.*, 521(7553):436–444, 2015.

[33] H. Lee, J. Im, S. Jang, H. Cho, and S. Chung. Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the International Conference on Knowledge Discovery & Data Mining*, pages 1073–1082, 2019.

[34] J. Li, K. Lu, Z. Huang, and H. T. Shen. On both cold-start and long-tail recommendation with social data. *IEEE Trans. Knowl. Data Eng.*, 33(1):194–208, 2021.

[35] K. Li, D. J. Kim, K. R. Lang, R. J. Kauffman, and M. Naldi. How should we understand the digital economy in asia? critical assessment and research agenda. *Electronic commerce research and applications*, 44:101004, 2020.

[36] P. Li and A. Tuzhilin. DDTCDR: deep dual transfer cross domain recommendation. In *The Thirteenth ACM International Conference on Web Search and Data Mining*, pages 331–339, 2020.

[37] V. Litvinenko. Digital economy as a factor in the technological development of the mineral sector. *Natural Resources Research*, 29(3):1521–1541, 2020.

[38] C. Liu, C. Zhou, J. Wu, Y. Hu, and L. Guo. Social recommendation with an essential preference space. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 346–353, 2018.

[39] H. Liu, Y. Zhu, T. Zang, J. Yu, and H. Cai. Jointly modeling individual student behaviors and social influence for prediction tasks. In *The 29th ACM International Conference on Information and Knowledge Management*, pages 865–874, 2020.

[40] S. Liu. Enhancing graph neural networks for recommender systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, page 2484, 2020.

[41] S. Luo, N. Yimamu, Y. Li, H. Wu, M. Irfan, and Y. Hao. Digitalization and sustainable development: How could digital economy development improve green

innovation in china? *Business Strategy and the Environment*, 32(4):1847–1871, 2023.

[42] C. Ma, P. Kang, and X. Liu. Hierarchical gating networks for sequential recommendation. In *Proceedings of the International Conference on Knowledge Discovery & Data Mining*, pages 825–833, 2019.

[43] C. Ma, S. Tschiatschek, K. Palla, J. M. Hernández-Lobato, S. Nowozin, and C. Zhang. EDDI: efficient dynamic discovery of high-value information with partial VAE. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 4234–4243, 2019.

[44] A. Merialdo. Improving collaborative filtering for new-users by smart object selection. In *Proceedings of International Conference on Media Features*, 2001.

[45] K. P. Neupane, E. Zheng, Y. Kong, and Q. Yu. A dynamic meta-learning model for time-sensitive cold-start recommendations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7868–7876, 2022.

[46] D. K. Panda and S. Ray. Approaches and algorithms to mitigate cold start problems in recommender systems: a systematic literature review. *Journal of Intelligent Information Systems*, 59(2):pp 341–366, 2022.

[47] S. Park and W. Chu. Pairwise preference regression for cold-start recommendation. In *Proceedings of the 2009 ACM Conference on Recommender Systems*, pages 21–28, 2009.

[48] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang. Pre-trained models for natural language processing: A survey. *CoRR*, abs/2003.08271, 2020.

[49] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl. Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*, pages 127–134, 2002.

[50] S. Roy and S. C. Guntuku. Latent factor representations for cold-start video recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 99–106, 2016.

[51] N. Rubens and M. Sugiyama. Influence-based collaborative active learning. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, pages 145–148, 2007.

[52] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 1257–1264. Curran Associates, Inc., 2007.

[53] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference*, pages 285–295, 2001.

[54] J. Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: The meta-meta-. hook*. PhD thesis, Technical University of Munich, Germany, 1987.

[55] S. Sedhain, S. Sanner, D. Braziunas, L. Xie, and J. Christensen. Social collaborative filtering for cold-start recommendations. In *Eighth ACM Conference on Recommender Systems*, pages 345–348, 2014.

[56] C. Sha, X. Wu, and J. Niu. A framework for recommending relevant and diverse items. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, volume 16, pages 3868–3874, 2016.

[57] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 27(3):379–423, 1948.

[58] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu. Heterogeneous information network embedding for recommendation. *IEEE Trans. Knowl. Data Eng.*, 31(2):357–370, 2019.

[59] T. J. Sturgeon. Upgrading strategies for the digital economy. *Global strategy journal*, 11(1):34–57, 2021.

[60] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009, 2009.

[61] F. O. Usman, N. L. Eyo-Udo, E. A. Etukudoh, B. Odonkor, C. V. Ibeh, and A. Adegbola. A critical review of ai-driven strategies for entrepreneurial success. *International Journal of Management & Entrepreneurship Research*, 6(1):200–215, 2024.

[62] A. van den Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems*, pages 2643–2651, 2013.

[63] M. Vartak, A. Thiagarajan, C. Miranda, J. Bratman, and H. Larochelle. A meta-learning perspective on cold-start recommendations for items. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, pages 6904–6914, 2017.

[64] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 5998–6008, 2017.

[65] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, pages 3630–3638, 2016.

[66] M. Volkovs, G. W. Yu, and T. Poutanen. Dropoutnet: Addressing cold start in recommender systems. In *Advances in Neural Information Processing Systems*, pages 4957–4966, 2017.

[67] C. Wang, Y. Zhu, H. Liu, W. Ma, T. Zang, and J. Yu. Enhancing user interest modeling with knowledge-enriched itemsets for sequential recommendation. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management*, pages 1889–1898, 2021.

[68] C. Wang, Y. Zhu, H. Liu, T. Zang, J. Yu, and F. Tang. Deep meta-learning in recommendation systems: A survey. 2022. Preprint at `https://arxiv.org/abs/2206.04415`.

[69] H. Wang, N. Wang, and D. Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 1235–1244. ACM, 2015.

[70] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 417–426, 2018.

[71] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, and Z. Wang. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 968–977, 2019.

[72] X. Wang, X. He, Y. Cao, M. Liu, and T. Chua. KGAT: knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 950–958, 2019.

[73] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu. Heterogeneous graph attention network. In *The World Wide Web Conference*, pages 2022–2032, 2019.

[74] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, 69:pp 29–39, 2017.

[75] L. Wu, X. He, X. Wang, K. Zhang, and M. Wang. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Trans. Knowl. Data Eng.*, 35(5):4425–4445, 2023.

[76] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T. Chua. Attentional factorization machines: Learning the weight of feature interactions via attention networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 3119–3125, 2017.

[77] H. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen. Deep matrix factorization models for recommender systems. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 3203–3209, 2017.

[78] S. Zhang, L. Yao, A. Sun, and Y. Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.*, 52(1):5:1–5:38, 2019.

[79] C. Zhao, C. Li, R. Xiao, H. Deng, and A. Sun. CATN: cross-domain recommendation for cold-start users via aspect transfer network. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 229–238, 2020.

[80] J. Zheng, Q. Ma, H. Gu, and Z. Zheng. Multi-view denoising graph auto-encoders on heterogeneous information networks for cold-start recommendation. In *The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2338–2348, 2021.

[81] Y. Zhu, J. Lin, S. He, B. Wang, Z. Guan, H. Liu, and D. Cai. Addressing the item cold-start problem by attribute-driven active learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 631–644, 2020.

[82] Z. Zhu, S. Sefati, P. Saadatpanah, and J. Caverlee. Recommendation for new users and new items via randomized training and mixture-of-experts transformation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 1121–1130, 2020.

[83] C. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32, 2005.

[84] L. Zou, L. Xia, Y. Gu, X. Zhao, W. Liu, J. X. Huang, and D. Yin. Neural interactive collaborative filtering. In *Proceedings of the International conference on research and development in Information Retrieval*, pages 749–758, 2020.

Published Papers

1 **Yu Li** and Tetsuya Furukawa. Information gain based dynamic support set construction for cold-start recommendation. *Journal of Intelligent Information Systems*, 61(3), pp.717–737, 2023.

2 **Yu Li**. Support Set Construction of Meta-Learning for Cold-Start Recommendation by Imitation Learning. *The Annual Report of Economic Science, Kyushu Association of Economic Science*, No.61, pp.131-138, 2023.

3 **Yu Li**, Yixiao Liu and Tetsuya Furukawa. Integrating Prior Knowledge from Meta-Learning and Large Language Models for Cold-Start Recommendation. In *Proceedings of international exchange and innovation conference on engineering & sciences*, pp.327-333, 2023.