# Continual Collaborative Filtering Through Gradient Alignment

JAIME HIEU DO, Singapore Management University, Singapore
HADY W. LAUW, Singapore Management University, Singapore

A recommender system operates in a dynamic environment where new items emerge and new users join the system, resulting in ever-growing user-item interactions over time. Existing works either assume a model trained offline on a static dataset (requiring periodic re-training with ever larger datasets); or an online learning setup that favors recency over history. As privacy-aware users could hide their histories, the loss of older information means that periodic retraining may not always be feasible, while online learning may lose sight of users' long-term preferences. In this work, we adopt a continual learning perspective to collaborative filtering, by compartmentalizing users and items over time into a notion of tasks. Of particular concern is to mitigate catastrophic forgetting that occurs when the model would reduce performance for older users and items in prior tasks even as it tries to fit the newer users and items in the current task. To alleviate this, we propose a method that leverages gradient alignment to deliver a model that is more compatible across tasks and maximizes user agreement for better user representations to improve long-term recommendations.

CCS Concepts: • **Information systems → Recommender systems**; • **Computing methodologies → Lifelong machine learning**.

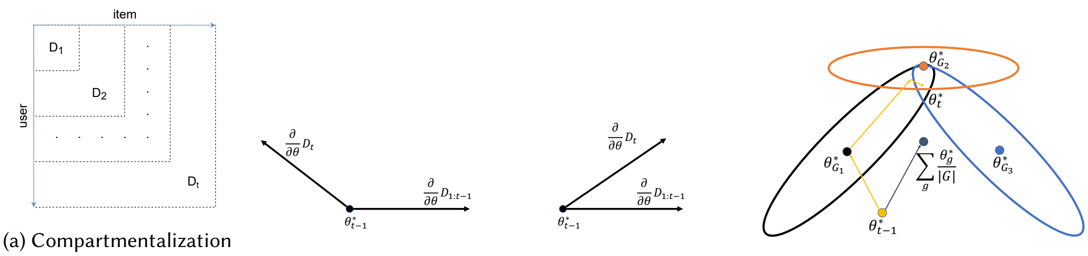
Additional Key Words and Phrases: continual learning, recommendation systems, collaborative filtering, gradient alignment

**ACM Reference Format:**
Jaime Hieu Do and Hady W. Lauw. 2023. Continual Collaborative Filtering Through Gradient Alignment. In *Seventeenth ACM Conference on Recommender Systems (RecSys '23), September 18–22, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3604915.3610648

## 1 INTRODUCTION

Collaborative filtering leverages users' interactions with items as a basis for recommendation. As new items emerge over time and new users appear, there are two common yet inadequate responses: retraining and online learning. The former is to re-train with increasingly larger datasets, a computationally expensive proposition. This is complicated by heightening privacy concerns, as more users hide their older histories, erasing data that could have been used for periodic retraining. The latter, online learning [12, 25], focuses on keeping the recommender systems current by favoring newer users and items. However, it is still inadequate for complex real-world recommendations, where ($i$) the interest of customers does not shift entirely, e.g., users prefer newer model of the same phone brand; ($ii$) in many domains items that emerge earlier in the timeline do not lose their relevance, e.g., a library recommends classics alongside new publications, a streaming service mixes up older movies and songs along with recent ones; ($iii$) shopping habits vary seasonally, e.g., this year's winter shopping behavior is more similar to last year's winter than to this year's summer. Historical user interests and older, yet still relevant items are crucial for modeling long-term preferences.

Given such dynamics, our objective is to capture the preference shift of existing users by modeling newly arising customers and products, while preserving the interests of previously learned users; then refine to a unified recommender system that distills both old and new preferences. These desiderata seem contradictory since it is hard for a fixed-capacity model to maintain perfect recall of learned tasks while absorbing a huge amount of upcoming information.




(a) Compartmentalization of rating data into tasks (b) Interference among tasks (c) Transfer between tasks (d) UACF desired learning path

Fig. 1. If two gradients give same loss for $D_t$, we prefer the gradient that leverages transfer between tasks (Fig. 1c). Figure 1d illustrates desired learning path of user alignment algorithm. Ovals are low-loss regions for individual groups. Normally, model updates towards average gradient over all groups of users, and finally ends up at the center of the triangle. Our algorithm learns group by group to find the optimal path to the intersection.

This reflects the trade-off between *stability* (to retain past preferences) and *plasticity* (to rapidly learn new examples). Recommendation systems need to balance these needs without growing capacity proportionally with streaming data. This, in a nutshell, is the crux of this paper: *designing a continual learning framework for collaborative filtering, investigating and analyzing methods for balancing learning ability and retention in continual collaborative filtering setting.*

## 2 CONTINUAL LEARNING FOR COLLABORATIVE FILTERING

To better assess the *stability-plasticity dilemma*, we propose a Continual Learning framework and evaluation protocol for recommender systems with clear definitions of tasks, goals, and metrics.

**Tasks and Goals** Over time, new users keep appearing and existing users continue to interact and explore new items. We could treat each group of users as a new "task". However, old items might still attract users even as new items are offered to users. Thus, we propose a novel continual learning setting for recommender systems, where interactions keep expanding on both user and item sides. We define a task as compartmentalization of users and items, as illustrated in Figure 1a. Suppose we order the dimensions of $\mathbf{R}$, i.e., users and items, according to the time they first emerge. In that case, a task corresponds to a block of users and items that appear over a specified period of time[1]. Within a new task, we see ratings corresponding to earlier users on new items, new users on earlier items, new users on new items, and potentially new ratings by existing users on old items. While observing the data periodically, the objective is to learn a generalized recommendation model that not only could capture recent preferences but also retain previous users' interest.

**Notation:** We index users by $u \in \{1, \ldots, U\}$ and items by $i \in \{1, \ldots, I\}$, which form the user-item interaction matrix $\mathbf{R} \in \mathbb{N}^{U \times I}$. We denote $T$ as the number of tasks in continual learning setting, which in theory is infinite. We refer to the set of observations within a task $t$ as $\mathcal{D}_t \in \mathbf{R}^{U_t \times I_t}$, where $U_t, I_t$ are the number of users and items in task $t$, respectively. $U_{t+1} > U_t, I_{t+1} > I_t$, which reflects ever-growing systems acquiring more users and items. We use $\ell$ as model-agnostic loss function for general collaborative filtering algorithm.

**"Multi-Task" Collaborative Filtering** In a typical offline batch learning manner, the objective is to learn the set of user and item parameters $\theta$ that minimize the loss over i.i.d. distribution samples within dataset $\mathcal{D} \in \mathbf{R}^{U \times I}$:

$$\theta^* = \arg\min_\theta \mathbb{E}_{r_{ui} \sim \mathcal{D}} \left[ \ell \left( r_{ui} \mid \theta \right) \right] \tag{1}$$

[1]Note that there is flexibility in how a task is defined, depending on specific application scenarios. For instance, it would also be possible to define a task in terms of only new items, with the assumption that all users are present throughout.

The above assumes a task-agnostic setting with an arbitrary loss function. Given multiple tasks, the learning objective turns into minimizing total loss for every observed user and item, over all the tasks:

$$\theta^* = \arg\min_{\theta} \sum_{t=1}^{T} \mathbb{E}_{r_{ui} \sim \mathcal{D}_t} \left[ \ell \left( r_{ui} \mid \theta \right) \right] \tag{2}$$

However, this loss function presumes that all tasks are observed at once.

**“Online" Task Learning** In the continual learning context of interest in this work, each task appears sequentially. Access to past tasks' data is prohibited. We learn each task consecutively. At each time step $t$, we leverage the prior parameters $\theta^*_{t-1}$ as initialization (i.e., transfer learning), to minimize loss over current dataset $\mathcal{D}_t$ to arrive at new parameters $\theta_t$. This is equivalent to the task-level *online learning* objective.

$$\theta^*_t = \arg\min_{\theta_t} \mathbb{E}_{r_{ui} \sim \mathcal{D}_t} \left[ \ell \left( r_{ui} \mid \theta^*_{t-1}, \theta_t \right) \right] \tag{3}$$

The sequential nature of tasks implies that as we handle tasks over time, the model parameters tend to favor recent tasks. Over time, they may no longer fit the earlier tasks well, resulting in lower performance for older users and items. This phenomenon is known as *catastrophic forgetting* [3, 16].

**Training and Test Protocol** By default, at task $t$, the model is initialized from previous model $f_{\theta_{t-1}}$ and can be trained in offline batch learning setting using data of current task $\mathcal{D}_t$, while access to past samples $\mathcal{D}_{1,\ldots,t-1}$ is forbidden. After learning, the model performs on test set of every seen task (i.e., $\mathcal{D}^{test}_1, \ldots, \mathcal{D}^{test}_t$) without revealing task identities (i.e., single-head setting). This protocol is called *fine-tuning*.

**Metrics** Conventionally for Collaborative Filtering, we have two sets of metrics: rating and ranking. In this paper, we adopt Mean Square Error (MSE) for rating metric. For ranking, Recall (Hit ratio) is used to measure the ranking effectiveness of recommendation model. Besides measuring performance for individual tasks, it is also crucial to monitor how the learning process *affects across tasks*. The goal is to quickly learn current task to satisfy immediate new users while preserving existing preferences.

For each base metric above, we construct the matrix $\mathbf{a} \in \mathbb{R}^{T \times T}$, where $a_{ij}$ is performance on task $j$ after observing task $i$. To aggregate performance across tasks, we derive these two metrics:

- Learning Average (LA) $\frac{1}{T} \sum_{i=1}^{T} a_{i,i}$: to measure learning ability, how good a model is at learning tasks and whether it benefits from previously learned tasks.
- Retained Average (RA) $\frac{1}{T} \sum_{i=1}^{T} a_{T,i}$: to measure learning retention and improvement after learning the final task. The overall average will be the average performance of every learned task up to the time of testing.

With limited capacity, one system cannot preserve perfect recall of previously learned tasks while also absorbing huge data from an arbitrary number of tasks, which refers to the stability-plasticity dilemma. Given that contradiction, we further propose a combined measure, which is the harmonic mean of learning average and retained average. This metric is the aggregated number that conveys the overall goodness of a Continual Collaborative Filtering model.

$$\text{Harmonic mean} = \frac{2 * LA * RA}{LA + RA}$$

Since rating metrics are better when lower, the corresponding continual metrics are better when lower. Conversely, ranking metrics are better when higher.

## 3 USER GRADIENT ALIGNMENT-BASED APPROACH FOR CONTINUAL COLLABORATIVE FILTERING

Continual learning optimization resembles a tug-of-war, where tasks, users, or instances desire to pull the parameters towards themselves. When numerous "players" participate and pull the parameters in different directions, the process may become unending. We propose an algorithm that groups similar users based on their preferences using $k$-means clustering and aligns the learning across these groups. By learning one group at a time, gradient alignment among groups helps mitigate interference.

The grouping offers several benefits: it enhances collaboration within each group, as users with similar preferences "agree" on parameter updates, facilitating modeling for less active users. It also reduces the number of "players" in the tug-of-war, leading to more stable task learning. Additionally, the algorithm can be extended to scenarios with immediate rewards, such as real-time systems where user-item interactions are streamed. In such cases, the system can learn and satisfy a group of users promptly through online learning before proceeding.

The transfer of user grouping occurs within individual tasks, so in addition to in-task transfer, we aim to maximize transfer across tasks by utilizing episodic memory $\mathcal{M}$, which is a fixed-size memory that represents all past tasks that we sample throughout the learning process by reservoir sampling [27]. We apply task alignment between current task $\mathcal{D}_t$ and episodic memory $\mathcal{M}$ to further reach greater "agreement" among users. In summary, ***U**ser **A**lignment for **C**ollaborative **F**iltering* (UACF) objective is:

$$\theta_t^* = \arg\min_{\theta_t} \mathbb{E}\left(\left(\sum_{g\in\mathbf{G}^t} \ell(G_g \mid \theta_t) - \beta \sum_{j=1}^{g-1} \frac{\partial \ell(G_g)}{\partial \theta_t} \times \frac{\partial \ell(G_j)}{\partial \theta_t}\right) - \alpha \frac{\partial \ell(\mathcal{D}_t)}{\partial \theta_t} \times \frac{\partial \ell(\mathcal{M}_t)}{\partial \theta_t}\right) \tag{4}$$

where $\mathbf{G}^t$ is the set of user groups at task $t$ derived by $k$-means clustering from $\mathcal{D}_t$ and $\beta$ is the control regularization term for group alignment. The intuition of this is illustrated in Figure 1d.

## 4 EXPERIMENTS

The main experimental objective is to show evidence that catastrophic forgetting indeed occurs, and how the continual learning approaches address it towards balancing the learning ability and retention of previously learned tasks.

### 4.1 Experimental Settings

**Datasets** We experiment on different sources of dataset: three categories of Amazon Product Review Dataset[2] (*Books, Kindle store, Movies and TV*) and MovieLens Ratings Dataset[3]. These are categories where older items remain relevant for a long time. To simulate the expansion in users and items over time, we split the datasets chronologically into five tasks as illustrated in Figure 1a. Five tasks were chosen, as they are commonly used in popular continual learning datasets (e.g., Split-MNIST, Split-SVHN, Split-CIFAR [1, 19]). Each task was divided to introduce a roughly equal number of new users and items. Table 1 summarizes the task-wise splits.

**Base Models** Our proposed methods are model-agnostic that can plug into different collaborative filtering models. In the experiments, we adopt two base models from basic linear model MF [10] to neural networks-based, NCF [5]. These are among the major methodologies for collaborative filtering.

**Comparative Methods** Under investigation is the framework of dealing with the continual learning setting itself. To that extent, the most apt baselines would be those considered classical approaches to continual learning (fine-tuning,

[2]http://jmcauley.ucsd.edu/data/amazon
[3]https://grouplens.org/datasets/movielens

Table 1. Task-wise statistics of four datasets

| Dataset | Stats | Total | Task 1 | Task 2 | Task 3 | Task 4 | Task 5 |
|---|---|---|---|---|---|---|---|
| books | #ratings | 1,960,674 | 228,399 | 348,435 | 365,906 | 455,571 | 562,363 |
| | #users | 38,121 | 7,624 | 15,061 | 19,721 | 22,796 | 27,916 |
| | #items | 35,736 | 7,147 | 13,153 | 18,487 | 23,719 | 27,593 |
| kindle | #ratings | 438,994 | 35,736 | 68,602 | 90,899 | 110,106 | 133,651 |
| | #users | 8,533 | 1,706 | 3,353 | 4,873 | 6,102 | 7,359 |
| | #items | 10,068 | 2,011 | 4,023 | 5,931 | 7,563 | 8,704 |
| movies | #ratings | 684,453 | 61,447 | 105,429 | 111,300 | 170,052 | 236,225 |
| | #users | 16,002 | 3,104 | 6,191 | 8,763 | 11,449 | 13,541 |
| | #items | 9,774 | 1,954 | 3,900 | 5,683 | 7,554 | 9,330 |
| movielens | #ratings | 1,000,209 | 39,962 | 137,431 | 202,555 | 299,256 | 321,005 |
| | #users | 6,040 | 1,198 | 2,408 | 3,610 | 4,820 | 6,007 |
| | #items | 3,706 | 699 | 1,391 | 2,039 | 2,808 | 3,485 |

experience replay). In addition, we compare the various adopted gradient alignment-based methods from the literature (Reptile [18], MER [22]). However, rather than simply measuring which method is better or worse, we set out to show and understand the trade-off between stability and plasticity that afflicts various models in different ways.

- *Finetuning*: standard transfer learning setting with objective as shown in Equation 3, where the model only favors current data. This way, the model is enriched by previous instances and can achieve good performance for immediate samples. However, without revision of previous users, the data-driven model quickly forgets and provides poor recommendations for distant users.
- *Experience Replay (ER)*: common baseline approach for continual learning [23], where we keep some samples from previous tasks as memory buffer and replay with novel input, with objective shown in Equation 5. In our Continual Collaborative Filtering context, each user-item rating is one instance to be sampled into the memory buffer $\mathcal{M}$ by reservoir sampling, i.e., the probability that each sample to be selected is equal in data stream.

$$\theta_t^* = \arg\min_{\theta_t} \mathbb{E}_{r_{ui} \sim \mathcal{D}_t \cup \mathcal{M}} \left[ \ell \left( r_{ui} \mid \theta_{t-1}^*, \theta_t \right) \right] \tag{5}$$

  The learned parameters are expected to fit the current tasks $\mathcal{D}_t$ as well as samples from earlier tasks in $\mathcal{M}$. Given that the remnants of each task in $\mathcal{M}$ is relatively small, this lessens forgetting though it may not avert it altogether. As the model capacity is limited, paying attention to revision would affect learning ability.
- *Reptile*: gradient alignment-based method which maximizes transfer between current task and memory buffer. The objective function is shown in Equation 6.

$$\theta_t^* = \arg\min_{\theta_t} \mathbb{E}\left( \ell(\mathcal{D}_t \mid \theta_{t-1}^*, \theta_t) - \alpha \frac{\partial \ell(\mathcal{D}_t)}{\partial \theta_t} \times \frac{\partial \ell(\mathcal{M}_t)}{\partial \theta_t} \right) \tag{6}$$

- *MER*: applies gradient alignment at two levels, instance- and user-level. MER approximately optimizes for:

$$\theta_t^* = \arg\min_{\theta_t} \mathbb{E}\left( \sum_u \sum_i \ell(r_{ui} \mid \theta_{t-1}^*, \theta_t) - \alpha \sum_{p=1}^{u-1} \sum_{q=1}^{i-1} \frac{\partial \ell(r_{ui})}{\partial \theta_t} \times \frac{\partial \ell(r_{pq})}{\partial \theta_t} \right) \tag{7}$$

**Metrics** Our objective is to closely monitor the base models in order to track performance changes over time. Consequently, for MF-based models, we employ the traditional rating metric, Mean Squared Error (MSE), as MF is renowned for rating prediction and optimized for MSE. Conversely, for NCF, which is optimized for ranking, we utilize

Table 2. Finetuning, MSE(↓) for MF-based model with MovieLens.

| FT | T1 | T2 | T3 | T4 | T5 |
|---|---|---|---|---|---|
| T1 | 1.1805 | - | - | - | - |
| T2 | 1.2140 | 1.0871 | - | - | - |
| T3 | 1.2674 | 1.2623 | 1.1464 | - | - |
| T4 | 1.2797 | 1.2749 | 1.2713 | 1.1514 | - |
| T5 | 1.2781 | 1.2740 | 1.3142 | 1.3313 | 1.1726 |

Table 3. Finetuning, Recall(↑) for NCF-based model with MovieLens.

| FT | T1 | T2 | T3 | T4 | T5 |
|---|---|---|---|---|---|
| T1 | 0.7691 | - | - | - | - |
| T2 | 0.0891 | 0.7100 | - | - | - |
| T3 | 0.0137 | 0.0180 | 0.7096 | - | - |
| T4 | 0.0035 | 0.0046 | 0.0022 | 0.7316 | - |
| T5 | 0.0014 | 0.0011 | 0.0005 | 0.0032 | 0.7254 |

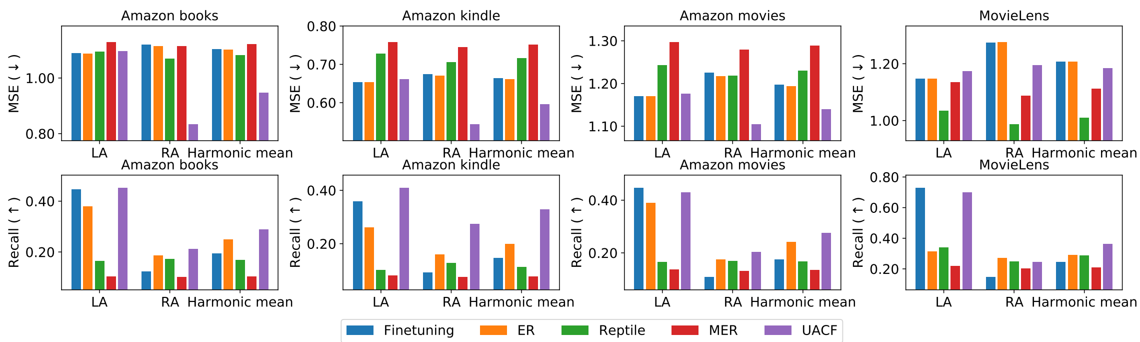



Fig. 2. Results of comparative methods on four datasets. MSE on MF-based model is on the left while Recall for NCF is on the right. Note that for MSE, the lower is the better performance and for Recall, the higher indicate better result.

the ranking-based Recall@5 metric. We follow training and test protocol described in Section 2 to obtain metric matrices. We report the overall matrices of metrics to observe model's behavior for each method and compare *Harmonic mean* that balances learning average (LA) and retained average (RA).

**Hyper-parameters tuning.** For all methods, at each task, the task's ratings are divided into train/validation/test sets. With rating prediction model, MF, we adopt proportional split in [7, 14] by ratio 60/20/20; while for NCF, we follow *leave-one-out* evaluation as original paper [5], the last item of each user is used for test and the second to last for validation. The number of user groups is fixed at 10 for all datasets. For each method, we carry out experiments with multiple (i.e., 5) random seeds and report the average over all runs.

### 4.2 Performance Comparison and Insights

**Research Question 1:** *Does the forgetting issue exist in Continual Collaborative Filtering setting?* Parameter sharing is the cause of catastrophic forgetting. Learning new tasks without proper revision of previous ones leads to forgetting of learned tasks. In MF, which is a linear model, users share parameters on representation of items; while neural networks-based NCF model, both user and item sides further share weights of layers. Intuitively, MF-based model would be affected less and NCF-based would suffer severe forgetting issue. We follow the setting in Section 2 and observe matrices of *MSE* performances with MF-based model and *Recall* results with NCF-based model for MovieLens dataset in Table 2 and 3, where each column $T_i$ is performance of task $i$ over time, while each row $T_j$ is result of all the task after training task $j$. As it proceeds to learn new tasks, the MF model slowly performs worse, which is shown in the gradual increase of MSE. On the other hand, NCF dramatically forgets all the learned tasks, as indicated by how Recall drops to near 0. This showcases that catastrophic forgetting indeed occurs in the context of collaborative filtering.

**Research Question 2:** *Overall, how does UACF perform?* Further results on four datasets, Amazon books, kindle, movies, and MovieLens are shown in Figure 2. For MF-based model, UACF achieves similar learning average and better retained average than other methods in three Amazon datasets out of the four.

With neural networks-based model, NCF, the results are consistent over all four datasets, where UACF acquires competitive learning ability as Finetuning and the best retention rate. Overall, UACF with NCF base performs the best on harmonic mean of LA and RA.

## 5 RELATED WORK

**Collaborative filtering** has mainly been oriented towards batch learning. A segment that addresses online learning [4, 6, 15] focuses primarily on incremental learning for efficiency, while orienting performance to recent observations.

There are a few works of incremental learning in general recommendation, but specific limitations in their settings and evaluation protocol preclude a bona fide continual learning setting. Ader [17] uses knowledge distillation for streaming session-based recommendation in an online learning setting without task boundaries. Conure [29], on the other hand, addresses multi-platform recommendations as a continual learning problem by parameter pruning. However, it has the drawback of being unable to accommodate new users. The experimental evaluations in [17, 20, 29] only assess post-learning performance, while ignoring across-task effects such as retention of previously learned tasks and the ability to learn new ones.

An orthogonal concept is reinforcement learning [2, 8], which has a very different focus in managing the explore-exploit trade-off of maximizing short-term payoff vs. long-term learning. In contrast, continual learning seeks to manage the stability-plasticity trade-off of preserving performance for earlier tasks vs. quickly adapting to new tasks.

Cold-start recommendation [24, 32] focuses on making recommendations for users or items with little prior information, while our continuous learning setting not only targets unaccustomed ones but also aims to improve recommendations for prior users and items. Unlike the multi-task or transfer learning setting in cross-domain recommendation [26, 31], our continual learning framework learns tasks sequentially in online manner.

**Continual learning** falls into three main categories. *Episodic memory* [13, 21, 23] keeps some samples from past tasks to utilize together with data from the current task. *Regularization* [9, 30] penalizes parameter changes to avoid forgetting previous tasks. *Dynamic architecture* [11, 28, 29] allocates specific capacity for each task and fuses local and global features for predictions. These approaches are studied in vision and have so far not been extensively explored as much in recommender systems.

## 6 CONCLUSION

We show that catastrophic forgetting indeed occurs in collaborative filtering and formulate a continual learning framework for recommendation systems. While Experience Replay stems the extent of forgetting, this comes at the cost of not being as quick to react to new observations. For a better balance, we explore gradient alignments novelly at user level. Experiments demonstrate that the proposed method, UACF, alleviates forgetting significantly while being flexible for learning new tasks, and together achieve a better balance between learning ability and retention.

## ACKNOWLEDGMENTS

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-020).

## REFERENCES


[1] Davide Abati, Jakub Tomczak, Tijmen Blankevoort, Simone Calderara, Rita Cucchiara, and Babak Ehteshami Bejnordi. 2020. Conditional channel gated networks for task-aware continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3931–3940.

[2] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 208–214.

[3] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211* (2013).

[4] Naiyang Guan, Dacheng Tao, Zhigang Luo, and Bo Yuan. 2012. Online nonnegative matrix factorization with robust stochastic approximation. *IEEE Transactions on Neural Networks and Learning Systems* 23, 7 (2012), 1087–1099.

[5] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[6] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of ACM SIGIR conference on Research and Development in Information Retrieval*. 549–558.

[7] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. 230–237.

[8] Jaya Kawale, Hung H Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. 2015. Efficient Thompson Sampling for Online Matrix-Factorization Recommendation. *Advances in neural information processing systems* 28 (2015).

[9] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.

[10] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[11] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. 2019. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International Conference on Machine Learning*. PMLR, 3925–3934.

[12] Guang Ling, Haiqin Yang, Irwin King, and Michael R Lyu. 2012. Online learning for collaborative filtering. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[13] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems* 30 (2017), 6467–6476.

[14] Hao Ma, Irwin King, and Michael R Lyu. 2009. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 203–210.

[15] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. 2010. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research* 11, 1 (2010).

[16] Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. Vol. 24. Elsevier, 109–165.

[17] Fei Mi, Xiaoyu Lin, and Boi Faltings. 2020. Ader: Adaptively distilled exemplar replay towards continual learning for session-based recommendation. In *Fourteenth ACM Conference on Recommender Systems*. 408–413.

[18] Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999* (2018).

[19] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. 2019. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11321–11329.

[20] Danni Peng, Sinno Jialin Pan, Jie Zhang, and Anxiang Zeng. 2021. Learning an adaptive meta model-generator for incrementally updating recommender systems. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 411–421.

[21] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2001–2010.

[22] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. 2019. Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference. In *In International Conference on Learning Representations (ICLR)*.

[23] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P Lillicrap, and Greg Wayne. 2019. Experience replay for continual learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 350–360.

[24] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. 253–260.

[25] Reema Sikka, Amita Dhankhar, and Chaavi Rana. 2012. A survey paper on e-learning recommender system. *International Journal of Computer Applications* 47, 9 (2012), 27–30.

[26] Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. 2012. Cross-domain collaboration recommendation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1285–1293.

[27] Jeffrey S Vitter. 1985. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)* 11, 1 (1985), 37–57.

[28] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. 2018. Lifelong Learning with Dynamically Expandable Networks. In *International Conference on Learning Representations*.

[29] Fajie Yuan, Guoxiao Zhang, Alexandros Karatzoglou, Joemon Jose, Beibei Kong, and Yudong Li. 2021. One person, one model, one world: Learning continual user representation without forgetting. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 696–705.

[30] Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*. PMLR, 3987–3995.

[31] Cheng Zhao, Chenliang Li, Rong Xiao, Hongbo Deng, and Aixin Sun. 2020. 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*. 229–238.

[32] Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. 2011. Functional matrix factorizations for cold-start recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 315–324.