Clustered Federated Learning for Recommendation Systems: Tackling Data Heterogeneity

Yiyang Lyu *

School of Data Science, The Chinese University of Hong Kong, Shenzhen, China

* Corresponding Author Email: 123090406@link.cuhk.edu.cn

Abstract. Recommendation systems have shifted to the federated paradigm (FedRS) to address privacy risks and data silos, yet non-IID data heterogeneity is still a bottleneck limiting FedRS performance. And Clustered Federated Learning (CFL) emerges as a potential solution for recommendation systems. This review attempts to sort out two core CFL frameworks—Iterative Federated Clustering Algorithm (IFCA, alternating cluster identity estimation and model optimization) and SnapCFL (decoupling pre-clustering and dynamic client selection)—and three CFL-based FedRS approaches: user clustering (PerFedRec, FedGWC, FedPCL), item clustering (CoFedRec, ClusterGCF), and user-item co-clustering (CdFed, CPF-GCN), while analyzing challenges like dynamic user preferences, data sparsity/cold-start, and large-system scalability. With the rapid growth of personalized services and increasing privacy concerns, understanding how CFL can enhance model personalization and efficiency has become crucial for advancing privacy-preserving recommender systems. This review tries to present key progress in this field, hoping to provide some reference for readers to grasp the current research status and clarify subsequent directions.

Keywords: Clustered federated learning, recommendation system, data heterogeneity.

1. Introduction

As information explodes in modern society, recommendation systems have become indispensable for users in wide real-world fields like e-commerce platforms, news aggregators, and Point-of-interest (POI) recommendation. Their core mission is to model users' preferences based on historical data and use this model to tailor the content shown to users, which alleviates the information overload problem. Traditionally, a recommendation system works based on a centralized paradigm where the central server collects, stores raw user data, and trains the model globally. However, increasing challenges like privacy safety risks, regulatory restrictions, and data silos have driven recommendation systems to shift from a centralized paradigm to a federated paradigm, which relies on sharing intermediate parameters instead of raw user data. Therefore, Federated Learning (FL) becomes a promising solution to address these challenges.

In FL, edge clients (e.g., mobile devices or data platforms) train local models using their own raw data and only send model parameters to the central server, which aggregates these local parameters to update the global model. This idea has been integrated with recommendation systems, leading to the emergence of Federated Recommendation Systems (FedRS) [1]. While FedRS effectively addresses privacy problems, data heterogeneity—especially non-independent and identically distributed(non-IID) data problems becomes a bottleneck, causing a single global model to be unable to adapt to different local patterns, thus undermining the recommendation's effect. Aimed at tackling heterogeneous problems, Clustered Federated Learning (CFL) integrates clustering algorithms and federated learning, grouping clients with similar data distributions and trains cluster-specific models [2]. FedRS with CFL strategies not only retains FedRS's core advantage of privacy preservation but also fits the heterogeneous data distributions of clients.

Until now, general CFL research toward heterogeneity problems has provided a solid algorithmic foundation for the development of FedRS. Distributed clustering mechanisms, with convergence guarantees for both convex/non-convex models, group clients precisely with similar patterns. FedRS integrated with CFL strategies can be mainly divided into three main streams: The first one is user clustering, which groups users by their preference similarity. The second one is item clustering, which

groups items by their features and guides users who interact frequently with a certain cluster of items to train the corresponding model. The third one is user-item co-clustering, which optimizes user-item clusters based on their interrelationship. All these three clustering strategies confront data heterogeneity challenges—divergent user preferences, sparse item features, and mismatched user-item relationships, requiring the update of CFL-based FedRS algorithms to optimize their performance.

This review focuses on Clustered Federated Learning for recommendation systems with the core goal of tackling data heterogeneity. First, it elaborates two fundamental CFL frameworks—the Iterative Federated Clustering Algorithm (IFCA) and SnapCFL—laying a theoretical foundation for subsequent recommendation system applications. Then it discusses three mainstreams of CFL-based FedRS and some outstanding studies addressing heterogeneity challenges: three on user clustering, two on item clustering, and two on user-item co-clustering. Finally, the review summarizes these and outlines future research directions.

2. Method

2.1. Fundamental Framework

In modern federated recommendation systems, the traditional independent and identically distributed (i.i.d.) assumption does not hold true since data from different clients exhibit diverse patterns, which leads to suboptimal performance of a single global model. To handle this challenge, CFL integrates the principles of clustering with FL, grouping clients with similar data distributions into clusters and training separate models for each cluster. This improves the overall recommendation performance.

2.1.1. Iterative Federated Clustering Algorithm (IFCA)

IFCA, proposed by Ghosh et al., is one of the foundational methods for implementing CFL in federated learning systems [3]. The core idea is alternating between estimating client cluster identities and optimizing each cluster's model parameters, which addresses data heterogeneity via a decentralized clustering method. First, each client is assigned an initial model for each cluster. Then in each iteration, each client receives the current models from the center machine, selects its model that minimizes its local loss function, does a local update (compute local gradient and send it to the center machine or update models locally and send updated model back). Finally, the center machine updates each cluster's model based on either averaged gradients or models. This process is repeated for a number specified earlier. Unlike traditional FL methods, IFCA does not rely on centralized clustering, and it is beneficial when there is data heterogeneity across clients.

2.1.2. Snap Cluster Federated Learning (SnapCFL)

SnapCFL is a pre-clustering-based CFL method that addresses the data heterogeneity challenge by decoupling the traditional process into two stages: pre-clustering and main FL training [4]. Before actual training, clients are grouped based on their data distributions. In this stage, SnapCFL models their similarity as a two-sample hypothesis testing problem, allowing for a more precise assessment of data similarity. After pre-clustering, clients within each cluster train their model collaboratively through a constraint-based client-based mechanism introduced by SnapCFL. This mechanism addresses system heterogeneity by dynamically selecting clients that minimize convergence time while meeting system constraints (latency and computational load). These separate stages provide a more flexible and efficient framework, helping avoid the local optima problem common in traditional coupled methods while also further improving performance by reducing system heterogeneity.

2.2. User Clustering

User clustering groups users based on their preferences and forms different client clusters with similar attributes. Recommendation systems use user clustering to recommend items to an entire group of clients who have similar interests.

2.2.1. Personalized Federated Recommendation (PerFedRec)

PerFedRec combines Graph Neural Networks (GNNs) with user clustering, aiming to enhance the performance while preserving clients' privacy [5]. The first step is to learn a representation for each user that captures their attributes and collaborative filtering data. In the raw embedding module, these features are processed through a linear layer and a feature crossing layer to generate feature embedding. Then, an attention mechanism is applied to result in each user's raw embedding, which is next processed by a local GNN module to capture collaborative user-item interactions and individual attributes. Once user representations are learned by PerFedRec Systems, it gets a compact and informative vector, and from this vector, the system clusters users by applying a clustering algorithm like K-means. After clustering users into groups, each cluster can be assigned a cluster-level federated model, which is trained based on the aggregated information from each cluster's users. Besides, a global federated model is created. Users are recommended by PerFedRec Systems'personalized model, combining local, cluster-level, and global models. The PerFedRec proposes an innovative perspective that efficiently reduces data heterogeneity by learning a personalized representation to cluster users.

2.2.2. Federated Gaussian Weighting Clustering (FedGWC)

FedGWC solves data heterogeneity challenges by clustering clients based on a Gaussian weighting mechanism instead of using direct model updates for clustering [6]. During the process of FedGWC, clients report their empirical loss values to the server, and based on the difference between each client's loss and the average loss of all clients, Gaussian rewards, which reflect the closeness of clients and global distribution, are computed. To model relationships between clients' data distributions, an interaction matrix, whose entries are updated by Gaussian rewards, is built, capturing pairwise similarities between clients. Then the interaction matrix is converted into an affinity matrix, and the FedGWC uses spectral clustering to divide clients into clusters. The Gaussian Weighting mechanism and spectral clustering process simplify the clustering process and effectively offer a practical approach for the CFL recommendation system.

2.2.3. Federated Personalized Contrastive Learning (FedPCL)

FedPCL addresses highly non-IID data and sparse user-item interactions issues by using structural contrastive learning and personalized model aggregation [7]. In FedPCL, each client maintains a local graph whose nodes represent users and items, and edges represent interactions between them. Then, a graph-based model, LightGCN, performs layer-wise aggregation to propagate information through the graph and eventually extract refined information from the graph. After getting the structural information, FedPCL introduces structural contrastive learning to use this information to enhance user-item representations. For each node, the model learns to pull its structural neighbors (similar users or items) closer while pushing other nodes further. During this step, FedPCL uses the InfoNCE loss function, which encourages positive samples to be close and negative samples to be distant, to minimize the contrastive loss measuring the similarity between nodes. Along with contrastive learning, the model also employs Bayesian Personalized Ranking (BPR), a ranking loss function that maximizes the likelihood of users interacting with the items they prefer, to make positive items (interact more often with users) rank higher than negative ones. With both contrastive loss and the BPR loss, the client updates the local model parameters' gradient and sends it to the server. The server receives these gradients and performs K-means clustering to group clients. Once the users are clustered, the server computes a cluster-specific model by averaging all users within each cluster and then aggregates them with the global model (average of all users) to produce a model for each cluster. This mixture balances personal and global information, improving the recommendation system's

efficiency. Over multiple federated rounds, the model reaches an acceptable threshold even with heterogeneous data.

2.3. Item Clustering

Item clustering refers to grouping items with similar features or rated similarly by users, which helps to relate a large number of items. When recommendation systems apply item clustering, they can provide item-based recommendations like movies of the same genre.

2.3.1. Co-clustering for Federated Recommendation (CoFedRec)

CoFedRec introduced co-clustering mechanisms to improve traditional Federated Recommender Systems (FRS), clustering users based on item categories [8]. First, the server initializes a global item embedded model and shares it with all clients. Then, in each communication round, users update their models and the server aggregates them to generate a global item representation. After applying K-means clustering on this global item representation, an item membership vector that categorizes items into clusters is generated. For a specific item category, the server selects a core user randomly and calculates cosine similarity between the core user and other users. And users are split into similar and dissimilar groups through elbow methods, evaluating their similarity scores. For users in a similar group, the server aggregated individual models into a group model and distributed it back, while the dissimilar group remains their model the same. Besides, during local training, each client minimizes binary cross-entropy loss and the supervised contrastive loss, updating the model based on both local data and global item categories. CoFedRec uses a co-cluster mechanism, grouping users based on specific item categories, to address the data heterogeneity challenge in FRS.

2.3.2. Cluster-Based Graph Collaborative Filtering (ClusterGCF)

ClusterGCF's reduction is applying high-order graph convolution on cluster-specific graphs constructed by soft node clustering, which captures multiple interests of users and reduces noise in high-order information [9]. ClusterGCF begins by performing soft node clustering on the user-item interaction graph, assigning item nodes to a probabilistic distribution across multiple clusters based on their features. Based on soft clustering results, cluster-specific graphs are formed by nodes of user and items weighted by their probability of belonging to a cluster, and edges represent interactions. Then ClusterGCF applies high-order graph convolutions on cluster-specific graphs, aggregating information within the cluster and reducing the noise from irrelevant neighbors. During the convolution process, item embeddings are updated based on interactions with users, focusing on cluster-specific neighbors. Finally, the model uses the final embedding to calculate recommendation scores by taking the dot product between user and item embeddings. ClusterGCF groups items based on shared features and user interactions, defining local graph structures used in a high-order convolution process, which propagates information better. Besides, soft clustering can capture complex characteristics of items, leading to more personalized recommendations.

2.4. User-Item Co-Clustering

User-item co-clustering means both users and items are clustered simultaneously based on the interaction matrix. This method finds the patterns of the interaction between users and items. Therefore, a recommendation system can use this latent relation to improve the recommendation accuracy, combining users'behaviors and items' characteristics to create clusters of user-item associations.

2.4.1. Cluster-driven GNN-based Federated Learning (CdFed)

CdFed combines FL with Graph Neural Networks for recommendation systems, targeting issues of data heterogeneity and overfitting [10]. In federated learning, clients hold their local subgraphs containing their own user-item interactions. These graphs are non-IID and only contain low-order interactions, which makes it difficult for normal GNN models to capture complex user-item relationships. Besides, overfitting is a challenge for local models trained on limited data. CdFed uses

Adaptive Model Clustering (AMC) to simultaneously cluster users and items to optimize recommendation performance by clustering models rather than clustering users and items directly. First, each client trains its local user-item model, using GNN to update node representations from neighbor nodes. After message propagation, CdFed applies Biased Message Dropout (BMD) to drop less important features, thus remaining only the most important features, which helps prevent overfitting problems. After each training round, AMC cluster models are based on their similarity in weight and feature distributions via calculating cosine similarity. Models in the same cluster share information to find missing user-item connections even when the data is sparse. Then these updated models are aggregated and optimized globally, and finally sent back to clients for further local updates.

2.4.2. Cluster-driven Personalized Federated Recommendation with Interest-aware Graph Convolution Network (CPF-GCN)

The CPF-GCN system combines FL with Graph Convolution Networks (GCNs) to tackle data heterogeneity challenges [11]. GCNs can capture complex user-item interactions that go beyond user-item matrix factorization techniques, while they face non-IID data and over-smoothing challenges (node representations become similar after multiple convolution layers). To address these challenges, CPF-GCN proposes a local interest-aware GCN module and an adaptive graph convolution method. In the beginning, each client trains a local model using their own user-item interaction data, and GCN learns user/item embeddings using interest-aware subgraphs, which ensures clients interact mainly with similar user/items. Then, after an adaptive graph convolution aiming to avoid over-smoothing problems, users are clustered based on their learned embeddings. The server selects a subset of representative clients from each cluster to generate cluster-level models along with a global model. Similarly, a personalized model is the combination of its cluster and global model. CPF-GCN addresses data heterogeneity problems by introducing interest-aware subgraphs, adaptive graph convolution, and cluster-driven aggregation.

3. Discussion

3.1. Challenges

3.1.1. Dynamic Shifts in User Preferences

Though clustering strategies address data heterogeneity problems at certain levels, in modern recommendation systems, heterogeneous data still remains one of the most significant challenges. Data across different clients can vary widely because of their preferences. This non-IID nature led to suboptimal performance when applying the model across diverse clients. Clustered Federated Learning mitigates this challenge by grouping clients with similar data distributions and training models for each cluster of clients. However, existing cluster methods in CFL, like IFCA and SnapCFL, assume that client data distributions remain relatively stable during the training process. In fact, data heterogeneity problems caused by dynamic drifts of user preference still exist since user behavior changes over time, causing static methods to be less effective.

3.1.2. Sparse Data and Cold-Start Problems

Sparse data in cold start scenarios is also a significant challenge in modern FRSs. New users typically have few interactions with the system, making it difficult for the recommendation system to learn their preferences. Similarly, newly introduced items lack sufficient ratings and interactions, whose relevance is hard to build in the recommendation process. In a cluster federated learning recommendation system, the cold-start problem is more complex than traditional systems due to several unique challenges. Clients in federated learning train their model locally and do not share raw data due to privacy protection, which limits the shared information used to mitigate the cold-start problem. In user cold-start scenarios, the recommendation systems have to rely on additional metadata, like general preferences, to make an initial user profile. And in item cold-start scenarios, content-based filtering is applied to recommend items with similar interactions. However, when users

or items are clustered, the additional data within a cluster becomes even sparser, making recommendations for new users or items less effective. Besides, cold-start problems also prevent the cluster algorithm from co-cluster user-item since their interaction data are insufficient.

3.1.3. Scalability of Federated Clustering in Large-Scale Systems

Scalability of federated clustering becomes a critical issue as FRS scales to a large number of diverse clients. Traditional clustering algorithms, such as K-means or spectral clustering, suffer from high computational complexity and communication overhead when dealing with a large number of clients. Especially for user-item co-clustering, the computational cost of similarities between large sets of users and items will significantly delay the model training process and consume large amounts of energy. Clustering is further complicated because of the non-IID nature of clients' data, forcing CFL techniques to balance accuracy and efficiency.

3.2. Future prospects

3.2.1. Adaptive Clustering Techniques

To address the dynamic shift in user preferences, future research can focus on developing adaptive clustering techniques that better respond to the changes in clients' preferences over time. Such techniques adjust clusters more frequently to make sure the model follows the evolution of its clients. A dynamic clustering algorithm that ensures clusters are non-static might be a promising solution to capture temporal patterns of users/items. Integrating contextual information, such as users' activity, can offer richer representations of clients and then predict clients' further preference shifts.

3.2.2. Integration of Auxiliary Data

To mitigate the sparse data and cold-start problems, auxiliary data like item metadata and user profiles can be integrated with CFL recommendation systems. In federated systems where raw data cannot be shared across clients due to privacy problems, such auxiliary data can provide additional information to help improve recommendation results. User-related auxiliary data helps enrich user profiles when user clustering lacks interaction data. These data can be demographic data, such as age, gender, location, or behavioral data (e.g., browsing history), and contextual data (e.g., mobile or desktop devices and browsing time of a day). This data can help with initial clustering steps when interactions of users are sparse, solving cold-start problems. As for new items, metadata such as genre, brand, and price are especially useful for grouping. Besides, content-based features like description in an e-commerce system help to cluster a new item.

3.2.3. More efficient clustering algorithms

Addressing the scalability of large-scale federated systems requires clustering algorithms to reduce computational complexity. Future work can explore distributed clustering techniques that reduce the clustering burden on the central server to different subsets of clients in parallel. FRSs that do not require high accuracy can also try approximate clustering algorithms like the mini-batch method to trade between accuracy and efficiency. Additionally, techniques like client selection and active learning focus on informative clients, reducing training data while keeping it robust for clustering. For Graph-based algorithms, pruning the graph reduces the model's size and information exchanged between edge devices and the server during the federated training process.

4. Conclusion

This review presents several Clustered Federated Learning methods for recommendation systems, with the core goal of addressing data heterogeneity challenges, which is a key bottleneck limiting the performance of traditional FedRS. First, it elaborates on two CFL frameworks and then explores several CFL-based federated recommendation methods divided by three main recommendation streams, including user clustering, item clustering, and user-item co-clustering. In this part, the review shows the core idea and main process of these methods, analyzing how each strategy is adapted to

heterogeneous data problems. After summarizing these methods, this paper discusses key challenges like dynamic user preference shifts, data sparsity with cold-start issues, and poor scalability in large systems. Future research may prioritize adaptive clustering techniques, integration of auxiliary data, and more efficient clustering algorithms.

References

[1] Yang L, Tan B, Zheng VW, Chen K, Yang Q. Federated recommendation systems. In: Federated Learning: Privacy and Incentive. Cham: Springer International Publishing; 2020. p. 225 – 39.

[2] Duan M, Liu D, Ji X, Wu Y, Liang L, Chen X, et al. Flexible clustered federated learning for client-level data distribution shift. IEEE Trans Parallel Distrib Syst. 2022; 33 (11): 2661 – 74.

[3] Ghosh A, Chung J, Yin D, Ramchandran K. An efficient framework for clustered federated learning. Adv Neural Inf Process Syst. 2020; 33: 19586 – 97.

[4] Cheng Y, Zhang W, Zhang Z, Kang J, Xu Q, Wang S, et al. SnapCFL: A pre-clustering-based clustered federated learning framework for data and system heterogeneities. IEEE Trans Mobile Comput. 2025.

[5] Luo S, Xiao Y, Song L. Personalized federated recommendation via joint representation learning, user clustering, and model adaptation. In: Proceedings of the 31st ACM International Conference on Information and Knowledge Management. 2022. p. 4289 – 93.

[6] Licciardi A, Leo D, Fanì E, Caputo B, Ciccone M. Interaction-aware Gaussian weighting for clustered federated learning. arXiv. 2025. arXiv: 2502.03340.

[7] Wang S, Zhou Y, Fan X, Li J, Lei Z, Gong M. Personalized federated contrastive learning for recommendation. IEEE Trans Comput Soc Syst. 2025.

[8] He X, Liu S, Keung J, He J. Co-clustering for federated recommender system. In: Proceedings of the ACM Web Conference 2024. 2024. p. 3821 – 32.

[9] Liu F, Zhao S, Cheng Z, Nie L, Kankan Halli M. Cluster-based graph collaborative filtering. ACM Trans Inf Syst. 2024; 42 (6): 1 – 24.

[10] Zhang R, Chen Y, Wu C, Wang F. Cluster-driven GNN-based federated recommendation with biased message dropout. In: 2023 IEEE International Conference on Multimedia and Expo (ICME). IEEE; 2023. p. 594 – 9.

[11] Mao X, Liu Y, Qi L, Duan L, Xu X, Zhang X, et al. Cluster-driven personalized federated recommendation with interest-aware graph convolution network for multimedia. In: Proceedings of the 32nd ACM International Conference on Multimedia. 2024. p. 5614 – 22.