SCIENCE CHINA
Information Sciences

• LETTER •

April 2025, Vol. 68, Iss. 4, 140106:1–140106:2
https://doi.org/10.1007/s11432-024-4338-4

Special Topic: Cloud-Edge Collaboration for On-Device Recommendation

PerFedKG: two-stage information-loop federated knowledge graph for personalized privacy-preserving recommendation systems

Fan WANG[1], Xuyun ZHANG[2], Weiming LIU[1], Li LI[3], Yuwen LIU[3],
Zhongyuan ZHANG[3], Guanfeng LIU[2*], Shengye PANG[1],
Xiaolong XU[4] & Lianyong QI[3]

[1]*College of Computer Science and Technology, Zhejiang University, Hangzhou 310013, China*
[2]*Department of Computing, Macquarie University, Sydney 2109, Australia*
[3]*College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China*
[4]*School of Software, Nanjing University of Information Science and Technology, Nanjing 210044, China*

Received 31 May 2024/Revised 14 September 2024/Accepted 11 March 2025/Published online 20 March 2025

Citation Wang F, Zhang X Y, Liu W M, et al. PerFedKG: two-stage information-loop federated knowledge graph for personalized privacy-preserving recommendation systems. Sci China Inf Sci, 2025, 68(4): 140106, https://doi.org/10.1007/s11432-024-4338-4

Knowledge graphs (KGs) effectively mitigate data sparsity in recommendation systems (RSs) by providing valuable auxiliary information [1]. However, traditional centralized KG-based RSs increase the risk of user privacy leakage. Federated learning (FL) enhances RS's privacy by enabling model training on decentralized data [2]. Although integrating KG and FL can address both data sparsity and privacy issues in RSs [3], several challenges persist. CH1, Each client's local model relies on a consistent global model from the server, limiting personalized deployment to end-users. In federated RSs, personalization is maintained locally through user embeddings, while item embeddings and the preference predictor are aggregated and shared with uniform parameters [4]. CH2, Existing solutions struggle to balance semantic completeness with low privacy overhead. Current methods [3, 4] allow each client to access only the end-user's immediate one-hop neighbors (interacted items). Privacy constraints prevent the exchange of user-item interaction data between clients, disrupting semantic information from commonly accessed items or associated entities. These fragmented semantics reduce KG effectiveness for RSs. Although encrypting user interaction data using techniques like differential privacy [5] can help access rich semantic information, it increases communication costs. Thus, it is challenging to achieve a solution that maintains both integral semantic information and low privacy overhead.

To address these challenges, we propose `PerFedKG`, a two-stage personalized information-loop federated knowledge graph framework for privacy-preserving recommendations. As illustrated in Figure 1, the framework consists of a central server and multiple user clients. Each client trains the model through three main modules, the recommendation module, the knowledge graph module, and the information exchange module, and a two-stage personalized information-loop mechanism. Finally, we apply local differential privacy (LDP) for parameter aggregation. Next, we will describe how `PerFedKG` works.

Problem formulation. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ and $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ be sets of users and items, respectively, where $|\mathcal{U}| = M$ and $|\mathcal{V}| = N$. User implicit feedback on items is represented by the interaction matrix $\boldsymbol{Y} \in \mathbb{R}^{M\times N}$. Additionally, we have a knowledge graph $\mathcal{G}$ composed of entity-relation-entity triplets (h, r, t), where each triplet describes a relationship r linking head h to tail t. The main problem is that given user-level private interactions $\boldsymbol{Y}_i$ stored locally on user u_i's client, along with the shared item set $\mathcal{V}$ and knowledge graph $\mathcal{G}$, how can we learn a user-specific model with complete semantics deployed on end devices in a privacy-preserving manner, thereby accurately predicting the personalized preference $\hat{y}_{i,j}$ of user u_i for item v_j.

Recommendation module. To begin with, we convert each item, relation, and entity into embeddings. For the j-th item embedding $\boldsymbol{v}_j$, we extract features by exploring high-order relationships with associated entities e:

$$\boldsymbol{v}_j^L = \frac{1}{|\mathcal{N}(v_j)|} \sum_{e\in\mathcal{N}(v_j)} \mathcal{F}^L(\boldsymbol{v}_j, \boldsymbol{e})\big|_v, \tag{1}$$

where $\mathcal{N}(v_j)$ is the set of entities linked to item v_j, and $\mathcal{F}^L|_v$ extracts information from entities within L hops. With the obtained $\boldsymbol{v}_j^L$, we use a personalized RS predictor $S_i^{\rm RS}$ to predict the rating $\hat{y}_{i,j}$ from user u_i to item v_j:

$$\hat{y}_{i,j} = \sigma(S_i^{\rm RS}(\boldsymbol{v}_j^L)), \tag{2}$$

where $S_i^{\rm RS}$ is an H-layer multi-layer perceptron (MLP) with

[*] Corresponding author (email: guanfeng.liu@mq.edu.au)

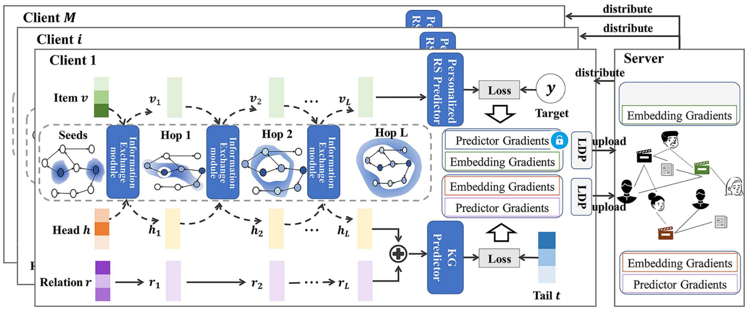

Figure 1 (Color online) Framework of our `PerFedKG` approach.

parameter θ_i^{RS} and $\sigma(\cdot)$ is the sigmoid function. The recommendation module's loss function for user u_i is

$$\mathcal{L}_i^{\mathrm{RS}}(y,\hat{y}) = -\sum_{(i,j)\in\mathcal{I}_i^+}\log\hat{y}_{i,j} - \sum_{(i,j')\in\mathcal{I}_i^-}\log(1-\hat{y}_{i,j'}), \quad (3)$$

where $\mathcal{I}_i^+$ and $\mathcal{I}_i^-$ are the sets of positive and negative samples for user u_i, respectively.

Knowledge graph module. For each knowledge triple (h,r,t), we extract the entity embedding $\boldsymbol{h}$ by exploring high-order relationships between h and its connected items:

$$\boldsymbol{h}^L = \frac{1}{|\mathcal{N}(h)|}\sum_{v\in\mathcal{N}(h)}\mathcal{F}^L(\boldsymbol{v},\boldsymbol{e})\big|_e, \quad (4)$$

where $\mathcal{N}(h)$ is the set of items linked to entity h, and $\mathcal{F}^L|_e$ extracts information from items within L hops of h, as detailed in the "Information exchange module." Next, for the relation embedding $\boldsymbol{r}$, we use an L-layer MLP to obtain its latent features:

$$\boldsymbol{r}^L = \mathcal{M}(\mathcal{M}(\cdots\mathcal{M}(\boldsymbol{r}))) = \mathcal{M}^L(\boldsymbol{r}). \quad (5)$$

We then concatenate $\boldsymbol{h}^L$ and $\boldsymbol{r}^L$ and use a KG predictor S^{KG} to predict the tail $\hat{\boldsymbol{t}}$:

$$\hat{\boldsymbol{t}} = S^{\mathrm{KG}}(\boldsymbol{h}^L \oplus \boldsymbol{r}^L), \quad (6)$$

where S^{KG} is an H-layer MLP with parameter θ^{KG} and $\oplus$ denotes concatenation. Finally, the knowledge graph module's loss function is defined as

$$\mathcal{L}^{\mathrm{KG}}(\boldsymbol{t},\hat{t}) = -\left(\sum_{(h,r,t)\in\mathcal{G}}\sigma(\boldsymbol{t}^{\mathrm{T}}\hat{\boldsymbol{t}}) - \sum_{(h',r,t')\notin\mathcal{G}}\sigma(\boldsymbol{t}'^{\mathrm{T}}\hat{\boldsymbol{t}})\right), \quad (7)$$

where $\boldsymbol{t}$ is the ground truth feature vector of t, and the inner product of $\boldsymbol{t}$ (or $\boldsymbol{t}'$) and $\hat{t}$ estimates their similarity.

Information exchange module. To extract the l-th order information for item v and entity e, we compute the outer product of their $(l-1)$-th order latent features [1], i.e., $\boldsymbol{v}^{l-1}(\boldsymbol{e}^{l-1})^{\mathrm{T}}$ or $\boldsymbol{e}^{l-1}(\boldsymbol{v}^{l-1})^{\mathrm{T}}$. We then apply trainable weights for the item side $(\boldsymbol{w}_{vv}^{l-1},\boldsymbol{w}_{ev}^{l-1},\boldsymbol{b}_v^{l-1})$ and the entity side $(\boldsymbol{w}_{ve}^{l-1},\boldsymbol{w}_{ee}^{l-1},\boldsymbol{b}_e^{l-1})$ to project and generate the l-th order latent features:

$$\begin{aligned}\boldsymbol{v}^l &= \mathcal{F}^l(\boldsymbol{v}^{l-1},\boldsymbol{e}^{l-1})\big|_v = \boldsymbol{v}^{l-1}(\boldsymbol{e}^{l-1})^{\mathrm{T}}\boldsymbol{w}_{vv}^{l-1} + \boldsymbol{e}^{l-1}(\boldsymbol{v}^{l-1})^{\mathrm{T}}\boldsymbol{w}_{ev}^{l-1} + \boldsymbol{b}_v^{l-1},\\ \boldsymbol{e}^l &= \mathcal{F}^l(\boldsymbol{v}^{l-1},\boldsymbol{e}^{l-1})\big|_e = \boldsymbol{v}^{l-1}(\boldsymbol{e}^{l-1})^{\mathrm{T}}\boldsymbol{w}_{ve}^{l-1} + \boldsymbol{e}^{l-1}(\boldsymbol{v}^{l-1})^{\mathrm{T}}\boldsymbol{w}_{ee}^{l-1} + \boldsymbol{b}_e^{l-1},\end{aligned} \quad (8)$$

where each weight vector $\boldsymbol{w}_{\cdot\cdot}^{l-1}$ and bias vector $\boldsymbol{b}_{\cdot}^{l-1}$ are in $\mathbb{R}^d$, ensuring dimensional consistency across projections.

Federated learning objective. Based on the three modules described above, we devise the overall optimization objective of `PerFedKG`:

$$\min_{\theta^v,\theta^e,\theta^r,\theta^{\mathrm{KG}},\{\theta_i\}_{i=1}^N} J = \sum_{i=1}^{L}\mathcal{L}_i^{\mathrm{RS}}(\theta_i;y,\hat{y}) + \mathcal{L}^{\mathrm{KG}}(\theta_i;\boldsymbol{t},\hat{t}),$$

$$\text{s.t.,}\quad \theta_i := (\theta^v - \nabla_{\theta^v}\mathcal{L}_i, \theta^e - \nabla_{\theta^e}\mathcal{L}_i, \theta^r - \nabla_{\theta^r}\mathcal{L}_i, \theta^{\mathrm{KG}} - \nabla_{\theta^{\mathrm{KG}}}\mathcal{L}_i, \theta_i^{\mathrm{RS}}), \quad (9)$$

where θ^v, θ^e, θ^r, and θ^{KG} are global parameters, collectively defined as $\theta := (\theta^v,\theta^e,\theta^r,\theta^{\mathrm{KG}})$. In contrast, $\theta_i := (\theta^v,\theta^e,\theta^r,\theta^{\mathrm{KG}},\theta^{\mathrm{RS}})$ represents the personalized parameter specific to the i-th client.

Two-stage personalized information-loop mechanism. Stage 1: Initialize θ_i with global θ from the server, fix θ, and learn the user-specific parameter θ_i^{RS} for S_i^{RS}. This yields a user-specific predictor enriched with preference information. Stage 2: Fix θ_i^{RS} and fine-tune θ by $\theta - \nabla_\theta\mathcal{L}_i$, ensuring global parameters become personalized to the user while incorporating their preference data. Overall, across the two stages, user personalization can be collectively preserved through four components (i.e., the predictor and the embeddings of the item, relation, and entity) for CH1. As user personalization is already encapsulated within the privacy-insensitive triplet embeddings, we can obtain a global model that contains integral semantic information through aggregating these triplet embeddings while protecting user privacy for CH2.

Result. We compared our method with six state-of-the-art approaches on MovieLens-100K and Lastfm-2K using normalized discounted cumulative gain (NDCG) and hit ratio (HR). On MovieLens-100K, our method improved NDCG and HR by averages of 78.98% and 34.23%, respectively, over other federated recommendation competitors. On Lastfm-2K, it achieved improvements of 43.83% (NDCG) and 5.91% (HR), respectively.

Conclusion. We propose a `PerFedKG` framework, which integrates federated learning with knowledge graphs to handle the user privacy and data sparsity problems. `PerFedKG`'s unique two-stage approach successfully provides a personalized model training with decentralized data, while maintaining comprehensive semantic integrity.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant No. 62372242).

References

1 Wang H, Zhang F, Zhao M, et al. Multi-task feature learning for knowledge graph enhanced recommendation. In: Proceedings of the ACM on Web Conference (WWW), 2019. 2000–2010
2 Zhang C, Long G, Zhou T, et al. Dual personalization on federated recommendation. In: Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI), 2023. 4558–4566
3 Yan B, Cao Y, Wang H, et al. Federated heterogeneous graph neural network for privacy-preserving recommendation. In: Proceedings of the ACM on Web Conference (WWW), 2024. 3919–3929
4 Perifanis V, Efraimidis P S. Federated neural collaborative filtering. Knowledge-Based Syst, 2022, 242: 108441
5 Wu C, Wu F, Lyu L, et al. A federated graph neural network framework for privacy-preserving personalization. Nat Commun, 2022, 13: 3091