# Federated News Recommendation with Fine-grained Interpolation and Dynamic Clustering

Sanshi Lei Yu, Qi Liu[1], Fei Wang, Yang Yu, Enhong Chen
University of Science and Technology of China

[1]Corresponding author.

# Contents

# Introduction

# News Recommendation

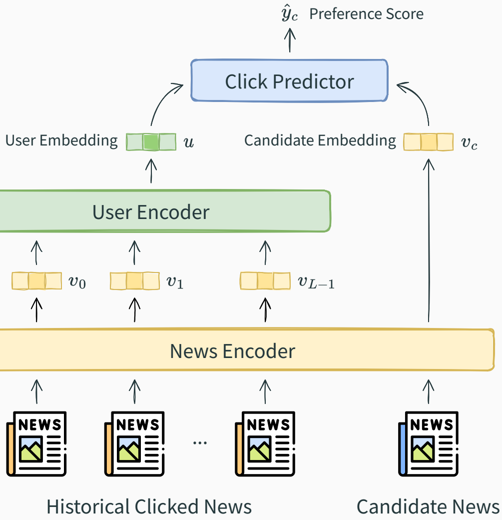



**Figure 1:** General news recommendation model structure

# Federated Learning

Federated Learning helps to protect user privacy in news recommendation tasks.

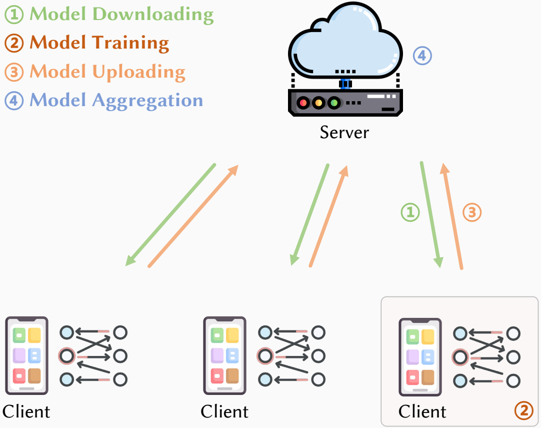



**Figure 2:** The workflow of Federated Learning

# Non-IID Problem Leads to Performance Degradation

The data of users are usually non-IID, which leads to model performance degradation in Federated Learning.

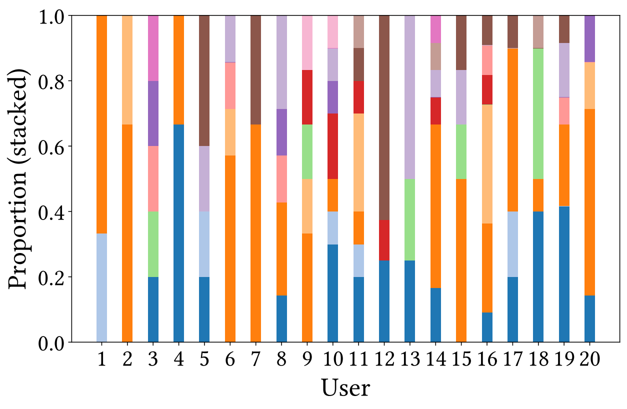



**Figure 3:** Category distribution of users' history news

# Model Interpolation

Model interpolation helps to solve this by interpolating the local personalized models with the global model.

$$
\begin{aligned}
\boldsymbol{w'}^{t}_{l_i} &= \lambda\, \boldsymbol{w}^{t-1}_{l_i} + (1-\lambda)\, \boldsymbol{w}^{t-1}_{g} \\
\boldsymbol{w}^{t}_{l_i} &= \boldsymbol{w'}^{t}_{l_i} - \eta\, \nabla \ell(\boldsymbol{w'}^{t}_{l_i}, d_i) \\
\boldsymbol{w}^{t}_{g} &= \boldsymbol{w}^{t-1}_{g} - \eta \sum_{i \in S_t} \frac{|d_i|}{\sum_{k \in S_t} |d_k|}\, \nabla \ell(\boldsymbol{w'}^{t}_{l_i}, d_i)
\end{aligned}
\tag{1}
$$

$\lambda \in [0, 1]$ is the interpolation coefficient, which controls how much the model is personalized.

# Model Interpolation

It's generally hard to determine the optimal interpolation coefficient $\lambda$.

$$\boldsymbol{w'}^{t}_{l_i} = \lambda \, \boldsymbol{w}^{t-1}_{l_i} + (1 - \lambda) \, \boldsymbol{w}^{t-1}_{g} \tag{2}$$

# Model Interpolation

It's generally hard to determine the optimal interpolation coefficient $\lambda$.

$$\boldsymbol{w'}^{t}_{l_i} = \lambda\, \boldsymbol{w}^{t-1}_{l_i} + (1-\lambda)\, \boldsymbol{w}^{t-1}_{g} \tag{2}$$

- Setting $\lambda$ per client by minimizing the empirical risk?

$$\begin{aligned} \lambda_i^t &= \arg\min_{\lambda}\, \ell(\boldsymbol{w'}^{t}_{l_i}, d_i) \\ &= \arg\min_{\lambda}\, \ell(\lambda\, \boldsymbol{w}^{t-1}_{l_i} + (1-\lambda)\, \boldsymbol{w}^{t-1}_{g}, d_i) \end{aligned} \tag{3}$$

$\Longrightarrow$ tremendously computational cost

# Model Interpolation

It's generally hard to determine the optimal interpolation coefficient $\lambda$.

$$\boldsymbol{w'}_{l_i}^{t} = \lambda\, \boldsymbol{w}_{l_i}^{t-1} + (1-\lambda)\, \boldsymbol{w}_g^{t-1} \tag{2}$$

- Setting $\lambda$ per client by minimizing the empirical risk?

$$\begin{aligned}\lambda_i^t &= \arg\min_{\lambda} \ell(\boldsymbol{w'}_{l_i}^{t}, d_i) \\ &= \arg\min_{\lambda} \ell(\lambda\, \boldsymbol{w}_{l_i}^{t-1} + (1-\lambda)\, \boldsymbol{w}_g^{t-1}, d_i)\end{aligned} \tag{3}$$

$\Longrightarrow$ tremendously computational cost

- Tuning $\lambda$ globally as a hyper-parameter? $\Longrightarrow$ non-optimal model performance

# Methodology

# Methodology

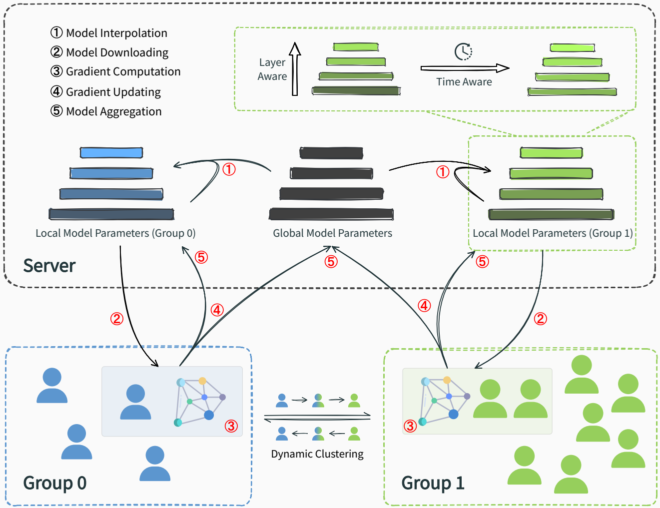



**Figure 4:** The framework of FINDING (Federated News Recommendation with **F**ine-grained **In**terpolation and **D**ynamic Cluster**ing**). It consists of two parts: 1. fine-grained model interpolation, 2. group-level personalization with dynamic user clustering.

# Methodology

## Fine-grained Model Interpolation

# Time-aware Interpolation

Recall that $\lambda \in [0, 1]$ controls how much the model is personalized.

$$\boldsymbol{w}'^{t}_{l_i} = \lambda\, \boldsymbol{w}^{t-1}_{l_i} + (1-\lambda)\, \boldsymbol{w}^{t-1}_{g} \tag{4}$$

Recall that $\lambda \in [0, 1]$ controls how much the model is personalized.

$$\boldsymbol{w'}^{t}_{l_i} = \lambda\, \boldsymbol{w}^{t-1}_{l_i} + (1-\lambda)\, \boldsymbol{w}^{t-1}_{g} \tag{4}$$

*Time-aware Interpolation*: the personalization coefficient $\lambda$ varies with the number of training rounds (the longer the training processes, the higher $\lambda$ should be).

$$\lambda \;\propto\; g(t) : [0, \infty) \rightarrow [0, 1] \tag{5}$$

## Time-aware Interpolation

Recall that $\lambda \in [0,1]$ controls how much the model is personalized.

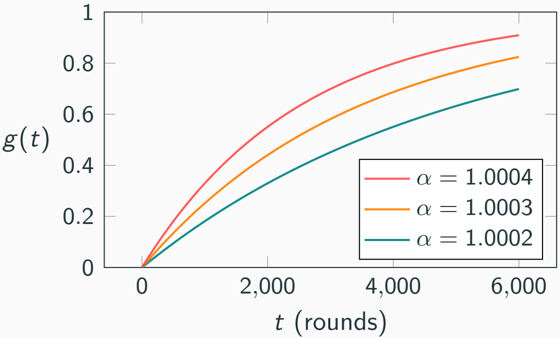
$$\boldsymbol{w'}^{t}_{l_i} = \lambda\, \boldsymbol{w}^{t-1}_{l_i} + (1-\lambda)\, \boldsymbol{w}^{t-1}_{g} \tag{4}$$

*Time-aware Interpolation*: the personalization coefficient $\lambda$ varies with the number of training rounds (the longer the training processes, the higher $\lambda$ should be).

$$\lambda \;\propto\; g(t) : [0,\infty) \to [0,1] \tag{5}$$

An example:

$$g(t) = 1 - \alpha^{-t} \quad (\alpha > 1) \tag{6}$$



# Layer-aware Interpolation

*Layer-aware Interpolation*: the personalization coefficient $\lambda$ depends on the layer depth (the shallower a layer is, the higher $\lambda$ should be).

$$\lambda \ \propto \ h(i) : [0, N-1] \rightarrow [0, 1] \tag{7}$$

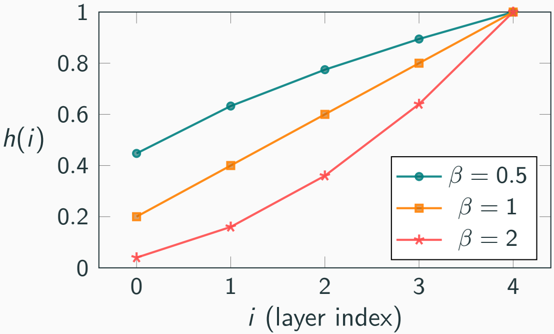
# Layer-aware Interpolation

*Layer-aware Interpolation*: the personalization coefficient $\lambda$ depends on the layer depth (the shallower a layer is, the higher $\lambda$ should be).

$$\lambda \propto h(i) : [0, N-1] \rightarrow [0, 1] \tag{7}$$

An example:

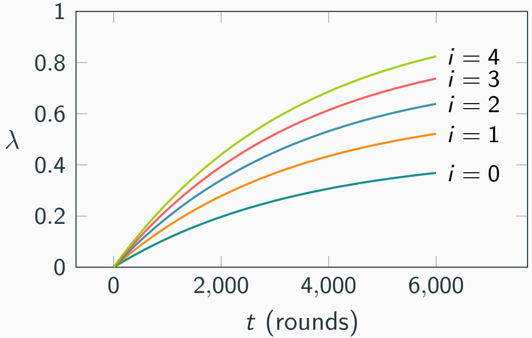
$$h(i) = (\frac{i+1}{N})^{\beta} \quad (\beta > 0) \tag{8}$$



# Fine-grained Model Interpolation

Integrating the two types of interpolation, we propose the *Fine-grained model Interpolation* strategy.

$$\lambda(t, i) = g(t)h(i) = (1 - \alpha^{-t})(\frac{i+1}{N})^{\beta} \quad (9)$$



# Methodology

## Group-level Personalization with Dynamic User Clustering

# Cold-user Problem

*cold* users $\Longrightarrow$ low-performance local models $\Longrightarrow$ limited gain from interpolation

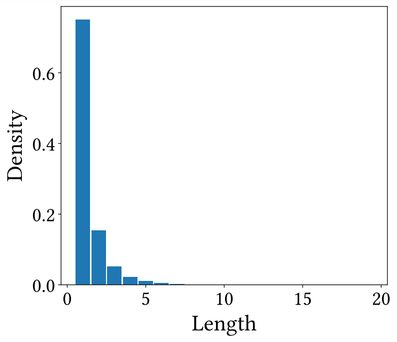



**Figure 5:** Length distribution of users' training samples

## Group-level Personalization with Dynamic User Clustering

initialize $\boldsymbol{w}_g^0 = \boldsymbol{w}_{l_0}^0 = \boldsymbol{w}_{l_1}^0, \dots, = \boldsymbol{w}_{l_{K-1}}^0$

$\boldsymbol{u}_0, \boldsymbol{u}_1, \cdots \leftarrow \text{InferUserVector}(\boldsymbol{w}_g^0, \{d_0, d_1, \dots\})$

$m \leftarrow \text{Cluster}(\boldsymbol{u}_0, \boldsymbol{u}_1, \dots)$ ▷ $m$ maps users to groups

**for** each round $t = 1, 2, \dots$ **do**

  **for** each group $i = 0, 1, \dots, K-1$ **do**

    ${\boldsymbol{w}'}_{l_i}^t \leftarrow \lambda\, \boldsymbol{w}_{l_i}^{t-1} + (1-\lambda)\, \boldsymbol{w}_g^{t-1}$ ▷ $\lambda$ from Eq. (9)

  **end for**

  $S_t \leftarrow$ (randomly select $C$ users)

  $\boldsymbol{w}_g^t \leftarrow \boldsymbol{w}_g^{t-1} - \eta \sum_{i \in S_t} \frac{|d_i|}{\sum_{k \in S_t} |d_k|} \nabla \ell({\boldsymbol{w}'}_{l_{m(i)}}^t, d_i)$

  **for** each group $i = 0, 1, \dots, K-1$ **do**

    $S_{t,i} \leftarrow \{j \in S_t \mid m(j) = i\}$

    $\boldsymbol{w}_{l_i}^t \leftarrow {\boldsymbol{w}'}_{l_i}^t - \eta \sum_{j \in S_{t,i}} \frac{|d_j|}{\sum_{k \in S_{t,i}} |d_k|} \nabla \ell({\boldsymbol{w}'}_{l_i}^t, d_j)$

  **end for**

  **if** $t \,\%\, T = 0$ **then** ▷ re-cluster periodically

    $\boldsymbol{u}_0, \boldsymbol{u}_1, \cdots \leftarrow \text{InferUserVector}(\boldsymbol{w}_g^t, \{d_0, d_1, \dots\})$

    $m \leftarrow \text{Cluster}(\boldsymbol{u}_0, \boldsymbol{u}_1, \dots)$

    $\boldsymbol{w}_{l_0}^t, \boldsymbol{w}_{l_1}^t, \dots, \boldsymbol{w}_{l_{K-1}}^t \leftarrow$ (reinitialize, see the paper for details)

  **end if**

**end for**

# Experiments

## Baseline Models

- **Centralized** denotes the plain centralized training method.
- **Vanilla FL** is the vanilla adaptation of federated learning to news recommendation tasks.
- **FedProx** addresses the heterogeneity issue with a proximal term that adjusts local model updates.
- **FedPer** trains the base layers of a deep model centrally, while the top layers (i.e., the personalization layers) are trained locally.
- **SCAFFOLD** proposes to tackle the client drift problem in federated learning with control variates.
- **pFedMe** makes use of the Moreau envelope function which helps decompose the personalized model optimization from global model learning.
- **CFL** iteratively splits the users into groups based on the similarity of the gradient updates.

# Performance Comparison

**Table 1:** Results of different methods on two datasets (in percent)

| | | Adressa | | | | MIND | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | AUC | MRR | nDCG@5 | nDCG@10 | AUC | MRR | nDCG@5 | nDCG@10 |
| NRMS | Centralized | **72.67** | **29.39** | 35.66 | 41.16 | 66.11 | **31.59** | **34.76** | 41.00 |
| | Vanilla FL | 71.13 | 26.10 | 32.03 | 37.49 | 65.04 | 30.78 | 33.58 | 40.04 |
| | FedProx | 71.25 | 27.30 | 32.55 | 38.33 | 65.14 | 30.49 | 33.42 | 39.75 |
| | FedPer | 71.39 | 27.64 | 34.02 | 39.17 | 65.43 | 31.03 | 34.06 | 40.41 |
| | SCAFFOLD | 71.50 | 27.66 | 34.59 | 39.28 | 65.48 | 30.81 | 33.95 | 40.26 |
| | pFedMe | 71.73 | 27.83 | 34.32 | 40.17 | 65.27 | 30.73 | 33.56 | 40.19 |
| | CFL | 71.60 | 27.79 | 34.62 | 40.04 | 65.32 | 30.92 | 33.80 | 40.39 |
| | FINDING | 72.51 | 28.89 | **35.81** | **41.28** | **66.14** | 31.30 | 34.62 | **41.03** |
| NAML | Centralized | **80.44** | **33.79** | **42.16** | 47.93 | 67.17 | **31.88** | **35.30** | 41.60 |
| | Vanilla FL | 78.71 | 32.84 | 41.04 | 46.75 | 66.01 | 30.96 | 34.38 | 40.70 |
| | FedProx | 78.69 | 33.26 | 41.74 | 47.01 | 66.15 | 31.16 | 34.41 | 40.66 |
| | FedPer | 79.01 | 33.11 | 41.88 | 47.43 | 66.78 | 31.56 | 34.92 | 41.02 |
| | SCAFFOLD | 79.44 | 33.22 | 41.34 | 47.15 | 66.42 | 31.37 | 34.69 | 40.94 |
| | pFedMe | 79.17 | 32.98 | 41.73 | 47.68 | 66.16 | 31.41 | 34.28 | 40.57 |
| | CFL | 79.44 | 33.12 | 41.60 | 47.58 | 66.23 | 31.25 | 34.50 | 40.94 |
| | FINDING | 80.35 | 33.59 | 42.13 | **48.06** | **67.26** | 31.85 | 35.19 | **41.64** |

# Conclusion

# Conclusion

**Problem**

How to address the data heterogeneity issue, namely the non-IID data problem, in federated news recommendation tasks?

# Conclusion

## Problem

How to address the data heterogeneity issue, namely the non-IID data problem, in federated news recommendation tasks?

## Solution

FINDING: Federated News Recommendation with **F**ine-grained **In**terpolation and **D**ynamic Cluster**ing**

1. Fine-grained model interpolation
    - Time-aware interpolation
    - Layer-aware interpolation
2. Group-level personalization with dynamic user clustering