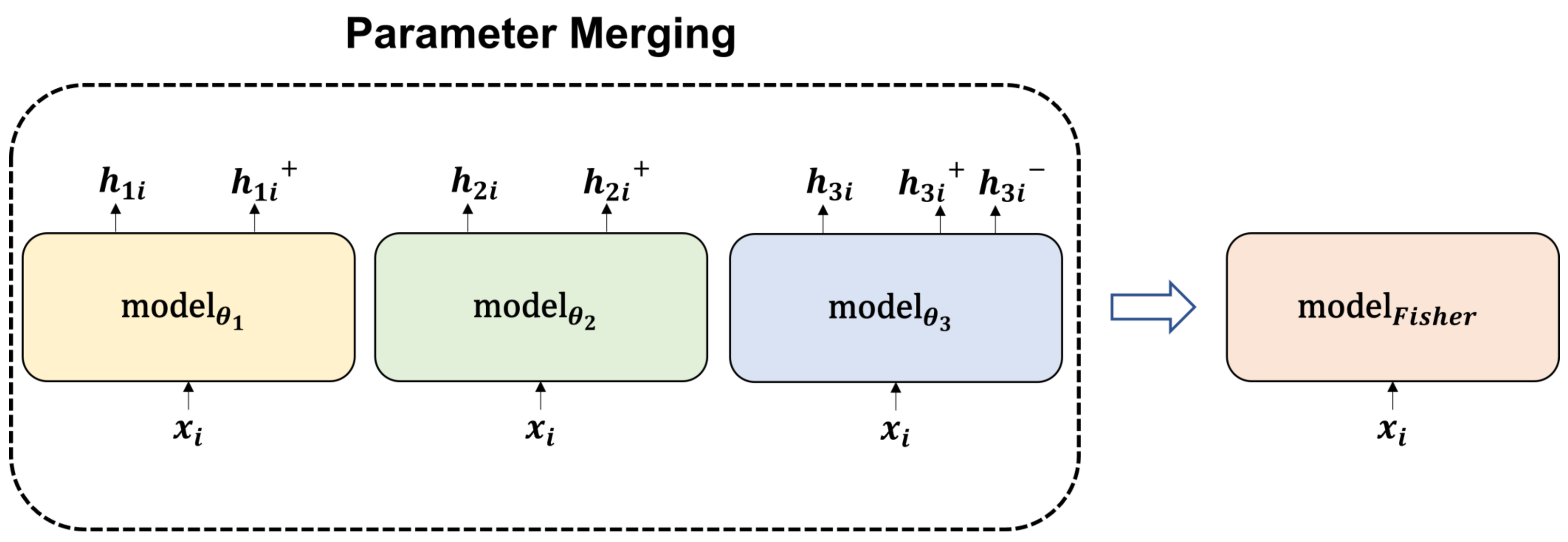
# Fisher-Weighted Merge of Contrastive Learning Models in Sequential Recommendation

Jung Hyun Ryu*, Jaeheyoung Jeon *, Jewoong Cho *, Myungjoo Kang

## Summary

We introduce a novel **ensemble technique for sequential models by applying Fisher merging in sequential recommendation systems.**

This approach ensures robust fine-tuning by merging the parameters of multiple models, resulting in improved overall performance.




## Background

*Task* Sequential recommendation aims to capture evolving user preferences over time. For user $u_i \in \mathcal{U}$ and item $v_i \in \mathcal{V}$, consider a chronological order of user-item interaction $s_i = \left[v_1^{u_i}, v_2^{u_i}, \cdots, v_t^{u_i}, \cdots, v_{n_{u_i}}^{u_i}\right]$. Our task is to predict the item with which user is interact in the next time step; $p(v_{n_{u_i}+1}^{u_i} = v \mid s_i)$ .

*Preliminaries* Perform model ensemble based on different types of loss.

Interpret the process of finding $\theta^*$ as maximizing the joint likelihood of the posteriors; $\sum_{\mathrm{m}} \log p(\theta|\theta_{\mathrm{m}})$.

(1) Assuming $p(\theta|\theta_{\mathrm{m}})$ follows Gaussian distribution $\mathcal{N}(\hat{\theta}_{\mathrm{m}}, I)$,

$$\theta^* = \operatorname{argmax}_\theta \sum_m \lambda_m \log p(\theta|\theta_m, I) = \frac{1}{M}\sum_m \theta_m.$$

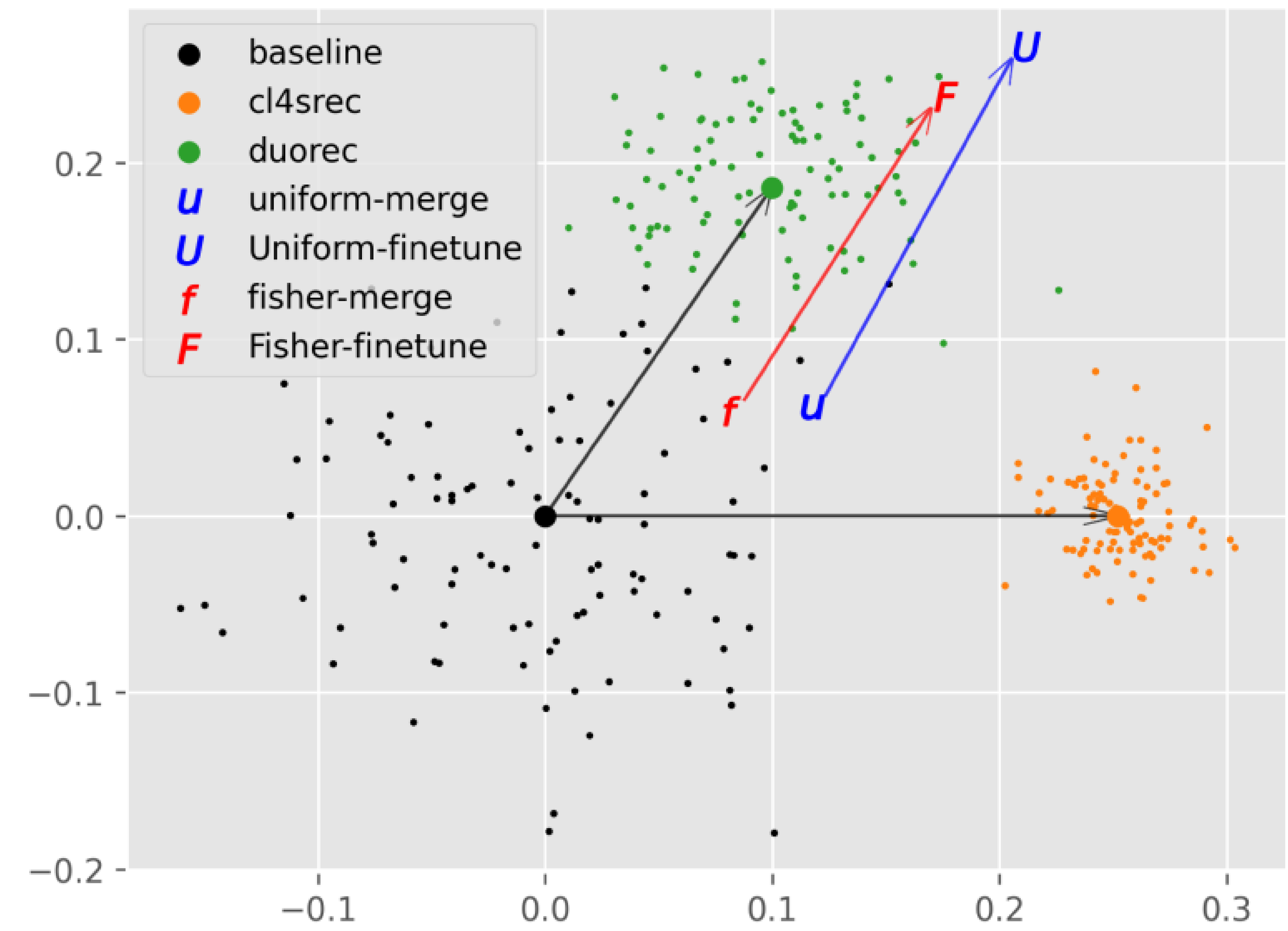
(2) Assume $p(\theta|\theta_{\mathrm{m}})$ follow a Gaussian distribution $\mathcal{N}(\hat{\theta}_{\mathrm{m}}, H^{-1})$, where $H$ corresponds to Hessian matrix of $\theta_{\mathrm{m}}$ obtained through the second-order Taylor expansion at the mode of the posterior(Fisher Matrix).

$$\theta^* = \operatorname{argmax}_\theta \sum_m \lambda_m \log p(\theta|\theta_m, F_{\mathrm{m}})$$

where $F_m = E_x E_{y \sim p_\theta(y|x)} \nabla_\theta \log p_\theta(v|s_i)\, \nabla_\theta \log p_\theta(v|s_i)^T$.

(closed form) $\theta^{*(j)} = \dfrac{\sum_m \lambda_m F_m^{(j)} \theta_m^{(j)}}{\sum_m \lambda_m F_m^{(j)}}$

## Visual Result




## Method

*Applying Model Ensemble*

$$\mathbb{E}_{s_i}\mathbb{E}_{v \sim p_\theta(v|s_i)}(\nabla_\theta \log p_\theta(v|s_i))^2$$
$$= \frac{1}{|\mathcal{U}|}\sum_i^{|\mathcal{U}|}\sum_j^{|\mathcal{V}|} p_\theta(v_j|s_i)\,(\nabla_\theta \log p_\theta(v_j|s_i))^2$$

To alleviate the computational burden, we employ batch-wise and sampling-based approach within the methodology.

*Batch-wise Computation* $\frac{1}{\mathrm{BS}_k}\sum_j^{|\mathcal{V}|}\left(\sum_i^{\mathrm{BS}_k} p_\theta(v_j|s_i)\right)\left(\nabla_\theta \sum_i^{\mathrm{BS}_k} \log p_\theta(v_j|s_i)\right)^2$

*How to Sample Items*

✓ Random Sampling : Randomly sample $j$ from the total number of items.

✓ Top-k Sampling : Selected a set of n items that are most likely to be of interest to the corresponding user, i.e., $p_\theta(v_j|s_i)$.

$$\sum_j^{\text{top}-k} p_\theta(v_j|\mathrm{s})\,(\nabla_\theta \log p_\theta(v_j|\mathrm{s}))^2$$

✓ Model-based Sampling : Randomly sample a subset of items $v_j$ based on the conditional probability $p_\theta(v_j|s_i)$.

$$\frac{1}{N}\sum_{v_j \sim p_\theta(v_j|s)}^{N} (\nabla_\theta \log p_\theta(v_j|s))^2$$

✓ Calculate with target item : Compute Fisher matrix based on the target item, disregarding other items with direct relevance.

$$p_\theta(v_j^*|\mathrm{s})\,(\nabla_\theta \log p_\theta(v_j^*|\mathrm{s}))^2$$

## Experimental Result

| | SAMPLE SIZE | FULL NDCG10 | FULL NDCG20 | RANDOM NDCG10 | RANDOM NDCG20 | POPULAR NDCG10 | POPULAR NDCG20 |
|---|---|---|---|---|---|---|---|
| BASELINE | | 0.135 | 0.1601 | 0.5573 | 0.5786 | 0.0426 | 0.0706 |
| CL4SREC | | 0.0585 | 0.0751 | 0.0513 | 0.043 | **0.0466** | 0.0701 |
| DUOREC (SUP.) | | 0.1346 | 0.1591 | 0.5547 | 0.58 | 0.0454 | 0.068 |
| DUOREC (UNSUP.) | | 0.1358 | 0.1609 | 0.5594 | 0.5782 | 0.0445 | 0.0742 |
| DUOREC (SUP.&UNSUP.) | | 0.1351 | 0.1599 | 0.554 | 0.5732 | 0.0423 | 0.0724 |
| RANDOM SAMPLING | 10 | 0.1379 | <u>0.1638</u> | 0.5606 | <u>0.5825</u> | 0.0457 | 0.0691 |
| | 30 | 0.1366 | 0.1624 | 0.5584 | 0.58 | 0.0477 | **0.0726** |
| | 50 | <u>0.1386</u> | 0.1636 | 0.5598 | 0.5813 | 0.0419 | 0.0419 |
| TOP-K SAMPLING | 10 | 0.1364 | 0.1624 | 0.5602 | 0.5817 | 0.0446 | 0.0689 |
| | 30 | 0.1373 | 0.1616 | **0.5637** | **0.5835** | 0.0457 | 0.0708 |
| | 50 | **0.1387** | 0.1635 | 0.5592 | 0.5807 | 0.0424 | 0.0672 |
| MODEL-BASED SAMPLING | 10 | 0.1358 | 0.1619 | 0.5564 | 0.5782 | 0.044 | 0.0696 |
| | 30 | 0.1385 | **0.1646** | 0.5579 | 0.5784 | 0.0446 | 0.0689 |
| | 50 | 0.138 | 0.1632 | 0.5605 | 0.5814 | <u>0.0465</u> | 0.0719 |
| CALCULATE ON TARGET ITEM | 1 | <u>0.1386</u> | 0.1628 | <u>0.5618</u> | 0.5806 | 0.0428 | <u>0.0725</u> |