AdaptiveRec : Adaptively Construct Pairs for Contrastive Learning in Sequential Recommendation

ICML International Conference On Machine Learning

Jaeheyoung Jeon*, Jung Hyun Ryu*, Jewoong Cho*, Myungjoo Kang
Contact : jan4021@snu.ac.kr

SEOUL NATIONAL UNIVERSITY

Summary

- We addresses the **false negative** issue in Sequential Recommendation System which is inevitable problem in Contrastive Learning.
- By introducing an advanced **adaptive** approach to contrastive learning, the proposed method improves the quality of item embeddings and performance in a variety of metrics.

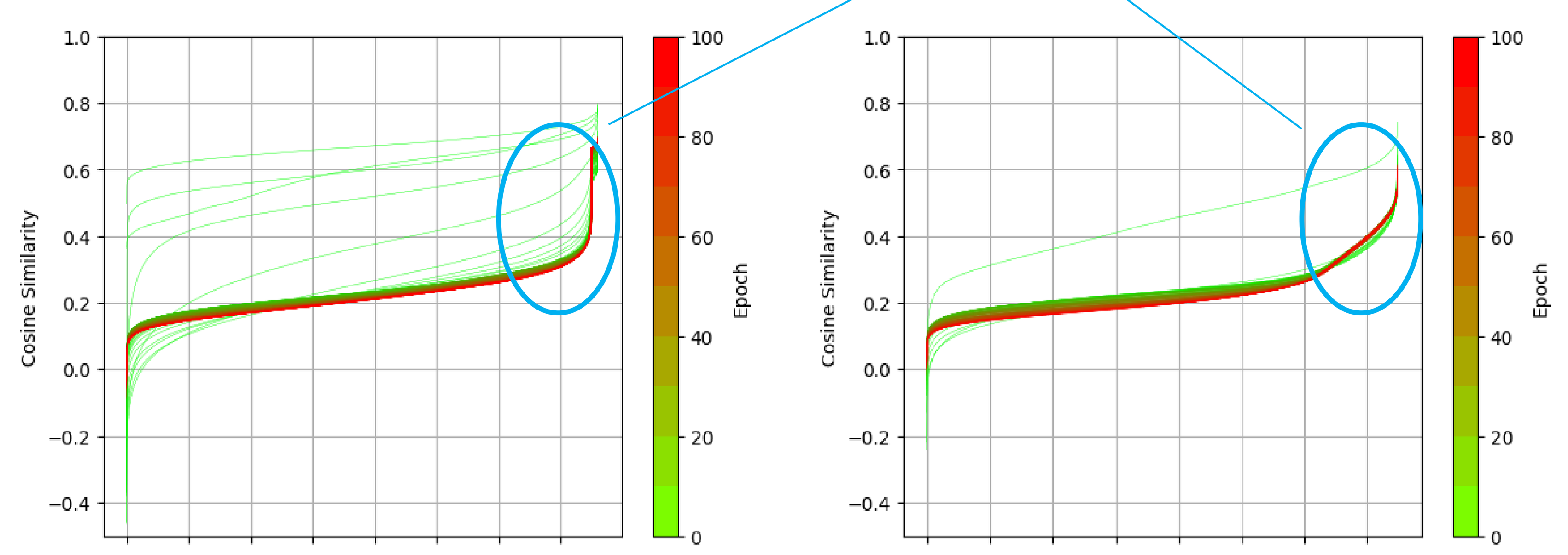
Background

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; $p(v_{n_{u_i}+1}^{u_i} = v \mid s_i)$.

Intuition

- We have observed that as the model continues to learn, it consistently produces high similarity values more clearly.
- Model already knows the positive pairs.

Model produce high similarity values



Method

| Existing Method and Limitation |

Augmented Batch (size : 2B-2)

Batch Size = B

0.1	0.3	0.5	0.7
0.1	0.2	0.3	0.4
0.1	0.7	0.8	0.9

calculated cosine similarity of feature pairs

0.6
0.25
0.95

learnable threshold (k_{learn}) for similarity

- Contrastive Loss

$$\mathcal{L}_{NCE} = \sum_{i \in I}\left[-\log \frac{e^{sim(f(s_i), f(s_i^+))}}{\sum_{j \in M(i)} e^{sim(f(s_i), f(s_j))}}\right] \qquad M(i) = \{i^+, j \mid j \in N(i)\}$$

$$N(i) = \text{negative pairs of } i$$

- Construct Negative Pairs

With batch size B and data augmentation there are $2B$ data. Each data has $(2B-2)$ pairs excluding itself and the augmented with itself.

- False Negative Problem

The existing method treats all pairs in the learning process as negative pairs. However, this approach inevitably results in false negative pairs, <u>where items that are semantically similar are considered different.</u>

| Proposed Method |

0.1	0.3	0.5	0.7
0.1	0.2	0.3	0.4
0.1	0.7	0.8	0.9

: considered as **negative** pairs

: considered as **positive** pairs

- Contrastive Loss

$$\mathcal{L}_{learn} = \sum_{i=1}^{2B}\left[-\log \frac{\frac{1}{2} e^{sim(f(s_i), f(s_i^+))}}{\sum_{j \in M^-(i) \cup \{i^+\}} e^{sim(f(s_i), f(s_j))}} + \sum_{j^+ \in M^+(i)} -\log \frac{\frac{1}{2|M^+(i)|} e^{sim(f(s_i), f(s_{j^+}))}}{\sum_{j \in M^-(i) \cup \{i^+\}} e^{sim(f(s_i), f(s_j))}}\right]$$

$$M^-(i) = \left\{j \in N(i) \mid sim\left(f(s_i), f(s_j)\right) \le k_{learn}[i]\right\} \qquad \mathcal{L}_{\mathrm{cl}} = \mathcal{L}_{\mathrm{learn}} + \lambda \mathcal{L}_{\mathrm{reg}}$$

$$M^+(i) = \left\{j \in N(i) \mid sim\left(f(s_i), f(s_j)\right) \ge k_{learn}[i]\right\} \qquad \mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{basic}} + \lambda_{\mathrm{cl}} \mathcal{L}_{\mathrm{cl}}$$

- Adaptive Threshold

Set **learnable threshold** (k_{learn}) as small model's output to deal with changing composition in every epoch.

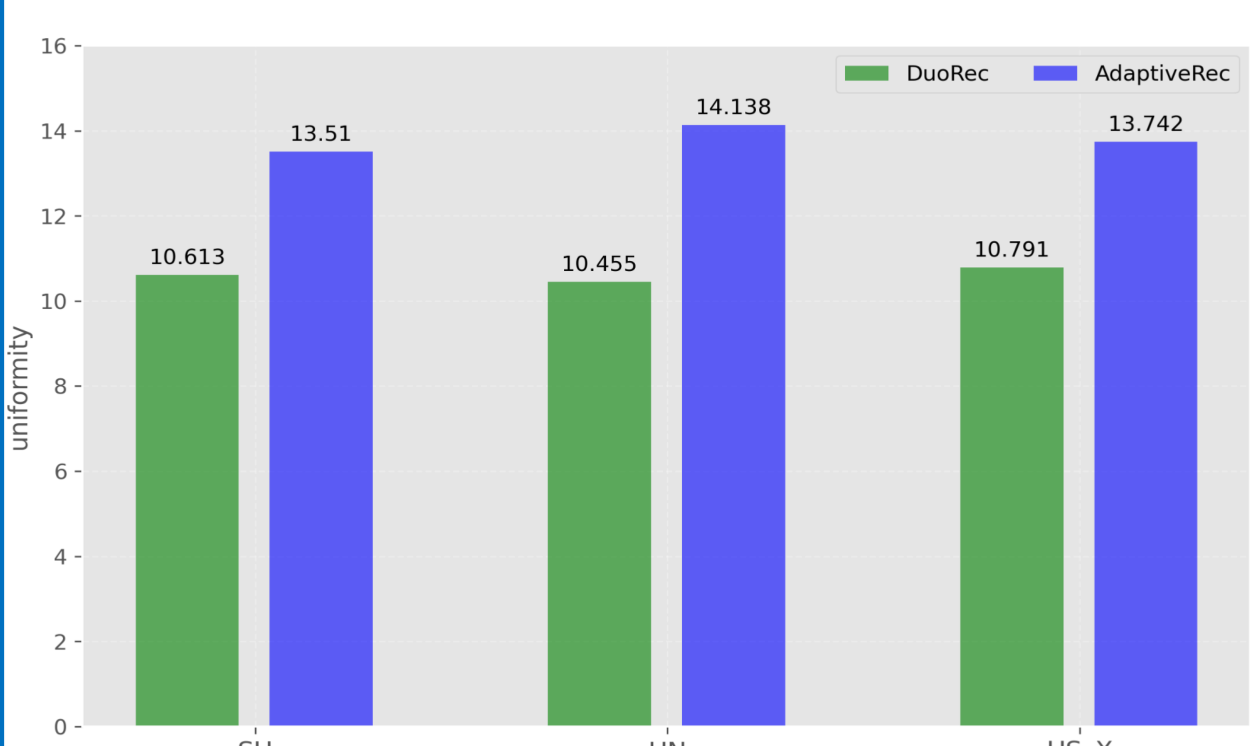
- Positive Pair Construction

Consider the pair having higher values than the learnable threshold (k_{learn}) as positive pair.

Result

(R) Performance Comparison

Our model achieve improvements in many metrics compared to the previous model.



Dataset	ML-1M						Amazon Beauty					
Metric	NDCG@ 5	NDCG@ 10	MRR@ 5	MRR@ 10	Recall@ 5	Recall@ 10	NDCG@ 5	NDCG@ 10	MRR@ 5	MRR@ 10	Recall@ 5	Recall@ 10
CL4SRec	0.0726	0.0955	0.0571	0.0665	0.12	0.1916						
DuoRec	0.1074	0.1348	0.089	0.0996	0.1632	0.2485	0.0329	0.0421	0.0252	0.0289	0.0564	0.0852
AdaptiveRec (ours)	**0.1135**	**0.1411**	**0.0945**	**0.1058**	**0.1738**	**0.2611**	**0.0335**	**0.0427**	**0.0257**	**0.0295**	**0.0574**	**0.0857**
Improv.	5.68%	4.67%	6.12%	6.22%	6.5%	5.07%	1.82%	1.43%	1.98%	2.08%	1.78%	0.59%

	Whole			Popular			Random		
	SU	UN	US_X	SU	UN	US_X	SU	UN	US_X
DuoRec	0.1386	0.1357	0.1389	0.0405	0.0442	0.0444	0.5621	0.5641	0.5668
AdaptiveRec (ours)	0.1367	0.1396	**0.1410**	0.0444	0.0433	**0.0463**	**0.5680**	0.5675	0.5643

(L) Uniformity

Our new approach has significantly increased uniformity, which is a metric that measures item embedding.

$$\ell_{\text{uniform}} \triangleq \log \mathbb{E}_{(x,y)\sim p_{\text{data}}} e^{-2\|f(x)-f(y)\|^2}$$