Method

CT4Rec: Simple yet Effective Consistency Training for Sequential Recommendation

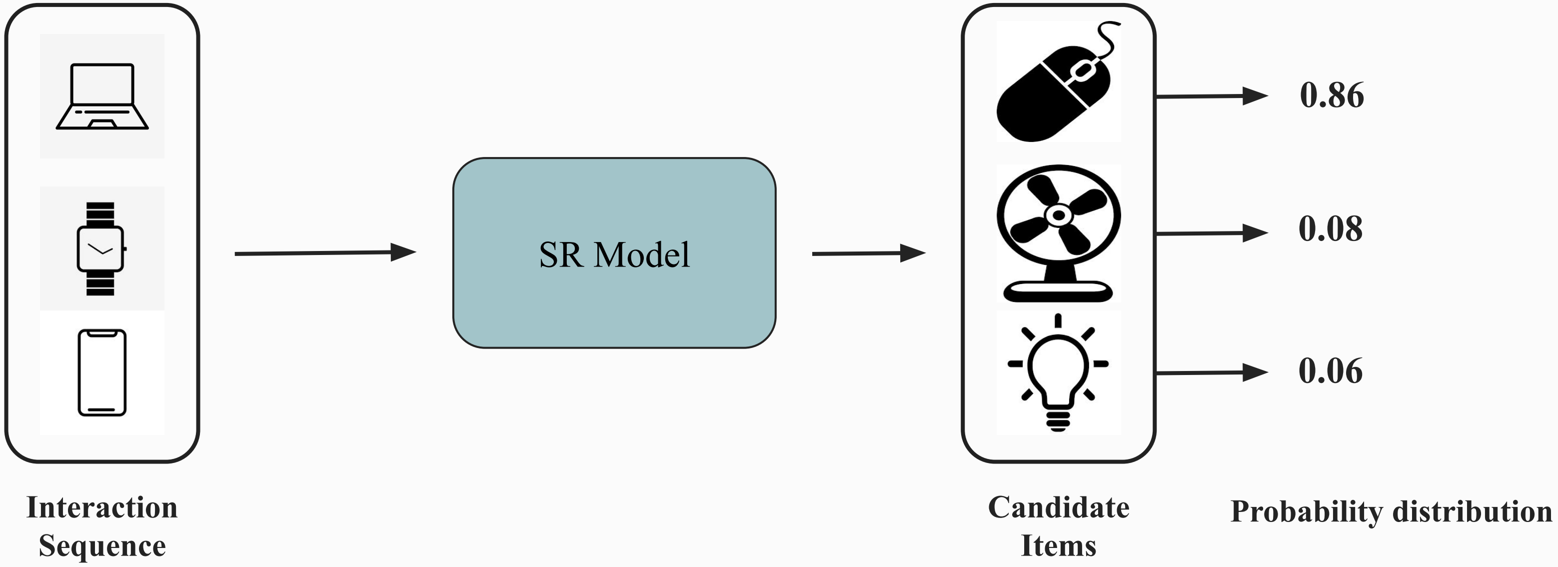
Task

Advisor : Jia-Ling, Koh
Speaker : Siao You, Chen
Source : KDD 2023
Date : 2023/12/05

Outline

- Introduction
- Related Work
- Method
- Experiment
- Conclusion

Sequential Recommendation

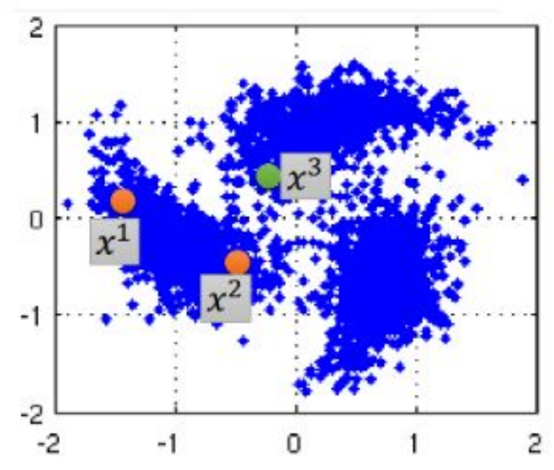


Consistency Training - Smoothness Assumption

- Similar $\boldsymbol{x}$ has the same ŷ

- x^1 and x^2 have the same label

  x^1 and x^3 have different labels

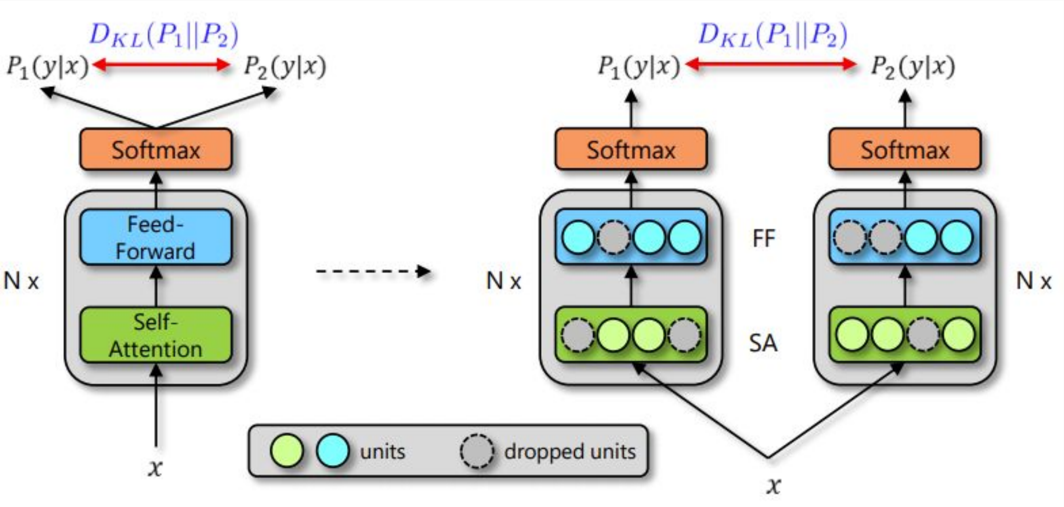


Outline

- Introduction
- Related Work
- Method
- Experiment
- Conclusion

R-Drop

- Regularization techniques are indispensable modules to prevent model overfitting.
- The Dropout technology has become the most widely used regularization technology.
- The randomness introduced by Dropout causes non negligible inconsistency.



Outline

- Introduction
- Related Work
- Method
- Experiment
- Conclusion

Notations

$\mathcal{U} = (u_1, u_2, ..., u_{|\mathcal{U}|})$ denote a set of users

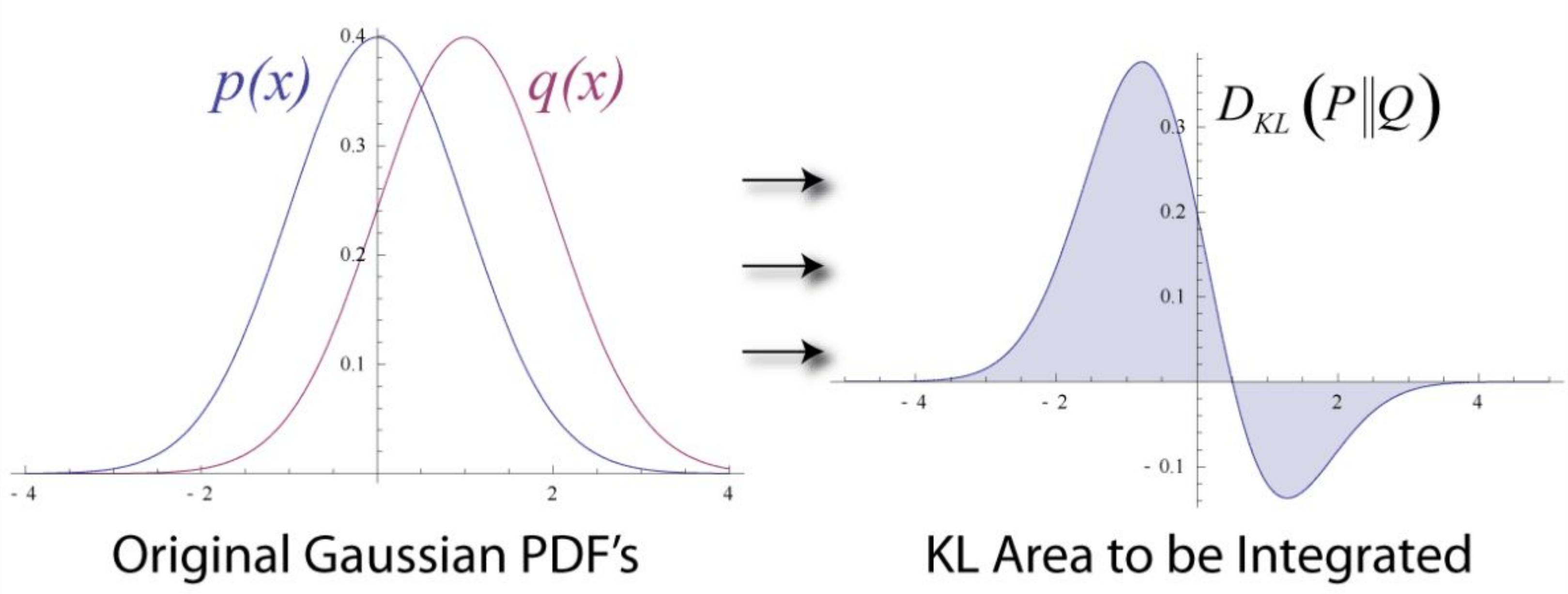
$\mathcal{V} = (v_1, v_2, ..., v_{|\mathcal{V}|})$ denote a set of items

The sequence for user $u \in \mathcal{U}$ is denoted as $s_u = (v_1^{(u)}, v_2^{(u)}, ..., v_t^{(u)}, ..., v_{|s_u|}^{(u)})$

The task of SR is to predict the probability of all alternative items to be interacted by user *u* at time step $|s_u| + 1$

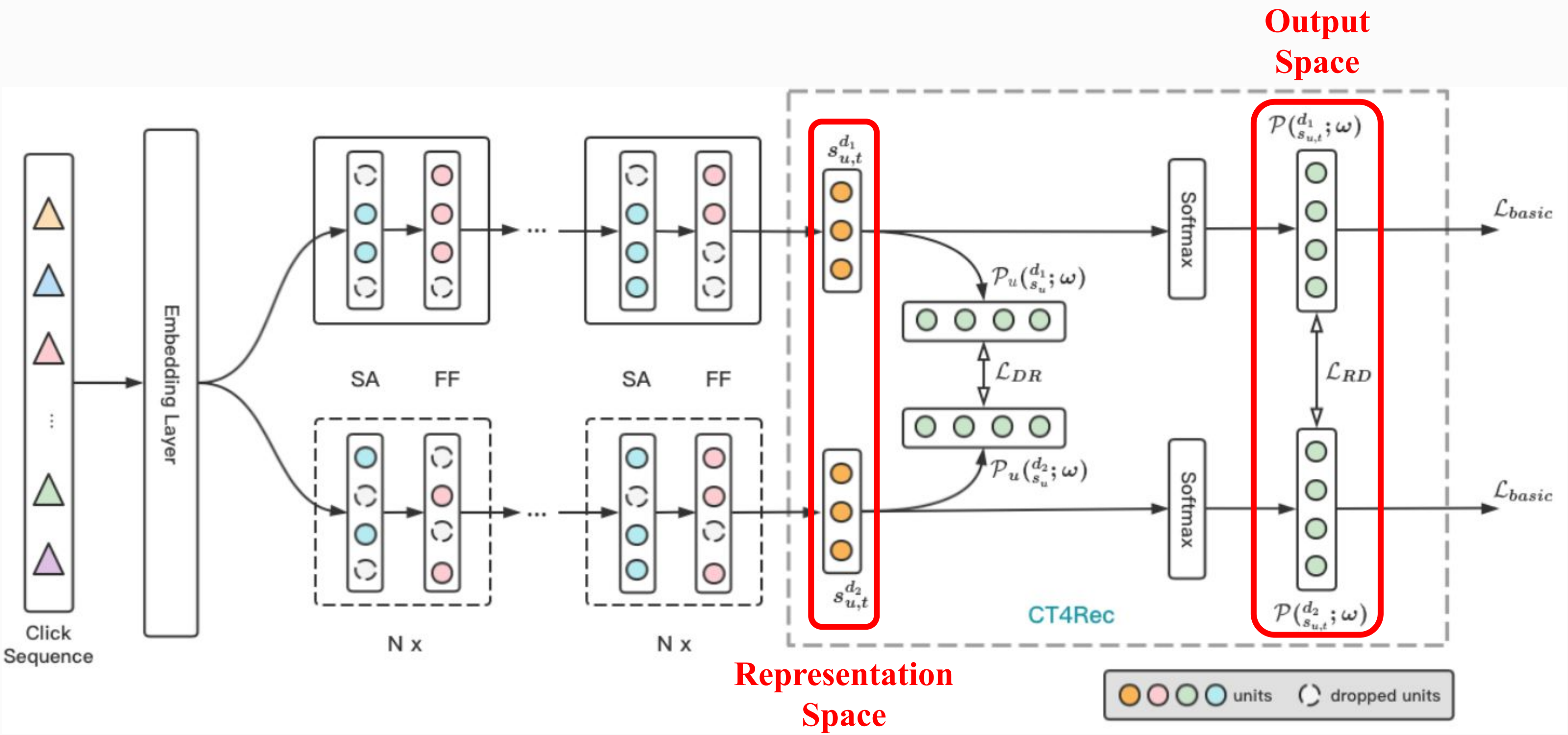
$$P(v_{|s_u|+1}^{(u)} = v | s_u)$$

Kullback-Leibler (KL) Divergence



$$D_{KL}(P(x)\,||\,Q(x)) = \sum_{x \in X} P(x) \cdot \log\left(\frac{P(x)}{Q(x)}\right)$$

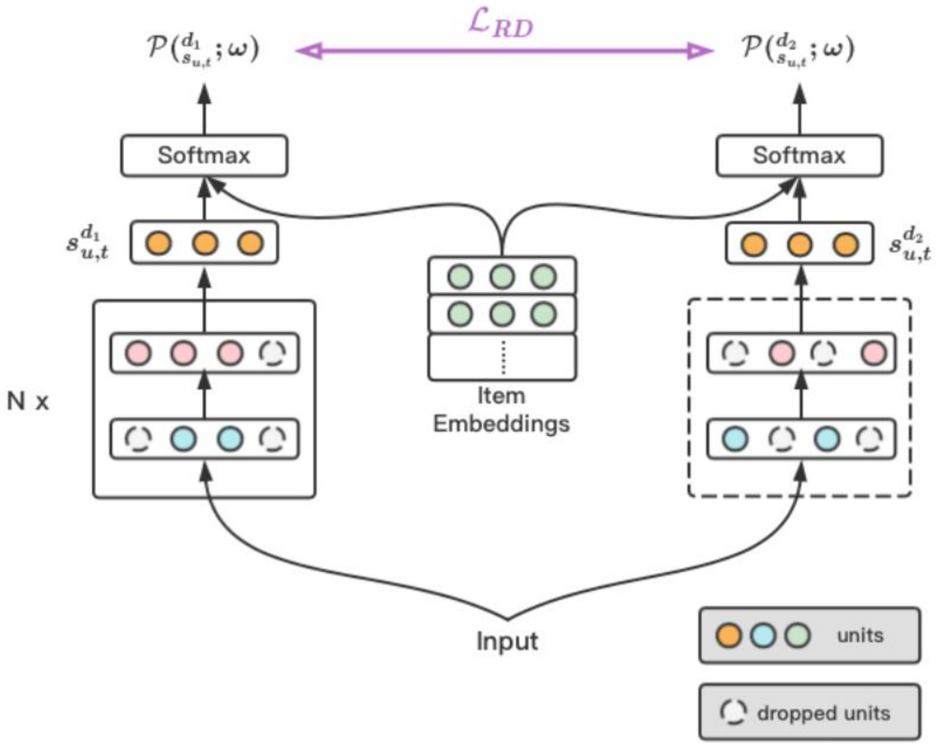
CT4Rec Model Structure



CT4Rec - Regularized Dropout (RD) Loss

- Constrain the output space from dropout

$$\mathcal{L}_{RD}(\boldsymbol{s}_{\boldsymbol{u},\boldsymbol{t}};\omega) = \frac{1}{2}(\mathcal{D}_{KL}(\mathcal{P}(\boldsymbol{s}^{\boldsymbol{d}_1}_{\boldsymbol{u},\boldsymbol{t}};\omega)||\mathcal{P}(\boldsymbol{s}^{\boldsymbol{d}_2}_{\boldsymbol{u},\boldsymbol{t}};\omega)) + \mathcal{D}_{KL}(\mathcal{P}(\boldsymbol{s}^{\boldsymbol{d}_2}_{\boldsymbol{u},\boldsymbol{t}};\omega)||\mathcal{P}(\boldsymbol{s}^{\boldsymbol{d}_1}_{\boldsymbol{u},\boldsymbol{t}};\omega))) \quad (3)$$

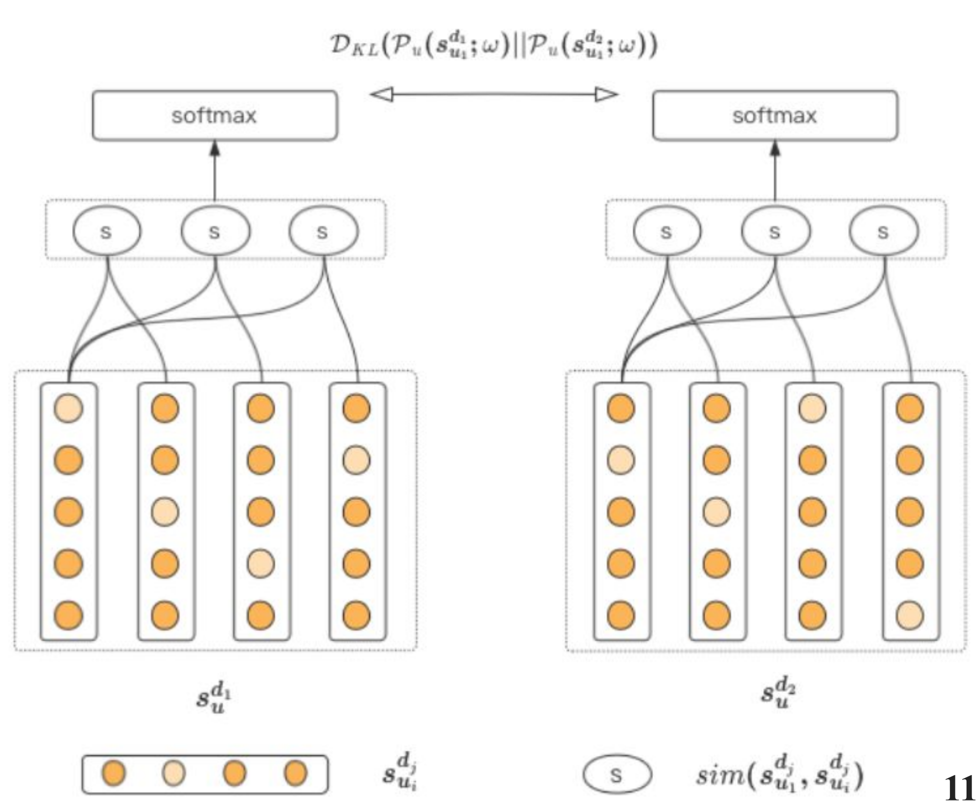


CT4Rec - Distributed Regularization (DR) Loss

- Regularize the representation space

$$\mathcal{P}_u(\boldsymbol{s}_{\boldsymbol{u}_1}^{\boldsymbol{d}_j};\omega) = softmax(sim(\boldsymbol{s}_{\boldsymbol{u}_1}^{\boldsymbol{d}_j}, \boldsymbol{s}_{\boldsymbol{u}_2}^{\boldsymbol{d}_j}), ..., sim(\boldsymbol{s}_{\boldsymbol{u}_1}^{\boldsymbol{d}_j}, \boldsymbol{s}_{\boldsymbol{u}_n}^{\boldsymbol{d}_j})) \tag{4}$$

$$\begin{aligned}\mathcal{L}_{DR}(\boldsymbol{s}_{\boldsymbol{u}};\omega) = \frac{1}{2}(&\mathcal{D}_{KL}(\mathcal{P}_u(\boldsymbol{s}_{\boldsymbol{u}}^{\boldsymbol{d}_1};\omega)||\mathcal{P}_u(\boldsymbol{s}_{\boldsymbol{u}}^{\boldsymbol{d}_2};\omega)) \\ &+\mathcal{D}_{KL}(\mathcal{P}_u(\boldsymbol{s}_{\boldsymbol{u}}^{\boldsymbol{d}_2};\omega)||\mathcal{P}_u(\boldsymbol{s}_{\boldsymbol{u}}^{\boldsymbol{d}_1};\omega)))\end{aligned} \tag{5}$$



CT4Rec - Basic & Final Loss

Backbone: SASRec*

$$\mathcal{P}(s_{u,t};\omega) = \frac{exp(s_{u,t}v^{+}_{t+1})}{exp(s_{u,t}v^{+}_{t+1}) + \sum_{v^{-}_{t+1}\in\mathcal{V}} exp(s_{u,t}v^{-}_{t+1})} \quad (1)$$

$$\mathcal{L}_{basic}(s_{u,t};\omega) = -log\mathcal{P}(s_{u,t};\omega) \quad (2)$$

CT4Rec

$$\mathcal{L}_{basic}(s_{u,t};\omega) = -\frac{1}{2}(log\mathcal{P}(s^{d_1}_{u,t};\omega) + log\mathcal{P}(s^{d_2}_{u,t};\omega)) \quad (6)$$

$$\mathcal{L}_{final} = \mathcal{L}_{basic} + \alpha\mathcal{L}_{RD} + \beta\mathcal{L}_{DR} \quad (7)$$

Outline

- Introduction
- Related Work
- Method
- Experiment
- Conclusion

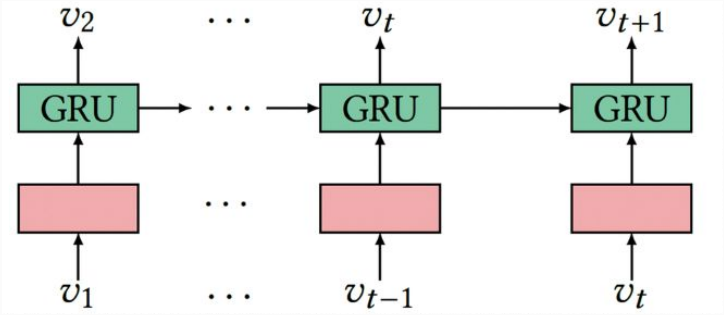
Datasets

Average action per user

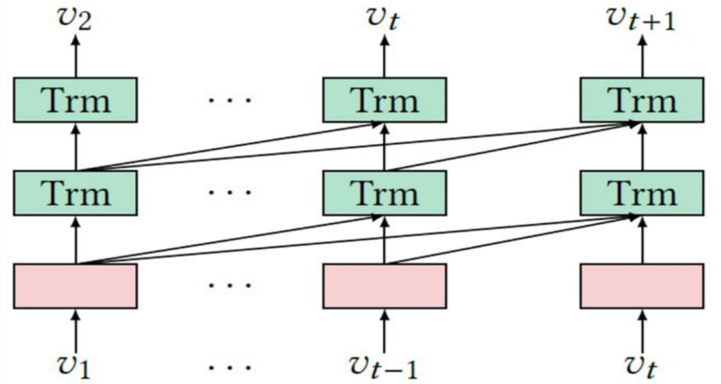
Dataset	#users	#items	#actions	avg.	density
Beauty	52,024	57,289	0.4M	7.6	0.01%
Sports	25,598	18,357	0.3M	8.3	0.05%
Yelp	30,431	20,033	0.3M	10.4	0.05%
WeChat	749,452	211,004	9.5M	12.7	0.006%

- Beauty & Sports: subsets of Amazon dataset's beauty and sports categories
- Yelp: business recommendation from the Yelp platform
- WeChat: new large-scale dataset collected from PC-WeChat Top Stories

Baselines

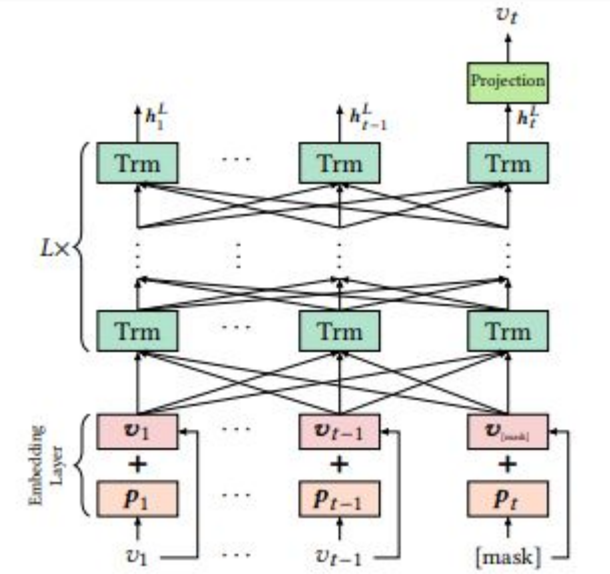


GRU4Rec

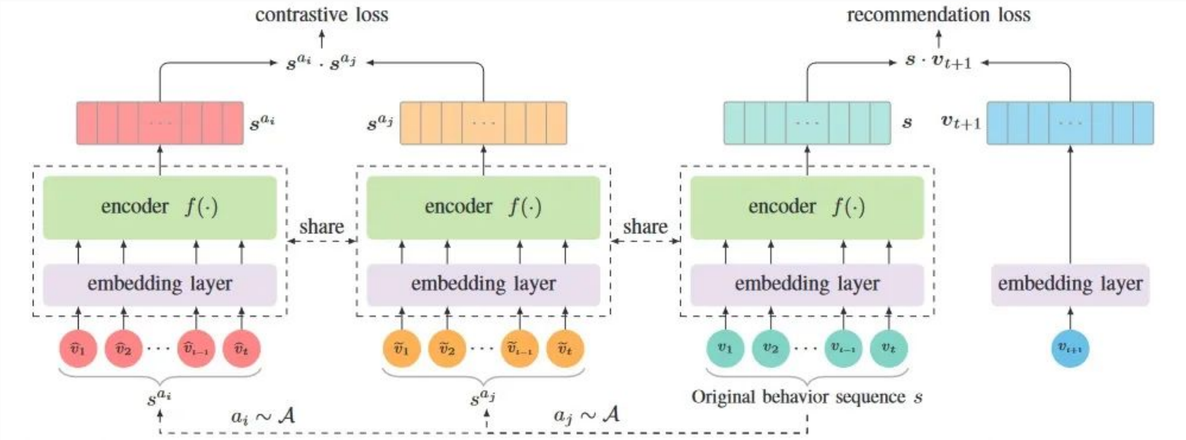


SASRec

Baselines

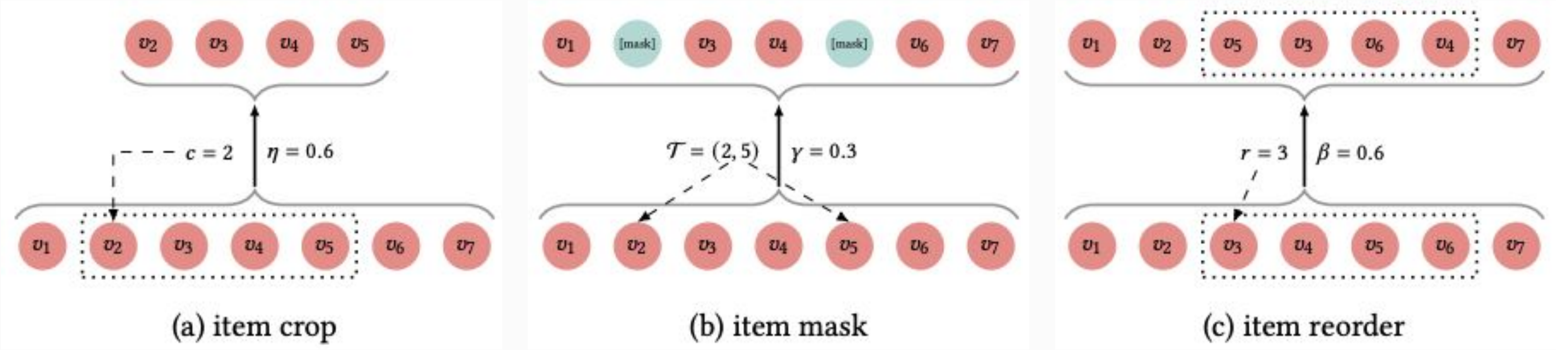


BERT4Rec



CL4SRec

Data Augmentation



Evaluation Metric

- NDCG@k

$$NDCG = \frac{1}{N} \sum \frac{1}{\log\left(p_i + 1\right)}$$

- HR@k

$$HR = \frac{1}{N} \sum hits(i)$$

Experimental Result

Datasets	Metric	SASRec (Backbone)	SASRec*	GRU4Rec (RNN)	BERT4Rec (Attention)	TiSASRec (Attention)	CL4SRec (Contrastive)	CT4Rec
Beauty	HR@5	0.2109	0.2194	0.1179	0.0860	0.2024	0.2275	**0.2556**
	HR@10	0.2759	0.2748	0.1582	0.1357	0.2746	0.2896	**0.3200**
	HR@20	0.3546	0.3392	0.2146	0.2036	0.3508	0.3624	**0.3891**
	NDCG@5	0.1523	0.1661	0.0869	0.0572	0.1413	0.1701	**0.1924**
	NDCG@10	0.1733	0.1840	0.1000	0.0732	0.1646	0.1901	**0.2132**
	NDCG@20	0.1932	0.2003	0.1141	0.0903	0.1839	0.2085	**0.2307**
Sports	HR@5	0.1912	0.1966	0.0961	0.0766	0.1703	0.2084	**0.2196**
	HR@10	0.2747	0.2683	0.1499	0.1267	0.2456	0.2834	**0.3010**
	HR@20	0.3751	0.3547	0.2295	0.2055	0.3352	0.3721	**0.3950**
	NDCG@5	0.1289	0.1398	0.0631	0.0494	0.1159	0.1488	**0.1556**
	NDCG@10	0.1558	0.1629	0.0804	0.0654	0.1402	0.1729	**0.1817**
	NDCG@20	0.1811	0.1847	0.1003	0.0852	0.1628	0.1953	**0.2055**
Yelp	HR@5	0.2834	0.3216	0.1457	0.1567	0.2935	0.3173	**0.3462**
	HR@10	0.4221	0.4469	0.2546	0.2623	0.4257	0.4451	**0.4784**
	HR@20	0.5975	0.5989	0.4257	0.4312	0.5839	0.5993	**0.6309**
	NDCG@5	0.1889	0.2283	0.0890	0.0996	0.2009	0.2236	**0.2443**
	NDCG@10	0.2335	0.2687	0.1239	0.1336	0.2435	0.2647	**0.2869**
	NDCG@20	0.2778	0.3070	0.1668	0.1759	0.2834	0.3036	**0.3253**
WeChat	HR@5	0.2756	0.3069	0.1836	0.1943	0.3193	0.3105	**0.3406**
	HR@10	0.4103	0.4366	0.2231	0.2247	0.4406	0.4511	**0.4861**
	HR@20	0.5291	0.5484	0.2884	0.2907	0.5539	0.5507	**0.5979**
	NDCG@5	0.1948	0.2131	0.1272	0.1266	0.2036	0.2195	**0.2361**
	NDCG@10	0.2357	0.2743	0.1447	0.1394	0.2615	0.2827	**0.3057**
	NDCG@20	0.2869	0.3013	0.1693	0.1526	0.2957	0.3089	**0.3314**

Dataset	#users	#items	#actions	avg.	density
Beauty	52,024	57,289	0.4M	7.6	0.01%
Sports	25,598	18,357	0.3M	8.3	0.05%
Yelp	30,431	20,033	0.3M	10.4	0.05%
WeChat	749,452	211,004	9.5M	12.7	0.006%

Experimental Result

Datasets	Metric	GRU4Rec
Beauty 0.01	HR@5	0.1179
	HR@10	0.1582
	HR@20	0.2146
	NDCG@5	0.0869
	NDCG@10	0.1000
	NDCG@20	0.1141
Sports 0.05	HR@5	0.0961
	HR@10	0.1499
	HR@20	0.2295
	NDCG@5	0.0631
	NDCG@10	0.0804
	NDCG@20	0.1003
Yelp	HR@5	0.1457
	HR@10	0.2546
	HR@20	0.4257
	NDCG@5	0.0890
	NDCG@10	0.1239
	NDCG@20	0.1668
WeChat	HR@5	0.1836
	HR@10	0.2231
	HR@20	0.2884
	NDCG@5	0.1272
	NDCG@10	0.1447
	NDCG@20	0.1693

- RNN-based methods usually exhibit worse performance for the data sparsity setting.

Experimental Result

Datasets	Metric	SASRec	SASRec*	CT4Rec	Improv.	Improv.*
Beauty	HR@5	0.2109	0.2194	**0.2556**	21.19%	16.50%
	HR@10	0.2759	0.2748	**0.3200**	15.98%	16.45%
	HR@20	0.3546	0.3392	**0.3891**	9.73%	14.71%
	NDCG@5	0.1523	0.1661	**0.1924**	26.33%	15.83%
	NDCG@10	0.1733	0.1840	**0.2132**	23.02%	15.87%
	NDCG@20	0.1932	0.2003	**0.2307**	19.41%	15.18%
Sports	HR@5	0.1912	0.1966	**0.2196**	14.85%	11.70%
	HR@10	0.2747	0.2683	**0.3010**	9.57%	12.19%
	HR@20	0.3751	0.3547	**0.3950**	5.31%	11.36%
	NDCG@5	0.1289	0.1398	**0.1556**	20.71%	11.30%
	NDCG@10	0.1558	0.1629	**0.1817**	16.62%	11.54%
	NDCG@20	0.1811	0.1847	**0.2055**	13.47%	11.26%
Yelp	HR@5	0.2834	0.3216	**0.3462**	22.16%	7.65%
	HR@10	0.4221	0.4469	**0.4784**	13.34%	7.05%
	HR@20	0.5975	0.5989	**0.6309**	5.59%	5.34%
	NDCG@5	0.1889	0.2283	**0.2443**	29.33%	7.01%
	NDCG@10	0.2335	0.2687	**0.2869**	22.87%	6.77%
	NDCG@20	0.2778	0.3070	**0.3253**	17.10%	5.96%
WeChat	HR@5	0.2756	0.3069	**0.3406**	25.58%	10.98%
	HR@10	0.4103	0.4366	**0.4861**	18.47%	11.34%
	HR@20	0.5291	0.5484	**0.5979**	13.00%	9.03%
	NDCG@5	0.1948	0.2131	**0.2361**	21.20%	10.79%
	NDCG@10	0.2357	0.2743	**0.3057**	29.70%	11.40%
	NDCG@20	0.2869	0.3013	**0.3314**	15.51%	9.99%

Dataset	#users	#items	#actions	avg.	density
Beauty	52,024	57,289	0.4M	7.6	0.01%
Sports	25,598	18,357	0.3M	8.3	0.05%
Yelp	30,431	20,033	0.3M	10.4	0.05%
WeChat	749,452	211,004	9.5M	12.7	0.006%

L2/ Cosine Regularization

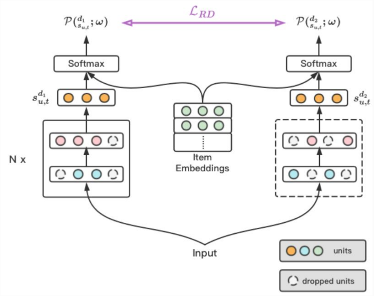
$$\ell_i = -\log \frac{e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau}}, \qquad (1)$$

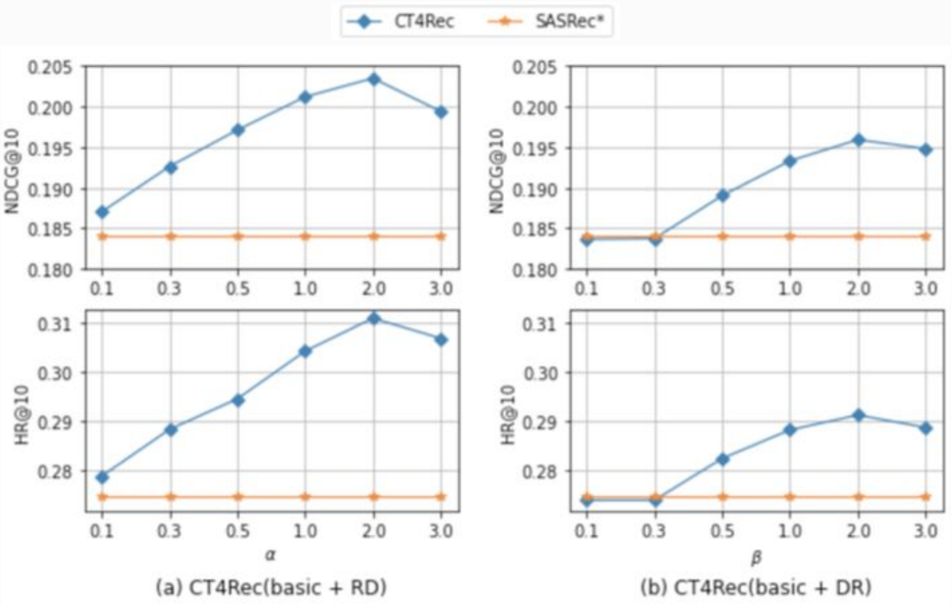
$$L2(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

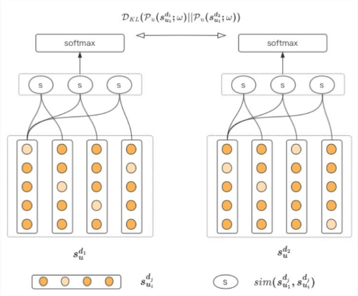
$$sim(h_1, h_2) = h_1 \cdot h_2 \implies \frac{h_1 \cdot h_2}{||h_1|| \cdot ||h_2||}$$

https://arxiv.org/pdf/2104.08821.pdf

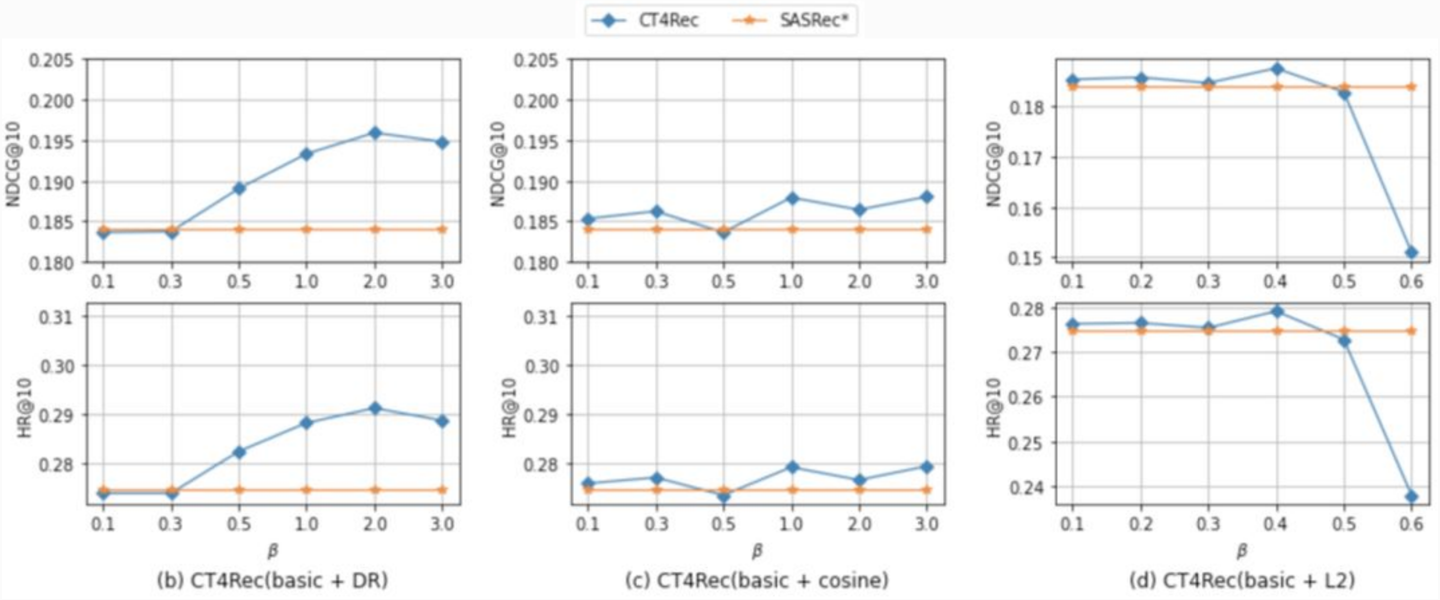
Ablation Study







Ablation Study



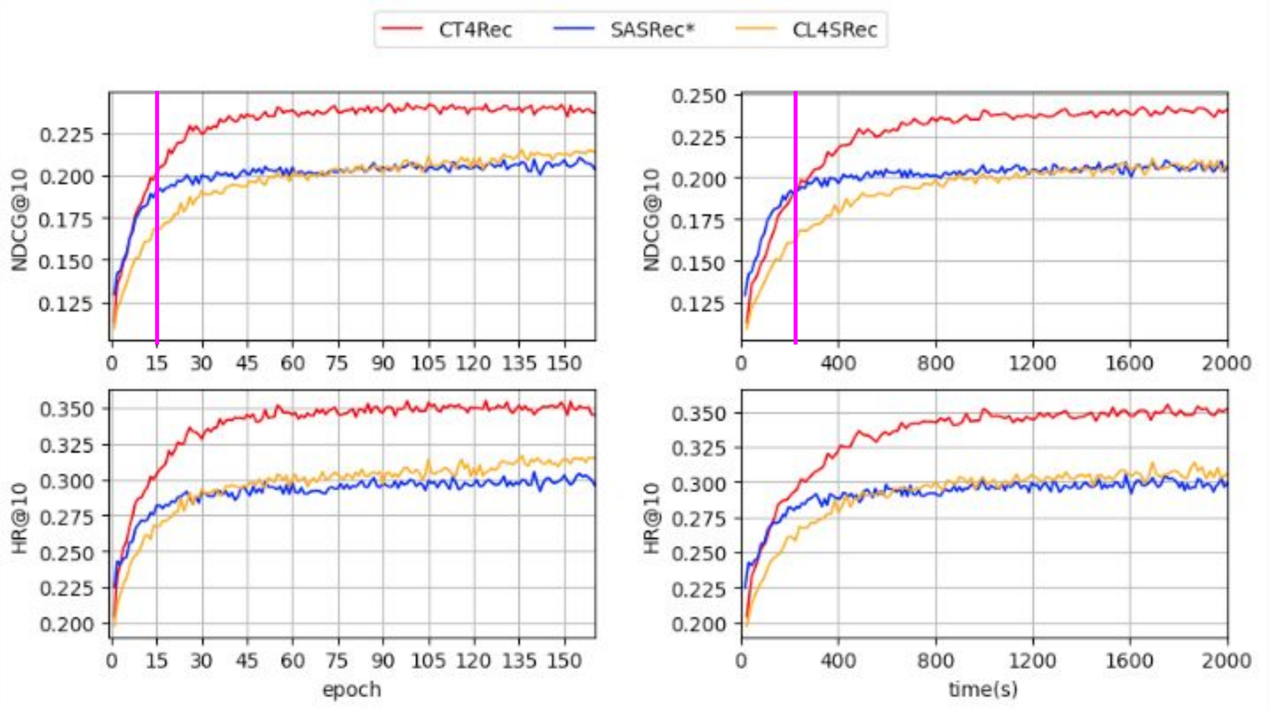
Extension to Data Augmentation

- Replace dropout with data augmentation method

Aug.[53]	Metrics	SASRec	SASRec*	+L2	+Cos	+DR	+RD	+CT4Rec	Improv.	Improv.*
Reorder	HR@10	0.2759	0.2748	0.2867	0.2856	0.2905	0.2857	**0.3076**	11.49%	11.94%
	HR@20	0.3546	0.3392	0.3552	0.3542	0.3558	0.3546	**0.3776**	6.49%	11.32%
	NDCG@10	0.1523	0.1661	0.1892	0.1900	0.1954	0.1899	**0.2047**	34.41%	23.24%
	NDCG@20	0.1932	0.2003	0.2064	0.2073	0.2119	0.2072	**0.2224**	15.11%	11.03%
Mask	HR@10	0.2759	0.2748	0.2831	0.2898	0.2892	0.2891	**0.3140**	13.81%	14.26%
	HR@20	0.3546	0.3392	0.3484	0.3553	0.3542	0.3609	**0.3868**	9.08%	14.03%
	NDCG@10	0.1523	0.1661	0.1879	0.1941	0.1948	0.1886	**0.2061**	35.33%	24.08%
	NDCG@20	0.1932	0.2003	0.2044	0.2106	0.2111	0.2067	**0.2244**	16.15%	12.03%

Training and Cost Analysis

- Better final optimum & less convergence time

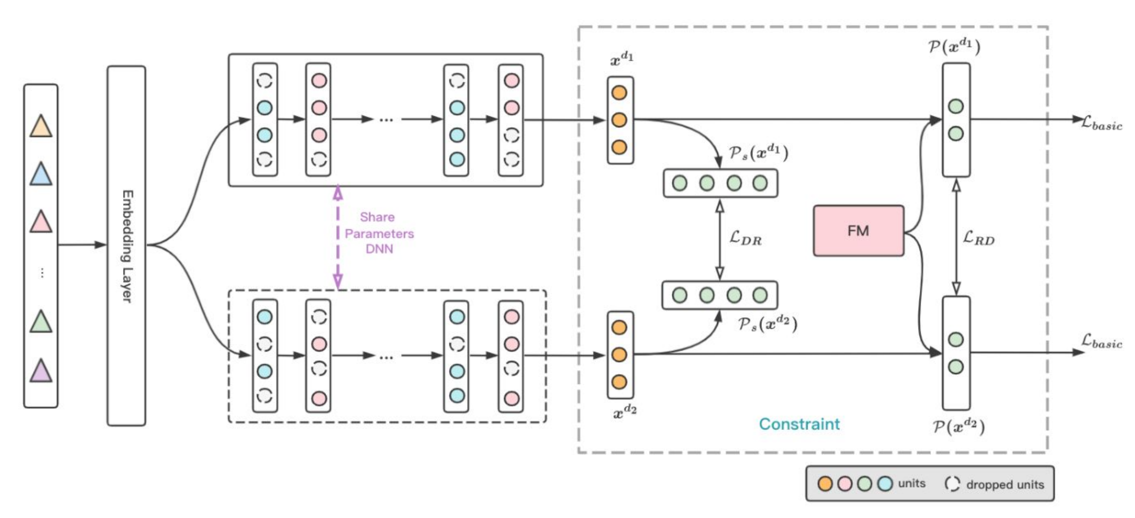


Outline

- Introduction
- Related Work
- Method
- Experiment
- Conclusion

Conclusion

- Introduce a top-performed regularization in the output space
- Design a novel consistency training term in the representation space
- Extensive experiment and analysis demonstrate its effectiveness, efficiency, etc.
- Further exploration on CTR prediction task



Further… CTR Prediction

- Definition:

  Click-through rate prediction is the task of predicting the likelihood that something on a website (or anything else) will be clicked.

WeChat-Video		WeChat-Article	
Model	AUC	Model	AUC
LR	0.7569	LR	0.7401
FM	0.7608	FM	0.7465
Wide&Deep	0.7695	Wide&Deep	0.7538
DeepFM	0.7719	DeepFM	0.7552
AutoInt	0.7703	AutoInt	0.7519
xDeepFM	0.7738	xDeepFM	0.7560
CT4CTR	0.7766	CT4CTR	0.7593