CROLoss: Towards a Customizable Loss for Retrieval Models in Recommender Systems

Tang Yongxiang[1*], Bai Wentao[2*], Li Guilin[3], Liu Xialong[4], Zhang Yu[5]

{yongxiang.tyx[1], baiwentao.bwt[2], xialong.lxl[4]}@alibaba-inc.com
{liguilin.lgl[3], daoji[5]}@lazada.com
*: co-first authors

Paper Link: https://arxiv.org/abs/2208.02971
Code Link: https://github.com/WDdeBWT/CROLoss

Notations

Notation	Meaning
u	user feature
v	item feature
I	candidate item set consists of distinct items
C	corpus consists of positive (u, v) samples
S(u, v)	the scoring function
R(u, v)	the ranking statistic, which is the rank number of item v for user u
ψ	the unit step function, $\psi(x) = 0$ when $x<0$ and $\psi(x)=1$ when $x>0$
Recall@N	the recall metric of the top-N recommend result of model S on corpus C

Recall@N Optimization in Top-N Recommender Systems

- Ranking statistic:

$$R_\psi(u, v; \theta) = 1 + \sum_{v_i \in I - \{v\}} \psi\left(S_\theta(u, v_i) - S_\theta(u, v)\right).$$

- Recall@N:

$$\text{Recall@N} = \frac{1}{|C|} \sum_{(u,v) \in C} \mathbf{1}_{\left\{R_\psi(u,v;\theta) \leq N\right\}}.$$

- Retrieval stage in recommender systems:
  - For each user u, retrieve top-N items.
  - We hope the top-N items cover more positive samples.
  - We need a loss function that directly minimizing Recall@N.

LTR Losses in Retrieval Stage

- Softmax loss (Shen et al. 2014, Yi et al. 2019):

$$L_{softmax} = -\sum_{(u,v)\in C} \log \frac{e^{S(u,v)}}{e^{S(u,v)} + \sum_{v_i \in I-\{v\}} e^{S(u,v_i)}}.$$

- Triplet loss (Schultze et al. 2003, Schroff et al. 2015, Huang et al. 2020):

$$L_{triplet} = \sum_{(u,v)\in C} \sum_{v_i \in I-\{v\}} (S(u, v_i) - S(u, v) + m)_+.$$

- BPR loss (Rendle et al. 2012):

$$L_{bpr} = -\sum_{(u,v)\in C} \sum_{v_i \in I-\{v\}} \log(\sigma(S(u, v) - S(u, v_i))).$$

Customizable Recall@N Optimization Loss

- The weighting function w

$$w\colon [1, |I| + 1) \to [0, +\infty), \quad \int_1^{|I|+1} w(x)\, dx = 1.$$

  - It is a PDF on [1, |I| + 1).
  - It resembles how much we want to optimize Recall@N for any N.
  - We usually set w monotonically decreasing, since we focus on the top retrieved items.

- The CROLoss

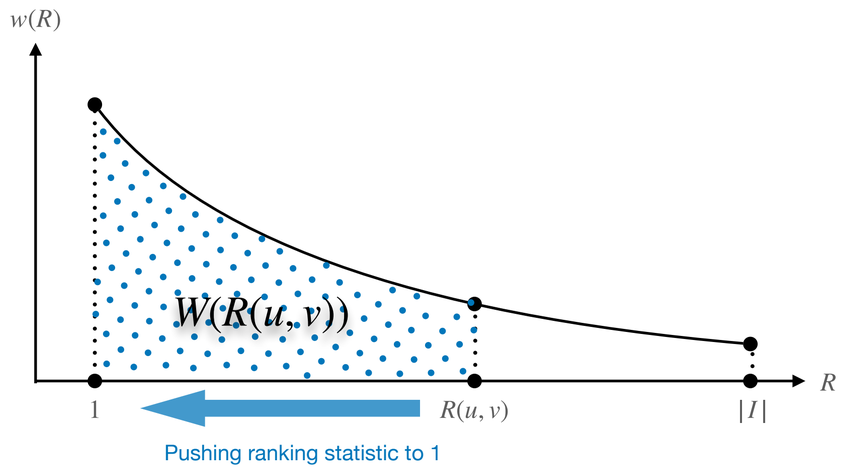
$$L(\theta) = -\,\mathbb{E}_w \,\mathrm{Recall}@x = -\int_1^{|I|+1} w(x)\mathrm{Recall}@x\, dx.$$

Customizable Recall@N Optimization Loss

- We proved that CROLoss is equivalent to

$$L(\theta) = \frac{1}{|C|} \sum_{(u,v)\in C} W(R_\psi(u, v; \theta)) - 1.$$

  - Here $W(x) = \int_1^x w(z)\, dz$.



- Using inverse power function as the weighting function for CROLoss.
  - $w_\alpha(x) = \frac{1}{Z} x^{-\alpha}$
  - If $\alpha = 1$, $W_\alpha(N) \sim \log(N)$
  - if $\alpha \neq 1$, $W_\alpha(N) \sim 1 - N^{1-\alpha}$.
  - Then we have

$$L_{\alpha,\psi}(\theta) = \sum_{(u,v)\in C} W_\alpha(R_\psi(u, v; \theta)).$$

Comparison Kernel Functions in CROLoss

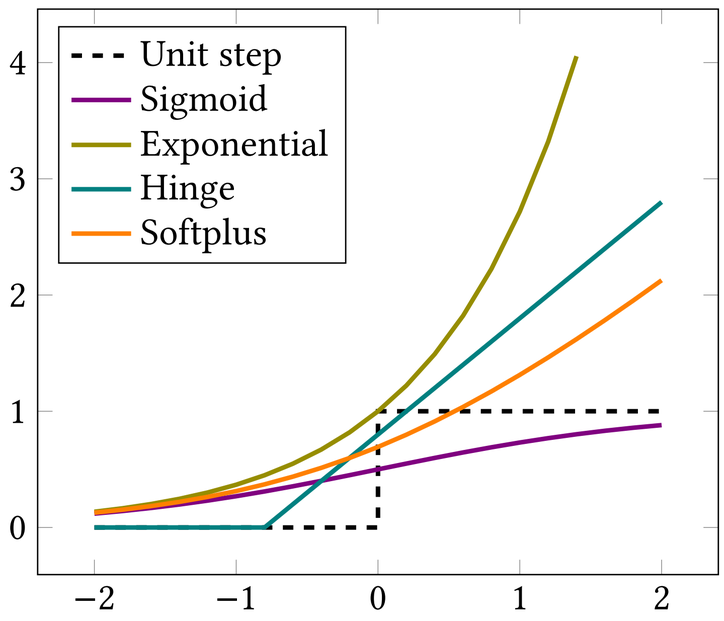
- The comparison kernel function
  - Use a differentiable function ϕ to replace the unit step function ψ.

- The approximation of ranking statistic:

$$R_\phi(u, v; \theta) = 1 + \sum_{v_i \in I - \{v\}} \phi\left(S_\theta(u, v_i) - S_\theta(u, v)\right).$$

- The CROLoss becomes:

$$L_{\alpha,\phi}(\theta) = \sum_{(u,v) \in C} W_\alpha\left(R_\phi(u, v; \theta)\right).$$



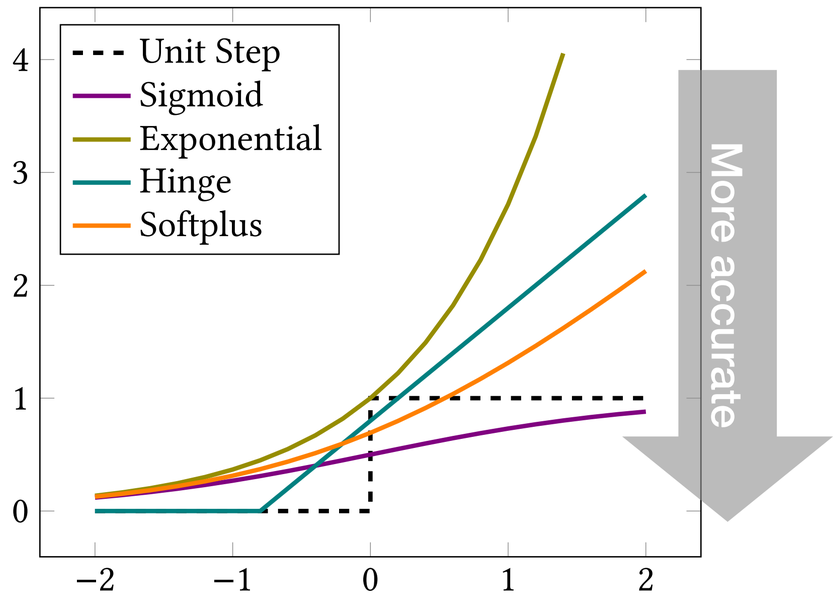
Relation to Common Loss Functions

- By taking different α and ϕ, we find out that the CROLoss $L_{\alpha,\phi}$ is equivalent to several conventional loss functions.

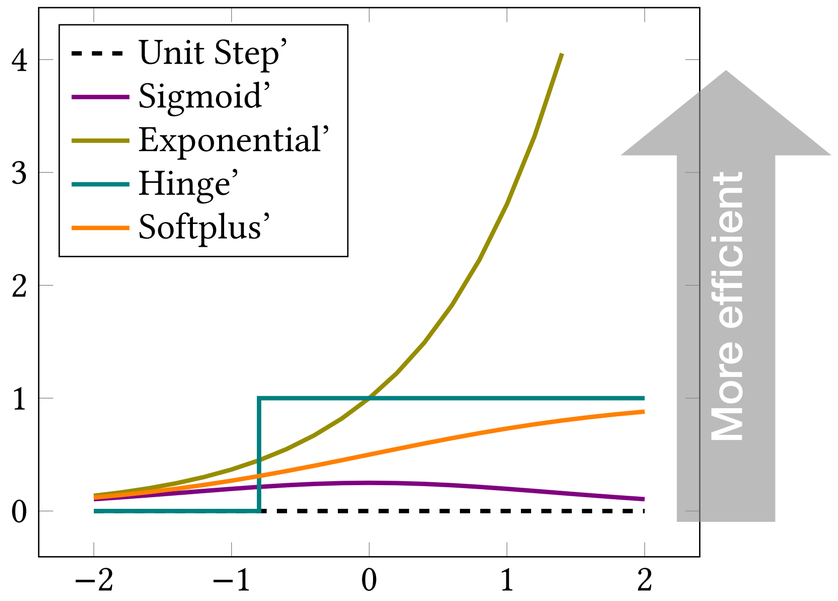
	α	$\phi(x)$	Expression
Softmax Loss	1	e^x	$\sum_{(u,v)\in C} \log\left(1 + \sum_{v_i \in I-\{v\}} e^{S(u,v_i)-S(u,v)}\right)$
Triplet Loss	0	$(x + m)_+$	$\frac{1}{\lvert I\rvert} \sum_{(u,v)\in C} \sum_{v_i \in I-\{v\}} \left(S(u,v_i) - S(u,v) + m\right)_+$
BPR Loss	0	$\log(1+ e^x)$	$\frac{1}{\lvert I\rvert} \sum_{(u,v)\in C} \sum_{v_i \in I-\{v\}} \log\left(1 + e^{S(u,v_i)-S(u,v)}\right)$

Lambda Method for Optimizing CROLoss

- The conflict between approximation accuracy and training efficiency:
  - Kernel closer to unit step → smaller gradient value → less training efficiency.
  - Larger gradient value → kernel further from unit step → less Recall@N approximation accuracy.



(a) The comparison kernel functions.



(b) The derivative of comparison kernel functions.

Lambda Method for Optimizing CROLoss

- The partial derivative of CROLoss:

$$\frac{\partial L_{\alpha,\phi}(\theta)}{\partial g_u^{(i)}} = \overbrace{\left\{ w_\alpha(R_\phi(u,v;\theta)) \right\}}^{\lambda_{\alpha,\phi}} \left\{ \phi'(g_u^{(i)}) \right\}$$

replace with more accurate kernel ϕ_1

replace with kernel ϕ_2 with larger derivative

  - Here $g_u^{(i)} = S_\theta(u, v_i) - S_\theta(u, v)$.
  - Similar to the Lambda-Rank (Burges et al. 2006).

- The CROLoss-lambda:

$$L_{\alpha,\phi_1,\phi_2}(\theta) = \sum_{(u,v)\in C} \overbrace{\mathrm{SG}\left(w_\alpha \left(R_{\phi_1}(u,v;\theta) \right) \right)}^{\lambda_{\alpha,\phi_1}} R_{\phi_2}(u,v;\theta).$$

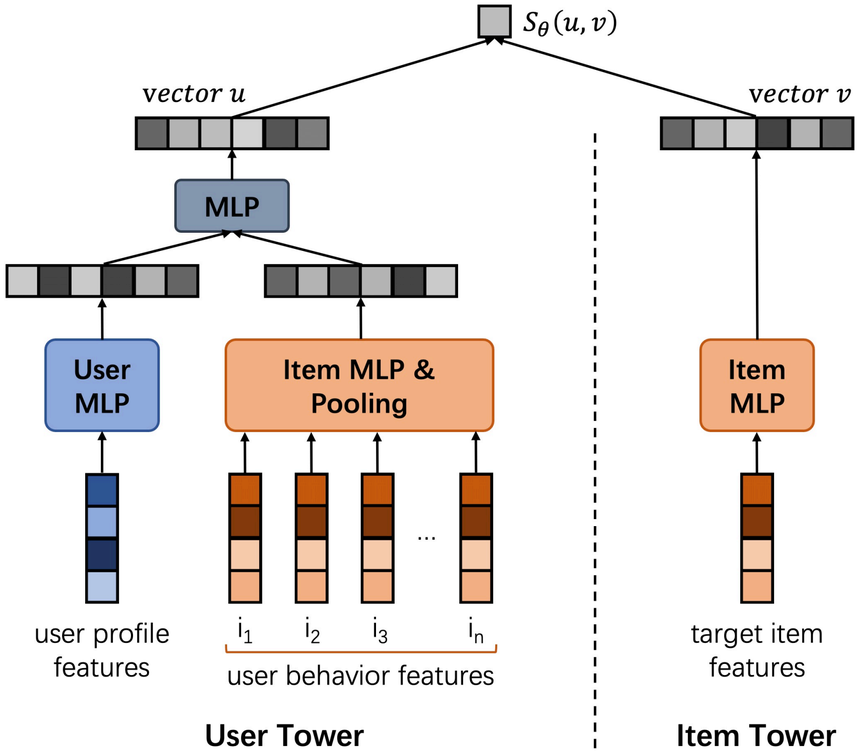
Experimental Setup

- The two-tower base model:
  - 10 * cosine($vector_u$, $vector_v$) as scoring function.
  - $Vector_u$ and $vector_v$ are output of two DNN towers.

- Dataset:

Dataset	#users	#items	#behaviors
Amazon Books	459,133	313,966	8,898,041
Taobao	976,779	1,708,530	85,384,110

- Compared methods:
  - Softmax/triplet/BPR loss.
  - Different choice of weighting function and comparison kernel.



Experiment Results

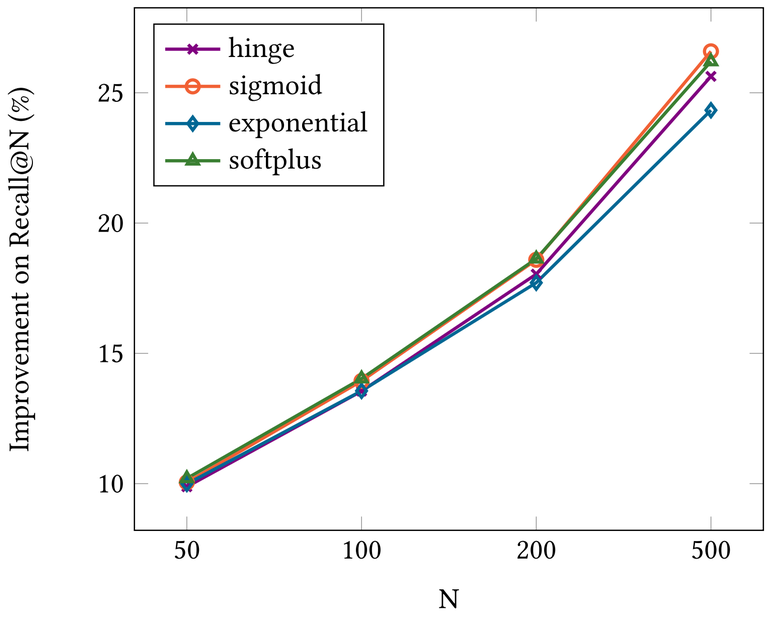
- CROLoss achieves the best Recall@N result on both datasets.

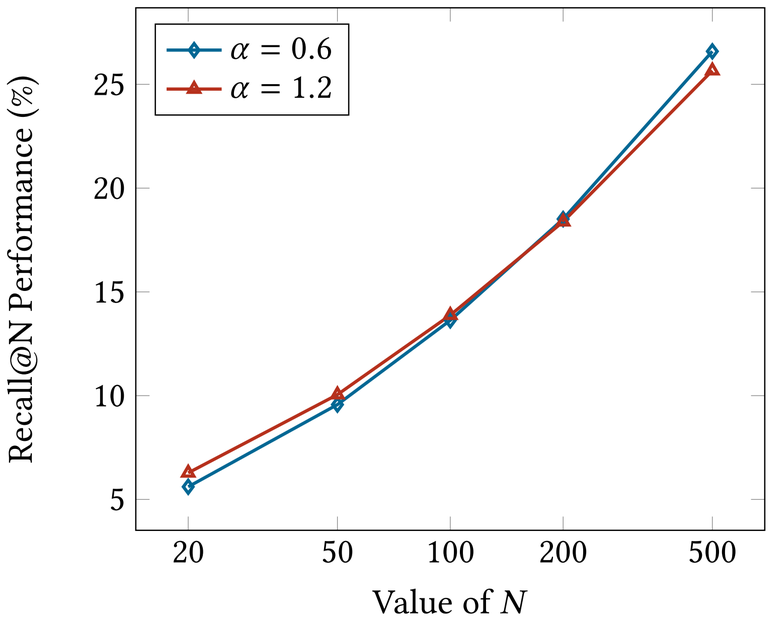
Datasets	Methods	R@50	R@100	R@200	R@500
Amazon	cross-entropy loss	9.68	13.24	17.46	24.26
	triplet loss	7.53	11.21	15.93	24.11
	BPR loss	8.24	12.08	16.96	25.21
	CROLoss[1]	**10.20**	14.03	18.63	26.06
	CROLoss-lambda[2]	10.17	**14.07**	**18.81**	**26.20**
Taobao	cross-entropy loss	4.71	6.59	9.01	13.13
	triplet loss	2.46	3.71	5.43	8.84
	BPR loss	2.89	4.33	6.35	10.25
	CROLoss[3]	4.75	6.65	9.06	13.13
	CROLoss-lambda[4]	**5.27**	**7.35**	**10.01**	**14.57**

1. Use softplus as kernel and set α to 1.0.
2. Use sigmoid as kernel 1 and softplus as kernel 2 and set α to 1.0.
3. Use exponential as kernel and set α to 1.4.
4. Use sigmoid as kernel 1 and exponential as kernel 2 and set α to 1.4.

Experiment Results

- Fixing α to 0.8 and we tested different comparison kernels.
- Fixing the comparison kernel to sigmoid function and we tested $\alpha \in \{0.6, 1.2\}$.





Experiment Results

- Choice of the two kernels in the CROLoss-lambda method:
  - We take $\alpha = 1$ for Amazon dataset and $\alpha = 1.4$ for Taobao dataset.
  - We compared the combinations of two kernels in the CROLoss-lambda method.

Table 1: Original CROLoss performance.

Dataset	Kernel ϕ	R@50	R@100	R@200	R@500
Amazon	hinge	9.88	13.55	17.94	24.99
	sigmoid	9.96	13.94	18.59	**26.09**
	exponential	9.68	13.24	17.46	24.26
	softplus	**10.20**	**14.03**	**18.63**	26.06
Taobao	hinge	4.02	5.62	7.68	11.18
	sigmoid	4.11	5.87	8.15	12.13
	exponential	**4.75**	**6.65**	**9.06**	**13.13**
	softplus	4.36	6.21	8.52	12.54

Table 2: CROLoss-lambda performance.

Dataset	Kernel ϕ_1	Kernel ϕ_2	R@50	R@100	R@200	R@500
Amazon	unit step	hinge	9.07	12.865	17.46	24.96
	unit step	exponential	8.74	12.15	16.35	23.20
	unit step	softplus	9.98	13.85	18.53	25.96
	sigmoid	hinge	9.435	13.305	17.995	25.69
	sigmoid	exponential	8.79	12.10	16.26	22.90
	sigmoid	softplus	**10.17**	**14.07**	**18.81**	**26.20**
Taobao	unit step	hinge	3.30	4.79	6.79	10.30
	unit step	exponential	4.85	6.81	9.36	13.70
	unit step	softplus	4.25	6.07	8.47	12.62
	sigmoid	hinge	3.82	5.49	7.70	11.60
	sigmoid	exponential	**5.27**	**7.35**	**10.01**	**14.57**
	sigmoid	softplus	4.74	6.71	9.26	13.62

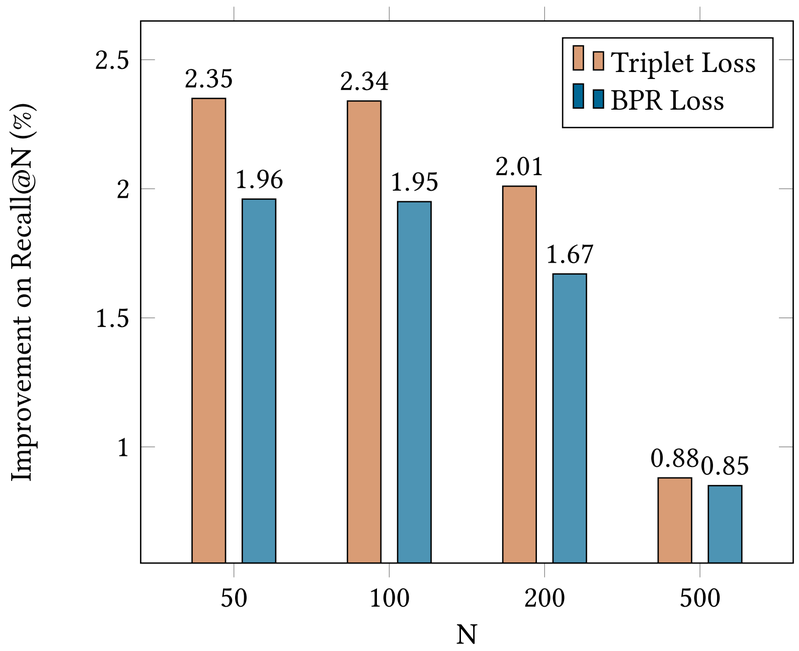
A New Perspective on Hard Negative Mining

- Our CROLoss method can be regarded as an importance sampling from the item set I, and the sampling probability of item v is proportional to

$$w_\alpha\left(R_\phi(u, v; \theta)\right).$$

- When $\alpha > 0$ the sampling probabilities are larger for hard samples.

- The improvement of Recall@N under hard negative sampling:
  - Set $\alpha = 1$ to simulate hard negative mining.
  - Set $\alpha = 0$ for uniformly random negative sampling.



Main Contributions

- The CROLoss:
  - Directly optimize Recall@N metrics.
  - Customizable retrieval size N.
  - Shows that CROLoss is a generalized form of conventional methods such as the softmax loss, the triplet loss, and the BPR loss.

- The Lambda method:
  - Solved the conflict between approximation accuracy and training efficiency of CROLoss.

- Experiments and a new perspective on hard negative:
  - The experiments prove our theory.
  - Shows that CROLoss provides an automatic and quantitative hard negative sampling method.

Q&A