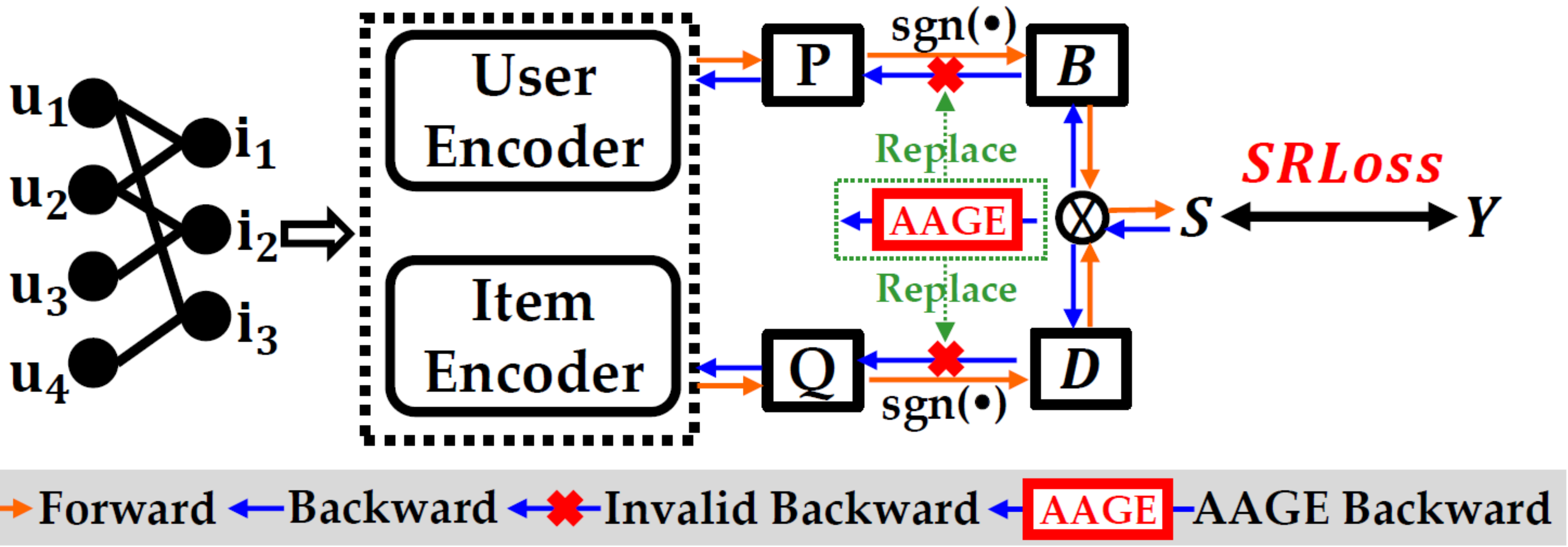
Optimizing Recall in Deep Graph Hashing Framework for Item Retrieval (Student Abstract)

Fangyuan Luo, Jun Wu
Beijing Jiaotong University, Beijing 100044, China

Motivation

However, existing HR methods fail to align the learning objective and evaluation metric, leading to unsatisfied recommendation performance. In addition, existing optimization strategies which are widely applied in deep graph hashing, such as straight-through estimation (STE), conducts a relative coarse gradient estimation, resulting in inaccurate optimization directions.

Method



Definition of Recall

$$\mathrm{Recall@N} = \frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} \mathbb{I}(\mathrm{R_{ui}} \leq \mathrm{N})$$

$$R_{ui} = 1 + \sum_{j=1\backslash i}^{n} \mathbb{I}(s_{ui} > s_{uj})$$

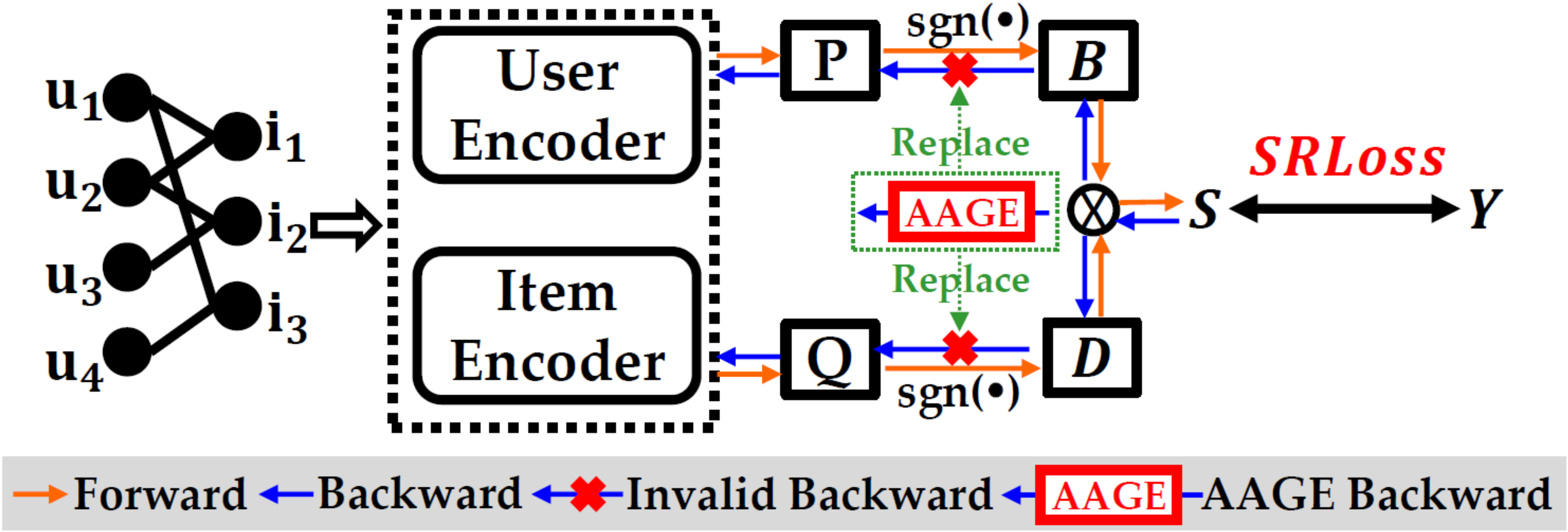
Formulation:

$$\mathcal{L} = -\frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} \mathbb{I}((1 + \sum_{j=1\backslash i}^{n} \mathbb{I}(\mathbf{b}_u^T \mathbf{d}_i > \mathbf{b}_u^T \mathbf{d}_j))) \leq N)$$
$$s.t.\ \mathbf{B} \in \{-1,1\}^{f\times m}, \mathbf{D} \in \{-1,1\}^{f\times n} \quad (2)$$

Smooth

$$\mathcal{L} = -\frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} \mathcal{K}_\phi((1 + \sum_{j=1\backslash i}^{n} \mathcal{K}_\psi(\mathbf{b}_u^T \mathbf{d}_i > \mathbf{b}_u^T \mathbf{d}_j))) - N)$$
$$s.t.\ \mathbf{B} \in \{-1,1\}^{f\times m}, \mathbf{D} \in \{-1,1\}^{f\times n} \quad (3)$$

Method



Approximation-Adjustable Gradient Estimation:

$$\mathrm{sgn}(\mathrm{x}) = \lim_{\beta\to\infty} 2\sigma(\beta \mathrm{x})(1+\beta \mathrm{x}(1-\sigma(\beta \mathrm{x}))) - 1;$$

$$\frac{\partial \mathrm{sgn}(\mathrm{x})}{\partial x} = \frac{2\cdot[(\beta^2 x + 2\beta)e^{-2\beta x} - (\beta^2 x - 2\beta)e^{-\beta x}]}{(1+e^{-\beta x})^3}$$

Experiments

➢ Datasets

Datasets	#Users	#Items	#Ratings	Density
Gowalla	29,858	40,981	1,027,370	0.084%
Yelp2018	31,831	40,841	1,666,869	0.128%

Table 2: Statistics of the datasets.

➢ Baselines

BGCH [WWW'2023], HashGNN [WWW'2020], HashRec [CIKM'2019]

➢ Metrics

NDCG@50, NDCG@100, Recall@50, Recall@100

Experiments

	Gowalla				Yelp2018			
	R@50	R@100	N@50	N@100	R@50	R@100	N@50	N@100
Proposed	**0.23082**	**0.31396**	**0.15109**	**0.17424**	**0.10140**	**0.16348**	**0.06016**	**0.08035**
BGCH (Chen et al. 2023)	0.19160	0.26590	0.12740	0.14840	0.08350	0.13450	0.05000	0.06700
HashGNN (Tan et al. 2020)	0.09481	0.15110	0.05112	0.06649	0.04692	0.08250	0.02661	0.03811
HashRec (Kang and McAuley 2019)	0.12060	0.18930	0.06160	0.08100	0.06307	0.10995	0.03590	0.05134

(1) Overall, our proposed method, BGCH and HashRec show superior performance to HashGNN. Such an observation illustrates the effectiveness of providing accurate gradient estimation.

(2) Among Proposed, BGCH, and HashRec, our proposed method demonstrates significant improvements. The performance improvements are attributed to the benefits of joint effect of the proposed smooth recall loss and the approximation-adjustable gradient estimator.

Thank you!

北京交通大學

BEIJING JIAOTONG UNIVERSITY