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

Fangyuan Luo, Jun Wu*

Key Laboratory of Big Data & Artificial Intelligence in Transportation of Ministry of Education, Beijing Jiaotong University, Beijing 100044, China

{fangyuanluo, wuj}@bjtu.edu.cn

Abstract

Hashing-based recommendation (HR) methods, whose core idea is mapping users and items into hamming space, are common practice to improve item retrieval efficiency. However, existing HR fails to align optimization objective (i.e., Bayesian Personalized Ranking) and evaluation metric (i.e., Recall), leading to suboptimal performance. In this paper, we propose a smooth recall loss (termed as *SRLoss*), which targets Recall as the optimization objective. Due to the existence of discrete constraints, the optimization problem is NP-hard. To this end, we propose an approximation-adjustable gradient estimator to solve our problem. Experimental Results demonstrate the effectiveness of our proposed method.

Introduction

Being able to provide personalized suggestions to each user, recommender systems (RS) have been widely used in countless online applications. With the rapidly growing number of users and items, the two-stage RS is widely adopted in industry, aiming to meet the stringent response requirement of RS. The two-stage RS is consist of the retrieval stage and the ranking stage. We mainly concentrate on the retrieval tasks. Embedding-based Retrieval (EBR) is a popular paradigm in retrieval tasks, where users and items are mapped into a shared real-valued latent space and then the retrieval task falls into a similarity search problem. Despite the effectiveness, the computational costs to filter hundreds of candidate items in real-valued embedding space are expensive. A promising solution is to learn hash codes for users and items, which gives birth to Hashing-based Recommendation (HR) methods. Then, the user-item similarities based on inner product can be replaced by hamming similarity based on bit operations. 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.

To this end, we propose a smooth recall loss (termed as *SRLoss*) for better retrieval accuracy, whose core idea is to replace the non-differentiable indicator functions with approximate differentiable functions (Luo, Wu, and Wang 2022). Furthermore, we introduce an approximation-adjustable gradient estimator (AAGE), where we take sign-swish function with parameter β to estimate the derivation of hash function in the backpropagation. β refers to the temperature, which controls the approximation between their derivations. Experimental results demonstrate the effectiveness of our proposed method.

*Corresponding Author

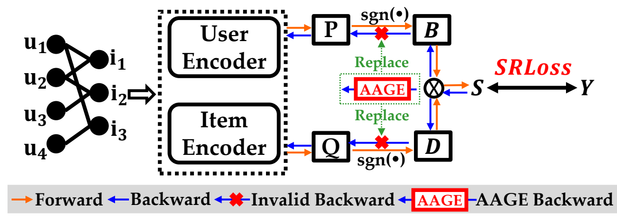

Figure 1: Illustration of the proposed framework,where $\mathbf{P}$ ($\mathbf{B}$) and $\mathbf{Q}$ ($\mathbf{D}$) are real-valued embeddings (hash codes) for users and items; $\mathbf{S}$ and $\mathbf{Y}$ denote the predicted scores and groundtruth; $\otimes$ denotes the inner product.

Method

The overall framework is shown in Figure 1, where the architecture is instantiated as LightGCN (He et al. 2020).

Smooth Recall Loss: Suppose the set of the index of observed user-item pairs is Ω, Recall@N is defined as:

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

where $R_{ui} = 1 + \sum_{j=1\backslash i}^{n} \mathbb{I}(s_{ui} > s_{uj})$ is the ranking position of positive item i corresponding to the relevance score $s_{ui} = \mathbf{b}_u^T \mathbf{d}_i$. $\mathbb{I}(\bullet)$ denotes an indicator function. Thus, our learning objective is formulated as:

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

where m and n are the number of users and items, f denotes the length of hash codes. Since the Recall metric in-

	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

Table 1: Recommendation performance on Gowalla and Yelp2018 dataset, where "R" and "N" denote the Recall and NDCG. The best performing method in each column is boldfaced, and the second best method is each column is underlined.

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.

cludes non-differentiable indicator functions, their optimization is notoriously difficult. To this end, we introduce a novel smooth recall loss whose core idea is to replace $\mathbb{I}(\bullet)$ with differentiable approximate functions (e.g., sigmoid, exponential, hinge and softplus). Eq.(2) can be approximated as:

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

where $\mathcal{K}_\phi$ and $\mathcal{K}_\psi$ are two approximate functions.

Approximation-Adjustable Gradient Estimation: Due to the existence of discrete constraints, solving Eq.(3) is generally NP-hard. To this end, we propose an approximation-adjustable gradient estimation method, which takes sign-swish function with parameter β (Darabi et al. 2018) to approximate hash function in the backpropagation:

$$\mathrm{sgn(x)} = \lim_{\beta\to\infty} 2\sigma(\beta\mathrm{x})(1+\beta\mathrm{x}(1-\sigma(\beta\mathrm{x}))) - 1; \quad (4)$$

where $\sigma(\bullet)$ is the sigmoid function and $\beta > 0$ controls how fast Eq.(4) asymptotes to -1 and $+1$. And the corresponding derivative can be derived accordingly as:

$$\frac{\partial \mathrm{sgn(x)}}{\partial x} = \frac{2\cdot[(\beta^2 x + 2\beta)e^{-2\beta x} - (\beta^2 x - 2\beta)e^{-\beta x}]}{(1+e^{-\beta x})^3} \quad (5)$$

In short, we apply $sgn(\bullet)$ function for forward propagation and estimate the gradients with Eq.(5) for backpropagation.

Experiments

We evaluate our method on Gowalla and Yelp2018 datasets, whose statistics are shown in Table 2. During training, the dataset is randomly divided into training, valid and test set based on 8:1:1. We adopt Recall and NDCG to evaluate recommendation performance. The length of hash codes is set as 64. We exclude early discrete coordinate descend (DCD)-based recommendation baselines, mainly because the competing models have validated the superiority.

From Table 1, we have the following observations: (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.

Conclusion

In this paper, we propose a novel Hashing-based recommendation model, which targets Recall as the optimization objective. Specifically, we introduce an approximation-adjustable gradient estimator, which can provide more accurate gradients. Experimental results on two datasets demonstrate the effectiveness of our proposed method.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities, under the Grant 2023YJS022, and the Open Research Fund from the Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen, under Grant B10120210117-OF01.

References

Chen, Y.; Fang, Y.; Zhang, Y.; and King, I. 2023. Bipartite Graph Convolutional Hashing for Effective and Efficient Top-N Search in Hamming Space. In *WWW*, 3164–3172.

Darabi, S.; Belbahri, M.; Courbariaux, M.; and Nia, V. P. 2018. BNN+: Improved Binary Network Training. *CoRR*, abs/1812.11800.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*, 639–648.

Kang, W.; and McAuley, J. J. 2019. Candidate Generation with Binary Codes for Large-Scale Top-N Recommendation. In *CIKM*, 1523–1532.

Luo, F.; Wu, J.; and Wang, T. 2022. Discrete Listwise Personalized Ranking for Fast Top-N Recommendation with Implicit Feedback. In *IJCAI*, 2159–2165.

Tan, Q.; Liu, N.; Zhao, X.; Yang, H.; Zhou, J.; and Hu, X. 2020. Learning to Hash with Graph Neural Networks for Recommender Systems. In *WWW*, 1988–1998.