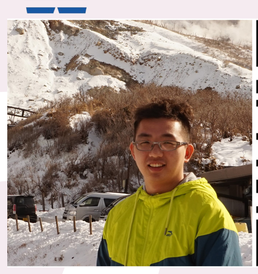
RNR: A Generic Bayesian-based Framework for Enhancing Top-N Recommender Systems

Rong-qin Xu[1] ✉, Mingyang Zhou[1] ✉, and Hao Liao[1];

[1]Shenzhen University, College of Computer Science and Software Engineering, Shenzhen, P.R. China
Correspondence to: xurongqin2018@email.szu.edu.cn, zmy@szu.edu.cn, jamesliao520@gmail.com

THE WEB CONFERENCE April 20-24 TAIPEI 2020



Abstract

In personalized top-N recommender systems, a core task is to design effective methods to measure user-item preference scores and then to suggest, for each user, a small set of personalized items with high scores. However, little attention was paid to the recommendation of low-score user-item links. In this work, based on the Bayesian estimation theory, we propose a novel metric **RNR** (**R**ecall-to-**N**oise **R**atio) to characterize the ability to recommend both high-score and low-score user-item links. Totally, there are three contributions in our work:

* We propose a generic framework that leverages **RNR** to transfer the link scores of the state-of-the-art recommendation methods.
* Besides,the RNR function could be efficiently calculated only by counting the score difference of existing and non-existing links, which does not improve the time complexity of the classical methods.
* Empirical experiments show that the proposed framework could optimally improve the recommendation accuracy.

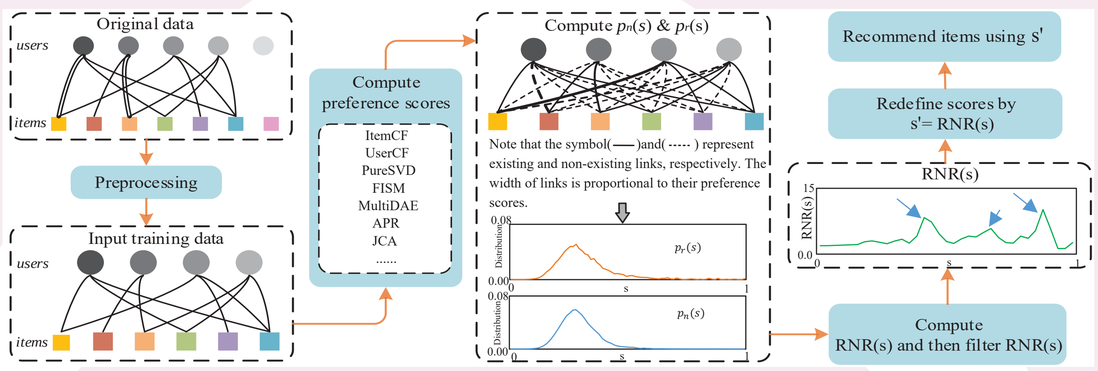


Figure 1. The architecture of our proposed RNR framework

1 The Proposed Framework

Supposing a user-item bipartite network $G = (U, I, L)$, where U and I denote the set of users and items respectively and $(u, i) \in L$ represents a link. If the user u provides feedback for item i, the entry $r_{ui} = 1$; otherwise $r_{ui} = 0$. The links L in graph G are divided into training set L^T and probe set L^P, $L^T \cup L^P = L$ and $L^T \cap L^P = \emptyset$. s_{ui} is the preference score of a user u towards an item i, obtained by an arbitrary recommendation method. We usually implicitly assume that a user-item link with higher score is more likely to be a potential future link, which could be rephrased as

$$p(r_{ui} = 1|s_{ui}) > p(r_{u'i'} = 1|s_{u'i'}), if \quad s_{ui} > s_{u'i'}, \tag{1}$$

where s_{ui} and $s_{u'i'}$ represent the preference scores of two non-existing links in the training set L^T respectively.

According to the bayesian estimation,

$$p(r_{ui} = 1|s_{ui}) = \frac{p(r_{ui} = 1) \cdot p(s_{ui}|r_{ui} = 1)}{p(s_{ui})}, \tag{2}$$

where $p(s_{ui})$ is the probability that a random chosen (existing or non-existing) link in the training set has score s_{ui}. It is obvious that $p(s_{ui})$ is mainly determined by the score distribution of non-existing links. That is to say, $p(s_{ui}) \approx p_n(s_{ui})$, where $p_n(s_{ui})$ is the probability that a random chosen non-existing link in the training set has score s_{ui}. Thus, we arrive at

$$p(r_{ui} = 1|s_{ui}) = \frac{p(r_{ui} = 1) \cdot p(s_{ui}|r_{ui} = 1)}{p_n(s_{ui})} = c_1 \cdot \frac{p(s_{ui}|r_{ui} = 1)}{p_n(s_{ui})}, \tag{3}$$

where $c_1 = p(r_{ui} = 1) = |L^T|/(|U| \times |I|)$ is a constant representing the probability of an existing link between a random pairwise user item in the training set. Neglecting the constant term, we propose **RNR (Recall-to-Noise Ratio)**,

$$RNR(s) = \frac{p(s|r_{ui} = 1)}{p_n(s)} = \frac{p_r(s)}{p_n(s)}, \tag{4}$$

where $p_r(s) = p(s|r_{ui} = 1)$ is the probability that a random chosen existing link in the training set has score s. Thus, our framework use $s' = RNR(s)$, an one-variable transferring function, to transfer original scores into s' and then use s the recommend items. Figure 1 shows the detailed structure of our proposed RNR framework.

2 Results

We conduct experiments on two datasets: (1) **ML100K** dataset has 943 users, 1682 items and 100000 user-item interactions (links). (b) **Delicious** dataset has 1867 users, 69227 items with 104799 user-item interactions (links). We randomly divide each dataset to 90% training set and 10% probe set. We compare our method with these state-of-the-art recommendation methods: **ItemCF** [6], **UserCF** [6], **PureSVD** [1], **FISM** [4], **MultiDAE** [5], **APR** [3] and **JCA** [7]. Finally, we use $Recall@N$ and $NDCG@N$ metrics to evaluate the results.

2.1 Empirical RNR(s) Analysis.

We first present an example of the score distributions $p_r(s)$, $p_n(s)$ and the $RNR(s)$ in Fig. 2. In Fig. 2(a), scores of most links are around 0.32 while $s > 0.6$ and $s < 0.1$, $p_n(s) \to 0$. In Fig. 2(b), apart from the high score $s \approx 0.97$, some low scores ($s \approx 0.78$ or 0.09) also have high RNR(s), demonstrating recognizing low-score links and the effectiveness of our proposed scheme as well as the failure of the classical assumption in eq. 1.

2.2 Accuracy Performance.

(1) Recall performance: Figure 3 reports the recall performance. In Fig. 3, our proposed RNR framework significantly outperforms the original methods in most cases, except the MultiDAE and APR on Delicious dataset. Detailed experiments show that in MultiDAE and APR methods, when we calculate $RNR(s) = p_r(s)/p_n(s)$, $p_n(s) \to 0$ introduces much fluctuation for $RNR(s)$. The fluctuation of $RNR(s)$ largely influences the recall accuracy for MultiDAE and APR methods on Delicious dataset. **(2) NDCG performance:** Figure 4 shows the $NDCG@N$ performance as a function of N. In Fig.4, RNR-based methods outperform the state-of-the-art methods in most cases. In practical scenarios, the forward positions usually mean a high click probability and high attraction. We can conclude that our method is inclined to push correct items to the forward positions.

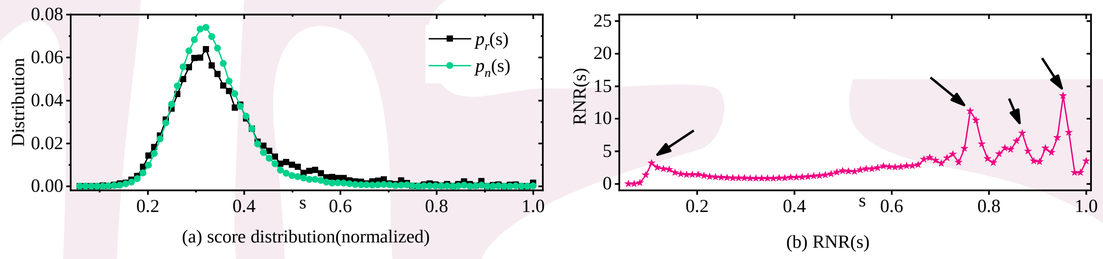


Figure 2. The score distributions of $p_r(s)$ and $p_n(s)$, and the corresponding $RNR(s)$ of Delicious dataset, where preference scores are obtained by FISM method.

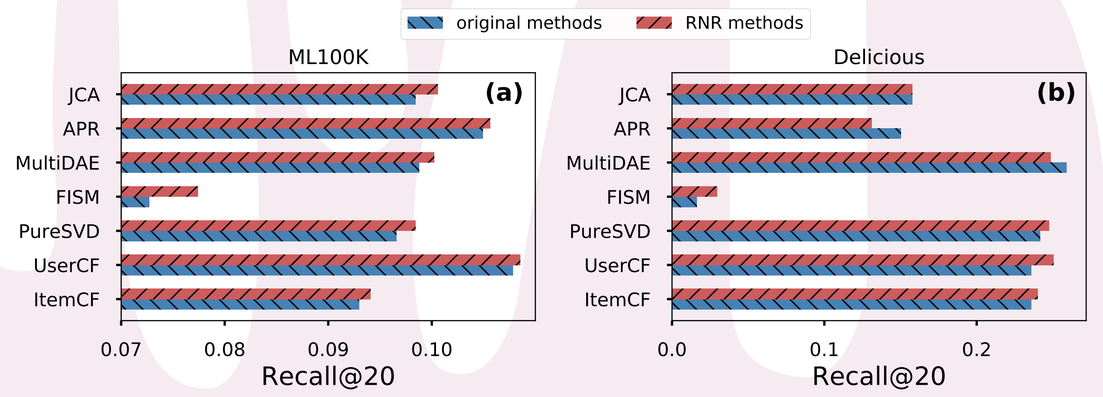


Figure 3. Comparisons of *Recall*@20 results of the RNR recommended links with that of the corresponding original methods on two datasets. (a) ML100K. (b) Delicious.

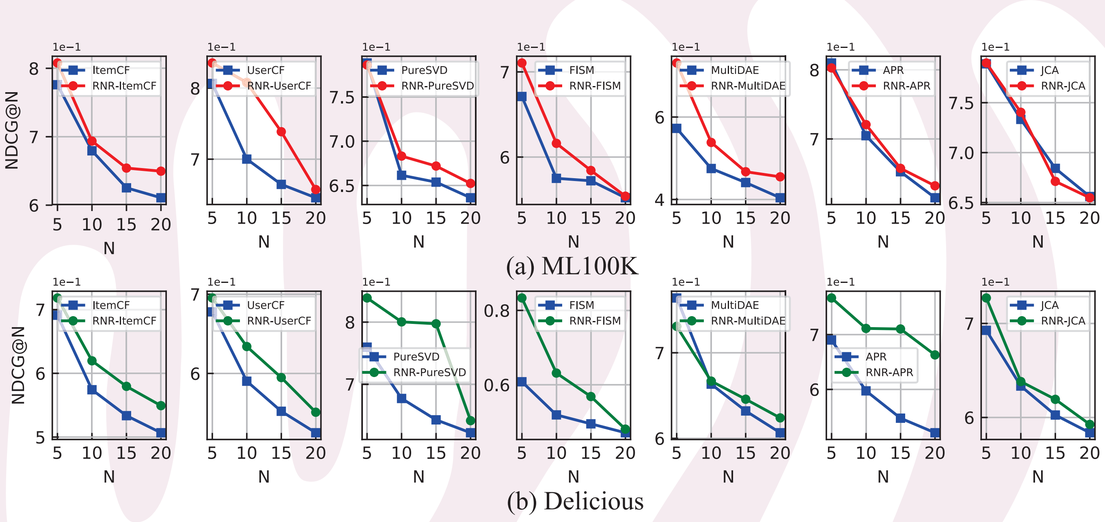


Figure 4. The $NDCG@N$ as a function of N by seven state-of-the-art methods and the corresponding RNR methods on two datasets. (a) ML100K. (b) Delicious.

References

[1] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46. ACM, 2010.

[2] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[3] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. Adversarial personalized ranking for recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 355–364. ACM, 2018.

[4] Santosh Kabbur, Xia Ning, and George Karypis. Fism: factored item similarity models for top-n recommender systems. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 659–667. ACM, 2013.

[5] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*, pages 689–698. International World Wide Web Conferences Steering Committee, 2018.

[6] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. Recommender system application developments: a survey. *Decision Support Systems*, 74:12–32, 2015.

[7] Ziwei Zhu, Jianling Wang, and James Caverlee. Improving top-k recommendation via joint collaborative autoencoders. In *The World Wide Web Conference*, pages 3483–3482. ACM, 2019.

Funded by

深圳大学 SHENZHEN UNIVERSITY
深圳市科技创新委员会
深圳市国家自主创新示范区管理委员会
深圳市高新技术产业园区管理委员会
深圳市外国专家局
Tencent 腾讯
NSFC