

A New Probabilistic Model for Top-k Ranking Problem

Shuzi Niu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng

niushuzi@software.ict.ac.cn, {lanyanyan, guojiafeng, cxq}@ict.ac.cn

Institute of Computing Technology, Chinese Academy of Sciences, Beijing, P.R.China

ABSTRACT

This paper is concerned with top-k ranking problem, which reflects the fact that people pay more attention to the top ranked objects in real ranking application like information retrieval. A popular approach to top-k ranking problem is based on probabilistic models, such as Luce model and Mallows model. However, whether the sequential generative process described in these models is a suitable way for top-k ranking remains a question. According to the riffled independence factorization proposed in recent literature, which is a natural structural assumption on top-k ranking, we propose a new generative process of top-k ranking data. Our approach decomposes distributions over the top-k ranking into two layers: the first layer describes the relative ordering between the top k objects and the rest $n-k$ objects, and the second layer describes the full ordering on the top k objects. On this basis, we propose a new probabilistic model for top-k ranking problem, called *hierarchical ordering model*. Specifically, we use three different probabilistic models to describe different generative processes of the first layer, and Luce model to describe the sequential generative process of the second layer, thus we obtain three different specific hierarchical ordering models. We also conduct extensive experiments on benchmark datasets to show that our proposed models can outperform previous models significantly.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Retrieval models

General Terms

Algorithm, Performance, Experimentation, Theory

Keywords

Learning to Rank, Top-k, Ranking Model

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...\$15.00.

1. INTRODUCTION

Ranking is a central problem in a number of machine learning applications, such as information retrieval, recommendation system and computational advertising etc. In these real applications, people pay more attention to the top ranked objects, which means that if two ranked lists of objects have the same ranking results for the top k positions, they will likely provide the same experience to the users. This characteristic of ranking problem has been explored in earlier studies in different setting [1, 9, 10, 7], which is referred to as the top-k ranking problem.

Many probabilistic models on permutation have been widely applied to solve the ranking problem, among which, the Luce model [11] and the Mallows Model [5] are the most popular ones. Xia et al. [10] propose Top-k ListMLE to model the top-k ranking data, which can be viewed as an extension of the Luce model to the top-k setting; Qin et al. [8] propose a coset-permutation distance based stagewise (CPS) model to inherit the advantages of both the Luce model and the Mallows model, which can also be extended to the top-k setting. Both generative processes of the above methods are sequential. At the i -th stage, an object is selected and assigned to position i with a certain probability. We refer to this approach as sequential generative process. Although these methods have been proven effective empirically, the rationality of the sequential generative process remains a question because it is heuristic and lacks theoretical support.

Riffled independence, which was introduced in recent literature by Huang et al. [2], is a generalized notion of probabilistic independence for ranked data. It corresponds to ranking disjoint sets of objects independently, then interleaving those rankings, which has been proven to be an appropriate structural assumption on the top-k ranking data theoretically [3, 4]. Inspired by the assumption of riffled independence, we argue that a more appropriate generative process of the top-k ranking data should be comprised of two layers: in the first layer, the relative ordering between the top k objects and the rest $n-k$ objects is generated, then in the second layer, the full ordering on the top k objects is further generated.

On the basis of the aforementioned hierarchical generative process, we propose a new probabilistic model for top-k ranking problem, which is called *hierarchical ordering model* (HOM for short). In HOM, the distribution of a top-k ranking is decomposed to the product of two terms, one term is the distribution of the relative ordering in the first layer and the other is the distribution of the full ordering in the second layer. Specifically, we introduce three different prob-

ability distributions to model the relative ordering in the first layer, and use Luce model to describe the full ordering in the second layer. Combining the two-layer probability distributions, we obtain three different specific hierarchical ordering models. To evaluate the performances of our proposed models, we conduct extensive experiments on benchmark datasets. Our experimental results show that our models consistently outperform previous top-k ranking models, such as Top-k ListMLE [10] and Top-k CPS [8].

2. BACKGROUNDS

In this section, we introduce some backgrounds of top-k ranking problem, including some notations, reviews on existing top-k probabilistic ranking models and riffled independence assumption.

2.1 Notations

Let $D = \{x_1, \dots, x_n\}$ be a set of objects to be ranked, where n is the number of objects. We use σ to denote a top-k ranking on D , which is comprised of a full ordering of top k objects and the relative ordering of top k objects and the rest $n - k$ objects. Let T be the set of top k objects and F be the set of the rest $n - k$ objects. The full ordering of T is denoted by σ_T . We use f to represent the ranking function to be learned. For evaluation, we relate the top-k ranking to relevance labels by defining $\mathbf{y} = \{y_1, \dots, y_n\}$ as position-aware relevance labels of the corresponding objects. As did in [7], y_j is defined as $k + 1 - \sigma_T(x_j)$ if $x_j \in T$, and $y_j = 0$ otherwise.

2.2 Top-k Probabilistic Ranking Models

Many probabilistic models on permutation have been widely applied to solve the ranking problem, such as the Luce model [11] and the Mallows Model [5]. Here we review two recently proposed probabilistic models based on them, one of which is directly a top-k ranking model and the other can be easily extended to the top-k ranking scenario.

Xia et al. [10] proposed Top-k ListMLE to model the top-k ranking data, which can be viewed as an extension of the Luce model to the top-k setting. In Top-k ListMLE, the probability of top-k ranking is formulated as below:

$$P(\sigma) = \prod_{i=1}^k \frac{\exp(f(x_{\sigma_T(i)}))}{\sum_{j=i}^k \exp(f(x_{\sigma_T(j)})) + \sum_{x_l \in F} \exp(f(x_l))}, \quad (1)$$

Qin et al. [8] introduced a coset-permutation distance based stagewise (CPS) model for rank aggregation, which inherits the advantages of both the Luce model and the Mallows model. It can also be applied to the setting of ranking by treating each feature as a weak ranker. As CPS organizes the probability in the form of Luce model, it can be easily extended to top-k ranking by similar technique as used in Top-k ListMLE. Thus we obtain Top-k CPS as in the following formulation:

$$P(\sigma) = \prod_{i=1}^k \frac{\exp(\sum_{m=1}^M \theta_m d(\sigma, \phi(h_m)))}{\sum_{j=i}^n \exp(\sum_{m=1}^M \theta_m d(S_{n-i}(\sigma_T, j), \phi(h_m)))}, \quad (2)$$

where M is the number of features, θ_m suggests the weight of feature h_m and $\phi(h_m)$ denotes the permutation generated by the ranking function h_m . σ in the right side can be directly viewed as a right coset as defined in [8], and $S_{n-i}(\sigma_T, j)$ denotes the right coset including all permutations that rank

objects $\sigma_T^{-1}(1), \dots, \sigma_T^{-1}(i-1)$ and $\sigma_T^{-1}(j)$ in the top i positions, respectively. The Kendall's tau coset-permutation distance $d(\cdot, \cdot)$ defined in [8] is adopted in this paper.

2.3 Riffled Independence

Riffled independence defines a novel class of independence structure on ranking [2], which corresponds to ranking disjoint sets of objects independently, then interleaving those rankings. The formal definition is as follows.

Definition 1. Given a ranking $\pi \in S_D$, S_D is the collection of all possible full rankings of object set D , a partition of D into disjoint sets A and B , let p be a distribution over S_D , the sets A and B are said to be *riffled independent* if p decomposes as:

$$p(\pi) = m_{AB}(\tau_{AB}(\pi)) \cdot f_A(\pi_A) \cdot g_B(\pi_B), \quad (3)$$

where interleaving distribution m_{AB} is defined over interleaving of A and B , distributions f_A and g_B are defined over relative rankings π_A and π_B , respectively.

In the top-k ranking scenario, riffled independence indicates that the distribution over top-k ranking has a natural decomposition, which provides us a new perspective to look at top-k ranking.

3. WAYS TO LOOK AT TOP-K RANKING

In this section, we will discuss the generative process of the top-k ranking data. First, we can see that both Top-k ListMLE [10] and Top-k CPS [8] take a sequential view of top-k ranking data. However, whether it is a good choice remains a question. Inspired by the assumption of riffled independence [2], we propose a new hierarchical view of top-k ranking data, which naturally captures the structural characteristics of the top-k ranking data.

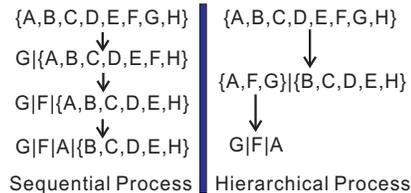


Figure 1: Different generative processes for top-3 ranking.

3.1 Sequential Generative Process

From Section 2.2, we can see that both Top-k ListMLE [10] and Top-k CPS [8] are stagewise models, which decompose the process of generating the top-k ranking data into k sequential stages. At the i -th stage, an object is selected and assigned to position i according to a probability distribution. For Top-k ListMLE, the probability is computed based on the scores of the unassigned objects by Luce Model. While for Top-k CPS, the probability is based on the distance between a location permutation and the right coset of the Top-k ranking by Mallows Model. Therefore, both generative processes of Top-k ListMLE and Top-k CPS are sequential, referred to as sequential generative process.

A typical sequential generative process on top-k ranking is like this. Given 8 objects $\{A, B, C, D, E, F, G, H\}$, the top-3 ranking $G|F|A|{B, C, D, E, H}$ is generated as follows: first,

we generate the object to be ranked at the 1-st position, and we get G ; second, we generate the object to be ranked at the 2-nd position, and we get F ; third, we generate the object to be ranked at the 3-rd position, and we get A . You will find a sketch map of the above generative process in Figure 1.

Although the sequential view of top-k ranking leads to efficient computing, the sequential generative process remains heuristic and lacks theoretical support. We argue that it is not the unique generative process and there is a more appropriate way to look at top-k ranking data.

3.2 Hierarchical Generative Process

According to the definition of riffled independence in Section 2.3, the distribution over the top-k ranking σ can be decomposed¹ into the interleaving distribution of T and F and the distribution over ranking σ_T . Inspired by the above idea, we propose a new two-layer generative process of top-k ranking, called hierarchical generative process. In the first layer, the relative ordering $T \prec F$ indicating all the k objects in T are ranked before all the other $n - k$ objects in F is generated; in the second layer, the full ordering σ_T on all the k objects in T is generated.

Take the aforementioned example of $\{A, B, C, D, E, F, G, H\}$ in Section 3.1, the top-3 ranking $G|F|A|\{B, C, D, E, H\}$ is generated as follows: first, we generate 3 objects which are ranked higher than the other 5 objects, and we get $\{A, F, G\}$; second, we generate the full ordering of $\{A, F, G\}$, and we get $G|F|A$. The above generative process is also depicted in Figure 1.

Compared to the sequential generative process, we can see that our hierarchical generative process can well capture the structural characteristics of top-k ranking. Furthermore, our hierarchical generative process inherits the empirical advantages of riffled independence and enjoys theoretical guarantees as in previous work on riffled independence [3, 4].

4. HIERARCHICAL ORDERING MODEL

According to the hierarchical generative process of top-k ranking as described in Section 3, we propose a new probabilistic model for top-k ranking problem, which is called hierarchical ordering model, HOM for short.

For each set of objects $\{x_1, \dots, x_n\}$, let P_{TF} be the probability distribution of the relative ordering in the first layer of the hierarchical generative process, and P_T be the probability distribution of the full ordering in the second layer of the hierarchical generative process. HOM defines the probability of a top-k ranking σ as the following form:

$$P(\sigma) = P_{TF}(T \prec F) \cdot P_T(\sigma_T), \quad (4)$$

It is easy to verify that the probability defined in HOM naturally forms a distribution. That is, for every possible top-k ranking σ , we always have $P(\sigma) \geq 0$, and $\sum_{\sigma} P(\sigma) = 1$.

The above hierarchical ordering model HOM is in fact a general framework. We can obtain different specific forms of HOM by defining different forms of distributions on relative ordering $T \prec F$ and full ordering σ_T , that is, P_{TF} and P_T . We discuss them separately and give three examples of HOM in this paper.

¹In top-k ranking problem, the ordering among the bottom $n - k$ objects do not matter. Thus, we can view the distribution over rankings of F as uniform, which can be ignored in the decomposition.

4.1 Distribution of Relative Ordering

For the probability distribution of the relative ordering $T \prec F$, we can model it in the following three different ways. (1) Taking T as a whole, the probability of $T \prec F$ can be viewed as the energy proportion of T to the whole object set, where the energy of an object set can be computed by adding the scores of all the objects in the set. This approach is like the competition between groups, which is called Group-to-Group. The formulation is as follows:

$$P_{TF}(T \prec F) \propto \frac{\sum_{x_i \in T} \exp(f(x_i))}{\sum_{j=1}^{|D|} \exp(f(x_j))}. \quad (5)$$

(2) Considering the relationship of each object in T and that in F , the probability of $T \prec F$ can be viewed as the event that each object in T is ranked higher than that in F . This approach is like the competition between objects, which is called One-to-One and can be well formulated by the widely used Bradley-Terry model [6]. The formulation is as follows:

$$P_{TF}(T \prec F) \propto \prod_{x_i \in T} \prod_{x_j \in F} \frac{\exp(f(x_i))}{\exp(f(x_i)) + \exp(f(x_j))}. \quad (6)$$

(3) Considering the relationship of each object in T and the whole F , the probability of $T \prec F$ can be viewed as a multi-step event that for each step an object in T is ranked higher than F . This approach is like the competition between each object and a whole group, which is called One-to-Group and can be well formulated by the widely used Plackett-Luce model [6]. The formulation is as follows:

$$P_{TF}(T \prec F) \propto \prod_{x_i \in T} \frac{\exp(f(x_i))}{\exp(f(x_i)) + \sum_{x_j \in F} \exp(f(x_j))}. \quad (7)$$

4.2 Distribution of Full Ordering

For the probability distribution of the full ordering in T , it can be computed by some widely used probabilistic models on permutation in previous work, such as Luce model [11] and Mallows model [5]. Considering computational efficiency, here we use Luce model as an example in this paper with the following formulation:

$$P_T(\sigma_T) = \prod_{i=1}^{|\mathcal{T}|} \frac{\exp(f(x_{\sigma_T(i)}))}{\sum_{j=i}^{|\mathcal{T}|} \exp(f(x_{\sigma_T(j)}))}. \quad (8)$$

If the probabilities distribution of P_T is fixed to Eq.(8), and P_{TF} takes the form of Eq.(5), Eq.(6) and Eq.(7), we will call the corresponding HOM as HOM-GG, HOM-OO and HOM-OG, respectively.

5. EXPERIMENTS

In this section, we compare the empirical performance of our proposed three kinds of hierarchical ordering models with state-of-the-art top-k ranking methods on benchmark datasets, including Top-k ListMLE and Top-k CPS. Note that k is set to 10 in our experiments.

5.1 Settings

We use the benchmark LETOR4.0 collection² in the experiments. There are two datasets containing permutation-level ground-truth in LETOR4.0, MQ2007-list and MQ2008-

²<http://research.microsoft.com/en-us/um/beijing/projects/letor/>

Top-10 MQ2007

Methods	κ -N@1	κ -N@2	κ -N@3	κ -N@4	κ -N@5	κ -N@6	κ -N@7	κ -N@8	κ -N@9	κ -N@10	κ -ERR
HOM-GG	0.4321	0.4642	0.4972	0.5231	0.5440	0.5618	0.5762	0.5873	0.5958	0.6037	0.5881
HOM-OO	0.4305	0.4640	0.4951	0.5221	0.5415	0.5606	0.5740	0.5873	0.5964	0.6028	0.5873
HOM-OG	0.4355	0.4686	0.4975	0.5250	0.5444	0.5633	0.5788	0.5901	0.5982	0.6050	0.5907
Top-k ListMLE	0.4110	0.4459	0.4730	0.4985	0.5207	0.5359	0.5512	0.5642	0.5740	0.5810	0.5688
Top-k CPS	0.3563	0.3945	0.4234	0.4447	0.4648	0.4819	0.4944	0.5063	0.5157	0.5238	0.5138

Top-10 MQ2008

Methods	κ -N@1	κ -N@2	κ -N@3	κ -N@4	κ -N@5	κ -N@6	κ -N@7	κ -N@8	κ -N@9	κ -N@10	κ -ERR
HOM-GG	0.2846	0.3166	0.3458	0.3696	0.3887	0.4045	0.4207	0.4340	0.4434	0.4525	0.4411
HOM-OO	0.2995	0.3238	0.3494	0.3723	0.3921	0.4060	0.4217	0.4370	0.4498	0.4593	0.4508
HOM-OG	0.2951	0.3257	0.3519	0.3803	0.3968	0.4134	0.4286	0.4420	0.4540	0.4665	0.4506
Top-k ListMLE	0.2650	0.2970	0.3202	0.3451	0.3672	0.3829	0.4003	0.4114	0.4220	0.4296	0.4194
Top-k CPS	0.2587	0.2779	0.3025	0.3189	0.3366	0.3556	0.3711	0.3821	0.3914	0.4014	0.4049

Table 1: Performance comparison on Top-10 MQ2007 and Top-10 MQ2008

list. Considering the computational cost of Top-k CPS, we only use top 100 documents for each query in the experiments. We construct the top-10 ground-truth by only preserving the total order of top 10 documents on each query. We obtain two datasets containing top-10 ground-truth, referred to as top-10 MQ2007 and top-10 MQ2008.

All the models including baselines and our methods, are optimized by gradient descent method. Parameters like the learning rate and stopping precision are tuned on the validation set in each trial of the 5-fold cross validation method. κ -NDCG and κ -ERR [7] are employed to measure the effectiveness of ranking performance, which are natural generalizations of NDCG and ERR to the top-k ranking scenario.

5.2 Results

The performances of different top-k ranking models on these two datasets are shown in Table 1, where κ -N@j stands for κ -NDCG@j. From the results, we can find that the proposed three kinds of HOMs can consistently and significantly outperform Top-k ListMLE and Top-k CPS in terms of both κ -NDCG and κ -ERR. For example, considering HOM-OG on Top-10 MQ2007, the relative performance improvements over top-k ListMLE, top-k CPS are 4.13%, 15.5% in terms of κ -NDCG@10 and 3.85%, 15.0% in terms of κ -ERR, respectively. Another example is HOM-OG model on Top-10 MQ2008, the relative performance improvements over Top-k ListMLE, Top-k CPS are 8.59%, 16.2% in terms of κ -NDCG@10 and 7.46%, 11.3% in terms of κ -ERR, respectively. Furthermore, we can see that the proposed three kinds of HOMs reach the best performances (denoted by numbers in bold) under all metrics on the two datasets.

The results indicate that our hierarchical ordering model is more effective than models based on the sequential generative process (i.e. Top-k ListMLE and Top-k CPS). It in some sense reflects the hierarchical generative process is superior to the sequential generative process for top-k ranking.

6. CONCLUSIONS

In this paper, we argue that the sequential generative process of top-k ranking adopted in many previous ranking models is not unique. Such a process is heuristic and lacks theoretical support. On the basis of the recently proposed riffled independence assumption of top-k ranking, we pro-

pose a new hierarchical generative process, which is a more appropriate way to look at top-k ranking data. According to the generative process, we propose a hierarchical ordering model, which is a new probabilistic model for the top-k ranking problem. We conducted extensive experiments on two benchmark datasets, and experimental results show that the proposed models can outperform previous top-k probabilistic ranking models significantly.

7. ACKNOWLEDGMENTS

This research work was funded by the National Natural Science Foundation of China under Grant No. 60933005, No. 61173008, No. 61003166 and 973 Program of China under Grants No. 2012CB316303.

8. REFERENCES

- [1] S. Cléménçon and N. Vayatis. Ranking the best instances. *JMLR*, 8:2671–2699, 2007.
- [2] J. Huang and C. Guestrin. Riffled independence for ranked data. NIPS’09, 2009.
- [3] J. Huang and C. Guestrin. Learning hierarchical riffle independent groupings from rankings. *JMLR’10*, 2010.
- [4] J. Huang, A. Kapoor, and C. Guestrin. Efficient probabilistic inference with partial ranking queries. *UAI2011*.
- [5] G. Lebanon and J. D. Lafferty. Cranking: Combining rankings using conditional probability models on permutations. *ICML ’02*, pages 363–370, 2002.
- [6] J. I. Marden. *Analyzing and Modeling Rank Data*. Chapman & Hall, 1995.
- [7] S. Niu, J. Guo, Y. Lan, and X. Cheng. Top-k learning to rank: Labeling, ranking and evaluation. *SIGIR’12*.
- [8] T. Qin, X. Geng, and T.-Y. Liu. A new probabilistic model for rank aggregation. NIPS’10, pages 1948–1956. 2010.
- [9] C. Rudin. Ranking with a p-norm push. *COLT’06*, pages 589–604, 2006.
- [10] F. Xia, T.-Y. Liu, and H. Li. Statistical consistency of top-k ranking. NIPS’09, pages 2098–2106. 2009.
- [11] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. *ICML ’08*, pages 1192–1199, 2008.