Information Sciences 457–458 (2018) 76–98

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

An efficient method for the reverse top k search

Jeong-hoon Park[b], Chin-Wan Chung[a,b,*]

[a] Chongqing Liangjiang KAIST International Program, Chongqing University of Technology (CQUT), Chongqing, China
[b] School of Computing, Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea

ARTICLE INFO

Article history:
Received 8 July 2016
Revised 18 February 2018
Accepted 2 May 2018
Available online 9 May 2018

Keywords:
Reverse top k search
RTOPk
Information retrieval

ABSTRACT

The reverse top-k search problem is a variant of the top-k search problem. Suppose that there are a set of objects O and a set of weight vectors W in the d-dimensional space. Then, given a query object, the reverse top-k(*RTOPk*) search is to find a subset W' of W such that, when the query object is inserted into O(or already included in O), the query object is one of the top-k objects for each weight vector in W'. Many studies on the *RTOPk* search have been conducted in the last few years. However, the existing methods for the *RTOPk* search only focus on how to efficiently process evaluations of the weight vectors, which check whether the weight vectors are included in the result without efficient pruning methods. In this paper, we propose an efficient method for the *RTOPk* search, that overcomes drawbacks of existing methods. In contrast to existing methods, the proposed method prunes each unnecessary weight vector without traversing objects, and filters out each unnecessary object without considering the weight vectors. In addition, the proposed method evaluates weight vectors by preserving search contexts for multiple traversals to tree-based indexes of weight vectors and objects. The experimental results based on synthetic datasets and a real dataset show that the proposed method is significantly more efficient than existing methods.

1. Introduction

The top-k search is, given a d-dimensional weight vector and a set of objects represented by d-dimensional vectors, to find the top-k ranked objects having better ranking scores than the others w.r.t the weight vector. Lots of studies on the top-k search [4] and its variants have been conducted for the last decade. Recently, as an important variant of the top-k search, the reverse top-k(*RTOPk*) search was introduced in [12].

In the *RTOPk* search, it is assumed that users represent their interests, by using a d-dimensional weight vector in which the relative importance of each attribute is assigned to the corresponding dimension of the weight vector. For example, suppose that *Car* has 2 attributes in a car database, *Price* and *Quality*. Then, if a user, Adam, represents his interest in *Car* using a 2-dimensional vector (2, 1), it means that Adam considers *Price* is 2 times more important than *Quality*. This model is in agreement with *preference* proposed in [9,17]. For each user, the weight vector representing the user's interest can be automatically made by the customer analysis based on the user information such as product ratings, favorite remarks and purchases. In addition, products are also represented by d-dimensional objects such that, for each dimension, a corresponding attribute value is assigned. Note that the lower the attribute value is, the better it is in this paper. For example, a car

* Corresponding author.
E-mail addresses: pjh83@kaist.ac.kr (J.-h. Park), chungcw@kaist.edu (C.-W. Chung).

https://doi.org/10.1016/j.ins.2018.05.010

Car	Price	Performance Level
x1	4	3
x2	2	4
x3	3	3
x4	5	2

Car Information

User	TOP2 Result (sorted)
u1	x4, x3
u2	x4, x3
u3	x2, x3

TOP2 Results

User	Price	Performance Level
u1	0.3	0.7
u2	0.1	0.9
u3	0.8	0.2

User Interests

Car	RTOP2 Result
x1	(Empty)
x2	u3
x3	u1, u2, u3
x4	u1, u2

RTOP2 Results

Fig. 1. An example of the *TOP*2 search and *RTOP*2 search.

for sale can be represented by (4, 3) meaning that the price is 40,000 dollars and the performance level is 3. As most of the studies on the top-*k* search, the *RTOPk* search uses a ranking function which takes an object and a weight vector as inputs. The ranking function returns the inner product of the input object and the input weight vector. Given a query object, the *RTOPk* search is to find a maximal set of weight vectors *W* such that, for each weight vector *w* in *W*, the query object is included in the result of the top-*k* search based on *w* when the query object is added to the set of objects. Fig. 1 shows an example of the top-*k* search and the *RTOPk* search where *k* is 2. For *u*1, the ranking score of *x*4 is lower than that of *x*3. Therefore, *x*4 is more attractive to *u*1 than *x*3. Since *x*4 is included in the top-2 results of *u*1 and *u*2, *RTOP*2 of *x*4 is {*u*1, *u*2}.

The applications of the *RTOPk* search include a profile-based marketing service as introduced in [12]. In the example of Fig. 1, the profile-based marketing service would be to advertise *x*2 to *u*3, and *x*4 to *u*1 and *u*2 because the query object is considered as a high ranked object for the users in the result of the *RTOPk* search. In addition, an empty result for *x*1 means that *x*1 is not attractive to anyone.

The studies related to the *RTOPk* search are conducted in [7,12–14], and [16]. The *RTOPk* search is firstly proposed in [12]. The method accesses each weight vector and evaluates whether the weight vector is in the result of the *RTOPk* search by performing the top-*k* search. For pruning a weight vector without the top-*k* search, the results of previous evaluations are utilized. However, for the pruning, it is necessary to optimize the access order of the weight vectors, which is an NP-hard problem. The most efficient existing *RTOPk* search method proposed in [16] is a branch and bound algorithm based on two R-trees for the objects and weight vectors, respectively. However, the evaluations of weight vectors always require traversals to the R-tree for objects although some weight vectors can be pruned without the traversals. In addition, each evaluation starts from the root and traverses the R-tree for objects. This causes redundant computations over multiple evaluations.

In this paper, we propose an efficient method for the *RTOPk* search that overcomes the drawbacks of the existing methods. Prior to the evaluations of weight vectors, the proposed method filters out unnecessary weight vectors by using three objects: 1) the object having the *k*th vector length, 2) the virtual object on the *k*th F_{max} that is introduced in Section 6, and 3) the query object. Given a weight vector *w*, the decision of pruning *w* is made by comparing the ranking scores of the three objects. By the comparisons, we can immediately realize whether there exist *k* or more objects which are better than the query object without identifying each of the better objects. In addition, some objects are also pruned without considering the weight vectors by using the domination relationships among objects. It is based on the fact that the objects dominated by the query object or a virtual object defined by the proposed method do not affect the evaluation of any weight vector. The method divides the universal area into multiple cells, and pre-computes the upper and lower bounds of ranking scores for each cell. In the query processing time, the information is used to reduce the search space for the evaluations of weight vectors. Since the pre-computed information is independent of the datasets, it is not necessary to update the information. In addition, during evaluations of weight vectors, the proposed method reuses search contexts to avoid unnecessary recomputations.

Our contributions are as follows:

- **Pruning Weight Vectors and Objects without Traversing Each Other** We devise effective methods for pruning weight vectors without traversing the objects, and for pruning objects without traversing the weight vectors. By pruning weight vectors and objects prior to the evaluation of weight vectors, the number of weight vectors to evaluate is significantly reduced, and the efficiency of each evaluation is improved.
- **An Efficient Algorithm for the *RTOPk* Search by Preserving Search Contexts** In order for the weight vector evaluations not to be performed from scratch, the proposed method maintains the search contexts and reuses them across the evaluations of weight vectors. By reusing the information, the estimations of score bounds for a large portion of objects become unnecessary. Therefore, the proposed method saves the computational cost for such unnecessary tasks.