Top-k Searching for More Users in Multidimensional B-tree with Lists

Matúš Ondrečka [*]
Supervised by Prof. Jaroslav Pokorný

Department of Software Engineering, Faculty of Mathematics and Physics
Charles University, Prague, Czech Republic
ondreicka@ksi.mff.cuni.cz

Abstract. In the last few years, research of top-k processing is in progress in various domains. The aim of our research is efficient top-k searching of the best k objects with more attributes according to complex user preferences. We use a model of user preferences based on fuzzy functions. In our previous research we focus on top-k searching in tree-oriented data structures and we developed various top-k algorithms, which solve top-k problem efficiently and support the model of user preferences. Furthermore, we assume that attributes of an object type can be distributed in various information resources. Therefore, integration of data is one of the topics of our research. We are developing a software system that will be including our new-developed technologies and experiments with real data from various information resources.

1 Introduction

In today's world, users of different systems are trying to find various objects, such as laptops, jobs, holidays, etc. In most cases, these objects are of the same type and have several attributes. Each object has different attribute values. According to the values of these attributes, users are searching for objects that best suit their preferences [9][7][3]. Each user prefers different attribute values. In general, the user is only looking for a few objects that are the best according to his/her preferences. Sometimes the user is looking for only one best object, for example, he/she is looking for a new job.

Nowadays, in many search engines a user can only restrict the values of some attributes. The result of searching of these search engines can be empty-set or a too-large-set of objects. The motivation for searching according to user preferences is to find a few best k objects for the user. In this work, we focus on efficient searching the best k objects with more attributes according to the user preferences.

The problem of searching the best k object according to values of different attributes simultaneously is indicated as a *top-k problem* [11][12].

[*] This work was supported by the grant SVV-2011-263312 and the grant number 9209 of Grant Agency of Charles University (GAUK) in Prague.

Moreover, we use a model of user preferences based on fuzzy functions. We focus on multi-user solution, where data is common for all users and each user can express his/her preferences for each attribute by fuzzy function and mutual relations between the attributes by an aggregation function [9][7].

It was needed to choose a suitable data structure for storing the set of objects, which supports the model of user preferences based on fuzzy functions and where it is possible to solve top-k problem efficiently. Therefore we focused on a using tree-oriented data structures. Furthermore we assume that attributes of an object type can be distributed in various information resources and integration of data is one of the topics of our research.

2 Related work

A trivial solution of the top-k problem is exhaustive searching, which requires to load all relevant objects together with the values of their attributes from the data resources, to evaluate every object's rating according to a rating function, and finally to select k objects with the highest rating.

The exhaustive searching is not efficient solution of top-k problem, because all the objects with attribute values have to be obtained from the information resources. It is a problem especially for a large set of objects.

Most of the research groups are focused on top-K processing techniques in relational databases [6]. In last few years, research of processing top-K queries is in progress in various domains such as XML [8], multimedia search [11], the Web [13], or distributed systems [10].

In our research, we focus on the family of Fagin's algorithms [12], which has been widely studied for efficient computing of top-k queries. These algorithms can find efficiently the best k objects according to an aggregate function @ without loading all the objects. Fagin's algorithms assume that set of objects is stored in lists and the aggregate function @ is monotone. We say that an @ is *monotone* if $@(p_1, ..., p_m) \leq @(q_1, ..., q_m)$, whenever $p_i \leq q_i$, for every $i = 1, ..., m$.

In our research, we focus on searching best k objects with usage of the model of user preferences based on fuzzy functions. The initial source of our research is PhD thesis [5]. In this thesis author uses user preference model [7] based on simple fuzzy functions and brings many improvements of Fagin's algorithms [12], which were published in [4].

There are not too many applications, which solve the top-k problem in environment of various information resources. Authors of [10] describe KLEE framework, which solves the efficient processing of top-k queries in wide-area distributed data repositories. KLEE framework uses approximate algorithms and searches for the best k documents, but only according to an aggregation function.

The goal of our research is to investigate searching the best k objects according to user preferences based on fuzzy functions.

3 Model of user preferences based on fuzzy functions

We use model of user preferences based on fuzzy functions, where each user can express his/her preferences for each attribute by a fuzzy function and mutual relations between the attributes by an aggregation function. We suppose a set of objects X with m attributes A_1, ..., A_m. Every object $x \in X$ has m values $A_1(x), ..., A_m(x)$ of these attributes.

In our work, we use a *rating function* (ranking function), which assigns a rating $R(x)$ for each object $x \in X$. In our work, $R(x)$ maps every object $x \in X$ into interval [0, 1], where 0 means no preference and 1 means the highest preference. According to object ratings it is possible to sort objects from X in descending order and determine the best k objects.

We differentiate between local and global preferences. *Local preferences* reflect how the object is preferred according to only one attribute. We use a fuzzy function f_i, which maps every value of actual attribute A_i domain into [0, 1] interval [9][1]. *Global preferences* express mutual relations between the attributes $A_1, ..., A_m$. For this purpose, we consider an monotone aggregation function [12][9][1].

For example, one user U prefers objects locally by three fuzzy functions (on Figure 1) and globally by aggregation function, which can be *weighted average*, where weights w_1, w_2, w_3 of single attributes determine how the user prefers single attributes, i.e. $R^U(x) = (w_1 \cdot f_1^U(x) + w_2 \cdot f_2^U(x) + w_3 \cdot f_3^U(x))/(w_1 + w_2 + w_3)$.

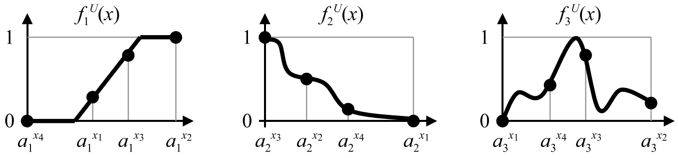


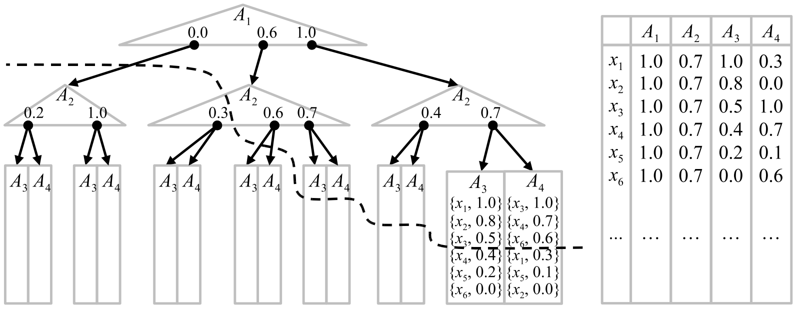
Fig. 1. An example of three fuzzy functions of user U.

4 Searching in multidimensional B-tree with lists

The main goal is to design new top-k algorithms and new data models for efficient top-k problem solving.

In our research, we applied the model of user preferences in combination with Fagin's algorithms [12]. It is possible, when we use a method for sorting objects according to a fuzzy function with using a B$^+$-tree [15].

Since the leaf nodes of the B$^+$-tree are linked in two directions, it is possible to cross the B$^+$-tree through the leaf level and to get all the objects. In this way, it is possible to obtain objects from B$^+$-tree in descending order according to course of user fuzzy function f^U [1][3].



	A_1	A_2	A_3	A_4
x_1	1.0	0.7	1.0	0.3
x_2	1.0	0.7	0.8	0.0
x_3	1.0	0.7	0.5	1.0
x_4	1.0	0.7	0.4	0.7
x_5	1.0	0.7	0.2	0.1
x_6	1.0	0.7	0.0	0.6
...	...	...	...	...

Fig. 2. MDB-tree with lists. Two nominal attributes are stored as MDB-tree and two ordinal attributes are stored as Fagin's lists. Under dotted line, there is a part of the data structure, into which the MXT-algorithm does not access during its computation.

Then we have studied intensively a possibilities of usage a Fagin's algorithms in tree-oriented data structures, in which is possible to index objects according to more attributes. We focused on *multidimensional B-tree* (*MDB-tree*) [14], which consists of the levels, which contain B$^+$-trees.

It was not possible to apply Fagin's algorithms in MDB-tree. Therefore we developed new algorithm *MD-algorithm* [3]. MD-algorithm is based on the depth-first-search and is able to find top-k object in MDB-tree without searching whole the MDB-tree. Furthermore, we have tested the MD-algorithm. Our tests show, that results depend on type of attributes. In general, we differentiate two possible attribute types, i.e., nominal attributes and ordinal attributes.

Therefore, we developed *MXT-algorithm* [2][1], which is based on integration of MD-algorithm and Fagin's algorithms. This new algorithm uses a new tree-oriented data structure *MDB-tree with lists* (see Figure 2). In MDB-tree with lists are nominal attributes stored like in MDB-tree and ordinal attributes like in Fagin's sorted lists [12]. The MXT-algorithm can find top-k objects without searching of all objects and according to the model user preferences.

The use of B$^+$-trees, MDB-tree and MDB-tree with lists allows to dynamise the environment while solving a top-k problem. Moreover, these tree-oriented data structures are independent from user's preferences, i.e. they are common for all users.

4.1 Experiments

Our new top-k algorithms and data models were implemented and tested on real data. The implementation of Fagin's algorithms, MD-algorithm and MXT-algorithm with a usage of tree-oriented data structures have been developed in Java. To evaluate a complexity of these algorithms it is important to estimate the number of obtained objects and the number of accesses into their data structures.

For example, we tested the set of 8 822 flats for rent in Prague [1]. For finding the best $k \leq 32$ objects, MXT-algorithm and MD-algorithm needed to search less than 10% of all objects. Figure 3 shows results of the experiment.

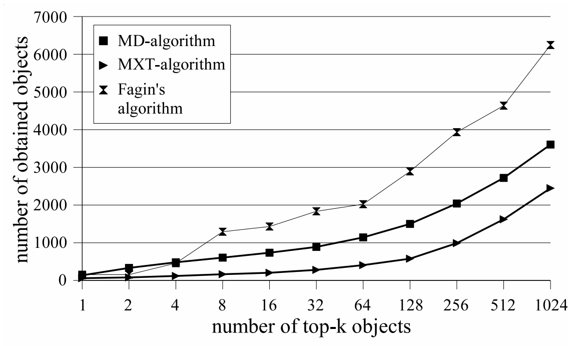


Fig. 3. A searching for the best flats in Prague.

5 Motivation and future plans of our research

There are a number of issues that need to be further investigated.

Improvements of previous solutions

In [5], the authors describe some useful heuristics for Fagin's algorithms. Similarly, a motivation for future research could be to develop some heuristics for our top-K algorithms, which would improve their performance. For example, it is possible to monitor a distribution of the key values in nodes for a set of objects stored in MDB-tree, etc. The use of parallel computing techniques could improve our MXT-algorithm. It could be interesting that some of instances of TA-algorithm would be computed concurrently.

Different models of user preferences

In our research, we used a model of preferences based on local and global user preferences. In future work, we can use user preferences based on different models. When a dependency exists between values of more attributes, a user has to set his/her one local preference for both attributes. In this case, we should evolve some modifications of our top-k algorithms or we should develop a new top-K algorithms, which would be based on absolutely different models of preferences.

Very large data sets

In future work, we want to find an application of our solutions in environments with very large data sets. The used tree-oriented data structures allow to dynamise the environment while solving a top-k problem. In these data structures it possible to update stored data very quickly. A challenging research topic is a top-k searching in data streams. For example, data with the same time stamp from more sensors can compose an object (event). Then we can use the tree-oriented data structures as a sliding window, in which we hold only the best objects according to a user preference.

In a real world of uncertain data, objects with unknown attribute values can exist. In this case, we can not determine the best k objects exactly and we can develope some versions of our top-k algorithms with usage of approximation. Naturally, our tree-oriented data structures will have to be necessarily modified in order to store an uncertain data.

Web environment

Next direction of our research is to experiment with our top-k algorithms in Web environment of more information resources. Some attribute values can be distributed in various information resources. Therefore, a integration of data is one of the topics of our research.

References

1. Ondreička, M., Pokorný J.: Efficient Top-K Problem Solvings for More Users in Tree-Oriented Data Structures. In: Proc. of IEEE Fifth International Conference on Signal Image Technologies and Internet-Based System, Marrakech, Morocco, 2009, pp. 345-354.
2. Ondreička, M., Pokorný J.: Combination of TA- and MD-algorithm for efficient solving of top-K problem according to user's preferences. In: Proc. of DATESO 2009, Špindleruv Mlýn, Czech Republic, April 2009.
3. Ondreička, M., Pokorný J.: Extending Fagin's algorithm for more users based on multidimensional B-tree. In: Proc. of ADBIS 2008, LNCS 5207, 2008, pp. 199-214.
4. Gurský P., Vojtáš P.: Speeding Up the NRA Algorithm, In: Proc. og SUM 2008, Scalable Uncertainty Management, Napoli , Italy, Springer, Sep. 2008, pp. 243-255.
5. Gurský, P.: Searching Top-k objects for many users. PhD Thesis, Pavol Jozef Šafŕik University in Košice, Institute of computer science, Slovakia, 2008.
6. Ilyas, I. F., Beskales, G., Soliman, M. A.: A survey of top-k query processing techniques in relational database systems. ACM Comput. Surv. 40, 4, 2008, 1-58.
7. Gurský P., Vaneková V., Pribolová J.: Fuzzy User Preference Model for Top-k Search. In: Proc. of IEEE World Congress on Computational Intelligence, Hong Kong, FS0377, 2008.
8. Theobald, M., Bast, H., Majumdar, D., Schenkel, R., Weikum, G.: TopX: efficient and versatile top-k query processing for semistructured data. The VLDB Journal, Vol. 17, No. 1, January 2008, pp. 81-115.
9. Eckhardt, A., Pokorný, J., Vojtáš, P.: A system recommending top-k objects for multiple users preference. In: Proc. of 2007 IEEE International Conference on Fuzzy Systems, July 24-26, 2007, London, England, pp. 1101-1106.
10. Michel, S., Triantafillou, P., and Weikum, G.: KLEE: a framework for distributed top-k query algorithms. In: Proc. of the 31st international Conference on Very Large Data Bases, Trondheim, Norway, 2005, VLDB Endowment, pp. 637-648.
11. Chaudhuri, S., Gravano, L., Marian, M.: Optimizing Top-k Selection Queries over Multimedia Repositories. IEEE Trans. On Knowledge and Data Engineering, Vol. 16, No. 8, 2004, pp. 992-1009.
12. Fagin, R., Lotem, A., Naor, M.: Optimal aggregation algorithms for middleware. Journal of Computer and System Sciences 66, 2003, pp. 614-656.
13. Bruno, N., L. Gravano, L., Marian, A.: Evaluating top-k queries over web-accessible databases. In: Proc. of ICDE, 2002, pp. 369 - 380.
14. Scheuerman, P., Ouksel, M.: Multidimensional B-trees for associative searching in database systems. Information systems, Vol. 34, No. 2, 1982, pp. 123-137.
15. Comer, D.: The Ubiquitous B-Tree. ACM Computing Surveys, Vol. 2, No. 11, June 1979, pp. 121-138.