Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 94 (2016) 191 – 198

Procedia Computer Science

The 13th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2016)

A Framework for Evaluating Skyline Queries over Incomplete Data

Yonis Gulzar[a], Ali A. Alwan[a1], Norsaremah Salleh[a], Imad Fakhri Al Shaikhli[a], Syed Idrees Mairaj Alvi[b]

[a]*International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia*
[b]*NXP Semiconductors, Manathaya Technology Park, Nagavara, Bengaluru, 560045, India*
yonis.gulzar@live.iium.edu.my, aliamer@iium.edu.my, norsaremah@iium.edu.my, imadf@iium.edu.my, idrisalvi@gmail.com

Abstract

Research interest in skyline queries has been significantly increased over the years, as skyline queries can be utilized in many contemporary applications, such as multi-criteria decision-making system, decision support system, recommendation system, data mining, and personalized systems. Skyline queries return data item that is not dominated by any other data items in all dimensions (attributes). Most of the existing skyline approaches assumed that database is complete and values are present during the skyline process. However, such assumption is not always to be true, particularly in a real world database where values of data item might not be available (missing) in one or more dimensions. Thus, the incompleteness of the data impacts negatively on skyline process due to losing the transitivity property which leads into the issue of cyclic dominance. Therefore, applying skyline technique directly on an incomplete database is prohibitive and might result into exhaustive pairwise comparison. This paper presents an approach that efficiently evaluates skyline queries in incomplete database. The approach aims at reducing the number of pairwise comparisons and shortens the searching space in identifying the skylines. Several experiments have been conducted to demonstrate that our approach outperforms the previous approach through producing a lower number of pairwise comparisons. Furthermore, the result also illustrates that our approach is scalable and efficient.

Peer-review under responsibility of the Conference Program Chairs

Keywords: Skyline; Skyline queries; Incomplete data; Preference queries; Query processing

[1] Ali A. Alwan. Tel.: +60-36196-6421; fax: +60-36196-5179.
E-mail address: aliamer@iium.edu.my

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.08.030

1. Introduction

Skyline queries try to retrieve the relevant data from the database based on the given user preferences in the submitted query. Given a set of dimensions, skyline queries attempt to identify the set of non-dominated data items called skylines. A data item *a* dominates another data item *b* if and only if *a* is not worse than *b* in all dimensions and better than *b* in at least one dimension. For instance, a tourist seeking for a hotel in a specific area that is near to a beach and at the same time is cheap in price, among the set of available hotels, skyline queries would return only those non-dominated hotels that meet the tourist's preferences.

It has been very obvious that skyline queries are very beneficial and most widely used in the contemporary database applications such as multi-criteria decision-making system, decision support system[1,2,3,4], personalized systems, recommendation system, data mining, e-commerce[5], hotel recommender[6], restaurant finder[7,8]. Apparently, due to the practical use of skyline queries that can be found in many real life modern database applications, a variety of skyline approaches have been proposed in the database literature. Many variations of skyline technique have been proposed to serve skyline queries such as *k*-dominance[2], top-*k* dominating[3], *k*-frequency[1]. Skyline technique has been frequently adopted in many applications due to its practical solutions over many applications and has many benefits including (i) does not involve any user-defined ranking function as each dimension is treated independently, (ii) The size of database and the number of dimensions has no significant impact on skyline result, (iii) Integrating skyline operator into SQL is extremely simple, (iv) Skyline queries relying on actual data, (v) skyline queries always retrieve result to the user. The main issue focus when computing skyline queries in a database is diminishing the searching space as low as possible and concentrating only on those data items with the high potential to be retrieved as skylines.

Most of the previous skyline approaches assumed that database is complete and values are present during skyline process. However, this is not always necessary the case, particularly for a database with a large number of dimensions and massive size of data, as values might not be available (missing) in one or more dimensions. Hence, the incompleteness of the data raises new challenges on processing skyline query due to losing the transitivity property and the facing the problem of cyclic dominance. Applying skyline technique directly on a database with incomplete data is impractical and produced a huge number of pairwise comparisons. For example, a person is looking for a hotel in a city that is closer to a beach with the lowest price and high rating. A hotel h_i consists of three dimensions (pr_i, ds_i, rt_i). Where pr_i is the price per night, ds_i is the distance from the hotel to beach and rt_i is the rating of the hotel. We assume that the hotel database consists of 3 data items (tuples) with missing dimension values namely h_1(150, 5, ?), h_2(?, 9, 2), and h_3(80, ?, 3). The symbol (?) represents the missing dimensions of the data items. Based on the common dimensions with non-missing values h_1 dominates h_2 as h_1 is better than h_2 (lower is better) (second dimension), while h_2 dominates h_3 as h_2 is less than h_3 (third dimension). However, comparing h_1 against h_3 indicates that h_3 dominates h_1. Thus, the hotel h_1 does not dominate h_3 which therefore means the dominance relation is not transitive. In addition, hotel h_3 dominates h_1 which means that the dominance relation is cyclic. From this example, all these three hotels are being dominated and thus the process of comparison failed to determine the best hotel in the database as skyline.

In this paper, we present a framework for evaluating skyline queries over an incomplete database. The framework comprises of four components, namely: Data Sorting and Arrays Constructor, Data Filter, Candidate Skyline Identifier and Final Skyline Identifier.

The rest of the paper is structured as follows. In Section 2, the previous works related to this research are reported. The basic definitions and notations, which are used in the rest of the paper, are set out in Section 3. The proposed framework is illustrated in Section 4. Experiment result has been reported in Section 5. The conclusion is explained in Section 6.

2. Related Works

Various approaches have been proposed in the literature for processing skyline queries. Most of the proposed works concentrated on enhancing the efficiency of the skyline processes and improving its performance. For this reason, most of the researches in the area of skyline queries have focused on developing approaches pruning the searching space of a large database into a small number of interested data by deleting dominated data items.