Chapter 3
Evaluating Top-k Skyline Queries on R-Trees

Marlene Goncalves
Universidad Simón Bolívar, Venezuela

Fabiana Reggio
Universidad Simón Bolívar, Venezuela

Krisvely Varela
Universidad Simón Bolívar, Venezuela

ABSTRACT

The Skyline queries retrieve a set of data whose elements are incomparable in terms of multiple user-defined criteria. In addition, Top-k Skyline queries filter the best k Skyline points where k is the number of answers desired by the user. Several index-based algorithms have been proposed for the evaluation of Top-k Skyline queries. These algorithms make use of indexes defined on a single attribute and they require an index for each user-defined criterion. In traditional databases, the use of multidimensional indices has shown that may improve the performance of database queries. In this chapter, three pruning criteria were defined and several algorithms were developed to evaluate Top-k Skyline queries. The proposed algorithms are based on a multidimensional index, pruning criteria and the strategies Depth First Search and Breadth First Search. Finally, an experimental study was conducted in this chapter to analyze the performance and answer quality of the proposed algorithms.

INTRODUCTION

In the last decade, many researchers have been interested in the problem of Skyline query evaluation because this kind of queries allows to filter relevant data from high volumes of data. A Skyline query selects those data that are non-dominated according to multiple user-defined criteria which induce a partial order over the data (Börzsönyi, Kossmann, & Stocker, 2001). It is said that one point *a* dominates another point *b* if *a* is as good or better than *b* for all criteria and strictly better than *b* in at least one cri-

DOI: 10.4018/978-1-4666-8767-7.ch003

terion. Skyline is also known as Pareto Curve or Maximal Vector Problem (Bentley, Kung, Schkolnick, & Thompson, 1978; Kung, Luccio & Preparata, 1975; Papadimitriou & Yannakakis, 2001; Preparata & Shamos, 1985).

However, the Skyline set may be huge because its size increases as the number of user-defined criteria augments (Bentley et al., 1978). The estimated Skyline size assuming independent dimensions is $O(\ln^{d-1}n)$ where n is the data size and d is the number of user-defined criteria (Bentley et al., 1978). Moreover, the user might require exactly k points on the result and, it is not possible for Skyline to discriminate among the answers because they are all optimal. To identify the best k Skyline points, Top-k Skyline has been proposed as a language that integrates Skyline and Top-k in order to retrieve exactly the best k points from the Skyline set based on a total order function (Goncalves & Vidal, 2009; Chan, Jagadish, Tan, Tung, & Zhang, 2006b; Lin, Yuan, Zhang, & Zhang, 2007). Particularly, Goncalves and Vidal (2012) define Top-k Skyline queries in terms of the Euclidean distance function with respect to a boundary condition defined by the user, i.e., a point belongs to the Top-k Skyline set if it is Skyline and it is one of the k nearest neighbors to the boundary condition. Also, k-Dominant Skyline (Chan, Jagadish, Tan, Tung, & Zhang, 2006a), Skyline Frequency (Chan et al., 2006b) and k Representative Skyline (Lin et al., 2007) are functions in order to measure the interestingness of each Skyline point. The Skyline Frequency ranks Skyline in terms of the number of times in which a Skyline point belongs to a non-empty subset or subspace of the multidimensional function; the user defined criteria is specified by a multidimensional function. The k-Dominant Skyline identifies Skyline points in $k \leq d$ dimensions of the multidimensional function. The k Representative Skyline produces the k Skyline points that have the maximal number of dominated points.

On the other hand, several existing algorithms make use of indexes defined on each user-defined criterion in order to evaluate a Top-k Skyline query (Goncalves & Vidal, 2012; Alvarado, Baldizan, Goncalves, & Vidal, 2013). In traditional databases, the use of multidimensional indexes has shown that can improve the query performance (Manolopoulos, Nanopoulos, Papadopoulos, & Theodoridis; 2013). In this chapter, R-tree based algorithms to evaluate Top-k Skyline queries are proposed where an R-tree is a multidimensional index structure that organizes the points by the closeness to each other and whose average-case search time is logarithmic (Göbel, 2007; Guttman, 1984). This index structure is a suitable to return points sorted by distance.

The proposed algorithms in this chapter apply two strategies for traversing the R-trees. These strategies are DFS (Depth First Search) and BFS (Breadth First Search) (Knuth, 1997). In addition, three pruning criteria are incorporated into the proposed algorithms in order to discard those R-tree regions in which there are not Skyline points. This way, if fewer regions are accessed because the R-tree is pruned using some pruning criterion, the algorithms will consume less time to return the response.

Finally, an experimental study on synthetic data applying our proposed algorithms was conducted in this chapter. Experimental results reveal that the BFS-based algorithms have better runtime than the DFS-based ones, except for the case with correlated data. Additionally, the pruning based algorithms typically require less time but can lose up to 18.9% of Skyline points. Therefore, the pruning criteria may reduce runtime of the algorithms although they may not produce complete answers.

This chapter is comprised of five sections in addition to section I that introduces the problem. Section II presents a motivating example and describes existing state-of-the-art approaches to compute Top-k Skyline queries. Section III defines the Top-k Skyline approach and the proposed algorithms that are able to identify the subset of the Skyline that will be required to produce the top-k objects. In Section IV, the quality and performance of the proposed techniques will be empirically evaluated. First, the execu-