Efficiently Producing the K Nearest Neighbors in the Skyline for Multidimensional Datasets

Marlene Goncalves and Maria-Esther Vidal

Universidad Simón Bolívar, Venezuela
{mgoncalves,mvidal}@ldc.usb.ve

Abstract. We propose a hybrid approach that combines Skyline and Top-k solutions, and develop an algorithm named k-NNSkyline. The proposed algorithm exploits properties of monotonic distance metrics, and identifies among the skyline tuples, the k ones with the lowest values of the distance metric, i.e., the k nearest incomparable neighbors. Empirically, we study the behavior of k-NNSkyline in both synthetic and real-world datasets; our results suggest that k-NNSkyline outperforms existing solutions by up to three orders of magnitude.

1 Introduction

Nowadays, Web based infrastructures have been developed and allow large datasets to be published and accessed from any node of the Internet. Although the democratization of the information provides the basis to manage large volumes of data, there are still applications where it is important to efficiently identify only the best tuples that satisfy a user requirement. Based on related work, we devised a solution to this ranking problem and developed techniques able to identify the nearest neighbors to a user query which are non-dominated by any other element in the dataset. The set of non-dominated points is known as a skyline, i.e., set of points such that, none of them is better than the rest [1–4]. We developed an algorithm named k-NNSkyline to decide which of the skyline points are the nearest points to the input query; properties of a distance metric are exploited by the algorithm to avoid the computation of the whole skyline and minimize execution time as well as the number of probes required to output the k-nearest neighbors. k-NNSkyline assumes that elements or points in a dataset are characterized by multidimensions. Points are stored following the vertical partition approach; points are maintained ordered and indexed. Additionally, the algorithm maintains information about the worst values seen so far; furthermore, the multidimensional values of the k-th best points seen so far are registered. These registered values are used to stop traversing the tables while completeness and correctness are ensured, and the number of probes and executing time are minimized. We empirically studied the properties of k-NNSkyline with respect to existing approaches; results suggest that k-NNSkyline may reduce execution time by up to three orders of magnitude in both real-world and correlated synthetic data.

P. Herrero et al. (Eds.): OTM 2012 Workshops, LNCS 7567, pp. 673–676, 2012.

This paper is composed of three additional sections. Section 2 summarizes our approach and section 3 reports the results of the empirical evaluation. Finally, we conclude in section 4 with an outlook to future work.

2 Our Approach

A *database* is a set of multidimensional points which are univocally identified. A *query* is comprised of: *i*) a *bound condition* or list of bounds on attributes of the multidimensional dataset, *ii*) a monotonic distance metric ρ, and *iii*) a natural number k. The *answer of a query* q corresponds to the k points in the multidimensional dataset D that are incomparable and that have the minimal values of the distance metric to the query bound condition. The set of incomparable points is known as *skyline*, and it is composed of all the points p, such that: *i*) there is not other point p' in D with values better or equal than p in all the attributes of p, and *ii*) other points in the skyline are better than p in at least one attribute.

Finally, the k *nearest incomparable neighbors* of a query q in D, correspond to a list L of k points in D, where: *i*) points in L are incomparable, i.e., they are part of the skyline, *ii*) there is no point p'' in the skyline, such that, p'' is not in L and the values of the distance metric ρ of p'' is lower or equal than at least one point in L, i.e., there is not point p''' in L and $\rho(p'') \geq \rho(p''')$.

To illustrate our approach suppose a customer is interested in selecting touristic packages for visiting the best museums and gardens in Rome; museums and gardens are both equally important for her. Further, she just wants to see a limited number of places, so a touristic package is preferred for her, if the number of museums and the number of gardens that can be visited are both lower than 10, i.e., the query bound condition establishes that (Museums $\leq$ 10, Gardens $\leq$ 10). Following these criteria a package with numbers of both museums and gardens lower than 10 will be preferred, if there is no other package with a greater number of museums and gardens. This set of packages will compose the skyline. In order to have a good selection of packages for taking a decision, the customer wants to check the 3 packages that best meet her conditions, i.e., she wants 3 nearest incomparable neighbors computed in terms of the Euclidean distance metric. The answer to this query corresponds to the 3 nearest incomparable neighbors such that, there is no package that dominates them, and there is no package with lower values of the Euclidean distance metric to the bound condition of the query.

We propose the k-NNSkyline algorithm to compute the k nearest incomparable neighbors of a query q in a multidimensional dataset D. k-NNSkyline exploits the properties of the query monotonic distance metric, and identifies among the set of skyline of points, the ones with the lowest values of this metric. The result of executing k-NNSkyline is the top-k incomparable points that satisfy the query bound condition, and with the lowest values of the distance metric, i.e., the k nearest incomparable neighbors with respect to q. k-NNSkyline assumes that data are stored following a vertically partitioned table representation, i.e., for each dimension a,

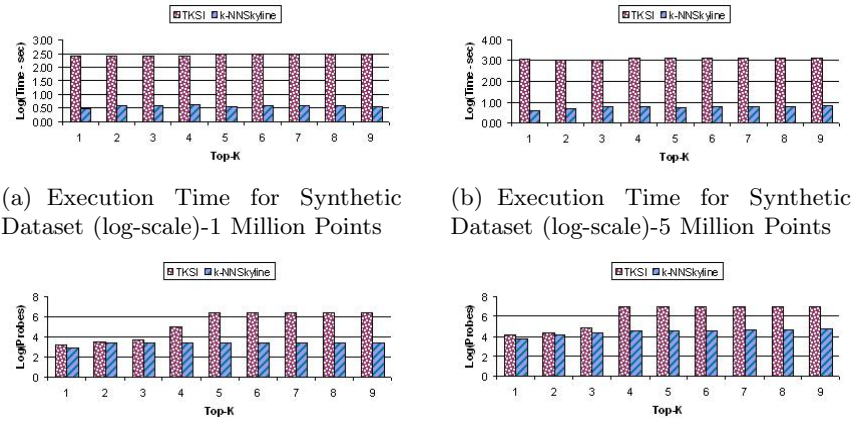
there exists a relation aR composed of two attributes, $PointId$ and $aValue$, that correspond to the point identifier and the value of the dimension a. Each table is indexed by two indices, one on the attribute $PointId$ and the other on $aValue$; points are ordered in aR based on the values of a. k-NNSkyline implements an index based algorithm to minimize the number of probes between data dimensions to compute the skyline as well as the number of evaluations of the monotonic distance metric. k-NNSkyline relies on the indices of the vertically partitioned tables to perform an index scan; also, the ordered scan operator is implemented to retrieve the points. k-NNSkyline iterates over the different vertically partitioned tables to identify the points that have the best values, i.e., there is no other point that dominates these points. The algorithm records in a structure named *WVSF*: *i*) the *worst values* seen so far for each dimension, *ii*) the *points associated with* these values, and *iii*) the k best points seen so far. k-NNSkyline works in iterations, where the best entry(ries) in each of the vertical tables is(are) considered in one iteration. *WVSF* is updated whenever: *i*) values of a seen point are worse than the values registered in *WVSF*, *ii*) new associated points are added to the structure, or *iii*) a point with lowest values of ρ is found. The algorithm stops when a fixed-point on *WVSF* is reached.

3 Experimental Study

We empirically analyze and report on the performance of the k-NNSkyline with respect to the Top-k Skyline algorithm TKSI [3]. We considered both synthetic and real-world data. A data generator was used to build 2 synthetic datasets with 1,000,000 and 5,000,000 of points, respectively. Points are univocally identified and have ten dimensions whose values range from 0.0 to 1.0. Five of the dimensions are highly correlated to the other five, i.e., the correlation between the i-th and (i+5)-th dimensions is higher than 90%. A point dimension may have duplicated values. Furthermore, real-world datasets are composed of 1,061,664 authors and their publications were available at the DBLP site in January 2012 [1]. Queries have 10 dimensions, and k varies from 1 to 10.

Figures 1(a) and (b) report on the execution time and Figures 1(c) and (d) show the number of probes. In both cases, k-NNSkyline is compared to TKSI. In general, we can observe that k-NNSkyline outperforms TKSI in the number of probes and execution time by up to three orders of magnitude. Based on information recorded in WVSF, the k-NNSkyline algorithm is able to reach a fixed-point by checking in average 50% of the Skyline points. Contrary, TKSI is not able to detect the top-k skyline points until the whole skyline is scanned. Additionally, in presence of small skylines and large number of dominated points, TKSI may probe a point with the others to detect if this is a dominated point; this increased the number of probes. Furthermore, TKSI requires to pre-compute the distance metric values for all data which represents up to 60% of the total

[1] `http://www.informatik.uni-trier.de/~ley/db/`



(a) Execution Time for Synthetic Dataset (log-scale)-1 Million Points

(b) Execution Time for Synthetic Dataset (log-scale)-5 Million Points

(c) Number of Probes for Synthetic Dataset (log-scale)-1 Million Points

(d) Number of Probes for Synthetic Dataset(log-scale)- 5 Million Points

Fig. 1. Execution Time and Number of Probes (log-scale)-Synthetic Data

execution time. We also evaluate these algorithms on research publications available from the DBLP website; our results confirm that k-NNSkyline outperforms TKSI algorithm by at least two orders of magnitude.

4 Conclusions and Future Work

We have casted the problem of locating the k nearest incomparable neighbors into Top-k Skyline, and proposed a ranking algorithm that provides an efficient solution to this problem. Empirically we studied the performance of our solution on real-world and synthetic data. Experimental results suggest that our algorithm may overcome existing approaches by up to three orders of magnitude. In the future we plan to exploit properties of variations of R-trees to improve the performance of our approach.

References

1. Chen, L., Lian, X.: Dynamic skyline queries in metric spaces. In: EDBT, pp. 333–343 (2008)
2. Fuhry, D., Jin, R., Zhang, D.: Efficient skyline computation in metric space. In: EDBT, pp. 1042–1051 (2009)
3. Goncalves, M., Vidal, M.-E.: Reaching the Top of the Skyline: An Efficient Indexed Algorithm for Top-k Skyline Queries. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2009. LNCS, vol. 5690, pp. 471–485. Springer, Heidelberg (2009)
4. Skopal, T., Lokoc, J.: Answering metric skyline queries by pm-tree. In: DATESO, pp. 22–37 (2010)