Chapter 7
Top-k queries

Abstract Recently there has been an increased interest in database management systems to incorporate and support more flexible query operators, such as top-k, that produce results of specified cardinality, thus avoiding huge and overwhelming result sets. Top-k queries retrieve the objects that best match the user requirements by employing user-specified scoring functions that result in an ordered set of objects containing the best k objects only [30, 75]. In this chapter, efficient processing of top-k queries in peer-to-peer systems is studied. To this end, the applicability of the skyline operator is investigated for efficiently answering top-k queries for a wide class of scoring functions, indicating user-specified preferences, in large P2P networks.

7.1 Overview

A number of applications can significantly benefit from support for top-k query processing, for example multimedia retrieval (including images) [31, 63], digital libraries [94, 95], web search [97], and e-commerce [96]. Consider for example online booking systems, e.g. travel and accommodation, where the user is only interested in the best offers (air-tickets, hotels) according to a set of dynamic, user-specified criteria. Due to applications and systems such as sensor networks, data streams, and peer-to-peer (P2P) systems, data generation and storage is becoming increasingly distributed. Thus an emerging challenge is to support top-k query processing over a highly distributed network of collaborative computers (i.e., servers or peers).

The main focus of this chapter is on top-k query processing in P2P systems. Each user may define his/her own arbitrary preferences for each query, therefore the queries are not necessarily re-occurring. The challenge is to provide efficient algorithms for processing top-k queries, i.e., queries that return only the exact best k results to the user. SPEERTO is a framework that supports top-k query processing over horizontally partitioned data stored on peers organized in a super-peer network. Users are allowed to specify a monotone function for each query that aggregates a
certain number of the objects characteristics into a single score that defines a total ordering, and therefore enables the retrieval of top-\(k\) results. SPEERTO supports a large class of scoring functions and uses the skyline set [21] for answering top-\(k\) queries. For a maximum value of \(K\), denoting an upper bound on the number of results requested by any top-\(k\) query \((k \leq K)\), each peer computes its \(K\)-skyband [103] as a pre-processing step. Each super-peer maintains and aggregates the \(K\)-skyband sets of its peers to answer any incoming top-\(k\) query. By exchanging skyline sets (a skyline is a subset of the \(K\)-skyband set) at super-peer level, SPEERTO always provides the exact and complete result set in a progressive way, while queries are deliberately routed to those super-peers that actually contribute to the top-\(k\) result.

The rest of this chapter is organized as follows: In Section 7.2, the local data summaries and query processing are described. In Section 7.3, the construction of the skyline-based routing mechanism for top-\(k\) query processing over a super-peer architecture is presented. Thereafter, in Section 7.4 the threshold-based top-\(k\) algorithm is presented. Section 7.5 reviews the related work, and finally Section 7.6 provides a brief summary of the contents of this chapter.

7.2 Local Data Summaries and Query Processing

The result of top-\(k\) queries for any increasingly monotone function can be answered using the \(K\)-skyband (where \(k \leq K\)). The \(K\)-skyband is a set of points, such that there exists no other point that can belong to the result of any top-\(k\) query for any increasingly monotone function. Therefore, the \(K\)-skyband can be used as data summary of a peer’s data, in the case of top-\(k\) queries with \(k \leq K\).

When a peer joins the P2P network, the peer computes the \(K\)-skyband of its local data. Each super-peer gathers the \(K\)-skyband sets from its peers and merges the individual \(K\)-skyband sets by discarding points that are dominated by more than \(K - 1\) points. In this way a super-peer maintains the aggregated \(K\)-skyband set of all data stored at its peers, and each super-peer is capable to answer any incoming top-\(k\) query over its peers’ data.

It should be stressed that even though the skyline operator and the \(K\)-skyband are CPU-intensive [29] and therefore more costly than a top-\(k\) query, they are only computed as a pre-processing step, i.e., their construction is a one-time cost, and then any top-\(k\) query with arbitrary \(k\) \((k \leq K)\) and scoring function can be processed. Thus, during local query processing a super-peer that is queried executes a top-\(k\) query on the locally stored \(K\)-skyband points.

7.3 Routing Summaries

Given the \(K\)-skyband at each super-peer, there exist two naive solutions to process top-\(k\) queries over the super-peer network. In the first, each super-peer broadcasts its