Querying and Mining Data Streams: You Only Get One Look

A Tutorial

Minos Garofalakis
Bell Labs, Lucent
minos@bell-labs.com

Johannes Gehrke
Cornell University
johannes@cs.cornell.edu

Rajeev Rastogi
Bell Labs, Lucent
rastogi@bell-labs.com

1. MOTIVATION AND SUMMARY

Traditional Database Management Systems (DBMS) software is built on the concept of *persistent data sets*, that are stored reliably in stable storage and queried/updated several times throughout their lifetime. For several emerging application domains, however, data arrives and needs to be processed on a continuous (24×7) basis, without the benefit of several passes over a static, persistent data image. Such *continuous data streams* arise naturally, for example, in the network installations of large Telecom and Internet service providers where detailed usage information (Call-Detail-Records (CDRs), SNMP/RMON packet-flow data, etc.) from different parts of the underlying network needs to be continuously collected and analyzed for interesting trends. Other applications that generate rapid, continuous and large volumes of stream data include transactions in retail chains, ATM and credit card operations in banks, financial tickers, Web server log records, etc. In most such applications, the data stream is actually accumulated and archived in the DBMS of a (perhaps, off-site) data warehouse, often making access to the archived data prohibitively expensive. Further, the ability to make decisions and infer interesting patterns *on-line* (i.e., as the data stream arrives) is crucial for several mission-critical tasks that can have significant dollar value for a large corporation (e.g., telecom fraud detection). As a result, recent years have witnessed an increasing interest in designing data-processing algorithms that work over continuous data streams, i.e., algorithms that provide results to user queries while looking at the relevant data items *only once and in a fixed order* (determined by the stream-arrival pattern).

Two key parameters for query processing over continuous data-streams are (1) the amount of *memory* made available to the on-line algorithm, and (2) the *per-item processing time* required by the query processor. The former constitutes an important constraint on the design of stream processing algorithms, since in a typical streaming environment, only limited memory resources are available to the query-processing algorithms. In these situations, we need algorithms that can summarize the data stream(s) involved in a concise, but reasonably accurate, *synopsis* that can be stored in the allotted (small) amount of memory and can be used to provide *approximate answers* to user queries along with some reasonable guarantees on the quality of the approximation. Such approximate, on-line query answers are particularly well-suited to the exploratory nature of most data-stream processing applications such as, e.g., trend analysis and fraud/anomaly detection in telecom-network data, where the goal is to identify generic, interesting or "out-of-the-ordinary" patterns rather than provide results that are exact to the last decimal.

The objective of this tutorial is to provide a comprehensive and, at the same time, clear and meaningful overview of the key research results surrounding data stream processing at this point in time. In addition to the systematic coverage of the most recent research results in the area, our discussion will focus on:

- One-pass algorithms for constructing the most popular types of data synopses over streams (namely, random samples, histograms, sketches, and wavelets).
- Use of the on-line data summaries for processing complex queries.
- An overview of advanced techniques and promising directions for future research.

2. TUTORIAL OUTLINE

- *Introduction:* Basic stream-processing models and architectures; motivating applications.
- *Basic Data Stream Summarization Algorithms:* Samples, quantiles/histograms, sketches, wavelets over streaming data.
- *Processing Queries on Streams:* Using sketches for self-joins, binary joins, and complex joins over data streams; estimating correlated aggregates; using histogram and wavelet synopses for approximate-query processing.
- *Mining High-speed Data Streams:* Single-pass algorithms for association-rule discovery, clustering, and decision-tree construction over data streams.
- *Advanced Topics and Future Research Directions:* Hot-list maintenance; estimating the number of distinct values; sliding-window processing models; multi-dimensional stream synopses; content-based filtering and routing of streaming XML documents.

ACM SIGMOD '2002 June 4-6, Madison, Wisconsin, USA