On Reservoir Sampling with Deletions

Rainer Gemulla[1], Wolfgang Lehner[1], Peter J. Haas[2]

[1] Technische Universität Dresden, Germany, {gemulla,lehner}@inf.tu-dresden.de
[2] IBM Almaden Research Center, USA, phaas@us.ibm.com

Abstract. Perhaps the most flexible synopsis of a database is a random sample of the data; such samples are widely used to speed up processing of analytic queries and data-mining tasks, enhance query optimization, and facilitate information integration. In this paper, we describe a recently proposed method for incrementally maintaining a uniform random sample of the items in a dataset in the presence of an arbitrary sequence of insertions and deletions. Our scheme, called "random pairing" (RP), maintains a bounded-size uniform sample by using newly inserted data items to compensate for previous deletions. The RP algorithm is the first extension of the almost 40-year-old reservoir sampling algorithm to handle deletions; RP reduces to the "passive" algorithm in [1] when the insertions and deletions correspond to a moving window over a data stream. We also prove that it is not possible to "resize" a bounded-size random sample upwards without accessing the base data.

Keywords: Reservoir sampling, Sample maintenance, Synopsis

1 Introduction

Because of its flexibility, sampling is widely used for quick approximate query answering, statistics estimation, data stream processing, data mining, and data integration. Uniform random sampling, in which all samples of the same size are equally likely, is the most basic of the available sampling schemes. Uniform sampling is ubiquitous in applications: most statistical estimators — as well as the confidence-bound formulas for these estimators — assume an underlying uniform sample. Thus uniformity is a must if it is not known in advance how the sample will be used. In this paper, we show how to maintain a bounded-size sample of a dataset defined by a stream of insertion and deletion transactions. Incremental sample maintenance is a powerful technique, because the abstract notion of the underlying "dataset" can be interpreted very broadly in applications. Indeed, the dataset can actually be an arbitrary view, e.g., over the result of an arbitrary SQL query. Samples over views are particularly good candidates for incremental maintenance, because producing such samples on the fly can require very expensive base-data accesses. The idea is to, in effect, compute the "delta" (set of insertions, updates, and deletions) to the view as the underlying tables are updated and then apply general sample-maintenance methods to the resulting sequence of view modifications.

In the context of a data stream management system (DSMS), often only a subset of the data stream is relevant for query processing. On the one hand, windowing techniques [1] restrict the relevant section of the stream to the most recent elements, where recency is defined either by the position of the elements in the stream or by a timestamp associated with each element. On the other hand, suppose that the stream itself consists of updates to a set of items such as the locations of cars on a highway. Many queries only focus on a certain area of interest, say, a specific section of the highway. In both cases, the scope of the query is continously evolving, that is, items enter and leave. Viewing the query scope as a dataset, each element of the stream can be seen as one or more insertions into or deletions from this dataset. Due to the vast number of data streams and/or the high arrival rate of their elements, it is often necessary to compress the relevant part of the data stream to fit into memory or to reduce processing cost. Again, sampling has proven to be a powerful tool for this type of dataset summarization.

This paper briefly describes a recently proposed method [2] for incrementally maintaining a uniform random sample of an evolving dataset. We also show that it is impossible to "resize" a bounded-size random sample upwards without accessing the base data.