Optimizing Sample Design for Approximate Query Processing

Philipp Rösch, Business Intelligence Practice, SAP Research, Dresden, Germany
Wolfgang Lehner, Database Technology Research Group, Dresden University of Technology, Dresden, Germany

ABSTRACT

The rapid increase of data volumes makes sampling a crucial component of modern data management systems. Although there is a large body of work on database sampling, the problem of automatically determining the optimal sample for a given query remained (almost) unaddressed. To tackle this problem, the authors propose a sample advisor based on a novel cost model. Primarily designed for advising samples of a few queries specified by an expert, the authors additionally propose two extensions of the sample advisor. The first extension enhances the applicability by utilizing recorded workload information and taking memory bounds into account. The second extension increases the effectiveness by merging samples in case of overlapping pieces of sample advice. For both extensions, the authors present exact and heuristic solutions. Within their evaluation, the authors analyze the properties of the cost model and demonstrate the effectiveness and the efficiency of the heuristic solutions with a variety of experiments.

Keywords: Advisor Tools, Approximate Query Processing, Computer Science, Information Systems, Random Sampling, Sampling Error Types

INTRODUCTION

Recent studies revealed a rapid multidimensional growth in current data warehouse databases: The size of the data triples every two years and the number of queries doubles each year. Additionally, the complexity of the queries increases—a lot of queries are complex aggregation queries with joins. These queries are of explorative nature, that is, users browse through the data and search for interesting regions. (Winter, 2008)

In order to make the data exploration reasonably applicable, short response times are essential. However, the rapid growth of data warehouse systems conflicts with the need for short response times. A common solution for this problem is the use of synopses. Synopses are concise representations that reflect the characteristics of the underlying data, and they allow for approximate query processing with significantly shorter response times. In the database literature, several kinds of synopses have been proposed like histograms (Ioannidis & Poosala, 1999; Poosala, Ioannidis, Haas, & Shekita, 1996), wavelets (Chakrabarti, Garo-
falakis, Rastogi, & Shim, 2000; Matias, Vitter, & Wang, 1998) or random samples (Vitter, 1985). From these synopses, random samples have proven to be a good choice: They provide probabilistic error bounds and they can easily be integrated into database systems. Initially, samples were mostly used for query optimization. In the last 10 years, however, the focus of database sampling has shifted more and more towards approximate query processing. Here, several techniques for online sampling have been proposed (Hellerstein, Haas, & Wang, 1997; Haas & Hellerstein, 1999; Jermaine, Dobra, Arumugam, Joshi, & Pol, 2005; Jermaine, Arumugam, Pol, & Dobra, 2007). However, the major part of research focuses on precomputed and materialized —that is, offline— sampling since online sampling has high overhead when drawing a sample and is only applicable for a small subset of queries.

In the field of offline sampling, a multitude of different sampling schemes have been proposed that are optimized for different query types, like aggregation (Chaudhuri, Das, Datar, Motwani, & Narasayya, 2001; Rösch, Gemulla, & Lehner, 2008), group-by (Acharya, Gibbons, & Poosala, 2000; Babcock, Chaudhuri, & Das, 2003; Rösch & Lehner, 2009) or foreign-key joins (Acharya, Gibbons, Poosala, & Ramaswamy, 1999; Gemulla, Rösch, & Lehner, 2008). While those sampling schemes provide great solutions for single (groups of) queries, the more general problem of automatic sample selection for an entire workload of a database remains almost unaddressed.

In this article, we address the problem of finding a set of samples that is to be materialized. We focus on simple random samples as those samples are easy to use and easy to maintain. Simple random samples are very general and can be used for a broad range of queries. Our solution is a sample advisor that suggests a set of samples for a set of queries specified by an expert. The sample advisor is based on a novel cost model to evaluate a sample for a given query. This cost model allows us to give advice on a sample for an individual query. To ease the usage of the sample advisor, we further propose an extension to utilize recorded workload information. In this scenario, the sample advisor selects from all the pieces of sample advices those that minimize the runtime of the workload and fit into a given memory bound. With a second extension, we are targeting the effectiveness of the sample advisor. Here, we consider the merge of samples in case of overlapping pieces of sample advice. As a result, more of the advised samples fit into the memory bound—the available memory is used more effectively—and thus, more queries can be answered very fast based on samples.

This article is an extended version of a previously published conference paper (Rösch & Lehner, 2010); we make the following (partly new) contributions:

- We analyze the database operations in the context of sample-based query processing.
- We propose a cost model for the evaluation of a sample for a given query. With this cost model, we can give a piece of sample advice for an individual query.
- Based on the cost model, we present our sample advisor in the setting of representative input queries specified by an expert.
- We propose two extensions for the sample advisor to ease the usage and to increase the effectiveness. For both extensions, we present an exact and a heuristic solution.
- With a variety of experiments, we analyze the properties of the cost model and compare the proposed algorithms. We further demonstrate the effectiveness and the efficiency of the heuristic solutions.

At the end of this article, we discuss related work and summarize the article in the conclusion. There, we also provide an outlook on future work.

DATABASE OPERATIONS AND SAMPLING

In this section, we discuss the interaction of database operations and (offline) sampling.
Related Content

The Study of the Entrepreneurial Leadership Style of Real Estate Industry in China: Based on the Content Analysis of Microblog
[www.igi-global.com/article/the-study-of-the-entrepreneurial-leadership-style-of-real-estate-industry-in-china/154910?camid=4v1a](www.igi-global.com/article/the-study-of-the-entrepreneurial-leadership-style-of-real-estate-industry-in-china/154910?camid=4v1a)

Framework for Managing Shared Knowledge in an Information Systems Outsourcing Context
[www.igi-global.com/article/framework-for-managing-shared-knowledge-in-an-information-systems-outsourcing-context/196542?camid=4v1a](www.igi-global.com/article/framework-for-managing-shared-knowledge-in-an-information-systems-outsourcing-context/196542?camid=4v1a)
Knowledge Management—the Second Generation: Creating Competencies with and Between Work Communities in the Competence Laboratory
www.igi-global.com/chapter/knowledge-management-second-generation/54265?camid=4v1a

The Evolution of the ISO/IEC 29110 Set of Standards and Guides
www.igi-global.com/article/the-evolution-of-the-isoiec-29110-set-of-standards-and-guides/169765?camid=4v1a