Available online at www.sciencedirect.com
ScienceDirect
The Journal of Systems and Software 81 (2008) 414–430
The Journal of Systems and Software
www.elsevier.com/locate/jss

Proactive and reactive multi-dimensional histogram maintenance for selectivity estimation ☆

Zhen He [a,*], Byung Suk Lee [b], X. Sean Wang [b]

[a] Department of Computer Science, La Trobe University, Bundoora, Vic. 3086, Australia
[b] Department of Computer Science, University of Vermont, Burlington, VT 05405, USA

Received 23 May 2006; received in revised form 22 November 2006; accepted 31 March 2007
Available online 19 April 2007

Abstract

Many state-of-the-art selectivity estimation methods use query feedback to maintain histogram buckets, thereby using the limited memory efficiently. However, they are "reactive" in nature, that is, they update the histogram based on queries that have come to the system in the past for evaluation. In some applications, future occurrences of certain queries may be predicted and a "proactive" approach can bring much needed performance gain, especially when combined with the reactive approach. For these applications, this paper provides a method that builds customized proactive histograms based on query prediction and mergers them into reactive histograms when the predicted future arrives. Thus, the method is called the *proactive and reactive histogram* (PRHist). Two factors affect the usefulness of the proactive histograms and are dealt with during the merge process: the first is the predictability of queries and the second is the extent of data updates. PRHist adjusts itself to be more reactive or more proactive depending on these two factors. Through extensive experiments using both real and synthetic data and query sets, this paper shows that in most cases, PRHist outperforms STHoles, the state-of-the-art reactive method, even when only a small portion of the queries are predictable and a significant portion of data is updated.

Keywords: Selectivity estimation; Proactive; Reactive; Histograms; Query optimization

1. Introduction

Extensive literature exists on the selectivity estimation problem. A variety of solutions have been proposed, but most popular ones are histogram-based solutions. Many state-of-the-art histogram methods adopt a "self-tuning" strategy, and construct and maintain histograms based on the feedback information provided in the form of query results (Aboulnaga and Chaudhuri, 1999; Bruno et al., 2001; Stillger et al., 2001). This allows them to keep more histogram buckets in regions queried more frequently. Due to the common spatial and temporal locality of user queries, these methods usually make more efficient use of the limited memory than methods based solely on data distribution.

However, all existing self-tuning histograms are "reactive" in nature, that is, they are updated based on the feedback information of the *past* queries only. This reactive approach does not consider the possibility that certain *future* query occurrences can be predicted. In this paper, by building and using "proactive histograms" based on query prediction, we show that a proactive approach can make more efficient use of histogram memory, when combined with the reactive approach. We believe this paper is the first to introduce the idea of *proactive histograms* and to provide a method to take advantage of these histograms.

Query prediction may be obtained by inspecting the query log and identifying patterns in it, or may be derived directly from the business practice of the enterprise. For example, (1) in a university database, queries on students'

☆ This work was partially done while the author was at the Department of Computer Science, University of Vermont.

* Corresponding author. Tel.: +61 394 793036; fax: +61 394 793060.
E-mail addresses: z.he@latrobe.edu.au (Z. He), Byung.Lee@uvm.edu (B.S. Lee), Sean.Wang@uvm.edu (X.S. Wang).

0164-1212/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2007.03.088

course grades occur more frequently at the end of every semester, and (2) in a financial company database, queries on financial records occur more frequently at the end of every financial quarter.

Queries are predictable to a varying degree, either because they follow patterns to a varying degree or the patterns are recognizable to a varying degree. It is thus important to handle queries with different degrees of predictability. To this end, we equip our method with the ability to adjust itself to be more proactive or reactive depending on query predictability. We call our method the *proactive and reactive histogram* (PRHist).

Within the PRHist method, we need to deal with two main issues: (1) predicting future queries, and (2) using the predicted queries to improve selectivity estimation. In order to address the first issue, we have PRHist find patterns in the query log.[1] Looking for patterns of individual query occurrences is not feasible because those occurrences may rarely repeat themselves (the classical overfitting phenomenon). Therefore, PRHist groups queries into "clusters" and looks for patterns considering all queries in the same cluster as a whole. Section 4.1 provides a formal definition of a cluster of queries and cluster patterns.

In order to address the second issue, we have PRHist operate in two phases: off-line and on-line. In the off-line phase, we divide the future into a sequence of consecutive subintervals of time which we call *f-subintervals*. Then we predict a set of queries for each *f*-subinterval. Then, it builds a sequence of proactive histograms, one histogram for each of the *f*-subintervals. In the on-line phase, PRHist maintains an on-line reactive histogram using query feedback as done in the STHoles. At the beginning of each *f*-subinterval, it loads the proactive histogram built for the *f*-subinterval and merges it with the on-line reactive histogram. At the time of the merge, PRHist assigns a weight to each histogram based on its "confidence" in the proactive histogram, thereby becoming more reactive (when the confidence is low) or more proactive (otherwise). The confidence is measured as a combination of two factors: the predictability of queries and the extent of data updates.

Our method incurs very small additional run-time overhead over above existing reactive histogram techniques. This small overhead is for loading the off-line proactive histogram into memory and merging it with the on-line reactive histogram at the beginning of each *f*-subinterval. For example, if a proactive histogram is loaded every 30 min as is the case in our experiments with the real query set, this amounts to one or two extra page loadings[2] plus one merge operation for every 30 min. The merge operation is fast since the histograms often contain only a small number of buckets.

[1] Deriving query patterns from the enterprise business practice is possible but beyond the scope of this paper.

[2] Histograms are typically very small, often fewer than one or two pages in size (Aboulnaga and Chaudhuri, 1999; Bruno et al., 2001; Poosala and Ioannidis, 1997).

We have conducted extensive performance comparisons between PRHist and the state-of-the-art reactive histogram method STHoles, using both real and synthetic data sets. The results show that PRHist outperforms STHoles for most test cases even when only a very small portion of the queries are predictable or even when there are a high percentage of data updates.

We make two key contributions through this paper. We (1) develop a novel histogram maintenance method, which uses proactive histograms complemented with reactive histograms to improve selectivity estimation accuracy, and (2) conduct an extensive performance study to show the performance advantage of our method in a variety of situations.

The rest of the paper is organized as follows. Following this introduction, we discuss related work in Section 2, and formally define the problem addressed by PRHist in Section 3. We describe the off-line proactive histogram construction in Section 4 and the on-line PRHist maintenance in Section 5. In Section 6, we present the experiments, and with Section 7, we conclude the paper.

2. Related work

In this section we discuss three areas of related work: histograms for selectivity estimation, forecasting methods for query prediction, and proactive optimization methods used in database systems.

2.1. Histograms

We classify histograms as shown in Fig. 1. Data-driven histograms are built and/or maintained solely based on the distribution of the data (Gunopulos et al., 2000; Lee et al., 1999; Muralikrishna and DeWitt, 1988; Poosala et al., 1996; Vitter and Wang, 1999), and are typically rebuilt or re-organized if the number of data updates exceeds a threshold or the estimation error is above a tolerance value (Donjerkovic et al., 2000; Gibbons et al., 1997; Matias et al., 2000; Thaper et al., 2002). These histograms have the drawback of assuming that all queries are equally likely at all times, which is rarely true as certain regions may be queried more frequently than others at certain times.

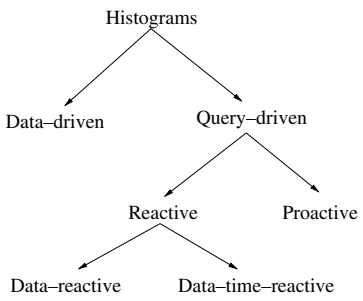

Fig. 1. Taxonomy of histogram methods.