Synopses for Massive Data: Samples, Histograms, Wavelets, Sketches
Synopses for Massive Data: Samples, Histograms, Wavelets, Sketches

Graham Cormode
AT&T Labs — Research
USA
graham@research.att.com

Minos Garofalakis
Technical University of Crete
Greece
minos@acm.org

Peter J. Haas
IBM Almaden Research Center
USA
peterh@almaden.ibm.com

Chris Jermaine
Rice University
USA
cmj4@cs.rice.edu
Foundations and Trends® in Databases
Editorial Board

Editor-in-Chief:
Joseph M. Hellerstein
Computer Science Division
University of California, Berkeley
Berkeley, CA
USA
hellerstein@cs.berkeley.edu

Editors
Anastasia Ailamaki (EPFL)
Michael Carey (UC Irvine)
Surajit Chaudhuri (Microsoft Research)
Ronald Fagin (IBM Research)
Minos Garofalakis (Yahoo! Research)
Johannes Gehrke (Cornell University)
Alon Halevy (Google)
Jeffrey Naughton (University of Wisconsin)
Christopher Olston (Yahoo! Research)
Jignesh Patel (University of Michigan)
Raghu Ramakrishnan (Yahoo! Research)
Gerhard Weikum (Max-Planck Institute)
Editorial Scope

Foundations and Trends® in Databases covers a breadth of topics relating to the management of large volumes of data. The journal targets the full scope of issues in data management, from theoretical foundations, to languages and modeling, to algorithms, system architecture, and applications. The list of topics below illustrates some of the intended coverage, though it is by no means exhaustive:

- Data Models and Query Languages
- Query Processing and Optimization
- Storage, Access Methods, and Indexing
- Transaction Management, Concurrency Control and Recovery
- Deductive Databases
- Parallel and Distributed Database Systems
- Database Design and Tuning
- Metadata Management
- Object Management
- Trigger Processing and Active Databases
- Data Mining and OLAP
- Approximate and Interactive Query Processing
- Data Warehousing
- Adaptive Query Processing
- Data Stream Management
- Search and Query Integration
- XML and Semi-Structured Data
- Web Services and Middleware
- Data Integration and Exchange
- Private and Secure Data Management
- Peer-to-Peer, Sensornet and Mobile Data Management
- Scientific and Spatial Data Management
- Data Brokering and Publish/Subscribe
- Data Cleaning and Information Extraction
- Probabilistic Data Management

Information for Librarians
Foundations and Trends® in Databases, 2011, Volume 4, 4 issues. ISSN paper version 1931-7883. ISSN online version 1931-7891. Also available as a combined paper and online subscription.
Synopses for Massive Data: Samples, Histograms, Wavelets, Sketches

Graham Cormode\textsuperscript{1}, Minos Garofalakis\textsuperscript{2}, Peter J. Haas\textsuperscript{3}, and Chris Jermaine\textsuperscript{4}

\textsuperscript{1}AT\&T Labs — Research, 180 Park Avenue, Florham Park, NJ 07932, USA, graham@research.att.com
\textsuperscript{2}Technical University of Crete, University Campus — Kounoupidiana, Chania, 73100, Greece, minos@acm.org
\textsuperscript{3}IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120-6099, USA, peterh@almaden.ibm.com
\textsuperscript{4}Rice University, 6100 Main Street, Houston, TX 77005, USA, cmj4@cs.rice.edu

Abstract

Methods for Approximate Query Processing (AQP) are essential for dealing with massive data. They are often the only means of providing interactive response times when exploring massive datasets, and are also needed to handle high speed data streams. These methods proceed by computing a lossy, compact synopsis of the data, and then executing the query of interest against the synopsis rather than the entire dataset. We describe basic principles and recent developments in AQP. We focus on four key synopses: random samples, histograms, wavelets, and sketches. We consider issues such as accuracy, space and time efficiency, optimality, practicality, range of applicability, error bounds on query answers, and incremental maintenance. We also discuss the trade-offs between the different synopsis types.
# Contents

1 Introduction 1

1.1 The Need for Synopses 2
1.2 Survey Overview 4
1.3 Outline 5

2 Sampling 11

2.1 Introduction 11
2.2 Some Simple Examples 12
2.3 Advantages and Drawbacks of Sampling 17
2.4 Mathematical Essentials of Sampling 21
2.5 Different Flavors of Sampling 32
2.6 Sampling and Database Queries 38
2.7 Obtaining Samples from Databases 56
2.8 Online Aggregation Via Sampling 65

3 Histograms 69

3.1 Introduction 71
3.2 One-Dimensional Histograms: Overview 79
3.3 Estimation Schemes 85
3.4 Bucketing Schemes 95
3.5 Multi-Dimensional Histograms 112
3.6 Approximate Processing of General Queries 127
3.7 Additional Topics 135
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Wavelets</td>
<td>145</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>145</td>
</tr>
<tr>
<td>4.2</td>
<td>One-Dimensional Wavelets and</td>
<td>146</td>
</tr>
<tr>
<td></td>
<td>Wavelet Synopses: Overview</td>
<td></td>
</tr>
<tr>
<td>4.3</td>
<td>Wavelet Synopses for Non-$L_2$ Error Metrics</td>
<td>156</td>
</tr>
<tr>
<td>4.4</td>
<td>Multi-Dimensional Wavelets</td>
<td>172</td>
</tr>
<tr>
<td>4.5</td>
<td>Approximate Processing of General Queries</td>
<td>182</td>
</tr>
<tr>
<td>4.6</td>
<td>Additional Topics</td>
<td>186</td>
</tr>
<tr>
<td>5</td>
<td>Sketches</td>
<td>203</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>203</td>
</tr>
<tr>
<td>5.2</td>
<td>Notation and Terminology</td>
<td>205</td>
</tr>
<tr>
<td>5.3</td>
<td>Frequency Based Sketches</td>
<td>212</td>
</tr>
<tr>
<td>5.4</td>
<td>Sketches for Distinct Value Queries</td>
<td>242</td>
</tr>
<tr>
<td>5.5</td>
<td>Other Topics in Sketching</td>
<td>259</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions and Future Research Directions</td>
<td>265</td>
</tr>
<tr>
<td>6.1</td>
<td>Comparison Across Different Methods</td>
<td>265</td>
</tr>
<tr>
<td>6.2</td>
<td>Approximate Query Processing in Systems</td>
<td>269</td>
</tr>
<tr>
<td>6.3</td>
<td>Challenges and Future Directions for Synopses</td>
<td>271</td>
</tr>
<tr>
<td></td>
<td>Acknowledgments</td>
<td>277</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>279</td>
</tr>
</tbody>
</table>
A synopsis of a massive dataset captures vital properties of the original data while typically occupying much less space. For example, suppose that our data consists of a large numeric time series. A simple summary allows us to compute the statistical variance of this series: we maintain the sum of all the values, the sum of the squares of the values, and the number of observations. Then the average is given by the ratio of the sum to the count, and the variance is ratio of the sum of squares to the count, less the square of the average. An important property of this synopsis is that we can build it efficiently. Indeed, we can find the three summary values in a single pass through the data.

However, we may need to know more about the data than merely its variance: how many different values have been seen? How many times has the series exceeded a given threshold? What was the behavior in a given time period? To answer such queries, our three-value summary does not suffice, and synopses appropriate to each type of query are needed. In general, these synopses will not be as simple or easy to compute as the synopsis for variance. Indeed, for many of these questions, there is no synopsis that can provide the exact answer, as is the case for variance. The reason is that for some classes of queries, the query
Introduction

answers collectively describe the data in full, and so any synopsis would effectively have to store the entire dataset.

To overcome this problem, we must relax our requirements. In many cases, the key objective is not obtaining the exact answer to a query, but rather receiving an accurate estimate of the answer. For example, in many settings, receiving an answer that is within 0.1% of the true result is adequate for our needs; it might suffice to know that the true answer is roughly $5 million without knowing that the exact answer is $5,001,482.76. Thus we can tolerate approximation, and there are many synopses that provide approximate answers. This small relaxation can make a big difference. Although for some queries it is impossible to provide a small synopsis that provides exact answers, there are many synopses that provide a very accurate approximation for these queries while using very little space.

1.1 The Need for Synopses

The use of synopses is essential for managing the massive data that arises in modern information management scenarios. When handling large datasets, from gigabytes to petabytes in size, it is often impractical to operate on them in full. Instead, it is much more convenient to build a synopsis, and then use this synopsis to analyze the data. This approach captures a variety of use-cases:

- A search engine collects logs of every search made, amounting to billions of queries every day. It would be too slow, and energy-intensive, to look for trends and patterns on the full data. Instead, it is preferable to use a synopsis that is guaranteed to preserve most of the as-yet undiscovered patterns in the data.
- A team of analysts for a retail chain would like to study the impact of different promotions and pricing strategies on sales of different items. It is not cost-effective to give each analyst the resources needed to study the national sales data in full, but by working with synopses of the data, each analyst can perform their explorations on their own laptops.
1.1 The Need for Synopses

- A large cellphone provider wants to track the health of its network by studying statistics of calls made in different regions, on hardware from different manufacturers, under different levels of contention, and so on. The volume of information is too large to retain in a database, but instead the provider can build a synopsis of the data as it is observed live, and then use the synopsis off-line for further analysis.

These examples expose a variety of settings. The full data may reside in a traditional data warehouse, where it is indexed and accessible, but is too costly to work on in full. In other cases, the data is stored as flat-files in a distributed file system; or it may never be stored in full, but be accessible only as it is observed in a streaming fashion. Sometimes synopsis construction is a one-time process, and sometimes we need to update the synopsis as the base data is modified or as accuracy requirements change. In all cases though, being able to construct a high quality synopsis enables much faster and more scalable data analysis.

From the 1990s through today, there has been an increasing demand for systems to query more and more data at ever faster speeds. Enterprise data requirements have been estimated [173] to grow at 60% per year through at least 2011, reaching 1,800 exabytes. On the other hand, users — weaned on Internet browsers, sophisticated analytics and simulation software with advanced GUIs, and computer games — have come to expect real-time or near-real-time answers to their queries. Indeed, it has been increasingly realized that extracting knowledge from data is usually an interactive process, with a user issuing a query, seeing the result, and using the result to formulate the next query, in an iterative fashion. Of course, parallel processing techniques can also help address these problems, but may not suffice on their own. Many queries, for example, are not embarrassingly parallel. Moreover, methods based purely on parallelism can be expensive. Indeed, under evolving models for cloud computing, specifically “platform as a service” fee models, users will pay costs that directly reflect the computing resources that they use. In this setting, use of Approximate Query Processing (AQP) techniques can lead to significant cost savings. Similarly, recent work [15] has pointed out that approximate processing

Full text available at: http://dx.doi.org/10.1561/1900000004
Introduction

techniques can lead to energy savings and greener computing. Thus AQP techniques are essential for providing, in a cost-effective manner, interactive response times for exploratory queries over massive data.

Exacerbating the pressures on data management systems is the increasing need to query streaming data, such as real time financial data or sensor feeds. Here the flood of high speed data can easily overwhelm the often limited CPU and memory capacities of a stream processor unless AQP methods are used. Moreover, for purposes of network monitoring and many other applications, approximate answers suffice when trying to detect general patterns in the data, such as a denial-of-service attack. AQP techniques are thus well suited to streaming and network applications.

1.2 Survey Overview

In this survey, we describe basic principles and recent developments in building approximate synopses (i.e., lossy, compressed representations) of massive data. Such synopses enable AQP, in which the user’s query is executed against the synopsis instead of the original data. We focus on the four main families of synopses: random samples, histograms, wavelets, and sketches.

A random sample comprises a “representative” subset of the data values of interest, obtained via a stochastic mechanism. Samples can be quick to obtain, and can be used to approximately answer a wide range of queries.

A histogram summarizes a dataset by grouping the data values into subsets, or “buckets,” and then, for each bucket, computing a small set of summary statistics that can be used to approximately reconstruct the data in the bucket. Histograms have been extensively studied and have been incorporated into the query optimizers of virtually all commercial relational DBMSs.

Wavelet-based synopses were originally developed in the context of image and signal processing. The dataset is viewed as a set of $M$ elements in a vector — that is, as a function defined on the set $\{0,1,2,\ldots,M-1\}$ — and the wavelet transform of this function is found as a weighted sum of wavelet “basis functions.” The weights,
or coefficients, can then be “thresholded,” for example, by eliminating coefficients that are close to zero in magnitude. The remaining small set of coefficients serves as the synopsis. Wavelets are good at capturing features of the dataset at various scales.

Sketch summaries are particularly well suited to streaming data. Linear sketches, for example, view a numerical dataset as a vector or matrix, and multiply the data by a fixed matrix. Such sketches are massively parallelizable. They can accommodate streams of transactions in which data is both inserted and removed. Sketches have also been used successfully to estimate the answer to COUNT DISTINCT queries, a notoriously hard problem.

Many questions arise when evaluating or using synopses.

- What is the class of queries that can be approximately answered?
- What is the approximation accuracy for a synopsis of a given size?
- What are the space and time requirements for constructing a synopsis of a given size, as well as the time required to approximately answer the query?
- How should one choose synopsis parameters such as the number of histogram buckets or the wavelet thresholding value? Is there an optimal, that is, most accurate, synopsis of a given size?
- When using a synopsis to approximately answer a query, is it possible to obtain error bounds on the approximate query answer?
- Can the synopsis be incrementally maintained in an efficient manner?
- Which type of synopsis is best for a given problem?

We explore these issues in subsequent chapters.

1.3 Outline

It is possible to read the discussion of each type of synopsis in isolation, to understand a particular summarization approach. We have tried to use common notation and terminology across all chapters, in order to
facilitate comparison of the different synopses. In more detail, the topics covered by the different chapters are given below.

1.3.1 Sampling

Random samples are perhaps the most fundamental synopses for AQP, and the most widely implemented. The simplicity of the idea — executing the desired query against a small representative subset of the data — belies centuries of research across many fields, with decades of effort in the database community alone. Many different methods of extracting and maintaining samples of data have been proposed, along with multiple ways to build an estimator for a given query. This chapter introduces the mathematical foundations for sampling, in terms of accuracy and precision, and discusses the key sampling schemes: Bernoulli sampling, stratified sampling, and simple random sampling with and without replacement.

For simple queries, such as basic SUM and AVERAGE queries, it is straightforward to build unbiased estimators from samples. The more general case — an arbitrary SQL query with nested subqueries — is more daunting, but can sometimes be solved quite naturally in a procedural way.

For small tables, drawing a sample can be done straightforwardly. For larger relations, which may not fit conveniently in memory, or may not even be stored on disk in full, more advanced techniques are needed to make the sampling process scalable. For disk-resident data, sampling methods that operate at the granularity of a block rather than a tuple may be preferred. Existing indices can also be leveraged to help the sampling. For large streams of data, considerable effort has been put into maintaining a uniform sample as new items arrive or existing items are deleted. Finally, “online aggregation” algorithms enhance interactive exploration of massive datasets by exploiting the fact that an imprecise sampling-based estimate of a query result can be incrementally improved simply by collecting more samples.

1.3.2 Histograms

The histogram is a fundamental object for summarizing the frequency distribution of an attribute or combination of attributes. The most
basic histograms are based on a fixed division of the domain (equi-width), or using quantiles (equi-depth), and simply keep statistics on the number of items from the input which fall in each such bucket. But many more complex methods have been designed, which aim to provide the most accurate summary possible within a limited space budget. Schemes differ in how the buckets are chosen, what statistics are stored, how estimates are extracted, and what classes of query are supported. They are quantified based on the space and time requirements used to build them, and the resulting accuracy guarantees that they provide.

The one-dimensional case is at the heart of histogram construction, since higher dimensions are typically handled via extensions of one-dimensional ideas. Beyond equi-width and equi-depth, end biased and high biased, maxdiff and other generalizations have been proposed. For a variety of approximation-error metrics, dynamic programming (DP) methods can be used to find histograms — notably the “v-optimal histograms” — that minimize the error, subject to an upper bound on the allowable histogram size. Approximate methods can be used when the quadratic cost of DP is not practical. Many other constructions, both optimal and heuristic, are described, such as lattice histograms, STHoles, and maxdiff histograms. The extension of these methods to higher dimensions adds complexity. Even the two-dimensional case presents challenges in how to define the space of possible bucketings. The cost of these methods also rises exponentially with the dimensionality of the data, inspiring new approaches that combine sets of low-dimensional histograms with high-level statistical models.

Histograms most naturally answer range–sum queries — for example, “compute total sales between July and September for adults from age 25 through 40” — and their variations. They can also be used to approximate more general classes of queries, such as aggregations over joins. Various negative theoretical and empirical results indicate that one should not expect histograms to give accurate answers to arbitrary queries. Nevertheless, due to their conceptual simplicity, histograms can be effectively used for a broad variety of estimation tasks, including set-valued queries, real-valued data, and aggregate queries over predicates more complex than simple ranges.
1.3.3 Wavelets

The wavelet synopsis is conceptually close to the histogram summary. The central difference is that, whereas histograms primarily produce buckets that are subsets of the original data-attribute domain, wavelet representations transform the data and seek to represent the most significant features in a wavelet (i.e., “frequency”) domain, and can capture combinations of high and low frequency information. The most widely discussed wavelet transformation is the Haar-wavelet transform (HWT), which can, in general, be constructed in time linear in the size of the underlying data array. Picking the $B$ largest HWT coefficients results in a synopsis that provides the optimal $L_2$ (sum-squared) error for the reconstructed data. Extending from one-dimensional to multi-dimensional data, as with histograms, provides more definitional challenges. There are multiple plausible choices here, as well as algorithmic challenges in efficiently building the wavelet decomposition.

The core AQP task for wavelet summaries is to estimate the answer to range sums. More general SPJ (select, project, join) queries can also be directly applied on relation summaries, to generate a summary of the resulting relation. This is made possible through an appropriately-defined AQP algebra that operates entirely in the domain of wavelet coefficients.

Recent research into wavelet representations has focused on error guarantees beyond $L_2$. These include $L_1$ (sum of errors) or $L_\infty$ (maximum error), as well as relative-error versions of these measures. A fundamental choice here is whether to restrict the possible coefficient values to those arising under the basic wavelet transform, or to allow other (unrestricted) coefficient values, specifically chosen to reduce the target error metric. The construction of such (restricted or unrestricted) wavelet synopses optimized for non-$L_2$ error metrics is a challenging problem.

1.3.4 Sketches

Sketch techniques have undergone extensive development over the past few years. They are especially appropriate for streaming data, in which large quantities of data flow by and the sketch summary must
continually be updated quickly and compactly. Sketches, as presented here, are designed so that the update caused by each new piece of data is largely independent of the current state of the summary. This design choice makes them faster to process, and also easy to parallelize.

“Frequency based sketches” are concerned with summarizing the observed frequency distribution of a dataset. From these sketches, accurate estimations of individual frequencies can be extracted. This leads to algorithms for finding approximate “heavy hitters” — items that account for a large fraction of the frequency mass — and quantiles such as the median and its generalizations. The same sketches can also be used to estimate the sizes of (equi)joins between relations, self-join sizes, and range queries. Such sketch summaries can be used as primitives within more complex mining operations, and to extract wavelet and histogram representations of streaming data.

A different style of sketch construction leads to sketches for “distinct-value” queries that count the number of distinct values in a given multiset. As mentioned above, using a sample to estimate the answer to a COUNT DISTINCT query may give highly inaccurate results. In contrast, sketching methods that make a pass over the entire dataset can provide guaranteed accuracy. Once built, these sketches estimate not only the cardinality of a given attribute or combination of attributes, but also the cardinality of various operations performed on them, such as set operations (union and difference), and selections based on arbitrary predicates.

In the final chapter, we compare the different synopsis methods. We also discuss the use of AQP within research systems, and discuss challenges and future directions.

In our discussion, we often use terminology and examples that arise in classical database systems, such as SQL queries over relational databases. These artifacts partially reflect the original context of the results that we survey, and provide a convenient vocabulary for the various data and access models that are relevant to AQP. We emphasize that the techniques discussed here can be applied much more generally. Indeed, one of the key motivations behind this survey is the hope that these techniques — and their extensions — will become a fundamental component of tomorrow’s information management systems.
References


279
References


References


References


References


Full text available at: http://dx.doi.org/10.1561/1900000004


References


References


References

References


References


References 293


References


References


