V

VERY LARGE DATABASES

A growing number of database applications require online, interactive access to very large volumes of data to perform a variety of data analysis tasks. As an example, large telecommunication and Internet service providers typically collect and store Gigabytes or Terabytes of detailed usage information (Call Detail Records, SNMP/RMON packet flow data, etc.) from the underlying network to satisfy the requirements of various network management tasks, including billing, fraud/anomaly detection, and strategic planning. Such large datasets are typically represented either as massive *alphanumeric data tables* (in the relational data model) or as massive *labeled data graphs* (in richer, semistructured data models, such as extensible markup language (XML)). In order to deal with the huge data volumes, high query complexities, and interactive response time requirements characterizing these modern data analysis applications, the idea of *effective, easy-to-compute approximations over precomputed, compact data synopses* has recently emerged as a viable solution. Due to the exploratory nature of most target applications, there are a number of scenarios in which a (reasonably accurate) fast approximate answer over a small-footprint summary of the database is actually preferable over an exact answer that takes hours or days to compute. For example, during a drill-down query sequence in ad hoc data mining, initial queries in the sequence frequently have the sole purpose of determining the truly interesting queries and regions of the database. Providing fast approximate answers to these initial queries gives users the ability to focus their explorations quickly and effectively, without consuming inordinate amounts of valuable system resources. The key, of course, behind such approximate techniques for dealing with massive datasets lies in the use of appropriate *data reduction techniques* for constructing compact *data synopses* that can accurately approximate the important features of the underlying data distribution. In this article, we provide an overview of date reduction and approximation methods for massive databases and discuss some of the issues that develop from different types of data, large data volumes, and applications-specific requirements.

APPROXIMATION TECHNIQUES FOR MASSIVE RELATIONAL DATABASES

Consider a relational table R with d data attributes X_1, X_2, ... X_d. We can represent the information in R as a d-dimensional array A_R , whose *jth* dimension is indexed by the values of attribute X_j and whose cells contain the count of tuples in R having the corresponding combination of attribute values. A_R is essentially the *joint frequency distribution* of all the data attributes of R. More formally, let $D = \{D_1, D_2, \ldots, D_d\}$ denote the set of dimensions of A_R, where dimension D_j corresponds to the *value domain* of attribute X_j . Without loss of generality, we assume that each dimension D_j is indexed by the set of integers $\{0, 1, \ldots, |D_j| - 1\}$, where $|D_j|$ denotes the size of dimension D_j. We assume that the attributes $\{X_1,\ldots,X_d\}$ are ordinal in nature, that is, their domains are naturally ordered, which captures all numeric attributes (e.g., age, income) and some categorical attributes (e.g., education). Such domains can always be mapped to the set of integers mentioned above while preserving the natural domain order and, hence, the locality of the distribution. It is also possible to map unordered domains to integer values; however, such mappings do not always preserve locality. For example, mapping countries to integers using alphabetic ordering can destroy data locality. There may be alternate mappings that are more locality preserving, (e.g., assigning neighboring integers to neighboring countries). (Effective mapping techniques for unordered attributes are an open research issue that lies beyond the scope of this article.) The d-dimensional joint-frequency array A_R comprises $N = \Pi_{i=1}^{d}|D_i|$ cells with cell $A_R[i_1, i_2,\ldots, i_d]$ containing the count of tuples in R having $X_j = i_j$ for each attribute $1 \leq j \leq d$.

The common goal of all relational data reduction techniques is to produce compact synopsis data structures that can effectively approximate the d-dimensional joint-frequency distribution A_R. In what follows, we give an overview of a few key techniques for relational data reduction, and discuss some of their main strengths and weaknesses as well as recent developments in this area of database research. More exhaustive and detailed surveys can be found elsewhere; see, for example, Refs. 1 and 2.

Sampling-Based Techniques

Sampling methods are based on the notion that a large dataset can be represented by a small *uniform random sample* of data elements, an idea that dates back to the end of the nineteenth century. In recent years, there has been increasing interest in the application of sampling ideas as a tool for data reduction and approximation in relational database management systems (3–8). Sample synopses can be either precomputed and incrementally maintained (e.g., Refs. 4 and 9) or they can be obtained progressively at run-time by accessing the base data using specialized data access methods (e.g., Refs. 10 and 11). Appropriate *estimator functions* can be applied over a random sample of a data collection to provide approximate estimates for quantitative characteristics of the entire collection (12). The adequacy of sampling as a data-reduction mechanism depends crucially on how the sample is to be used. Random samples can typically provide accurate estimates for aggregate quantities (e.g., `COUNT`s or `AVERAGE`s) of a (sub)population (perhaps determined by some selection predicate), as witnessed by the long history of successful applications of random sampling in population surveys

Wiley Encyclopedia of Computer Science and Engineering, edited by Benjamin Wah.

(12,13). An additional benefit of random samples is that they can provide probabilistic guarantees (i.e., *confidence intervals*) on the quality of the approximation (7,11). On the other hand, as a query approximation tool, random sampling is limited in its query processing scope, especially when it comes to the "workhorse" operator for correlating data collections in relational database systems, the *relational join*. The key problem here is that a join operator applied on two uniform random samples results in a *nonuniform* sample of the join result that typically contains *very few tuples*, even when the join selectivity is fairly high (9). Furthermore, for *nonaggregate* queries, execution over random samples of the data is guaranteed to always produce a small subset of the exact answer, which is often *empty* when joins are involved (9,14). The recently proposed "join synopses" method (9) provides a (limited) sampling-based solution for handling *foreign-key joins that are known beforehand* (based on an underlying "star" or "snowflake" database schema). Techniques for appropriately biasing the base-relation samples for effective approximate join processing have also been studied recently (3).

Histogram-Based Techniques

Histogram synopses or approximating *one-dimensional* data distributions have been extensively studied in the research literature (15–19), and have been adopted by several commercial database systems. Briefly, a histogram on an attribute X is constructed by employing a *partitioning rule* to partition the data distribution of X into a number of mutually disjoint subsets (called *buckets*), and approximating the frequencies and values in each bucket in some common fashion. Several partitioning rules have been proposed for the bucketization of data distribution points—some of the most effective rules seem to be ones that explicitly try to minimize the overall variance of the approximation in the histogram buckets (17–19). The summary information stored in each bucket typically comprises (1) *the number of distinct data values* in the bucket, and (2) *the average frequency* of values in the bucket, which are used to approximate the actual bucket contents based on appropriate *uniformity assumptions* about the spread of different values in the bucket and their corresponding frequencies (19).

One-dimensional histograms can also be used to approximate a (multidimensional) joint-frequency distribution A_R through a *mutual-independence assumption* for the data attributes $\{X_1, \ldots, X_d\}$. Mutual independence essentially implies that the joint-frequency distribution can be obtained as a product of the one-dimensional marginal distributions of the individual attributes X_i. Unfortunately, experience with real-life datasets offers overwhelming evidence that this independence assumption is almost always invalid and can lead to gross approximation errors in practice (20,21). Rather than relying on heuristic independence assumptions, *multidimensional histograms* [originally introduced by Muralikrishna and DeWitt (22)] try to directly approximate the joint distribution of $\{X_1, \ldots, X_d\}$ by strategically partitioning the data space into d-dimensional buckets in a way that captures the variation in data frequencies and values. Similar to the one-dimensional case, uniformity assumptions are made to approximate the distribution of frequencies and values within each bucket (21). Finding optimal histogram bucketizations is a hard optimization problem that is typically *NP*-complete even for two dimensions (23). Various greedy heuristics for multidimensional histogram construction have been proposed (21,22,24) and shown to perform reasonably well for low to medium data dimensionalities (e.g., $d = 2$–5).

Recent work has demonstrated the benefits of histogram synopses (compared with random samples) as a tool for providing fast, approximate answers to both *aggregate* and *nonaggregate* (i.e., "set-valued") user queries over low-dimensional data (14). Other studies have also considered the problem of *incrementally* maintaining a histogram synopsis over updates (25,26) or using query feedback (27,28), and the effectiveness of random sampling for approximate histogram construction (29). Unfortunately, like most techniques that rely on space partitioning (including the wavelet-based techniques of the next section), multidimensional histograms also fall victim to the "*curse of dimensionality*," which renders them ineffective above 5–6 dimensions (24).

Wavelet-Based Techniques

Wavelets are a mathematical tool for the hierarchical decomposition of functions with several successful applications in signal and image processing (30,31). Broadly speaking, the wavelet decomposition of a function consists of a coarse overall approximation along with detail coefficients that influence the function at various scales (31). A number of recent studies have also demonstrated the effectiveness of the wavelet decomposition (and Haar wavelets, in particular) as a data reduction tool for database problems, including selectivity estimation (32) and approximate query processing over massive relational tables (33–35).

Suppose we are given the one-dimensional data frequency vector A containing the $N = 8$ values $A = [2, 2, 0, 2, 3, 5, 4, 4]$. The Haar wavelet decomposition of A can be computed as follows. We first average the values together pairwise to get a new "lower-resolution" representation of the data with the following average values [2,1,4,4]. In other words, the average of the first two values (that is, 2 and 2) is 2, that of the next two values (that is, 0 and 2) is 1, and so on. Obviously, some information has been lost in this averaging process. To be able to restore the original values of the frequency array, we need to store some *detail coefficients* that capture the missing information. In Haar wavelets, these detail coefficients are simply the differences of the (second of the) averaged values from the computed pairwise average. Thus, in our simple example, for the first pair of averaged values, the detail coefficient is 0 because $2 - 2 = 0$; for the second sample, we again need to store -1 because $1 - 2 = -1$. Note that no information has been lost in this process—it is fairly simple to reconstruct the eight values of the original data frequency array from the lower-resolution array containing the four averages and the four detail coefficients. Recursively applying the above pairwise averaging and differencing

process on the lower-resolution array containing the averages, we get the following full decomposition:

Resolution	Averages	Detail Coefficients
3	[2, 2, 0, 2, 3, 5, 4, 4]	—
2	[2, 1, 4, 4]	[0, −1, −1, 0]
1	[3/2, 4]	[1/2, 0]
0	[11/4]	[−5/4]

The *Haar wavelet decomposition* of A is the single coefficient representing the overall average of the frequency values followed by the detail coefficients in the order of increasing resolution. Thus, the one-dimensional Haar wavelet transform of A is given by $W_A = [11/4, -5/4, 1/2, 0, 0, -1, -1, 0]$. Each entry in W_A is called a *wavelet coefficient*. The main advantage of using W_A instead of the original frequency vector A is that for vectors containing similar values most of the detail coefficients tend to have very small values. Thus, eliminating such small coefficients from the wavelet transform (i.e., treating them as zeros) introduces only small errors when reconstructing the original data, resulting in a very effective form of lossy data compression (31). Furthermore, the Haar wavelet decomposition can also be extended to *multidimensional* joint-frequency distribution arrays through natural generalizations of the one-dimensional decomposition process described above (33,35). Thus, the key idea is to apply the decomposition process over an input dataset along with a thresholding procedure in order to obtain a compact data synopsis comprising of a selected small set of *Haar wavelet coefficients*. The results of several research studies (32–37) have demonstrated that fast and accurate approximate query processing engines (for both aggregate and nonaggregate queries) can be designed to operate solely over such compact *wavelet synopses*.

Other recent work has proposed probabilistic counting techniques for the efficient online maintenance of wavelet synopses in the presence of updates (38), as well as time- and space-efficient techniques for constructing wavelet synopses for datasets with multiple measures (such as those typically found in OLAP applications) (39). All the above-mentioned studies rely on conventional schemes for eliminating small wavelet coefficients in an effort to minimize the overall sum-squared error (SSE). Garofalakis and Gibbons (34,36) have shown that such conventional wavelet synopses can suffer from several important problems, including the introduction of severe bias in the data reconstruction and wide variance in the quality of the data approximation, as well as the lack of nontrivial guarantees for individual approximate answers. In contrast, their proposed *probabilistic wavelet synopses* rely on a probabilistic thresholding process based on *randomized rounding* that tries to *probabilistically* control the maximum relative error in the synopsis by minimizing appropriate probabilistic metrics.

In more recent work, Garofalakis and Kumar (40) show that the pitfalls of randomization can be avoided by introducing efficient schemes for *deterministic* wavelet thresholding with the objective of optimizing a *general class of error metrics* (e.g., maximum or mean relative error). Their optimal and approximate thresholding algorithms are based on novel Dynamic-Programming (DP) techniques that take advantage of the *coefficient-tree structure* of the Haar decomposition. This turns out to be a fairly powerful idea for wavelet synopsis construction that can handle a broad, natural class of *distributive error metrics* (which includes several useful error measures for approximate query answers, such as maximum or mean weighted relative error and weighted L_p-norm error) (40). The above wavelet thresholding algorithms for non-SSE error metrics consider only the *restricted* version of the problem, where the algorithm is forced to select values for the synopsis from the standard Haar coefficient values. As observed by Guha and Harb (41), such a restriction makes little sense when optimizing for non-SSE error, and can, in fact, lead to sub-optimal synopses. Their work considers *unrestricted* Haar wavelets, where the values retained in the synopsis are specifically chosen to optimize a general (weighted) L_p-norm error metric. Their proposed thresholding schemes rely on a DP over the coefficient tree (similar to that in (40) that *also iterates over the range of possible values for each coefficient*. To keep time and space complexities manageable, techniques for bounding these coefficient-value ranges are also discussed (41).

Advanced Techniques

Recent research has proposed several sophisticated methods for effective data summarization in relational database systems. Getoor et al. (42) discuss the application of *Probabilistic Relational Models (PRMs)* (an extension of Bayesian Networks to the relational domain) in computing accurate selectivity estimates for a broad class of relational queries. Deshpande et al. (43) pro-posed *dependency-based histograms*, a novel class of histogram-based synopses that employs the solid foundation of *statistical interaction models* to explicitly identify and exploit the statistical characteristics of the data and, at the same time, address the dimensionality limitations of multidimensional histogram approximations. Spiegel and Polyzotis (44) propose the Tuple-Graph synopses that view the relational database as a semi-structured data graph and employ summarization models inspired by XML techniques in order to approximate the joint distribution of join relationships and values. Finally, Jagadish et al. (45) and Babu et al. (46) develop semantic compression techniques for massive relational tables based on the idea of extracting *data mining models* from an underlying data table, and using these models to effectively compress the table to within user-specified, per-attribute error bounds.

Traditional database systems and approximation techniques are typically based on the ability to make multiple passes over *persistent datasets* that are stored reliably in stable storage. For several emerging application domains, however, data arrives at high rates 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* occur 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. As a result, we are witnessing a recent surge of interest in data stream computation, which has led to several (theoretical and practical) studies proposing novel one-pass algorithms for effectively summarizing massive relational data streams in a limited amount of memory (46–55).

APPROXIMATION TECHNIQUES FOR MASSIVE XML DATABASES

XML (56) has rapidly evolved from a markup language for web documents to an emerging standard for data exchange and integration over the Internet. The simple, self-describing nature of the XML standard promises to enable a broad suite of next-generation Internet applications, ranging from intelligent web searching and querying to electronic commerce. In many respects, XML represents an instance of *semistructured data* (57): The underlying data model comprises a labeled graph of *element* nodes, where each element can be either an atomic data item (i.e., raw character data) or a composite data collection consisting of references (represented as graph edges) to other elements in the graph. More formally, an XML database can be modeled as a directed graph $G(V_G, E_G)$, where each node $u \in V_G$ corresponds to a document element, or an element attribute, with label `label`(u). If u is a leaf node, then it can be associated with a value `value`(u). An edge (u, v) denotes either the nesting of v under u in the XML document, or a reference from u to v, through `ID/IDREF` attributes or XLink constructs (58–60).

XML query languages use two basic mechanisms for navigating the XML data graph and retrieving qualifying nodes, namely, *path expressions* and *twig queries*. A path expression specifies a sequence of navigation steps, where each step can be predicated on the existence of sibling paths or on the value content of elements, and the elements at each step can be linked through different structural relationships (e.g., parent-child, ancestor-child, or relationships that involve the order of elements in the document). As an example, the path expression `//author[/book//year = 2003]//paper` will select all paper elements with an author ancestor, which is the root of at least one path that starts with book and ends in year, and the value of the ending element is 2003. The example expression is written in the XPath (61) language, which lies at the core of XQuery (62) and XSLT (63), the dominant proposals from W3C for querying and transforming XML data.

A twig query uses multiple path expressions in order to express a complex navigation of the document graph and retrieve combinations of elements that are linked through specific structural relationships. As an example, consider the following twig query, which is expressed in the XQuery (62) language: `for $a in //author, $p in $a//paper/title, $b in $a//book/title`. The evaluation of the path expressions proceeds in a nested-loops fashion, by using the results of "parent" paths in order to evaluate "nested" paths. Thus, the first expression retrieves all authors, and, for each one, the nested paths retrieve the titles of their papers and books. The final result contains all possible combinations of an author node, with a paper title node and a book title node that it reaches. Twig queries represent the equivalent of the SQL `FROM` clause in the XML world, as they model the generation of element tuples, which will eventually be processed to compute the final result of the XML query.

The goal of existing XML data reduction techniques is to summarize, in limited space, the key statistical properties of an XML database in order to provide *selectivity estimates* for the result size of path expressions or twig queries. Selectivity estimation is a key step in the optimization of declarative queries over XML repositories and is thus key for the effective implementation of high-level query languages (64–66). Given the form of path expressions and twig queries, an effective XML summary needs to capture accurately both the path structure of the data graph and the value distributions that are embedded therein. In that respect, summarizing XML data is a more complex problem than relational summarization, which focuses mainly on value distributions. As with any approximation method, the proposed XML techniques store compressed distribution information on specific characteristics of the data, and use statistical assumptions in order to compensate for the loss of detail due to compression. Depending on the specifics of the summarization model, the proposed techniques can be broadly classified in three categories: (1) techniques that use a graph synopsis, (2) techniques that use a relational summarization method, such as histograms or sampling, and (3) techniques that use a Markovian model of path distribution. It should be noted that, conceptually, the proposed summarization techniques can also be used to provide approximate answers for XML queries; this direction, however, has not been explored yet in the current literature and it is likely to become an active area of research in the near future.

Graph-Synopsis-Based Techniques

At an abstract level, a graph synopsis summarizes the basic path structure of the document graph. More formally, given a data graph $G = (V_G, E_G)$, a graph synopsis $S(G) = (V_S, E_S)$ is a directed node-labeled graph, where (1) each node $v \in V_S$ corresponds to a subset of element (or attribute) nodes in V_G (termed the *extent* of) that have the *same label*, and (2) an edge in $(u, v) \in E_G$ is represented in E_S as an edge between the nodes whose extents contain the two endpoints u and v. For each node u, the graph synopsis records the common tag of its elements and a count field for the size of its extent.

In order to capture different properties of the underlying path structure and value content, a graph synopsis is augmented with appropriate, localized distribution information. As an example, the structural XSKETCH-summary mechanism (67), which can estimate the selectivity of simple path expressions with branching predicates, augments the general graph-synopsis model with localized

per-edge stability information, indicating whether the synopsis edge is *backward-stable or forward-stable*. In short, an edge (u, v) in the synopsis is said to be *forward-stable* if all the elements of u have at least one child in v; similarly, (u, v) is backward-stable if all the elements in v have at least one parent in u [note that backward/forward (B/F) stability is essentially a localized form of *graph bisimilarity* (68)]. Overall, edge stabilities capture key properties of the connectivity between different synopsis nodes and can summarize the underlying *path structure* of the input XML data.

In a follow-up study (69), the structural XSketch model is augmented with localized per-node value distribution summaries. More specifically, for each node u that represents elements with values, the synopsis records a summary $H(u)$, which captures the corresponding value distribution and thus enables selectivity estimates for value-based predicates. Correlations among different value distributions can be captured by a multidimensional summary $H(u)$, which approximates the *joint* distribution of values under u and under different parts of the document. It should be noted that, for the single-dimensional case, $H(u)$ can be implemented with any relational summarization technique; the multidimensional case, however, imposes certain restrictions due to the semantics of path expressions, and thus needs specialized techniques that can estimate the number of *distinct* values in a distribution [examples of such techniques are range-histograms (69), and distinct sampling (70)].

The TwigXSketch (71) model is a generalization of the XSketch synopses that deals with selectivity estimation for twig queries. Briefly, the key idea in TwigXSketches is to capture, in addition to localized stability information, the distribution of document edges for the elements in each node's extent. In particular, each synopsis node records an *edge histogram*, which summarizes the distribution of child counts across different stable ancestor or descendant edges. As a simple example, consider a synopsis node u and two emanating synopsis edges (u, v) and (u, w); a two-dimensional edge histogram $H_u(c_1, c_2)$ would capture the fraction of data elements in `extent`(u) that have exactly c_1 children in `extent`(v) and c_2 children in `extent`(w). Overall, TwigXSketches store more fine-grained information on the path structure of the data, and can thus capture, in more detail, the *joint* distribution of path counts between the elements of the XML dataset.

Recent studies (73–75) have proposed a variant of graph-synopses that employ a clustering-based model in order to capture the path and value distribution of the underlying XML data. Under this model, each synopsis node is viewed as a "cluster" and the enclosed elements are assumed to be represented by a corresponding "centroid," which is derived in turn by the aggregate characteristics of the enclosed XML elements. The TreeSketch (72) model, for instance, defines the centroid of a node u_i as a vector of average child counts $(c_1, c_2, \cdots, c_n)$, where c_j is the average child count from elements in A_i to every other node u_j. Thus, the assumption is that each element in u_i has exactly c_j children to node u_j. Furthermore, the clustering error, that is, the difference between the actual child counts in u_i and the centroid, provides a measure of the error of approximation. The TreeSketch study has shown that a partitioning of elements with low clustering error provides an accurate approximation of the path distribution, and essentially enables low-error selectivity estimates for structural twig queries. A follow-up study has introduced the XClusters (73) model that extends the basic TreeSketch synopses with information on element content. The main idea is to augment the centroid of each cluster with a value-summary that approximates the distribution of values in the enclosed elements. The study considers three types of content: numerical values queried with range predicates, string values queried with substring predicates, and text values queried with term-containment predicates. Thus, the key novelty of XClusters is that they provide a unified platform for summarizing the structural and heterogeneous value content of an XML data set. Finally, Zhang et al. have proposed the XSeed (74) framework for summarizing the recursive structure of an XML data set. An XSeed summary resembles a TreeSketch synopsis where all elements of the same tag are mapped to a single cluster. The difference is that each synopsis edge may be annotated with multiple counts, one per recursive level in the underlying data. To illustrate this, consider an element path $/e_1/e'_1/e_2/e'_2$ where e_1 and e_2 correspond to cluster u and e'_1 and e'_2 to cluster u'. The sub-path e_1/e'_1 will map to an edge between u and u' and will contribute to the first-level child count. The sub-path e_2/e'_2 will map to the same edge, but will contribute to the second-level child count from u to u'. Hence, XSeed stores more fine-grained information compared to a TreeSketch synopsis that uses a single count for all possible levels. This level-based information is used by the estimation algorithm in order to approximate more accurately the selectivity of recursive queries (i.e., with the "//" axis) on recursive data.

Histogram- and Sampling-Based Techniques

Several XML-related studies attempt to leverage the available relational summarization techniques by casting the XML summarization problem into a relational context. More specifically, the proposed techniques represent the path structure and value content of the XML data in terms of flat value distributions, which are then summarized using an appropriate relational technique.

The StatiX (75) framework uses histogram-based techniques and targets selectivity estimation for twig queries over tree-structured data (note, however, that StatiX needs the schema of the XML data in order to determine the set of histograms, which makes the technique nonapplicable to the general case of schema-less documents). StatiX partitions document elements according to their schema type and represents each group as a set of (*pid*, *count*) pairs, where *pid* is the id of some element p and *count* is the number of elements in the specific partition that have p as parent. Obviously, this scheme encodes the joint distribution of children counts for the elements of each partition. This information is then compressed using standard relational histograms, by treating *pid* as the value and *count* as the frequency information.

A similar approach is followed in *position histograms* (76), which target selectivity estimation for two-step path

expressions of the form A//B. In this technique, each element is represented as a point (s, e) in 2-dimensional space, where s and e are the start and end values of the element in the depth-first traversal of the document tree; thus, (s_a, e_a) is an ancestor of (s_b, e_b), if $s_a < s_b < e_b < e_a$. The proposed summarization model contains, for each tag in the document, one spatial histogram that summarizes the distribution of the corresponding element points. A spatial join between histograms is then sufficient to approximate the ancestor-descendant relationship between elements of different tags.

A recent study (77) has introduced two summarization models, namely the Position Model and the Interval Model, which are conceptually similar to position histograms but use a different encoding of structural relationships. Again, the focus is on selectivity estimation for two-step paths of the form A//B. The Position Model encodes each element in A as a point (s_a, e_a), and each element in B as a point s_b; the selectivity is then computed as the number of s_b points contained under an (s_a, e_a) interval. In the Interval Model, a covering matrix C records the number of points in A whose interval includes a specific start position, whereas a position matrix P includes the start positions of elements in B; the estimate is then computed by joining the two tables and summing up the number of matched intervals. Clearly, both models reduce the XML estimation problem to operations on flat value distributions, which can be approximated using relational summarization techniques.

Markov-Model-Based Techniques

At an abstract level, the path distribution of an XML dataset can be modeled with the probability of observing a specific tag as the next step of an existing path. Recent studies have investigated data reduction techniques that summarize the path structure by approximating, in limited space, the resulting path probability distribution. The principal idea of the proposed techniques is to compress the probability distribution through a Markovian assumption: If p is a path that appears in the document and l is a tag, then the probability that p/l is also a path depends only on a suffix $\bar{p}$ of p (i.e., the next step is not affected by distant ancestor tags). Formally, this assumption can be expressed as $P[p/l] = P[p] \cdot P[l|\bar{p}]$, where $P[q]$ is the probability of observing path q in the data. Of course, the validity of this independence assumption affects heavily the accuracy of the summarization methods.

Recent studies (78,79) have investigated the application of a k-order Markovian assumption, which limits the statistically correlated suffix of p to a maximum predefined length of k. Thus, only paths of length up to k need to be stored in order to perform selectivity estimation. The proposed techniques further compress this information with a *Markov histogram*, which records the most frequently occurring such paths. Less frequent paths are grouped together, either according to their prefix (if the aggregate frequency is high enough), or in a generic '*' bucket. In order to estimate the occurrence probability of a longer path, the estimation framework identifies subpaths that are present in the Markov histogram and combines the recorded probabilities using the Markovian independence assumption.

Correlated Suffix Trees (CSTs) (80) employ a similar Markovian assumption in order to estimate the selectivity of twig queries. The path distribution of the document is stored in a tree structure, which records the most frequently occurring suffixes of root-to-leaf paths; thus, the tree encodes frequent paths of variable lengths, instead of using a predefined fixed length as the Markov Histogram approach. In addition to frequent path suffixes, the summary records a *hash signature* for each outgoing path of a tree node, which encodes the set of elements in the node's extent that have at least one matching outgoing document path. Intuitively, an "intersection" of hash signatures, where each signature corresponds to a different label path, approximates the number of elements that have descendants along all represented paths. Combined with path frequency information, this information yields an approximation of the joint path-count distribution for different subsets of document elements.

BIBLIOGRAPHY

1. D. Barbarà, W. DuMouchel, C. Faloutsos, P. J. Haas, J. M. Hellerstein, Y. Ioannidis, H. V. Jagadish, T. Johnson, R. Ng, V. Poosala, K. A. Ross, and K. C. Sevcik, The New Jersey data reduction report, *IEEE Data Eng. Bull.*, **20**(4): 3–45, 1997, (Special Issue on Data Reduction Techniques).
2. M. Garofalakis and P. B. Gibbons, Approximate query processing: Taming the Terabytes, Tutorial in *27th Intl. Conf. on Very Large Data Bases, Roma, Italy*, September 2001.
3. S. Chaudhuri, R. Motwani, and V. Narasayya, On random sampling over joins, in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, Philadelphia, PA, May 1999, pp. 263–274.
4. P. B. Gibbons and Y. Matias, New sampling-based summary statistics for improving approximate query answers, in *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, Seattle, WA, June 1998, pp. 331–342.
5. P. J. Haas and A. N. Swami, Sequential sampling procedures for query size estimation, in *Proceedings of the 1992 ACM SIGMOD International Conference on Management of Data*, San Diego, CA, June 1992, pp. 341–350.
6. R. J. Lipton, J. F. Naughton, and D. A. Schneider, Practical selectivity estimation through adaptive sampling, in *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, Atlantic City, NJ, May 1990, pp. 1–12.
7. R. J. Lipton, J. F. Naughton, D. A. Schneider, and S. Seshadri, Efficient sampling strategies for relational database operations, *Theoret. Comp. Sci.*, **116**: 195–226, 1993.
8. F. Olken and D. Rotem, Simple random sampling from relational databases, in *Proceedings of the Twelfth International Conference on Very Large Data Bases*, Kyoto, Japan, August 1986, pp. 160–169.
9. S. Acharya, P. B. Gibbons, V. Poosala, and S. Ramaswamy, Join synopses for approximate query answering, in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, Philadelphia, PA, May 1999, pp. 275–286.
10. P. J. Haas and J. M. Hellerstein, Ripple joins for online aggregation, in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, Philadelphia, PA, May 1999, pp. 287–298.

11. J. M. Hellerstein, P. J. Haas, and H. J. Wang, Online aggregation, in *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, Tucson, AZ, May 1997.
12. W. G. Cochran, *Sampling Techniques*, 3rd ed. New York: John Wiley & Sons, 1977.
13. C.-E. Särndal, B. Swensson, and J. Wretman, *Model Assisted Survey Sampling*, New York: Springer-Verlag (Springer Series in Statistics), 1992.
14. Y. E. Ioannidis and V. Poosala, Histogram-based approximation of set-valued query answers, in *Proceedings of the 25th International Conference on Very Large Data Bases*, Edinburgh, Scotland, September 1999.
15. Y. E. Ioannidis, Universality of serial histograms, in *Proceedings of the Nineteenth International Conference on Very Large Data Bases*, Dublin, Ireland, August 1993, pp. 256–267.
16. Y. E. Ioannidis and S. Christodoulakis, Optimal histograms for limiting worst-case error propagation in the size of join results, *ACM Trans. Database Sys.*, **18**(4): 709–748, 1993.
17. Y. E. Ioannidis and V. Poosala, Balancing histogram optimality and practicality for query result size estimation, in *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, May 1995, pp. 233–244.
18. H. V. Jagadish, N. Koudas, S. Muthukrishnan, V. Poosala, K. Sevcik, and T. Suel, Optimal histograms with quality guarantees, in *Proceedings of the 24th International Conference on Very Large Data Bases*, New York City, NY, August 1998.
19. V. Poosala, Y. E. Ioannidis, P. J. Haas, and E. J. Shekita, Improved histograms for selectivity estimation of range predicates, in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, Montreal, Quebec, June 1996, pp. 294–305.
20. C. Faloutsos and I. Kamel, Beyond uniformity and independence: Analysis of R-trees using the concept of fractal dimension, in *Proceedings of the Thirteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Minneapolis, MN, May 1994, pp. 4–13.
21. V. Poosala and Y. E. Ioannidis, Selectivity estimation without the attribute value independence assumption, in *Proceedings of the 23rd International Conference on Very Large Data Bases*, Athens, Greece, August 1997, pp. 486–495.
22. M. Muralikrishna and D. J. DeWitt, Equi-depth histograms for estimating selectivity factors for multi-dimensional queries, in *Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data*, Chicago, IL, June 1988, pp. 28–36.
23. S. Muthukrishnan, V. Poosala, and T. Suel, On rectangular partitionings in two dimensions: Algorithms, complexity, and applications, in *Proceedings of the Seventh International Conference on Database Theory (ICDT'99)*, Jerusalem, Israel, January 1999.
24. D. Gunopulos, G. Kollios, V. J. Tsotras, and C. Domeniconi, Approximating multi-dimensional aggregate range queries over real attributes, in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas, TX, May 2000.
25. D. Donjerkovic, Y. Ioannidis, and R. Ramakrishnan, Dynamic histograms: Capturing evolving data sets, in *Proceedings of the Sixteenth International Conference on Data Engineering*, San Diego, CA, March 2000.
26. P. B. Gibbons, Y. Matias, and V. Poosala, Fast incremental maintenance of approximate histograms, in *Proceedings of the 23rd International Conference on Very Large Data Bases*, Athens, Greece, August 1997, pp. 466–475.
27. A. Aboulnaga and S. Chaudhuri, Self-tuning histograms: Building histograms without looking at data, in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, Philadelphia, PA, May 1999, pp. 181–192.
28. N. Bruno, S. Chaudhuri, and L. Gravano, STHoles: A Multidimensional workload-aware histogram, in *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, Santa Barbara, CA, May 2001.
29. S. Chaudhuri, R. Motwani, and V. Narasayya, Random sampling for histogram construction: How much is enough?, in *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, Seattle*, WA, June 1998.
30. B. Jawerth and W. Sweldens, An overview of wavelet based multiresolution analyses, *SIAM Rev.*, **36**(3): 377–412, 1994.
31. E. J. Stollnitz, T. D. DeRose, and D. H. Salesin, *Wavelets for Computer Graphics—Theory and Applications*, San Francisco, CA: Morgan Kaufmann Publishers, 1996.
32. Y. Matias, J. S. Vitter, and M. Wang, Wavelet-based histograms for selectivity estimation, in *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, Seattle, WA, June 1998, pp. 448–459.
33. K. Chakrabarti, M. Garofalakis, R. Rastogi, and K. Shim., Approximate query processing using wavelets, in *Proceedings of the 26th International Conference on Very Large Data Bases*, Cairo, Egypt, September 2000, pp. 111–122.
34. M. Garofalakis and P. B. Gibbons, Wavelet synopses with error guarantees, in *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, Madison, WI, June 2002, pp. 476–487.
35. J. S. Vitter and M. Wang, Approximate computation of multidimensional aggregates of sparse data using wavelets, in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, Philadelphia, PA, May 1999.
36. M. Garofalakis and P. B. Gibbons, Probabilistic wavelet synopses, *ACM Trans. Database Syst.*, **29** (1): 2004. (SIGMOD/PODS Special Issue).
37. R. R. Schmidt and C. Shahabi, ProPolyne: A fast wavelet-based algorithm for progressive evaluation of polynomial range-sum queries, in *Proceedings of the 8th International Conference on Extending Database Technology (EDBT'2002)*, Prague, Czech Republic, March 2002.
38. Y. Matias, J. S. Vitter, and M. Wang, Dynamic maintenance of wavelet-based histograms, in *Proceedings of the 26th International Conference on Very Large Data Bases*, Cairo, Egypt, September 2000.
39. A. Deligiannakis and N. Roussopoulos, Extended wavelets for multiple measures, in *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, San Diego, CA, June 2003.
40. M. Garofalakis and A. Kumar, Wavelet synopses for general error metrics, *ACM Trans. Database Syst.*, **30**(4), 2005.
41. S. Guha and B. Harb, Wavelet synopsis for data streams: Minimizing non-euclidean error, in *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, IL, August 2005.
42. L. Getoor, B. Taskar, and D. Koller, Selectivity estimation using probabilistic models, in *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, Santa Barbara, CA, May 2001.
43. A. Deshpande, M. Garofalakis, and R. Rastogi, Independence is good: Dependency-based histogram synopses for high-dimensional data, in *Proceedings of the 2001 ACM SIGMOD*

International Conference on Management of Data, Santa Barbara, CA, May 2001.

44. J. Spiegel and N. Polyzotis, Graph-based synopses for relational selectivity estimation, in *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, Chicago, IL, 2006, pp. 205–216.
45. H. V. Jagadish, J. Madar, and R. Ng, Semantic compression and pattern extraction with fascicles, in *Proceedings of the 25th International Conference on Very Large Data Bases*, Edinburgh, Scotland, September 1999, pp. 186–197.
46. S. Babu, M. Garofalakis, and R. Rastogi, SPARTAN: A model-based semantic compression system for massive data tables, in *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, Santa Barbara, CA, May 2001.
47. N. Alon, P. B. Gibbons, Y. Matias, and M. Szegedy, tracking join and self-join sizes in limited storage, in *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Philadeplphia, PA, May 1999.
48. N. Alon, Y. Matias, and M. Szegedy, The space complexity of approximating the frequency moments, in *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, Philadelphia, PA, May 1996, pp. 20–29.
49. A. Dobra, M. Garofalakis, J. Gehrke, and R. Rastogi, Processing complex aggregate queries over data streams, in *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, Madison, WI, June 2002, pp. 61–72.
50. J. Feigenbaum, S. Kannan, M. Strauss, and M. Viswanathan, An approximate L^1-difference algorithm for massive data streams, in *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, New York, NY, October 1999.
51. S. Ganguly, M. Garofalakis, and R. Rastogi, Processing set expressions over continuous update streams, in *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, San Diego, CA, June 2003.
52. M. Greenwald and S. Khanna, Space-efficient online computation of quantile summaries, in *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, Santa Barbara, CA, May 2001.
53. A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss, Surfing wavelets on streams: One-pass summaries for approximate aggregate queries, in *Proceedings of the 27th International Conference on Very Large Data Bases*, Roma, Italy, September 2001.
54. A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss, How to summarize the universe: Dynamic maintenance of quantiles, in *Proceedings of the 28th International Conference on Very Large Data Bases*, Hong Kong, China, August 2002, pp. 454–465.
55. P. Indyk, Stable distributions, pseudorandom generators, embeddings and data stream computation, in *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, Redondo Beach, CA, November 2000, pp. 189–197.
56. N. Thaper, S. Guha, P. Indyk, and N. Koudas, Dynamic multidimensional histograms, in *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, Madison, WI, June 2002, pp. 428–439.
57. T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler, Extensible Markup Language (XML) 1.0, 2nd ed. W3C Recommendation. Available: http://www.w3.org/TR/REC-xml/).
58. S. Abiteboul, Querying semi-structured data, in *Proceedings of the Sixth International Conference on Database Theory (ICDT'97)*, Delphi, Greece, January 1997.
59. R. Goldman and J. Widom, DataGuides: Enabling query formulation and optimization in semistructured databases, in *Proceedings of the 23rd International Conference on Very Large Data Bases*, Athens, Greece, August 1997, pp. 436–445.
60. R. Kaushik, P. Shenoy, P. Bohannon, and E. Gudes, Exploiting local similarity for efficient indexing of paths in graph structured data, in *Proceedings of the Eighteenth International Conference on Data Engineering*, San Jose, CA, February 2002.
61. T. Milo and D. Suciu, Index structures for path expressions, in *Proceedings of the Seventh International Conference on Database Theory (ICDT'99)*, Jerusalem, Israel, January 1999.
62. J. Clark, and S. DeRose, XML Path Language (XPath), Version 1.0, W3C Recommendation. Available: http://www.w3.org/TR/xpath/.
63. D. Chamberlin, J. Clark, D. Florescu, J. Robie, J. Siméon, and M. Stefanescu, XQuery 1.0: An XML query language, W3C Working Draft 07. Available. http://www.w3.org/TR/xquery/).
64. J. Clark, XSL Transformations (XSLT), Version 1.0, W3C Recommendation. Available: http://www.w3.org/TR/xslt/).
65. Z. Chen, H. V. Jagadish, L. V. S. Lakshmanan, and S. Paparizos, From tree patterns to generalized tree patterns: On efficient evaluation of XQuery, in *Proceedings of the 29th International Conference on Very Large* Data Bases, Berlin, Germany, September 2003.
66. A. Halverson, J. Burger, L. Galanis, A. Kini, R. Krishnamurthy, A. N. Rao, F. Tian, S. Viglas, Y. Wang, J. F. Naughton, and D. J. DeWitt, Mixed mode XML query processing, in *Proceedings of the 29th International Conference on Very Large Data Bases*, Berlin, Germany, September 2003.
67. J. McHugh and J. Widom, Query optimization for XML, in *Proceedings of the 25th International Conference on Very Large Data Bases*, Edinburgh, Scotland, September 1999.
68. N. Polyzotis and M. Garofalakis, Statistical synopses for graph-structured XML databases, in *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, Madison, WI, June 2002.
69. R. Milner, *Communication and Concurrency*, Englewood Cliffs, NJ: Prentice Hall (Intl. Series in Computer Science), 1989.
70. N. Polyzotis and M. Garofalakis, Structure and value synopses for XML data graphs, in *Proceedings of the 28th International Conference on Very Large Data Bases*, Hong Kong, China, August 2002.
71. P. B. Gibbons, Distinct sampling for highly-accurate answers to distinct values queries and event reports, in *Proceedings of the 27th International Conference on Very Large Data Bases*, Roma, Italy, September 2001.
72. N. Polyzotis, M. Garofalakis, and Y. Ioannidis, Selectivity estimation for XML twigs, in *Proceedings of the Twentieth International Conference on Data Engineering*, Boston, MA, March 2004.
73. N. Polyzotis, M. Garofalakis, and Y. Ioannidis, Approximate XML query answers, in *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, 2004, pp. 263–274.
74. N. Polyzotis and M. Garofalakis, XCluster synopses for structured XML content, in *Proceedings of the 22nd International Conference on Data Engineering*, 2006.
75. N. Zhang, M.T. Ozsu, A. Aboulnaga, and I.F. Ilyas, XSEED: Accurate and fast cardinality estimation for XPath queries, in *Proceedings of the 22nd International Conference on Data Engineering*, 2006.

76. J. Freire, J. R. Haritsa, M. Ramanath, P. Roy, and J. Siméon, StatiX: Making XML count, in *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, Madison, WI, June 2002.

77. Y. Wu, J. M. Patel, and H. V. Jagadish, Estimating answer sizes for XML queries, in *Proceedings of the 8th International Conference on Extending Database Technology (EDBT'2002)*, Prague, Czech Republic, March 2002.

78. W. Wang, H. Jiang, H. Lu, and J. X. Yu, Containment join size estimation: Models and methods, in *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, San Diego, CA, June 2003.

79. A. Aboulnaga, A. R. Alameldeen, and J. F. Naughton, Estimating the selectivity of XML path expressions for internet scale applications, in *Proceedings of the 27th International Conference on Very Large Data Bases*, Roma, Italy, September 2001.

80. L. Lim, M. Wang, S. Padamanabhan, J. S. Vitter, and R. Parr, XPath-Learner: An on-line self-tuning Markov histogram for XML path selectivity estimation, in *Proceedings of the 28th International Conference on Very Large Data Bases*, Hong Kong, China, August 2002.

81. Z. Chen, H. V. Jagadish, F. Korn, N. Koudas, S. Muthukrishnan, R. Ng, and D. Srivastava, Counting twig matches in a tree, in *Proceedings of the Seventeenth International Conference on Data Engineering*, Heidelberg, Germany, April 2001.

MINOS GAROFALAKIS
Yahoo! Research
Berkeley, California, and
University of California, Berkeley
Santa Clara, California

NEOKLIS POLYZOTIS
University of California,
Santa Cruz
Santa Cruz, California