

Hierarchical Data Summarization

Egemen Tanin

Department of Computer Science and Software Engineering

University of Melbourne

Victoria, Australia

egemen@csse.unimelb.edu.au

SYNONYMS

Hierarchical Data Summarization

DEFINITION

Given set of records data summaries on different attributes are frequently produced in data management systems. Commonly used examples are the number of records that fall into a set of ranges of an attribute or the minimum values in these ranges. To improve the efficiency in accessing summaries at different resolutions or due to a direct need for investigating a hierarchy that is inherent to the data type, such as dates, hierarchical versions of data summaries can be used. A data structure or algorithm is labelled as hierarchical if that structure or algorithm uses the concept of subcomponents to systematically obtain conceptually larger components. The method of obtaining a larger component is regularly induced by the user's understanding of the domain, such as dates in a year, as well as the fact that hierarchies can also be created automatically by a set of rules embedded into the system. Thus, rules used in a data structure's creation, e.g., B+-trees, are also considered as a means for hierarchical data summarization. In fact, different variants of popular data structures are used in hierarchical data summarization. Various algorithms for data reduction and aggregation have also adopted hierarchical processing techniques.

HISTORICAL BACKGROUND

From a data structures point of view, foundations of hierarchical data summarization (HDS) techniques can be found in indexing literature for databases. Although many of the indexing techniques, e.g., B+-trees, are used for efficiently selecting records stored on a disk, they can also be considered as hierarchical summaries on large amounts of data. For multidimensional and spatial data, indices such as R-trees and quadtrees can be used for HDS.

Today, many versions of popular indexing techniques that directly target retrieval of summary information exist. Some indices are also used in query optimization due to their HDS capabilities, e.g., using a space decomposition one can guess the number of records in a certain region of data before a join operation can take place. More recently, spatial indexing techniques, for example quadtrees, were developed for distributed settings such as sensor networks for HDS.

Historically, histograms are the most basic structures that could be used for data summarization. They are frequently utilized in query optimization decisions. They are also used in data warehousing. Hierarchical versions of histograms were recently built and are of interest for HDS.

From an algorithmic point of view, techniques such as wavelet transformations, sketches, and data clustering with aggregation, when run in a hierarchical fashion, can be considered as HDS techniques. These techniques are extensively deployed in data management as well as in other fields of computer science over many years.

In recent years, for distributed data processing, variants of known algorithms have become popular in HDS. For example, researchers have introduced data aggregation techniques on sensor networks that can be considered as HDS techniques that rely on sketches. In this context, random-tree-based data aggregation algorithms in sensor networks can also be considered as basic HDS techniques. All of these different roots and aspects of hierarchical data summarization are visited in this article.

SCIENTIFIC FUNDAMENTALS

B+-trees are frequently used in databases. A B+-tree is given in Figure 1 (only some parts of the tree are shown to simplify the presentation). B+-trees are hierarchical structures where internal nodes store keys and the leaf nodes contain the records attached to these keys. Due to their high fanout they are commonly shallow as well as balanced, i.e., in comparison to binary trees. They are used for efficient selection of a range of records from disks. The lowest level contains links between neighboring nodes to allow for sequential access to consecutive data items.

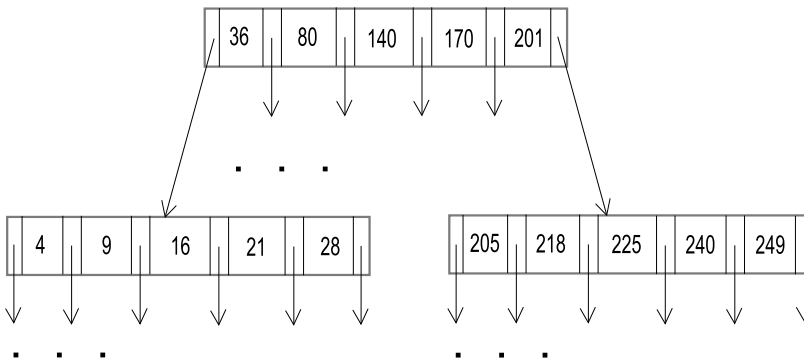


Figure 1: An example B+-tree

The B+-tree and related data structures can be considered as basic means of keeping hierarchical summary information. Given a fanout for a B+-tree, upper levels of the tree can be easily used for approximate HDS. One can refer to these levels to find the approximate number of items in a range. To make this HDS method more accurate, extra information should be maintained within the tree structure. For example, counts can be kept with each link in this hierarchy [10]. This requires extra space and maintenance costs as each count needs to be stored and updated with insertion and deletion operations. This can cause problems if many levels and nodes need to be maintained per update operation and if updates occur frequently for a given tree. If small errors in counters are tolerated then these overheads can be significantly reduced [3]. Counts form only one form of data summaries. Thus, the idea of counts is extended to other types of data summaries in [4] and [9].

For a set of queries and objects in space, such as range queries and a set of waterways in a country, spatial data structures can be used to efficiently store the data and answer queries on this data. For example, quadtrees are well-known, space-partitioning based structures. They are used with many different types of spatial data and thus many quadtree variants can be found in the literature. For example, a PR quadtree is given in Figure 2. In this example, the space is recursively divided into four quadrants until a single data item is left in each quadrant. In Figure 2, the space partitioning is shown (on the left) with its mapping tree structure (on the right). The positions of all the data items are also stored in the structure (not shown in the figure). Another related space partitioning method is the k-d tree. For k-d trees different dimensions of the underlying space is partitioned in turns at different levels of the tree. (Note that some of these methods are better named as tries, however, due to historical reasons, they are referred to as trees.)

If spatial objects are grouped together using bounding boxes and then a hierarchy of these bounding boxes

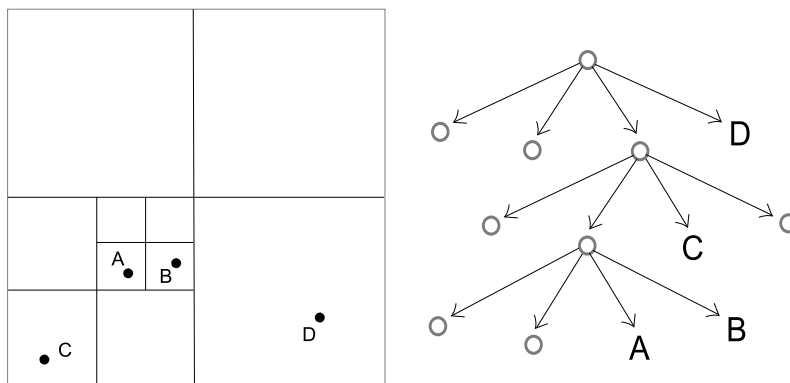


Figure 2: An example quadtree with point objects

are created, one can obtain an index called the R-tree. R-trees also have many variants. In comparison to quadtrees, they are commonly more balanced indexing schemes. However, many variants suffer from the fact that multiple bounding boxes, defining the tree nodes, can overlap in space. This nature of R-trees can reduce the pruning capability of this structure as a query may have to investigate multiple branches for the same space. Although disjoint-bounding-box based versions of R-trees exist, these variants could partition the data items into multiple boxes. There are many other spatial indices that are not presented in this article for the brevity of the presentation. The techniques mentioned are used to present HDS methods based on spatial indexing.

Similar to the case in B+-trees, spatial indices can also be viewed as HDS techniques. Moreover, data summaries can be explicitly maintained with these spatial data structures. This information can then be used for query processing, e.g., aggregate queries. Recently, spatial indexing is used in distributed settings for data summarization. For example, [7] introduces fractional cascading in sensor networks. In this approach, each sensor maintains detailed readings that it has obtained as well as data from its nearby neighbors. Information regarding other sensors are not kept as accurately. The space of sensors is partitioned using a distributed quadtree that is overlaid onto the sensor network. The partitioning is done in a similar manner to the PR quadtree example in Figure 2. With increasing distance to the rest of the sensors in the network (i.e., to faraway quadrants) the amount of data collected from them drops with a function, e.g., a logarithmic function. This paradigm utilized the fact that data and queries in sensor networks are spatially and temporally correlated. Thus, this structure can be used to efficiently serve routing requests using locally summarized data as well as to answer queries. The distributed structure can be seen as a multi-rooted HDS technique as each sensor uses the same summarization scheme independently.

Similar to fractional cascading, [6] introduces the DIMENSIONS system that uses a pyramid-based space decomposition to aggregate and summarize data in a sensor network. Each quadrant finds a “local” leader node for building a distributed pyramid of nodes with their data. Other similar systems are DIM and DIFS systems [8, 11] that use spatial indices on sensor networks for processing selection queries as well as resorting to summaries for user interest elimination. In [8] a k-d tree based structure is introduced while [11] introduces a multi-rooted quadtree type for avoiding bottlenecks from having a single root node.

Indices such as quadtrees and R-trees are also utilized in query optimization (e.g., [1]). As they represent summary information about the space they cover, they can easily be used in estimating the runtime costs of a query before it is executed.

In comparison to sophisticated indexing methods, a simple technique for summarizing data is the histogram. Histograms have long been employed in query optimization as they are compact and easy to maintain. With the emergence of data warehousing and On-line Analytical Processing (OLAP) technologies, they have also become

crucial components from a new angle in data management. For data sets that explicitly contain hierarchies, e.g., years-months-weeks, histograms can easily be used.

From a processing cost estimation and query optimization point of view, Bruno et al. [5] introduced the concept of nested buckets with histograms. This can be seen as the first form of HDS using histograms. Later, Reiss et al. [14] built on this concept for distributed settings for bandwidth usage reduction. Reiss et al. present hierarchical histograms for aggregate query processing on identification data, i.e., RFIDs.

From an algorithmic point of view many methods that have long been used in approximating and summarizing data can be considered as hierarchical approaches to summarization. For example, wavelet transformations are well-established techniques in signal processing that can be used and considered as HDS methods. Lets consider wavelets in the context of spatial data to give a simple example.

A three dimensional (3D) object in space can be approximated using a triangular mesh. One can use different sets of triangles, i.e., small or large, to give a more or less detailed approximation of the surface of the 3D object. Thus, an object can be represented in different resolutions using different meshes. If these meshes are related to each other geometrically, one can easily progressively update the details for this object on demand. Therefore, if M^I denotes a triangular mesh at resolution I , one can then represent an object as a series of meshes, M^0, M^1, \dots, M^J , where, M^0 is the base mesh and M^J is the final mesh. Figure 3(a) shows a triangular mesh with one triangle, $M^0, (1, 2, 3)$, for a 2D object. The triangle is the coarse approximation for the surface of the given circle.

Lets consider a simple transformation. To obtain a higher resolution approximation of the given surface in the figure, the triangle $(1, 2, 3)$ is divided into four sub-faces by introducing new vertices $(4', 5', 6')$, Figure 3(b). The new set of vertices are now displaced to make the mesh better fit to the surface of the circle. The new, finer resolution mesh M^1 , is shown in Figure 3(c). This operation can be done recursively and can be represented with a simple transformation function. The coefficients that represent the difference between M^0 and M^1 are d_4^0, d_5^0 , and d_6^0 . In this simple wavelet transformation, for example, the wavelet coefficient d_4^0 is obtained by $v_4^1 - \frac{v_1^0 + v_2^0}{2} = v_4^1 - v_4^0$. Thus, the wavelet-based decomposition of a mesh M^J produces a base mesh M^0 and the sets, $\{W_0, W_1, \dots, W_{J-1}\}$, of coefficients. From an HDS point of view, the recursive execution of the above mentioned method can be seen as a hierarchical summarization of a detailed polygonal representation of a complex data set. Various further HDS methods can be derived using the base concept of wavelets. For example, recently, [2] uses wavelets with R-trees to progressively retrieve and refine spatial data from a remote database.

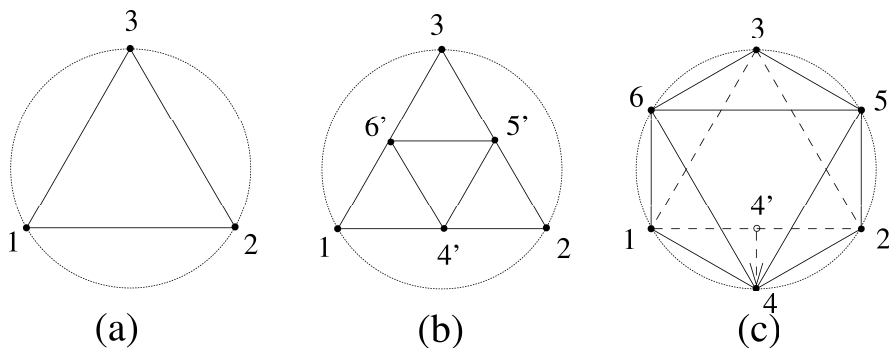


Figure 3: A wavelet-based approximation

With the emergence of distributed systems such as sensor networks, aggregate query processing itself can now also be considered as a HDS technique. For example, to process an aggregate query in sensor networks, [12] uses a random-tree with in-network aggregation. Each node in this tree can compute an aggregate from its sub-trees,

such as a minimum, and then pass this information to the higher-levels of the tree along the data collection path. The base-station, root, can then present a summary of the sensor data to the user.

Random-trees, however, are not robust. A single failure can cause significant problems (especially when a node that is close to the root fails). To address these problems, researchers have been working on multi-path data aggregation methods. In this scheme, multiple reports, due to the wireless coverage advantage in sensor networks, can be sent through different routes for increasing the robustness of the data collection method. However, this can cause deviations in certain aggregation operations, e.g., counts, as the same data is incorporated to the result multiple times. In [13], Nath et al. introduce a sketch theory-based HDS method to address this issue. They map aggregate functions, e.g., counts, to a set of order and duplicate insensitive synopsis generation and fusion functions.

Data clustering forms another area of research that, when applied using data aggregates and hierarchies, can be considered as a source of HDS techniques. For example, [15] introduces the STING system which uses a hierarchy of cells that contain aggregate information about the individual data items. Thus, for many query types, they can resort to these cells, rather than items, to answer queries efficiently. For queries that cannot be answered using summary data, individual data items can still be used as a backup strategy.

KEY APPLICATIONS*

Key applications of HDS techniques are aggregate query processing and query optimization. If many queries are interested in retrieving summary data, e.g., aggregate queries, then maintaining a hierarchical summary would be efficient. For example, for data warehousing applications with hierarchical data, the benefits for maintaining a hierarchical summary could be significant. Data summaries have also long been used in query optimization. For example, an optimizer can use HDS techniques for selectivity estimation on attributes.

FUTURE DIRECTIONS

There is a significant amount of activity in using data summaries in distributed settings and especially in sensor networks. In addition, with the emerging research directions in location-based services and VANETs, readers may expect to see the use of spatial HDS techniques more frequently. In distributed settings, bandwidth savings on the communication optimization front using HDS could be significant.

URL TO CODE*

Demos for many of the spatial indices that are mentioned in this article can be found at <http://www.cs.umd.edu/~hjs/quadtrees/index.html>.

CROSS REFERENCE*

B+-TREES, QUADTREES, R-TREES, HISTOGRAMS, WAVELETS, SKETCHES, INDEXING, AGGREGATE QUERIES, QUERY PROCESSING AND OPTIMIZATION, SPATIAL AND MULTIDIMENSIONAL DATABASES, SENSOR NETWORKS

RECOMMENDED READING

- [1] A. Aboulmaga and W. G. Aref. Window query processing in linear quadtrees. *Distributed and Parallel Databases*, 10(2):111–126, 2001.
- [2] M. E. Ali, R. Zhang, E. Tanin, and L. Kulik. A motion-aware approach to continuous retrieval of 3d objects. In *ICDE*, Cancun, Mexico, 2008.
- [3] G. Antoshenkov. Query processing in DEC RDB: Major issues and future challenges. *IEEE Data Engineering Bulletin*, 16(4):42–45, 1993.

- [4] P. M. Aoki. Generalizing "search" in generalized search trees. In *ICDE*, pages 380–389, Orlando, FL, 1998.
- [5] N. Bruno, S. Chaudhuri, and L. Gravano. STHoles: A multidimensional workload-aware histogram. *SIGMOD Record*, 30(2):211–222, 2001.
- [6] D. Ganesan, D. Estrin, and J. Heidemann. DIMENSIONS: Why do we need a new data handling architecture for sensor networks? In *ACM Workshop on Hot Topics in Networks*, Princeton, NJ, 2002.
- [7] J. Gao, L. J. Guibas, J. Hersberger, and L. Zhang. Fractionally cascaded information in a sensor network. In *IPSN*, pages 311–319, Berkeley, CA, 2004.
- [8] B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy, and S. Shenker. DIFS: A distributed index for features in sensor networks. In *IEEE Workshop on Sensor Network Protocols and Applications*, Anchorage, AK, 2003.
- [9] J. M. Hellerstein, J. F. Naughton, and A. Pfeffer. Generalized search trees for database systems. In *VLDB*, pages 562–573, Zurich, Switzerland, 1995.
- [10] D. E. Knuth. *Sorting and Searching, The Art of Computer Programming*, volume 3. Addison Wesley Publishing, Redwood City, CA, 1973.
- [11] X. Li, Y. J. Kim, R. Govindan, and W. Hong. Multi-dimensional range queries in sensor networks. In *SenSys*, pages 5–7, Los Angeles, CA, 2003.
- [12] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TinyDB: An acquisitional query processing system for sensor networks. *ACM Transactions on Database Systems*, 30(1):122–173, 2005.
- [13] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson. Synopsis diffusion for robust aggregation in sensor networks. In *SenSys*, pages 250–262, Baltimore, MD, 2004.
- [14] F. Reiss, M. Garofalakis, and J. M. Hellerstein. Compact histograms for hierarchical identifiers. In *VLDB*, pages 870–881, Seoul, Korea, 2006.
- [15] W. Wang, J. Yang, and R. Muntz. STING: A statistical information grid approach to spatial data mining. In *VLDB*, pages 186–195, Athens, Greece, 1997.