Chapter 1.11
Histogram-Based Compression of Databases and Data Cubes

Alfredo Cuzzocrea
University of Calabria, Italy

INTRODUCTION

Histograms have been extensively studied and applied in the context of *Selectivity Estimation* (Kooi, 1980; Muralikrishna & DeWitt, 1998; Piatetsky-Shapiro et al., 1984; Poosala et al., 1996; Poosala, 1997), and are effectively implemented in commercial systems (e.g., Oracle Database, IBM DB2 Universal Database, Microsoft SQL Server) to **query optimization** purposes. In statistical databases (Malvestuto, 1993; Shoshani, 1997), histograms represent a method for approximating **probability distributions**. They have also been used in data mining activities, intrusion detection systems, scientific databases, that is, in all those applications which (*i*) operate on huge numbers of detailed records, (*ii*) extract useful knowledge only from condensed information consisting of summary data, (*iii*) but are not usually concerned with detailed information. Indeed, histograms can reach a surprising efficiency and effectiveness in approximating the actual distributions of data starting from **summarized information**. This has led the research community to investigate the use of histograms in the fields of database management systems (Acharya et al., 1999; Bruno et al., 2001; Gunopulos et al., 2000; Ioannidis & Poosala, 1999; Kooi, 1980; Muralikrishna & DeWitt, 1998; Piatetsky-Shapiro et al., 1984; Poosala, 1997; Poosala & Ioannidis, 1997), *online analytical processing* (OLAP) systems (Buccafurri et al., 2003; Cuzzocrea, 2005a; Cuzzocrea & Wang, 2007; Furfaro et al., 2005; Poosala & Ganti, 1999), and data stream management systems (Guha et al., 2001; Guha et al., 2002; Thaper et al., 2002), where, specifically, compressing data is mandatory in order to obtain fast answers and manage the endless arrival of new information, as no bound can be given to the amount of information which can be received.

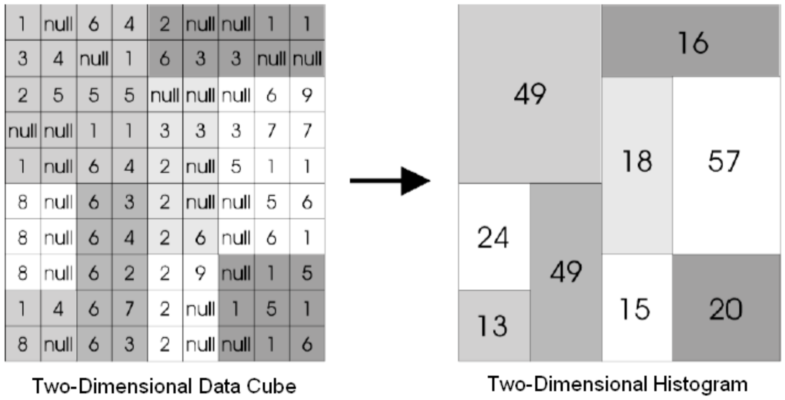
Histograms are data structures obtained by partitioning a **data distribution** (or, equally, a data domain) into a number of mutually disjoint blocks, called ***buckets***, and then storing, for each bucket, some **aggregate information** of the corresponding range of values, like the sum of values in that range (i.e., applying the SQL aggregate operator SUM), or the number of occurrences (i.e., applying the SQL aggregate operator COUNT), such that this information retains a certain "summarizing content."

Figure 1 shows an instance of a histogram built on a two-dimensional **data cube** (left-side of the figure), represented as a matrix. The corresponding (two-dimensional) histogram (right-side of the figure) is obtained by (*i*) partitioning the matrix into some rectangular buckets which do not overlap, and (*ii*) storing for each so-obtained bucket the sum of the measure attributes it contains.

Histograms are widely used to support two kinds of applications: (*i*) selectivity estimation inside *Query Optimizers* of DBMS, as highlighted before, and (*ii*) *approximate query answering* against databases and data cubes. In the former case, the data distribution to be compressed consists of the frequencies of values of attributes in a relation (it should be noted that, in this vest, histograms are mainly used within the core layer of DBMS, thus dealing with databases properly). In the latter case, the data distribution to be compressed consists of the data items of the target domain (i.e., a database or a data cube) directly, and the goal is to provide fast and approximate answers to resource-intensive queries instead of waiting-for time-consuming exact evaluations of queries. To this end, a widely-accepted idea is that of evaluating (with some approximation) queries against ***synopsis data structures*** (i.e., succinct, compressed representations of original data) computed over input data structures (i.e., a database or a data cube) instead of the same input data structures. Histograms are a very-popular class of synopsis data structures, so that they have been extensively used in the context of approximate query answering techniques. Some relevant experiences concerning this utilization of histograms are represented by the work of Ioannidis and Poosala (Ioannidis & Poosala, 1999), that propose using histograms to provide approximate answers to set-valued queries, and the work of Poosala and Ganti (Poosala & Ganti, 1999), that propose using histograms to provide approximate answers to *range-queries* (Ho et al., 1997) in OLAP.

In both utilizations, a relevant problem is how to reconstruct the original data distribution form the compressed one. In turn, this derives from the fact that the original data distribution summarized within a bucket cannot be reconstructed exactly, but can be approximated using some estimation

Figure 1. A two-dimensional data cube and its corresponding two-dimensional histogram



1	null	6	4	2	null	null	1	1
3	4	null	1	6	3	3	null	null
2	5	5	5	null	null	null	6	9
null	null	1	1	3	3	3	7	7
1	null	6	4	2	null	5	1	1
8	null	6	3	2	null	null	5	6
8	null	6	4	2	6	null	6	1
8	null	6	2	2	9	null	1	5
1	4	6	7	2	null	1	5	1
8	null	6	3	2	null	null	1	6

Related Content