Query-Condition-Aware Histograms in Selectivity Estimation Method

Dariusz R. Augustyn

Abstract. The paper shows an adaptive approach to the query selectivity estimation problem for queries with a range selection condition based on continuous attributes. The selectivity factor estimates a size of data satisfying a query condition. This estimation is calculated at the initial stage of the query processing for choosing the optimal query execution plan. A non-parametric estimator of probability density of attribute values distribution is required for the selectivity calculation. Most of known approaches use equi-width or equi-height histograms as representations of attribute values distributions. The proposed approach uses a new type of histogram based on either an attribute values distribution or a distribution of range bounds of a query selection condition. Applying query-condition-aware histogram lets obtain more accurate selectivity values than using a standard histogram. The approach may be implemented as some extension of query optimizer of DBMS Oracle using ODCI Stats module.

Keywords: database query optimization, selectivity estimation, query-condition-aware histogram.

1 Introduction

The database queries are processed in two phases—a prepare (parse) phase and an execution one. During the prepare phase a database management system (DBMS) finds the optimal method of query performing so-called the execution plan. This is done by a module of DBMS—the cost query optimizer. Finding the optimal query execution method requires an estimation of size of the query result set (this must be done before the query is executed). This is the reason why a query selectivity

Dariusz R. Augustyn
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: draugustyn@polsl.pl

T. Czachórski et al. (Eds.): Man-Machine Interactions 2, AISC 103, pp. 437–446.
springerlink.com

estimator was introduced. Query selectivity is a number of rows satisfying the query condition divided by a total number of rows of whole input set. For a single-table query selectivity is a fraction of table rows satisfying the query predicate.

For a single-table query (Q) with a simple range selection condition ($a < X < b$) based on a table attribute (X) with continuous domain the selectivity is defined as follows:

$$sel(Q(a < X < b)) = \int_a^b f(x)\,\mathrm{d}x, \tag{1}$$

where $f(x)$ is a probability density function (PDF) of X values distribution.

As we can see in (1) obtaining the selectivity value requires some non-parametric estimator of PDF. Commonly equi-width histograms or equi-height ones are used is DBMS as representations of an attribute values distribution. However, there are many other unconventional approaches for representing an attribute values distribution which are suitable for the effective selectivity estimation like these ones based on kernel estimator [6], Bayesian Network [5], Cosine Transform [8], Discrete Wavelets Transform [3].

In above–mentioned approaches the distribution representation is obtained by scanning set of values of selected attribute. Some other methods are based of self-tuning histograms [2]. Here the representation is created or updated during executions of queries. A database server uses information about a result size of a query and updates on-line the representation structure. It makes that the created representation takes into account set of query conditions. However this approach assumes on-line rebuilding the representation structure. The another problem is that commercial database servers don't support a programming interface suitable for an implementation of this solution.

The proposed mechanism based on 1-dimensional query-condition-aware histogram collects on-line simple information about distribution of query conditions (obtaining the query result size is not required). This approach may be applied for queries with selection conditions based on only one attribute.

For efficient reason a simple approximate 1-dimensional representation of a 2-dimensional query conditions distribution was proposed. The main workload of creating the representation (i.e. the query-condition-aware histogram) is deferred—it may be done when a user want to. The another advantage of this approach is that some commercial DBMS (i.e. Oracle DBMS) allows to extend the functionality of optimizer module that the proposed solution may be easy implemented.

2 The Idea of Query-Condition-Aware Histograms

The main idea of the proposed approach based on query-condition-aware histogram will be presented using a simple example described in this section.

Let's assume that domain of X attribute is $[0, 1]$. The distribution of X values is a superposition of two Gaussian clusters i.e. the probability density function of X values is:

$$f(x) = 0.5\,PDF(N(0.5, 0.06)) + 0.5\,PDF(N(0.6, 0.002)). \quad (2)$$

The *PDF* and the histogram $H_x(1000)$ of X values based on a sample with $N_x = 1000$ buckets are shown in Fig. 1.

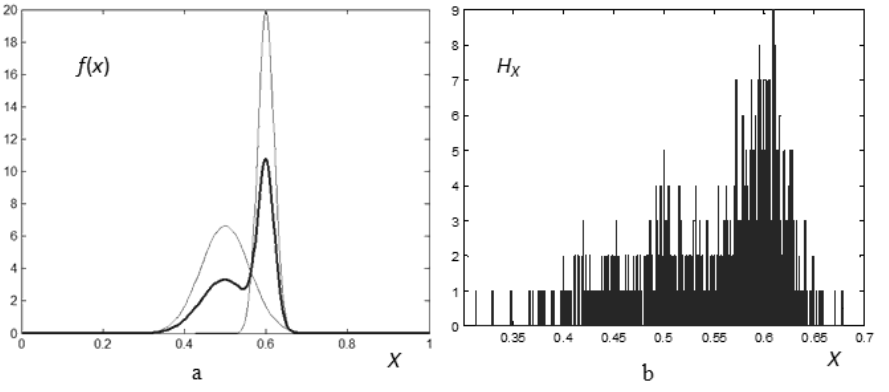

Fig. 1 (**a**) *PDF* of X values distribution (bold curve), (**b**) Histogram H_x of X values distribution

Let's make some assumptions for the distribution of condition bounds—a and b. These assumptions are based on observations of real information systems. Let's assume that some values of X are more interesting for a user then the others (e.g. recently inserted data are more required by a user then older one). So left query bounds mainly concentrate near a some value and the distribution of a values is a Gaussian cluster e.g. this one shown in Fig. 2a.

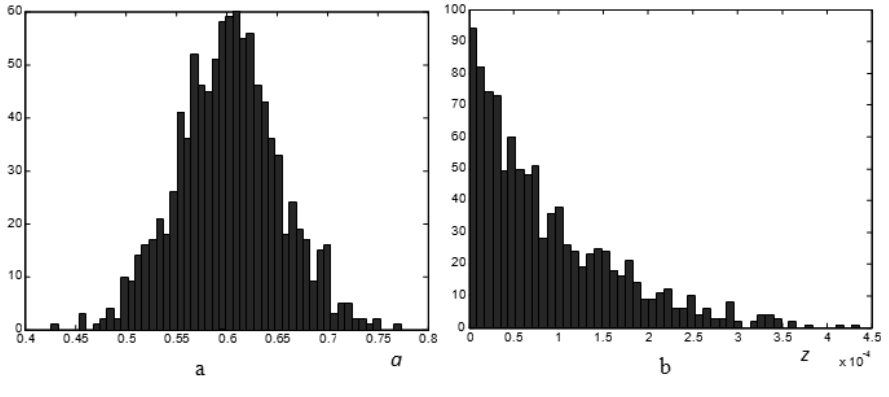

Fig. 2 (**a**) Histogram of a values distribution (a—left query bound), (**b**) Histogram of $z = b - a$ values distribution (z—length of query condition interval)

Let's assume that query intervals $z = b - a$ are rather narrow (e.g. commonly a user requires data 'placed near' the actual data). The distribution of z is assumed as a truncated exponent distribution shown in Fig. 2b, so rather small values of z are preferred.

Both histograms shown in Fig. 2 were made for 1000 sample query conditions.

According to the distribution of a and z shown in Fig. 2 the joint distribution of $a \times b$ can be presented as 2D view of a and b pairs (Fig. 3a) or as the smoothed bivariate *PDF* (Fig. 3b).

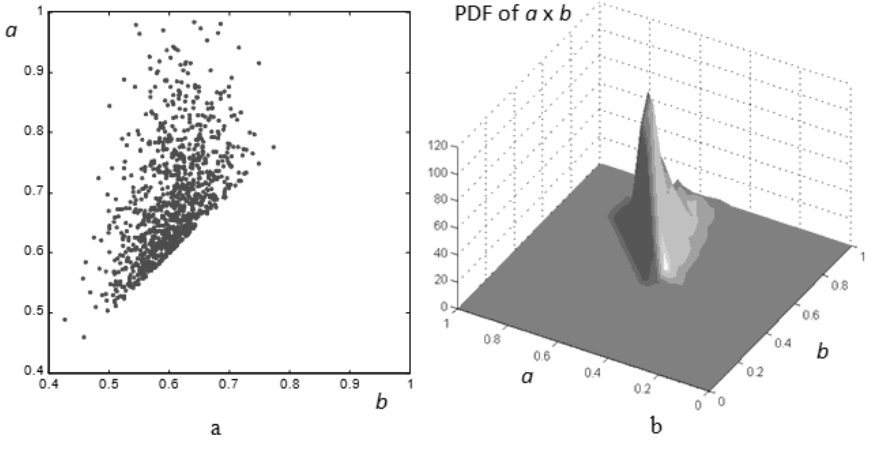

Fig. 3 (**a**) 2D view of the $a \times b$ distribution, (**b**) Bivariate *PDF* of the joint a and b distribution

A region of user interest of X values can be described by a new type histogram based on overlapped intervals of query conditions. This equi-width-type histogram will be called H_{QCD} (histogram of a query conditions distribution). A query condition interval (a, b) overlaps some subset of H_{QCD} buckets, so values in those buckets are incremented by 1. This process is illustrated in Fig. 4. The method of overlapping is shown in Fig. 4.b where bold segments correspond to these H_{QCD} buckets which will be incremented.

H_{QCD} may be treated as a 1D approximate representation of bivariate *PDF* of $a \times b$ values distribution (e.g. the histogram H_{QCD} from Fig. 5a is the 1D representation of *PDF* form Fig. 3b)

The transform $a \times b$ distribution into H_{QCD} is not invertible. The number of buckets of H_{QCD} (from Fig. 5) is equal 100 ($N_{\mathrm{QCD}} = 100$).

A standard equi-width-type histogram called H_{STD} was also made for data described by distribution of X values form Fig. 1. This $H_{\mathrm{STD}}(12)$ with $N_b = 12$ buckets was shown in Fig. 6a. Such type of histogram is commonly used by DBMS as a representation of the attribute values distribution.

A new type of histogram for representing X attribute values distribution—a query-condition-aware histogram is proposed in this paper. This type of histogram

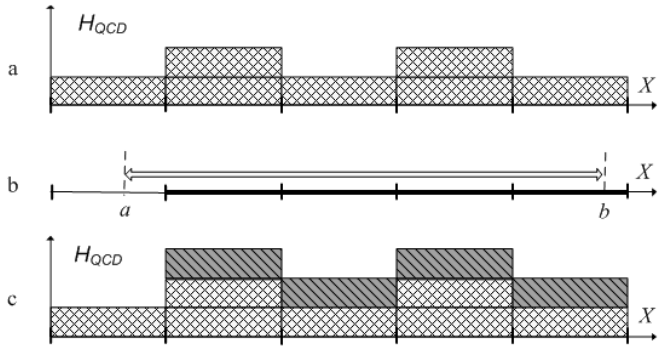

Fig. 4 Method of a H_{QCD} building: (**a**) initial state of H_{QCD}, (**b**) some interval of a query condition, (**c**) H_{QCD} state after taking into account the query condition interval

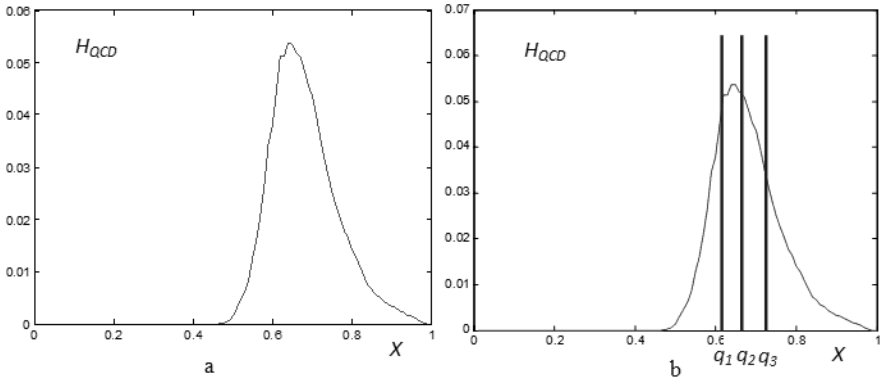

Fig. 5 (**a**) H_{QCD}—the histogram of query condition distribution as the approximate representation of the $a \times b$ distribution from Fig. 3, (**b**) H_{QCD} and $1/4$-th quantiles (3 bold vertical lines)

named H_{QCA} takes into account either a distribution of attribute values (H_x) or a distribution of query conditions (H_{QCD}). This lets to reflect user requirements described by the a and b bounds distributions.

p-th quantiles are obtained using H_{QCD} histogram at the first stage of the H_{QCA} construction algorithm. p is equal $1/N_q$, where N_q is the number of the first level of X domain splitting in H_{QCA}. In our example N_q is equal 4 so p-th quantiles are simply quartiles. p-th quantiles designate bounds of superhistogram intervals. As we can see in Fig. 5b three vertical bold lines separate the X domain into four intervals in the superhistogram. Intervals of superhistogram are narrow in the region of the user interest (see intervals: $[q_1, q_2)$ and $[q_2, q_3)$).

This stage of H_{QCA} construction algorithm allows to reflect the region of the user interest.

A small equi-width-type subhistogram is build for every bucket of the superhistogram at the second stage of the H_{QCA} construction algorithm. There are N_q subhistograms. Every subhistogram has N_{eqb} buckets, so the whole query-condition-aware histogram $H_{QCA}(N_q, N_{\mathrm{eqb}})$ has $N_q \times N_{\mathrm{eqb}} = N_b$ buckets. In our example the subhistogram has $N_{\mathrm{eqb}} = 3$ buckets so $H_{\mathrm{QCA}}(4,3)$ histogram has $4 \times 3 = 12$ buckets. This H_{QCA} histogram is shown in Fig. 7a.

The second stage of the H_{QCA} construction algorithm allows to reflect the table attribute values distribution. The resolution of H_{QCA} is high in the region of the user high interest (widths of subhistogram buckets are small in such region). We can see it in Fig. 7b where zoomed part of H_{QCA} is shown (3 buckets in very small interval $[q_1, q_2)$).

We can notice that a trivial $H_{\mathrm{QCA}}(1, N_b)$ is equivalent to some $H_{\mathrm{STD}}(N_b)$ and then a distribution of query range bounds (H_{QCD}) is not taken into account.

For ranking H_{QCA} and H_{STD} histogram-based selectivity estimation methods, the relative error of selectivity estimation was defined as follows:

$$RE(Q(a < X < b)) = \frac{|sel(Q) - sel\hat{}(Q)|}{sel(Q)}\ 100\%, \tag{3}$$

were *sel* is an exact value of selectivity of a range query Q and *sel^* is an approximate selectivity value obtained using a histogram. All values of *REs* for all queries Q (1000 query conditions according to the $a \times b$ distribution) were used for ranking H_{QCA} and H_{STD}.

Two histograms (with the same number of buckets)—$H_{\mathrm{STD}}(12)$ and $H_{QCA}(4,3)$ were taken into account. Mean values of relative errors—*MREs* (small squares in Fig. 6b), median values (bold horizontal lines), and distances between the 3rd and the 1st quartile (heights of boxes) were calculated for set of *REs* using both histograms. Selectivity estimation based on the proposed H_{QCA} is better than this one bases on H_{STD} as we can see in Fig. 6. Error statistics values for H_{QCA} are less than these ones for H_{STD}, i.e. means are equal 31 % and 211 %, medians—29 % and 41 %, quartile distances—29 % and 132 %.

Let's consider the problem of finding the error-optimal H_{QCA} for given the total number of buckets ($N_b = 12$). This is equivalent to choose some $H_{\mathrm{QCA}}(N_q, N_{\mathrm{eqb}})$ for $1 \le N_q \le N_b$ with the smallest mean error value (*MRE*) were $N_q \times N_{\mathrm{eqb}} = N_b$ and N_q is a positive divisor of N_b. Figure 8 shows that the pair $(N_q, N_{\mathrm{eqb}}) = (4,3)$ is optimal.

3 The Proposed Selectivity Estimation Method

The method of selectivity estimation based on proposed query-condition-aware histogram is presented in this section.

Obtaining a selectivity value based on the proposed histograms does not differ from these ones based on standard histograms. Thus, the effectiveness of the selectivity calculation method is comparable to these ones well-known approaches. Thus, there is no need experimental results to prove this.

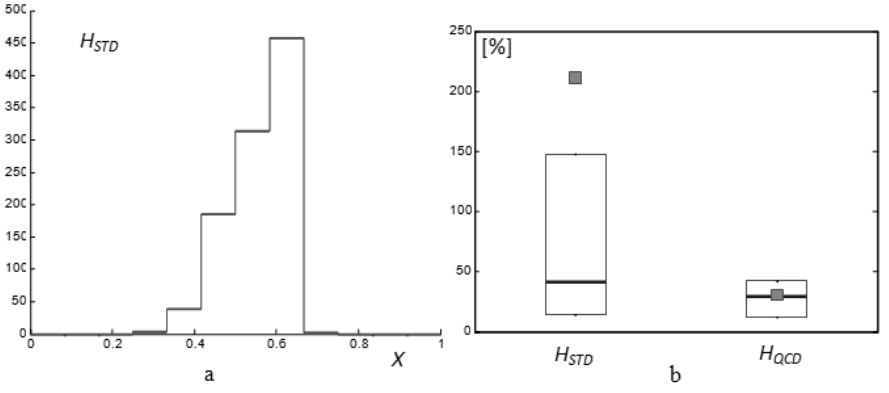

Fig. 6 (**a**) Standard histogram $H_{\mathrm{STD}}(12)$ made for X attribute values, (**b**) Accuracy rating of selectivity estimation methods based on $H_{\mathrm{STD}}(12)$ and $H_{\mathrm{QCA}}(4,3)$

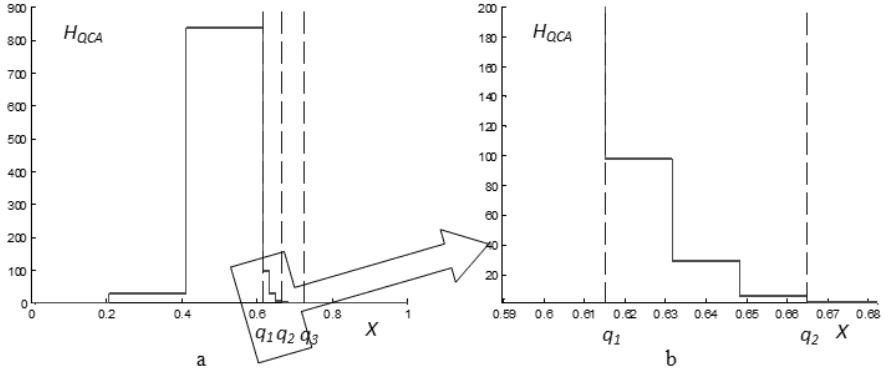

Fig. 7 (**a**) $H_{QCA}(4,3)$—the final query-condition-aware histogram, (**b**) Zoomed view of the third subhistogram (the region of the user high interest)

N_q	N_{eqb}	MRE [%]
1	12	211
2	6	181
3	4	34
4	3	31
6	2	35
12	1	39

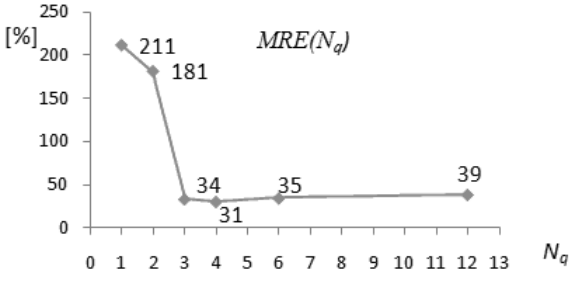

Fig. 8 Mean relative error $MRE(N_q)$ of the selectivity estimation based on $H_{\mathrm{QCA}}(N_q, 12/N_q)$ for $N_q = 1,2,3,4,6,12$

The advantage of the method results form a conformity to the query bounds distribution. This makes that selectivity approximation errors for H_{QCA}-based method are less than these ones based on H_{STD}.

The main issue of the presented method is a problem of H_{QCA} construction. The construction of H_{QCA} is rather not time-critical. Only the simple operation for H_{QCD} construction i.e. gathering data about the $a \times b$ distribution is time-critical because it is done during every query execution.

However, because of an efficiency reason the final version of the method of H_{QCA} creating is a little differ than this one described in the previous section. Instead of calculating the mean relative selectivity error for a sample set of queries this approach is directly based on matching histograms.

The method of finding the optimal $H_{\mathrm{QCA}}(N_q, N_{\mathrm{eqb}})$ for a given N_b (where $N_b = N_q \times N_{\mathrm{ebq}}$) will be presented in the following subsection. The score function is different than this one based on *MRE*.

3.1 The H_{QCD}-Based Histogram Construction Algorithm

Before H_{QCA} creation the relevant should be created. H_{QCD} describes a query bounds distribution and it has N_{QCD} buckets ($N_{\mathrm{QCD}} >> N_q$).

At first a equi-height histogram $H_{\mathrm{QCD}}\hat{}$ is created. This $H_{\mathrm{QCD}}\hat{}$ is an approximation of H_{QCD} based on $1/N_q$-th quantiles. Intervals bounds of $H_{\mathrm{QCD}}\hat{}$ are determined by values of quantiles.

A histogram H_x is created next. H_x describes a X values distribution and it has N_x buckets ($N_x >> N_b$).

The main procedure of the algorithm finds such pair (N_q, N_{eqb}) for H_{QCA} that some score function value is the smallest. The score function is a sum of two addends (both taken with the same weight in the assumed approach):

1. a sum of squares of distances between values of H_{QCD} and $H_{\mathrm{QCD}}\hat{}$,
2. a sum of squares of distances between values of H_x and H_{QCA}.

A value of the first addend reflects matching H_{QCD} and $H_{\mathrm{QCD}}\hat{}$ (i.e. mismatching to the query bounds values distribution). A value of the second addend reflects matching H_x and H_{QCA} (i.e. mismatching to the attribute value distribution).

This method uses the absolute error based on values from histograms (the previous method uses the mean relative error of selectivity estimation). However, for the data presented in the previous section, this method gives the same error-optimal parameter values of the result H_{QCA} i.e. $(N_q, N_{\mathrm{eqb}}) = (4,3)$.

3.2 Implementing the Method in Oracle DBMS

The presented method of selectivity estimation may be implemented and integrated into Oracle DBMS using ODCIStats (**O**racle **D**ata **C**artridge **I**nterface **Stat**istics) module [7].

ODCIStats enables the interface for implementing creation of a user-defined attribute values distribution representation so-called a user-defined statistics. It also enables the interface for implementing a user-defined selectivity function (which operates on the user-defined statistics). This allows the query optimizer to invoke the user-defined selectivity function during evaluation a query condition (in the prepare phase of the processed query). Such application of ODCIStats module was presented in [4, 1].

The above-mentioned ODCIStats functionality also allows to gather on-line information about query conditions i.e. query range bounds values distribution. Such purposed mechanism allows to create H_{QCD} histogram. This is unconventional usage of ODCIStats module (i.e. invoking *ODCISelectivity* function causes not only selectivity calculation but it updates H_{QCD} too).

The proposed implementation may allow DBMS to work in two modes. In the first mode named the gathering mode (or the learning mode) DBMS may update H_{QCD} for each processed query. By invoking the selectivity function the query optimizer also increments values in histogram buckets of H_{QCD} (Fig. 4).

After then DBMS is switched into the second mode named the normal mode (and H_{QCD} is not updated). Now the user-defined statistics (H_{QCA} histogram) may be created using standard command like *dbms_stats.gather_table_stats*. This invokes implicitly the algorithm for creating a new H_{QCA} from H_x and H_{QCD} which was described in the previous subsection.

From this moment the query optimizer may use implicitly the user-defined selectivity method based on the created H_{QCA}.

4 Conclusions

The problem of selectivity estimation for range query conditions is considered in this paper. Known approaches to obtaining selectivity are based on histograms. These histograms are only non-parametric estimators of an attribute values distribution.

The new approach which reflects a query conditions distribution is presented in this paper. A new type histogram named query-condition-aware one (H_{QCA}) is proposed in the article. The query-condition-aware histogram reflects either an attribute values distribution or a range query bounds values distribution. The algorithm of construction of the error-optimal query-condition-aware histogram was introduced.

Future works will concentrate on increasing effectiveness of the H_{QCA} construction algorithm by applying dynamic programming technique for efficient scoring histograms matching.

References

1. Augustyn, D.: Applying advanced methods of query selectivity estimation in Oracle DBMS. In: Cyran, K., Kozielski, S., Peters, J., Stanczyk, U., Wakulicz-Deja, A. (eds.) Man-Machine Interactions. Advances in Intelligent and Soft Computing, vol. 59, pp. 585–593. Springer, Heidelberg (2009)

2. Bruno, N., Chaudhuri, S., Gravano, L.: STHoles: a multidimensional workload-aware histogram. SIGMOD Record 30, 211–222 (2001)
3. Chakrabarti, K., Garofalakis, M., Rastogi, R., Shim, K.: Approximate query processing using wavelets. The VLDB Journal 10, 199–223 (2001)
4. Döller, M., Kosch, H.: The MPEG-7 multimedia database system (MPEG-7 MMDB). Journal of Systems and Software 81, 1559–1580 (2008)
5. Getoor, L., Taskar, B., Koller, D.: Selectivity estimation using probabilistic models. SIGMOD Record 30, 461–472 (2001)
6. Gunopulos, D., Kollios, G., Tsortas, V.J., Domeniconi, C.: Selectivity estimators for multidimensional range queries over real attributes. The VLDB Journal 14, 137–154 (2005)
7. Oracle®Corporation: Using Extensible Optimizer, `http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28425/ext_optimizer.htm`
8. Yan, F., Hou, W.C., Jiang, Z., Luo, C., Zhu, Q.: Selectivity estimation of range queries based on data density approximation via cosine series. Data & Knowledge Engineering 63, 855–878 (2007)