

Обработка и исполнение запросов в СУБД (Лекция 3)

Классические системы: гистограммы и оценка промежуточных результатов

v6

Георгий Чернышев

Высшая Школа Экономики

chernishev@gmail.com

16 сентября 2020 г.

История [Ioannidis, 2003]:

- *ιστος* (istos, “мачта”) + *γραμμα* (gram-ma, “надпись”);
- Термин придумал Karl Pearson, есть ссылки с лекции по статистике 1892 г;
- Много источников указывают на то, что такие объекты использовались гораздо раньше;
- Bar charts: “Commercial and Political Atlas (London 1786), William Playfair”, подвид гистограмм.

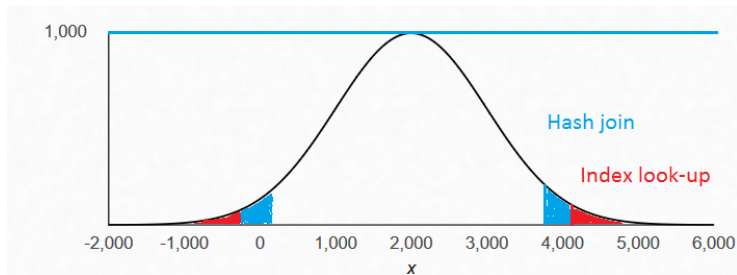
Применение в информатике:

- в обработке изображений и системах компьютерного зрения;
- в геоинформационных системах;
- в базах данных: сжатие данных и аппроксимация распределений
 - оценка селективности;
 - приближенные ответы на запросы (для оптимизации);
 - фидбек на пользовательские запросы, получаемый перед выполнением (профилирование запросов).

- Это набор пар вида (значение, частота);
 - Знание распределения позволит оптимизировать запросы;
 - Оно большое, если хранить всё, то выигрыша не будет :(
- Надо **дешево** аппроксимировать распределение данных.

Для этого и используются гистограммы.

Как оптимизировать?



Q1: SELECT X FROM T_1 WHERE $X = \alpha$;

- Если выбирается мало записей — индекс по T_1 ,
- Иначе — последовательный просмотр.

Q2: SELECT Z FROM T_1 , T_2 WHERE $T_1.X = T_2.Y$ AND $T_1.X = \alpha$;

- Если выбирается мало записей — можно hash join по T_1 ,
- Иначе — что-то другое.

Гистограмма над атрибутом X строится с помощью разбиения распределения данных в X на $\beta(\geq 1)$ попарно различных подмножеств (называемых ведрами) и аппроксимации частот и значений в каждом ведре единым образом.

OLYMPIAN		
Name	Salary	Department
Apollo	60K	Energy
Aphrodite	60K	Domestic Affairs
Aris	50K	Defense
Artemis	60K	Energy
Athena	70K	Education
Demeter	60K	Agriculture
Ermis	60K	Commerce
Hefestus	50K	Energy
Hera	90K	General Management
Hestia	50K	Domestic Affairs
Poseidon	80K	Defense
Pluto	80K	Justice
Zeus	100K	General Management

Table 1: The OLYMPIAN relation

Department	Frequency
General Management	2
Defense	2
Education	1
Domestic Affairs	2
Agriculture	1
Commerce	1
Justice	1
Energy	3

Table 2: Frequency distribution of Department

¹Изображение взято из [Ioannidis and Poosala(a), 1995]

Гистограммы по данным

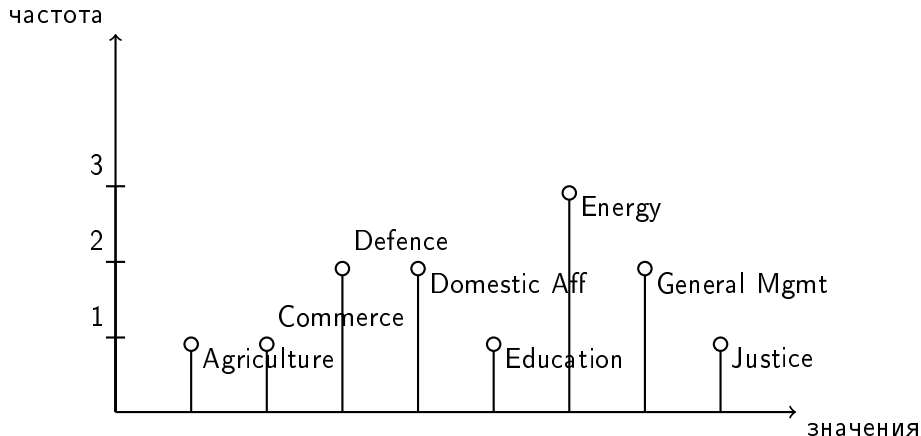
Department	Histogram H1		Histogram H2		Histogram H3	
	Frequency in Bucket	Approximate Frequency	Frequency in Bucket	Approximate Frequency	Frequency in Bucket	Approximate Frequency
Agriculture	1	1.50	1	1.33	1	1.43
Commerce	1	1.50	1	1.33	1	1.43
Defense	2	1.50	2	1.33	2	1.43
Domestic Affairs	2	1.50	2	2.50	2	1.43
Education	1	1.75	1	1.33	1	1.43
Energy	3	1.75	3	2.50	3	3.00
General Management	2	1.75	2	1.33	2	1.43
Justice	1	1.75	1	1.33	1	1.43

Table 3: Three types of histograms for the Department attribute

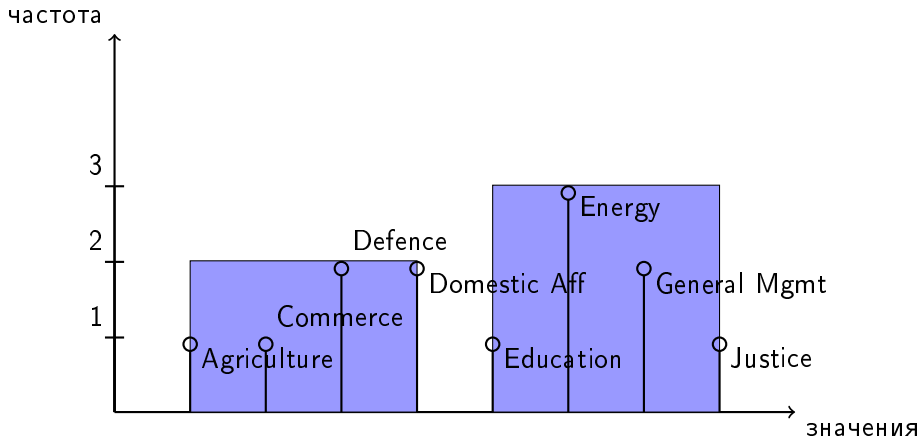
2

²Изображение взято из [Ioannidis and Poosala(a), 1995]

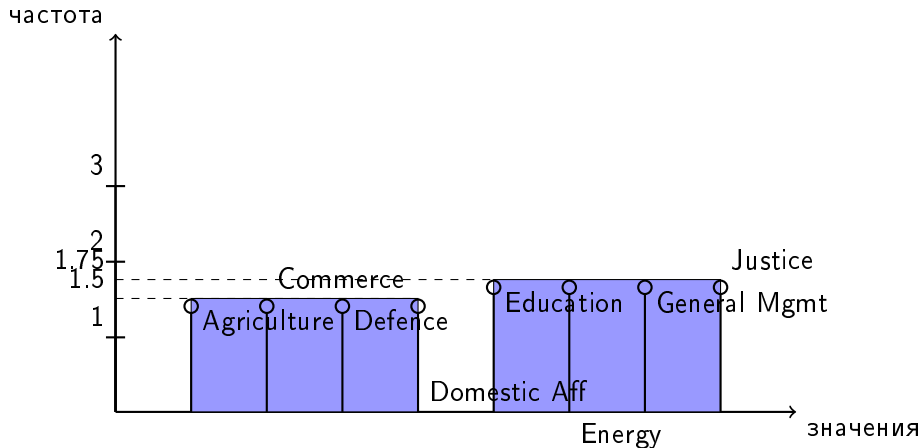
Данные, визуально I



Гистограммы по данным, визуально I



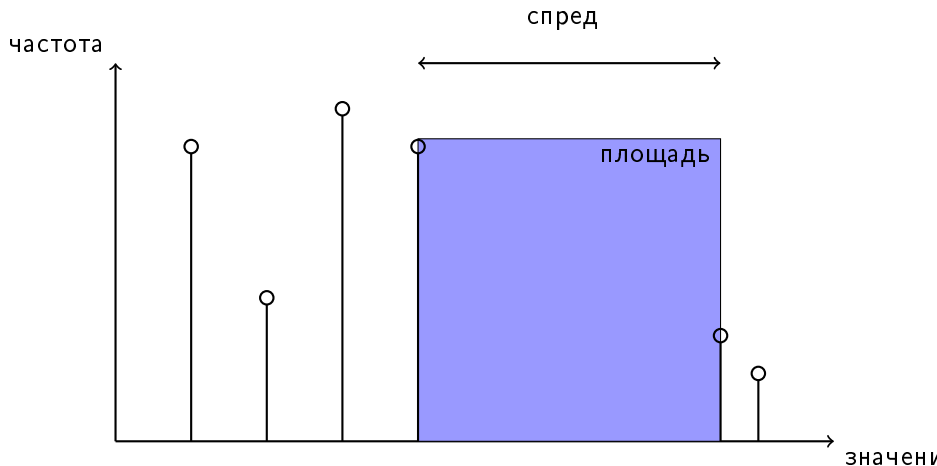
Гистограммы по данным, визуально II



Базовые определения формально

- Отношение R имеет n атрибутов, обозначаемых $X_i, i \in (1, n)$;
- Множество значений V_i атрибута X_i — значения присутствующие в R ;
- Пусть $V_i = \{v_i(k) : 1 \leq k < D_i\}$, где $v_i(k) < v_i(j)$ когда $k < j$, тогда:
 - **спред** $s_i(k)$ для $v_i(k)$ определяется как $s_i(k) = v_i(k+1) - v_i(k)$ для $1 \leq k < D_i$, при этом положим $s_i(D_i) = 1$;
 - **частота** $f_i(k)$ для $v_i(k)$ определяется как количество записей в R , у которых $X_i = v_i(k)$
 - **площадь** $a_i(k)$ определяется как $a_i(k) = f_i(k) * s_i(k)$
- Распределение данных X_i это множество пар $T_i = \{((v_i(1), f_i(1)), (v_i(2), f_i(2)), \dots, (v_i(D_i), f_i(D_i)))\}$
- Объединенная частота $f(k_1, \dots, k_n)$ комбинации значений $\langle v_1(k_1), \dots, v_n(k_n) \rangle$ это число записей в R которые содержат $v_i(k_i)$ в атрибуте X_i , по всем i .
- Объединенное распределение $T_{1, \dots, n}$ для X_1, \dots, X_n это всё множество пар (комбинация значений, объединенная частота).

Базовые определения, визуально



System R:

- хранила минимум и максимум по каждому атрибуту;
- использовала предположение о равномерном распределении.

Тоже “гистограмма” :)

Оценки неточны.

Появление гистограмм в СУБД (equi-width)

Первое появление гистограмм:

- Диссертация Kooi [Kooi, 1980];
- Суть: множество значений разделенное на диапазоны одинаковой длины (equi-width гистограммы);
- Внутри ведра значения и частоты аппроксимируются исходя из: непрерывности значений + равномерного распределения частот;
- Встроил в СУБД Ingres, позже подхвачены и другими СУБД;

Непрерывность значений [Poosala et al., 1996]: предполагаем что все значения из V_i есть в указанном интервале.

Пример: просто гистограммы

For example, given an attribute with the values {1, 1, 2, 2, 2, 3, 6, 6, 6, 6, 6}, two of the many possibilities for representing the

R	C
0	.0
1	.2
2	.3
3	.1
5	.0
6	.4

(a)

R	C
0	.0
2	.5
3	.1
5	.0
6	.4

(b)

Figure 6-1. Possible histograms for the set {1, 1, 2, 2, 2, 3, 6, 6, 6, 6, 6}.

³ Изображение взято из [Kooi, 1980]

Пример (6.1a), визуализация

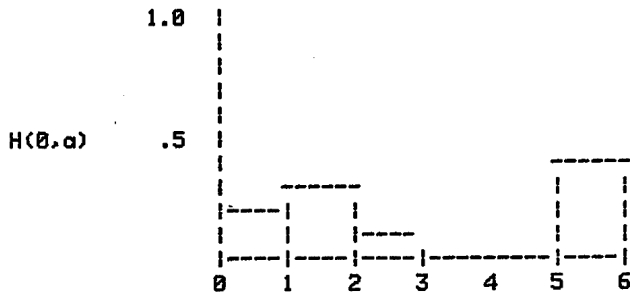


Figure 6-2. Graphic representation of histogram of Figure 6-1(a).

4

Равноширинная, если представить что есть ведра 3-4 и 4-5.

⁴ Изображение взято из [Kooi, 1980]

Свойства equi-width гистограмм

- Позволяют отвечать на запросы диапазона, $x < 100$;
(простой проход по вёдрам)
- Плюсы:
 - сохраняют порядок;
 - дешевы в хранении;
 - понятно как реализовывать;
- Лучше чем подход System R;
- Минусы получаются из предположения о равномерности значений в ведре:
 - большой разброс;
 - не оценить ошибку;

Пример

данные: (1, 2000), (2, 300), (3, 100), (4, 1), (5, 1)... (10, 1)

гистограмма: ([1-2], 1150), ([3-4], 50), ([5-6], 1), ...

А подобных распределений много и они часто встречаются в реальных данных: закон ципфа, нормальное, ...

Первая альтернатива: equi-depth [Kooi, 1980] [Piatetsky-Shapiro and Connel, 1984]

- Выравниваем не границы ведер, а количество записей в каждом;
- Как пришли [Piatetsky-Shapiro and Connel, 1984]: надо ограничивать высоту ведра на графике (частоту);
- Тоже страдают от сложных распределений;
- Занимают столько же места сколько equi-width, но сложно обновлять;
- Тем не менее, было показано что у них лучше оценка ошибки в среднем и худшем случаях [Ioannidis and Poosala(a), 1995];
→ индустрия стала использовать их [Ioannidis, 2003].

Practically, we compute distribution steps by the following procedure:

1. Collect the values of A from all the tuples in a relation and sort them in ascending order according to the *intrinsic* ordering of the domain. We note that this ordering must exist and be unique, for otherwise comparison $A < X$ is not meaningful.
2. Select, depending on the desired accuracy and available storage, the number of distribution steps S . Select $S+1$ positions (including the first and the last) in the sorted list of attribute values, such that there is the same number of attribute values between any⁴ two successive positions. These positions are $1, 1+N, 1+2N, \dots, 1+(S-1)*N, 1+S*N=T$, where $N = (T-1)/S$.
3. Take values found in these positions in the sorted list of all values and let them be the distribution steps $STEP(0), STEP(1), \dots, STEP(S)$.

5

⁵ Изображение взято из [Piatetsky-Shapiro and Connel, 1984]

Иллюстрация работы алгоритма

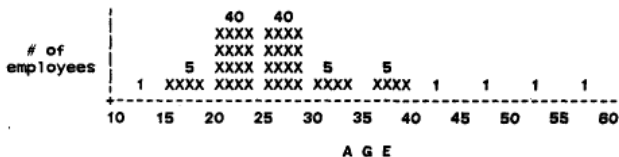


Figure 4-1: Scheme 1: Histogram

6

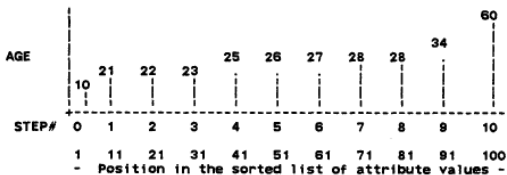


Figure 4-2: Scheme 2: Distribution Steps

7

⁶ Изображение взято из [Piatetsky-Shapiro and Connel, 1984]

⁷ Изображение взято из [Piatetsky-Shapiro and Connel, 1984]

Как выполнять запросы?

To estimate the same selectivity - $SEL(<29)$ - we first find where "29" falls relative to distribution steps. Since

$$STEP(8) = 28$$

we know that more than 80 employees are 28 or younger, so $SEL(<29) > 0.80$. Since

$$STEP(9) = 34$$

we know that 90 or fewer employees are younger than 34, so $SEL(<29) \leq 0.90$. Therefore

$$0.80 < SEL(<29) \leq 0.90$$

Again choosing the midpoint of the range (0.85) as our estimate of $SEL(<29)$, the maximum possible error is 0.05, 4 times less than in scheme 1.

8

⁸ Изображение взято из [Piatetsky-Shapiro and Connel, 1984]

Оффтоп: поразительный факт про сэмплинг

Computing the distribution steps even for *one* attribute of a large relation is expensive, since it requires reading and sorting all the values for that attribute. Computing the distribution steps for all attributes in all relations (a typical example may be 50 relations with 100,000 tuples each with 100 attributes each) would take many days. This might be acceptable, if the database is read-only or if it changes very slowly over time. However there are cases when the database changes rapidly, and there is no time to recompute all statistics in full.

A solution to this dilemma is to compute distribution steps using a *sample* of the tuples, rather than all of them. We lose guaranteed precision but we are still able to maintain very good accuracy. We rely on the theory of nonparametric statistics (see, e.g., [Dixon 79], chapter 17) for analysis of estimation errors.

Kolmogorov's statistic tells us that if we take a sample of size 1064 tuples and β is the fraction of the tuples in the sample with $attr < V$, then with confidence 99% the fraction of the tuples in the *entire* relation with $attr < V$ is in the interval $[\beta-0.05, \beta+0.05]$. If we take a sample of size 740 then the confidence would decrease to 95% with the same interval. The full relationship between the sample size, the confidence level and the interval is given by the table of Kolmogorov's statistic ([Dixon 79], p. 550). Interestingly, the sample size does not depend on the number of tuples in the entire relation.

9

⁹ Изображение взято из [Piatetsky-Shapiro and Connel, 1984]

Общая схема описания гистограмм

<Ограничение на фрагментирование> (<Параметр Сортировки>, <Параметр Источника>):

- Класс фрагментирования (некоторые ограничения на ведра). Пример: серийный класс (serial, не пересекаются по параметру) и подкласс серийных — смещенных на концах (end-biased);
- **Ограничение на фрагментирование**: ограничение на **параметр источника**, уникально идентифицирующий гистограмму в классе фрагментирований. Пример: equi-sum, v-optimal, maxdiff, и compressed.
- **Параметр сортировки**: некоторый параметр, значения которого для каждого элемента в распределении получаются из соответствующих значений атрибутов и частот. Пример: значение атрибута (V), частота (F), площадь (A);
- **Параметр источника**: некоторый параметр, отражающий наиболее важное (с точки зрения задачи оценки) свойство распределения данных. Вместе с ограничением фрагментирования однозначно определяет фрагментирование. Пример: спред (S), частота (F), площадь (A).

Примеры: Equi-sum(V,S), Equi-sum(V,F), V-optimal (F,F) ...

<Ограничение на фрагментирование> (<Параметр Сортировки>, <Параметр Источника>)

- Равноширинные: серийный класс, $\text{equi-sum}(V, S)$, ограничение на фрагментирование = значения исходного параметра (спреды) у каждого ведра одинаковы.
- Равноглубинные: серийный класс, $\text{equi-sum}(V, F)$, ограничение на фрагментирование = количество элементов в ведре одинаково.

<i>SORT PARAMETER</i>	<i>SOURCE PARAMETER</i>		
	SPREAD (S)	FREQUENCY (F)	CUM. FREQ(C)
VALUE(V)	EQUI-SUM	EQUI-SUM	SPLINE-BASED
FREQUENCY(F)		V-OPTIMAL	

Figure 2: Histogram Taxonomy.

10

Серийные гистограммы

Department	Histogram H1		Histogram H2		Histogram H3	
	Frequency in Bucket	Approximate Frequency	Frequency in Bucket	Approximate Frequency	Frequency in Bucket	Approximate Frequency
Agriculture	1	1.50	1	1.33	1	1.43
Commerce	1	1.50	1	1.33	1	1.43
Defense	2	1.50	2	1.33	2	1.43
Domestic Affairs	2	1.50	2	2.50	2	1.43
Education	1	1.75	1	1.33	1	1.43
Energy	3	1.75	3	2.50	3	3.00
General Management	2	1.75	2	1.33	2	1.43
Justice	1	1.75	1	1.33	1	1.43

Table 3: Three types of histograms for the Department attribute

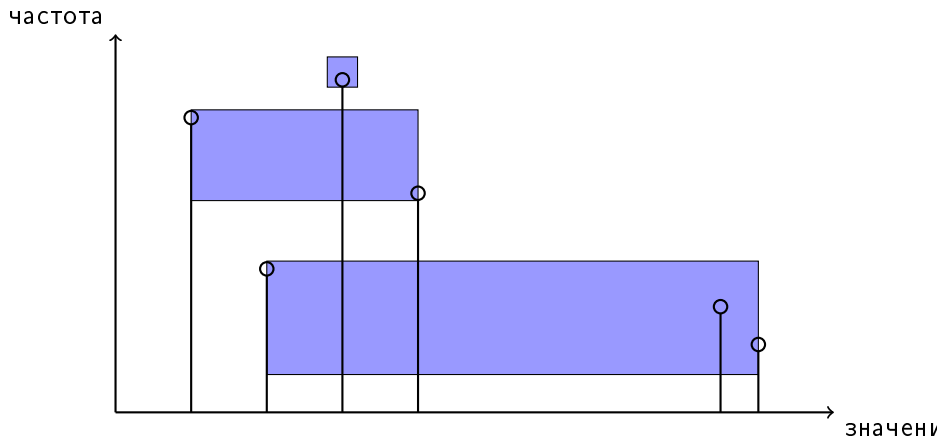
11

Серийная гистограмма: параметры (частоты) ассоциированные с каждым ведром больше или меньше параметров (частот) любого другого ведра. Т. е. ведра серийной гистограммы хранят вместе параметры (частоты) близкие друг к другу и не допускают пересечение.

H1 – нет; H2, H3 – да.

¹¹Изображение взято из [Ioannidis and Poosala(a), 1995]

Серийный класс, визуально



Класс, серийный по частоте.

Какие есть серийные?

- $\text{equi-sum}(V, S) = \text{equi-width}$; $\text{equi-sum}(V, F) = \text{equi-depth}$;
- v -optimal;
- spline-based;

Свойства:

- Доказано, что оптимальны для минимизации ошибок для определенных запросов;
- Дорого хранить, для оптимальности при выборках и соединениях требуют хранения в ведре всего списка значений;
- Часто, дополнительно приходится пользоваться индексом [Ioannidis and Poosala(a), 1995].

нет порядковой корреляции между значениями и частотами \rightarrow а частоту каждого атрибута считать надо! \rightarrow нужны многомерные индексы при обобщении на несколько атрибутов.

V-Optimal (F, F)¹²

- Группируют частоты, минимизируют дисперсию в ведре;
- Минимизируют взвешенную дисперсию:

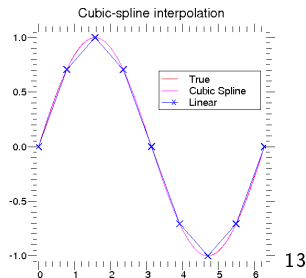
$$\sum_{j=1}^{\beta} n_j * V_j,$$

где n_j количество записей в j ведре, V_j дисперсия частот в j ведре;

- Канонический алгоритм построения требует полного перебора, поэтому, пользуются эвристиками;
- Оптимальны для дерева с соединением по "=", выборках и без функций.

¹²Наглядный пример построения:

Spline-Based (V, C)



- Кусочно-линейная **аппроксимация** T_{C+} ;
- Лучше **аппроксимация** → меньше ошибка;
- Задача оптимальной расстановки узлов (optimal knot placement problem) — используют эвристический алгоритм.

V-Optimal-End-Biased(F,F)

- Класс end-biased это подкласс серийного;
 - Идея: некоторые самые большие и некоторые самые маленькие частоты хранятся в отдельных ведрах. Остальные хранятся в одном ведре, аппроксимируются.
 - Дешевы в построении: полный перебор за почти линейное время;
 - Дешевы в хранении, не нужен индекс. В ведрах-синглтонах храним и значения;
 - Подробно разобраны здесь: [Ioannidis and Poosala(b), 1995].
- популярны в индустрии.

Что лучше?

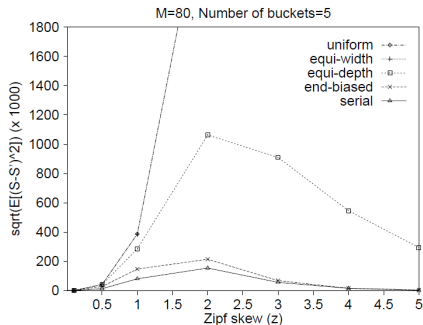
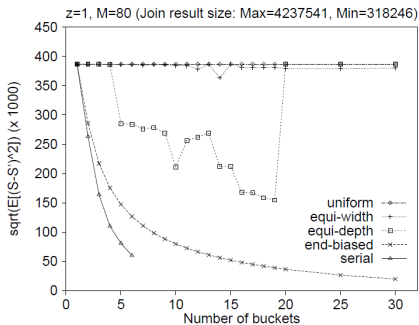


Figure 1: Error as a function of the number of buckets. Figure 2: Error as a function of skew (z parameter of Zipf).

14

SORT PARAMETER	SOURCE PARAMETER			
	SPREAD (S)	FREQUENCY (F)	AREA (A)	CUM. FREQ (C)
VALUE (V)	EQUI-SUM	EQUI-SUM <div>V-OPTIMAL MAX-DIFF COMPRESSED</div>	<div>V-OPTIMAL MAXDIFF COMPRESSED</div>	SPLINE-BASED <div>V-OPTIMAL</div>
FREQUENCY (F)		V-OPTIMAL <div>MAXDIFF</div>		
AREA (A)			<div>V-OPTIMAL MAXDIFF</div>	

Figure 3: Augmented Histogram Taxonomy.

15

Maxdiff:

- ставим границу между ведрами по двум значениям параметра-источника: если разница между ними относится к $\beta - 1$ самых больших разниц;
- идея: не группировать значения с сильно разными значениями параметра-источника в одно ведро;
- вычисляются эффективно, считаем разницу между ближайшими параметрами.

Compressed:

- Самые большие значения параметра-источника хранятся отдельно в ведрах-синглтонах, остальные equi-sum;
- Хорошая точность при аппроксимации неравномерных распределений и/или спредов.

Histogram	Time Taken (msec)	
	Space = 160b	Space = 400b
Compressed	5.9	9.3
Equi-sum	6.2	10.9
MaxDiff	7.0	12.8
V-optimal-end-biased	7.2	10.9
Spline-Based	20.3	41.7
V-optimal	42.9	67.0
Equi-Depth: by P^2	4992	10524

Table 1: Construction cost for various histograms

16

Что лучше-3?

Histogram	Error (%)
Trivial	60.84
Equi-depth: P^2	17.87
V-optimal(A,A)	15.28
V-optimal(V,C)	14.62
Equi-width	14.01
V-optimal(F,F)	13.40
V-optimal-end-biased(A,A)	12.84
V-optimal-end-biased(F,F)	11.67
Equi-depth:Precise	10.92
Spline-based(V,C)	10.55
Compressed(V,A)	3.76
Compressed(V,F)	3.45
Maxdiff(V,F)	3.26
V-Optimal(V,F)	3.26
Maxdiff(V,A)	0.77
V-Optimal(V,A)	0.77

Table 2: Errors due to histograms

17

Альтернативы гистограммам

- сэмплинг — кажется это то, что сейчас популярно и чему сейчас гистограммный подход проигрывает;
- вейвлеты;
- нишевые методы;
- параметрические методы.

Гистограммы используются на практике: дешевы, могут занимать 200 байт [Ioannidis and Poosala(a), 1995] и при этом быть полезными!

Оценка размера результатов: выборки

- Если на колонке есть ограничение типа UNIQUE, а предикат на равенство то гистограммы не нужны 😊;
- Если ограничения нет, а есть предикат \rightarrow гистограмма;
- Отрицание: $SEL(!p) = 1 - SEL(p)$;
- Если есть два предиката на одну таблицу и
 - они объединены через AND то $SEL(p_1 \cup p_2) = SEL(p_1) * SEL(p_2)$
 - они объединены через OR то
$$SEL(p_1 \cap p_2) = SEL(p_1) + SEL(p_2) - SEL(p_1) * SEL(p_2)$$

Это работает если распределения независимы то есть, атрибуты не скореллированы.

Пример: пусть есть тип одежды и тип материала. Формула не сработает если оценивается селективность запроса “шуба из меха” (шуб не из меха не бывает!): селективность будет равна не произведению, а селективности только шубы.

Оценка размера результатов: соединения

R, S — отношения, $|R|$ — количество записей в R , $|R_A|$ — количество уникальных значений в R . Тогда:

$$joinSize \approx \frac{|R| * |S|}{max(|R_A|, |S_A|)}$$

Здесь A это атрибут соединения в общем случае.

Как к этому пришли? Рассмотрели по-отдельности каждую запись из R и прикинули сколько будет результатов в S . Потом наоборот.

На самом деле эта формула почти бесполезна.

Оценка размера результатов: количество уникальных значений атрибута

Наивно:

- Посчитать влоб: пройти по всем данным, держать в памяти все уникальные записи и количество.
- Отсортировать и пройти.

Недостатки серьезны → на практике применяют HyperLogLog.

Основывается на probabilistic counting: 1.5 килобайта в памяти могут дать 2% ошибки на 10^9 + данных.

Очень хорошая демонстрация работы¹⁸.

Есть и другие методы [Harmouch and Naumann, 2017].

¹⁸<http://content.research.neustar.biz/blog/hll.html>

Что в реальности с оценками? I

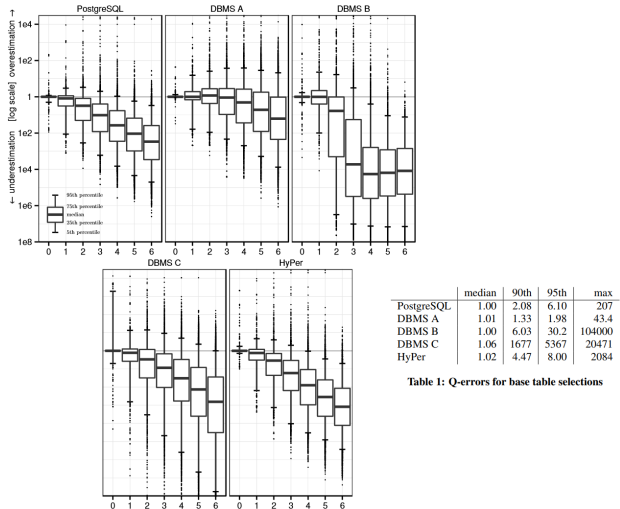


Table 1: Q-errors for base table selections

Figure 3: Quality of cardinality estimates for multi-join queries in comparison with the true cardinalities. Each boxplot summarizes the error distribution of all subexpressions with a particular size (over all queries in the workload)

Что в реальности с оценками? II

Это очень важная статья. Она экспериментально показывает что модель « оценок.

	<0.9	[0.9,1.1)	[1.1,2)	[2,10)	[10,100)	>100
PostgreSQL	1.8%	38%	25%	25%	5.3%	5.3%
DBMS A	2.7%	54%	21%	14%	0.9%	7.1%
DBMS B	0.9%	35%	18%	15%	7.1%	25%
DBMS C	1.8%	38%	35%	13%	7.1%	5.3%
HyPer	2.7%	37%	27%	19%	8.0%	6.2%

Ускорение с истинными оценками

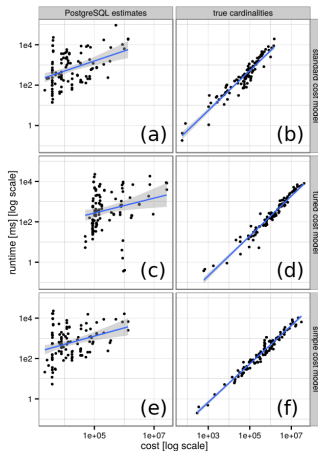


Figure 8: Predicted cost vs. runtime for different cost models

Изображения взяты из статьи [Leis et al, 2015].



Yannis Ioannidis. 2003. The history of histograms (abridged). In Proceedings of the 29th international conference on Very large data bases - Volume 29 (VLDB '03), Johann Christoph Freytag, Peter C. Lockemann, Serge Abiteboul, Michael J. Carey, Patricia G. Selinger, and Andreas Heuer (Eds.), Vol. 29. VLDB Endowment 19–30.



Y. Ioannidis and V. Poosala. Histogram Based Solutions to Diverse Database Estimation Problems, IEEE Data Engineering, Vol. 18, No. 3, pp. 10–18, September 1995.



Viswanath Poosala, Peter J. Haas, Yannis E. Ioannidis, and Eugene J. Shekita. 1996. Improved histograms for selectivity estimation of range predicates. In Proceedings of the 1996 ACM SIGMOD international conference on Management of data (SIGMOD '96), Jennifer Widom (Ed.). ACM, New York, NY, USA, 294–305. DOI=<http://dx.doi.org/10.1145/233269.233342>



Yannis E. Ioannidis and Viswanath Poosala. 1995. Balancing histogram optimality and practicality for query result size estimation. In Proceedings of the 1995 ACM SIGMOD international conference on Management of data (SIGMOD '95), Michael Carey and Donovan Schneider (Eds.). ACM, New York, NY, USA, 233–244. DOI=<http://dx.doi.org/10.1145/223784.223841>



Robert Philip Kooi. The Optimization of Queries in Relational Databases. PhD Thesis, Case Western Reserve University (1980).



Gregory Piatetsky-Shapiro and Charles Connell. 1984. Accurate estimation of the number of tuples satisfying a condition. In Proceedings of the 1984 ACM SIGMOD international conference on Management of data (SIGMOD '84). ACM, New York, NY, USA, 256–276. DOI=<http://dx.doi.org/10.1145/602259.602294>



Harmouch, H., Naumann, F.: Cardinality Estimation: An Experimental Survey. Proceedings of the VLDB Endowment (PVLDB). pp. 499–512 (2017).



Viktor Leis, Andrey Gubichev, Atanas Mirchev, Peter Boncz, Alfons Kemper, and Thomas Neumann. 2015. How good are query optimizers, really? Proc. VLDB Endow. 9, 3 (November 2015), 204–215.
DOI:<https://doi.org/10.14778/2850583.2850594>