DEPARTAMENTO
DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 2

13 de Junio de 2014 — Base de Datos

Bobby Tables

Integrante	LU	Correo electrónico
Mancuso Emiliano	597/07	emiliano.mancuso@gmail.com
Mataloni Alejandro	706/07	amataloni@gmail.com
Gauder María Lara	027/10	marialaraa@gmail.com
Reartes Marisol	422/10	marisol.r5@hotmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2160 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (54 11) 4576-3359
http://www.fcen.uba.ar

Índice

0. Classic Histogram and Distribution Steps

0.1. Introducción

Tanto Classic Histogram como Distrbution Steps son estimadores de tuplas que satisfacen una condicion. Se puede realizar una búsqueda por aquellas tuplas que cumplan una condición de igualdad, por menor o menor igual y por mayor o mayor igual a cierto valor. Los estimadores son utilizados en las bases de datos en el proceso de seleccionar un plan de ejecucion óptimo al momento de realizar una consulta.

Son varios los factores que se involucran en el cálculo de estimadores. Alguno de ellos son la precisión, la cantidad de información que contenga la base de datos, los errores en la estimación, el espacio que consume el estimador y la estructura para consultar la información requerida. Cada uno de estos factores se ajustan al modificar algunas variables que son pasadas por parámetros a los algoritmos de los estimadores.

Las variables en cuestión son:

- La tabla a la que se desea realizar las consultas,
- Una columna
- Una variable denominada PARAM que representan la cantidad en la que se va a agrupar los datos de la base, es decir, la cantidad de steps.

Los steps denotan la cantidad en la que se dividen los conjuntos de datos. Por lo tanto, es claro que al aumentar el valor de steps se aumenta la presición y se disminuye el error en la estimación.

El beneficio que aporta el estimador Distrbution Steps podrá ser observado al compararlo con Classic Histogram y evaluando diferentes factores. Principalmente se tiene en cuenta el costo temporal y espacial de la construcción de las estructuras requeridas por el estimador, el costo temporal de la consulta y el error que comete en la misma.

0.2. Análisis

Para entender el funcionamiento de ambos estimadores, se creó el siguiente caso simple de prueba.

Número	0	1	2	3	4	5	6	7	8	9	10	11	12
Cantidad	1	1	0	1	12	1	0	2	0	1	0	0	1

Cuadro 1: New Table

Esperamos que ‘Distrution steps‘ tenga mejor performance pues, como bien dice el *paper*, viene a solucionar el problema que tenemos con el ‘Classic Histogram‘ manejando la altura de los **bins**.

Tomemos como ejemplo la *selectividad de 5*. En la *figura 1* vemos que la selectividad es de 0,65 pues queda agrupado junto con *4* que es un valor que aparece muchas veces. Sin embargo, los datos reales nos indican que deberia ser *0.05* ya que tiene pocas apariciones.

Como vemos este es un caso **no favorable** para **Classic Histogram** y muestra la ventaja de controlar la altura de los *bins*.

Para el mismo ejemplo, examinamos la *selectividad de 5* nuevamente con el nuevo histograma. Podemos observar que esta vez, la selectividad es de 0,25. Si bien tambien se desvia del valor real (0,05), la diferencia del error es significativa comparada con **Classic Histogram**.

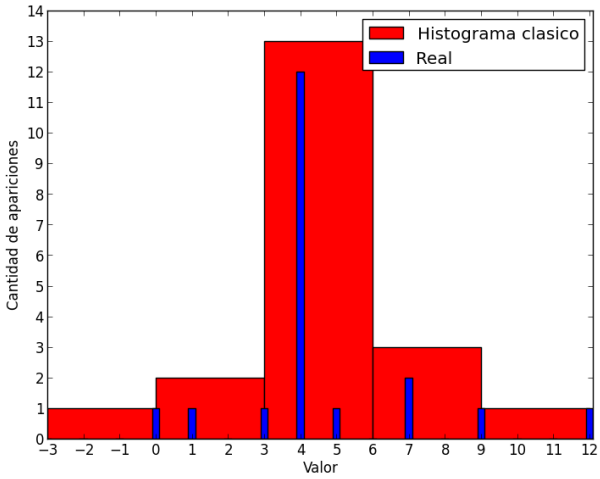

Figura 1: Comparacion Classic vs. Real

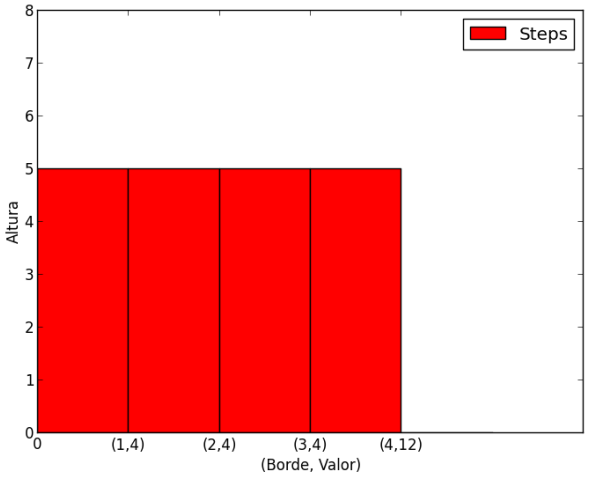

Figura 2: Distribution Steps

Basándose en el ejemplo planteado, se observa que el comportamiento de los distintos estimadores depende de la distribición de los datos. Por ejemplo, cuando los datos están agrupados, **Distribucion Steps** brinda un menor error en la estimación pero, cuando los datos estan dispersos el error del estimador es mucho mayor que el que presenta **Classic Histogram**.

A continuación se experimentará con datos que siguen una distribución uniforme o normal.

[Ale dice q aclaremos q esto se explica mas adelante]

0.2.1. Generación casos de prueba

Los casos de prueba que se van a utilizar para el análisis del comportamiento de ambos estimadores se generarán de manera aleatoria. Se define una función que genera números aleatorios que siguen una distribucion normal o uniforme.

0.2.2. Factores

Debido a la estructura que se utiliza para construir los estimadores y el algoritmo de construcción de los mismos, se podrá afirmar que:

Classic Histogram

- Costo de creación de estimador:

$$O(b * n + n) \tag{1}$$

$$O(b * n) \tag{2}$$

  Siendo n la cantidad total de tuplas existentes en la base de datos. Por el otro lado, b representa el valor pasado en **PARAM**. Como se recorre toda la tabla por cada dato que se requiere para la tabla del estimador, se deberá considerar un costo de $b * n$. Por otro lado, también se requiere almacenar en la estructura del estimador, el máximo, mínimo y la cantidad total de tuplas, lo cual es una búsqueda que realiza el motor de bases de datos en tiempo lineal.

- Costo espacial del estimador:

$$O(2 * b + 3) \tag{3}$$

$$O(2 * b) \tag{4}$$

  Se utiliza un diccionario que presenta un costo espacial del valor pasado en PARAM por dos, ya que por cada bin se almacen dos valores más (la cantidad y el acumulado hasta ese valor). Por otro lado, se requere almacenar el máximo, mínimo y la cantidad total.

- Costo de consulta:

$$O(1) \tag{5}$$

  Lo único se se requere es el acceso al diccionario, lo cual presenta un costo constante.

Distribution Steps

- Costo de creación del estimador:

$$O(n * log(n) + n + n) \tag{6}$$

$$O(n * log(n)) \tag{7}$$

  El algoritmo consiste en el ordenamiento de las tuplas de acuerdo a la columna pasada por parámetro. El ordanamiento lo realiza el motor de la base de datos en costo $O(n * log(n))$. Por otro lado se deberá calcular el máximo, mínimo y total de tuplas existentes en la base.

- Costo espacial del estimador:

$$O(b + 1) \tag{8}$$

$$O(b) \tag{9}$$

  Se requiere unicamente almacenar un array con las tuplas que se encuentran en las distintas posiciones de los steps. Además, se deberá almacenar la cantidad de tuplas totales existentes en la tabla pasada por parámetro.

- Costo de consulta:

$$O(n) \tag{10}$$

  Tanto para una búsqueda por igualdad o por menor, se deberá recorrer el array buscando el valor de tupla deseado.

0.3. Ejercicio 2 - Cambiar nombre

0.3.1. Classic Histogram

Como explicamos anteriormente, el **param** indica la cantidad de bins que generamos.

Suponemos que cuando **param** toma un valor mas grande, disminuye el error en la estimación, pues agrupa menor cantidad de valores por *bin*.

Ademas, suponemos que la distribucion de los datos no influencia en el resultado. Si bien el histograma puede tener menor error con una determinada distribucion, la cantidad de *bins* suponemos que produciria una reduccion de error mas significativa.

Veamos el siguiente gráfico que muestra el error promedio que comete el histograma variando la cantidad de *bins*.

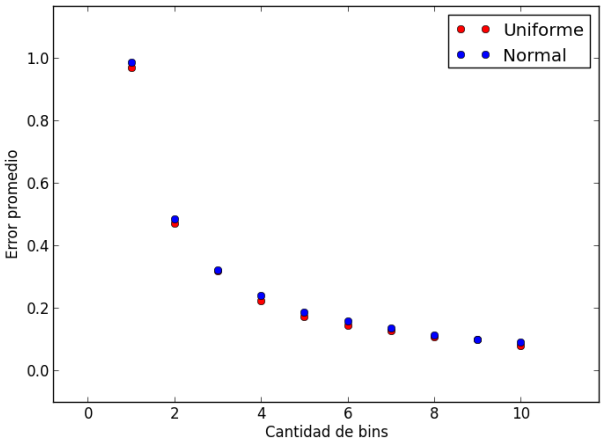

Figura 3: Classic Histogram - Variacion del parametro

Como era de esperar, mejora a medida que agrandamos el **param**. Esto es porque separa mejor los casos y entonces se tiene una estimacion mas precisa. Sin embargo, por cada incremento de *bin*, el costo se incrementa pues se requiere mas espacio para almacenar las estadisticas.

Si hacemos que $param \to n$ (siendo n la cantidad de valores distintos), nos daria la estimación perfecta, pero con un costo altisimo tanto espacial como computacional.

0.3.2. Distribution Steps

Para este histograma, partimos con una suposicion similar, pues la diferencia es que mientras mas *steps* se crean, mas chicos son los *bins* que agrupan valores.

Veamos el siguiente gráfico que muestra el error promedio que comete el histograma variando la cantidad de *bins*.

Nuevamente nuestra hipotesis se confirma. El costo tambien se ve incrementado por la cantidad de *steps*.

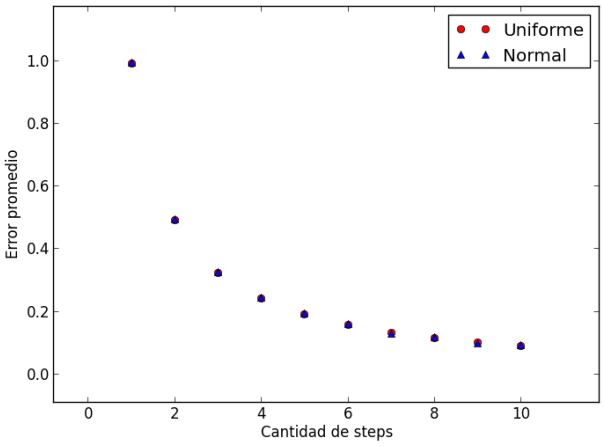

Figura 4: Distribution Steps - Variacion del parametro

0.3.3. Group

para este tenemos bins y la variacion (error)

mejora de clasic histogram

A diferencia de los histogramas anteriores, este cuenta con dos parametros.

- size - cantidad de bins
- threshold - maximo error permitido

Como este Histograma es una modificacion de *Classic Histogram*, obtendriamos la misma conclusion al variar el **size**, el equivalente de **params**.

Por lo tanto, nos queda ver como evoluciona la estimacion cuando modificamos el **threshold**, es decir aumentando y disminuyendo el maximo error permitido.

Esperamos ver, que a medida que se achica el **threshold**, la selectividad es mas precisa.

Veamos el siguiente gráfico que muestra el *maximo error* que comete el histograma variando el **threshold**.

En el caso de la distribucion uniforme, para sorpresa nuestra, el maximo error llega a un techo y no sigue creciendo. No lo supusimos desde el principio, pero cobra sentido si recordamos que los datos provienen de una distribucion uniforme, donde la varianza no es grande. Por eso solo podemos apreciar las grandes mejoras, con un **threshold** muy pequeño.

Para la distribucion normal, el grafico si muestra como mejora la presicion o como se pierde para los distintos valores del **threshold**.

0.3.4. Distribución Uniforme

Partimos con una base de datos cuya información cumple con una distribución uniforme.

Con respecto al error de factor en la estimacion, ambos estimadores no difieren significativamente

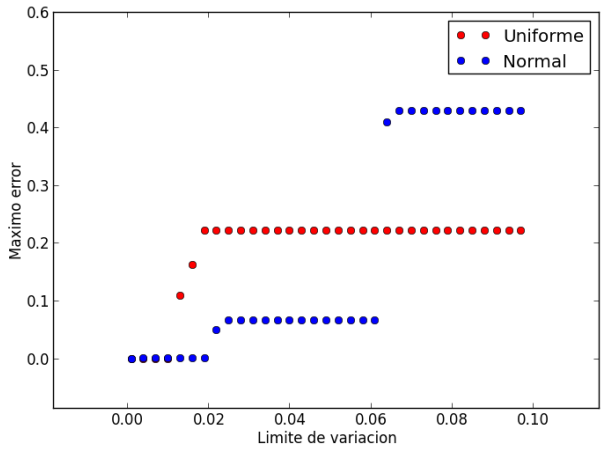

Figura 5: Distribution Group - Variacion del threshold

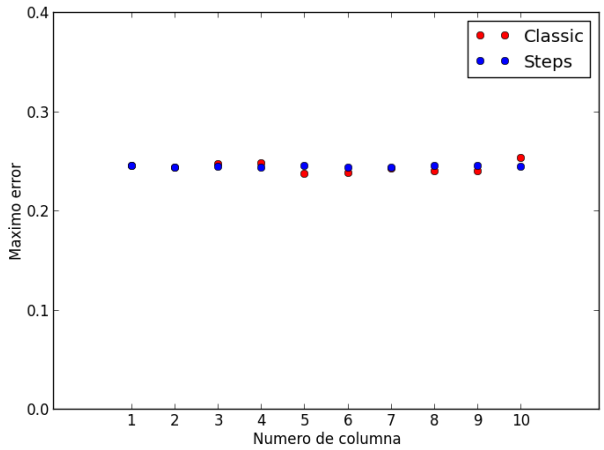

Figura 6: Comparacion Classic vs. Step para distribucion uniforme

en sus resultados. Dado que al tener los datos distribuidos de forma uniforme, la probabilidad de cada tupla es $1/n$, siendo n la cantidad total de tuplas de la tabla.

(Para rango —a-b—/(max - min))

Debido a que se los compara con igual cantidad de steps, entonces la altura, en el caso de **Classic Histogram**, y el ancho, en el **Distribution Steps**, son similares.

Sin embargo, teniendo en cuenta el factor de costos de creación, el estimador **Distribution Steps** es más costoso. El motivo principal es debido al ordenamiento de los datos antes de construir el histograma requerido.

En conclusión, para datos que cumplen una distribución uniforme, sugerimos implementar **Classic Histogram**, debido al menor costo de construcción del histograma.

0.3.5. Distribución Normal

Partimos con una base de datos que la información sigue una distribución normal.

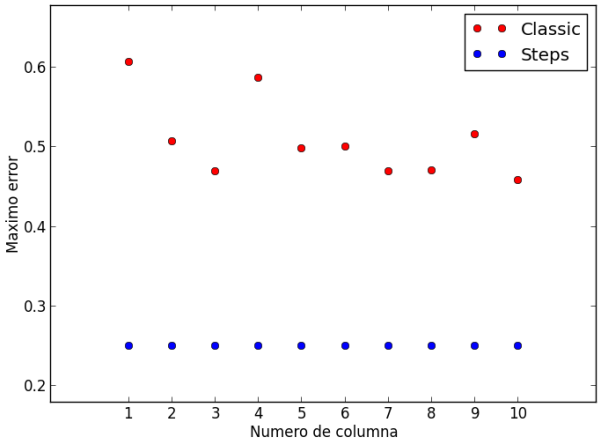

Figura 7: Comparacion Classic vs. Step para distribucion normal