공간 선택률 추정을 위한 압축 히스토그램 기법

정재두°, 지정희, 류근호
충북대학교 데이터베이스 연구실
°chungjaedu@hanmail.net, {jhchi, khryu}@dblab.cbu.ac.kr

A Compressed Histogram Technique for Spatial Selectivity Estimation

Jae Du Chung°, Jeong Hee Chi, Keun Ho Ryu
Database Laboratory of Chungbuk National University

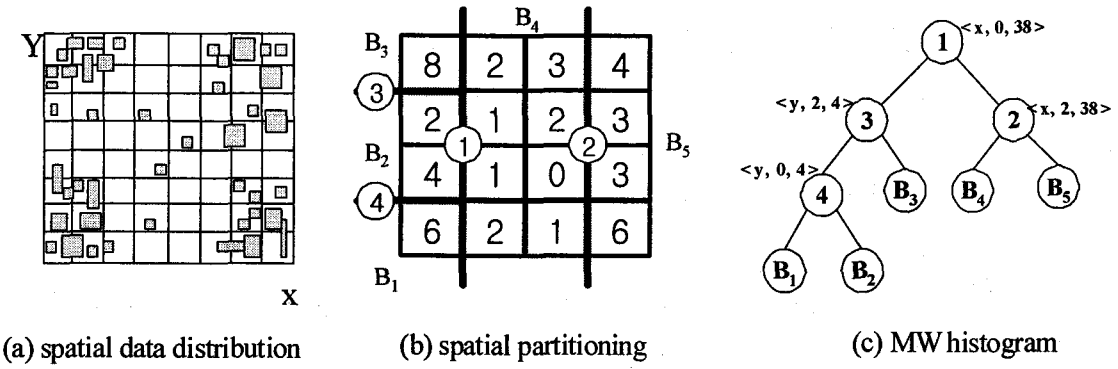
Abstract

Selectivity estimation for spatial query is very important process in finding the most efficient execution plan. Many works have been performed to estimate accurately selectivity. Although they deal with some problems such as false-count, multi-count, they require a large amount of memory to retain accurate selectivity, so they can not get good results in little memory environments such as mobile-based small database. In order to solve this problem, we propose a new technique called MW histogram which is able to compress summary data and get reasonable results. It also has a flexible structure to react dynamic update. The experimental results showed that the MW histogram has lower relative error than MinSkew histogram and gets a good selectivity in little memory.

1. Introduction

Most of database management systems (DBMS) use summary data for efficient processing of a query. The summary data consists of paired component values and frequencies, which implicitly represents the data distribution of the specific component value of records stored in the database [9]. The summary data has been primarily used for selectivity estimation in query optimization and physical database design [10,11]. With the recent dramatic increase in the scale of the database, the significance of the summary data has further intensified. However, major differences between the general histogram and spatial histogram exist. First of all, the original data has different structures. Spatial object has its own domain with each object being different in size. Also, its input values are not very diverse with respect to the domain, unevenly distributed in a specified domain. With the consideration of the unique characteristics of the spatial objects, the summary data can be generated by histogram methods by using spatial partitioning, graphic theory, dimension transformations, and trees. Among these methods, the spatial histogram is the simplest method that preserves the buckets produced by spatial partitioning to be adjusted under any circumstances. It also has an advantage of requiring a small storage space. Various histograms that have thus far been proposed in the field display variety of functions according to bucket partitioning methods and the information kept within the buckets. The existing studies on selectivity estimation methods for spatial

* 본 연구는 대학 IT 연구센터 육성 및 지원사업의 연구결과로 수행되었음



(a) spatial data distribution (b) spatial partitioning (c) MW histogram

Fig.1. Structure of MW histogram

query include [1,2,3,4,5,6,7,8]. However, these methods are capable of producing good selectivity estimation only when sufficient memory space is available. If such methods are used in given small memory capacity, good selectivity cannot be obtained. Also recent advancements in computing and mobile technology make it possible to provide information services on the user's position and geography using small size database, thus increasing the importance, in practical as well as in theoretical aspects, of selectivity estimation method for small database.

Therefore, in this paper we propose a compressed histogram called MW histogram that yields a good selectivity estimation using a small space. This method exploits the special properties of the space, allocating more buckets to place with more spatial distribution, thus ensuring each bucket to have uniform density. The proposed histogram method is designed to maintain summary data using a small space that results from compression of these partitioned buckets.

The rest of this paper is organized as follows. In the next section we summarize related work. The proposed structure and algorithm of MW Histogram is presented in section 3. In section 4 we describe the strengths and weakness of the proposed method through experiments. Finally, we draw conclusions and give a future work in Section 5.

2. Related Work

Selectivity estimation is a well-studied problem for traditional data types such as integers. Histograms are most widely used form for doing selectivity estimation in relational database systems. Many different histograms have been proposed in the literature and some have been deployed in commercial RDBMSs. In case selectivity estimation in spatial databases, some techniques for range queries have been proposed in the literature [2,5,7].

In [2], Acharya et. al. proposed the MinSkew algorithm. The MinSkew algorithm starts with a density histogram of the dataset, which effectively transforms region objects to point data. The density histogram is further split into more buckets until the given bucket count is reached or the sum of the variance in each bucket cannot be reduced by additional splitting. In result, the MinSkew algorithm constructs a spatial histogram to minimize the spatial-skew of spatial objects. The CD (Cumulative Density) Histogram is proposed in [7]. Typically when building a histogram for region objects, an object may be counted multiple times if it spans across several buckets. The CD algorithm address this problem by keeping four sub-histogram stores the number of corresponding corner points that fall in the buckets, so even if a rectangle spans several buckets, it is counted exactly one in a each sub-histogram. The Euler Histogram is proposed in [5]. The mathematical foundation of the Euler Histogram is based on Euler's Formula in graph theory.

As in the CD Histogram, Euler Histogram also addresses the multiple-count problem.

Though these techniques are efficient methods to approximate range query selectivity estimation in spatial databases. These techniques require a large amount of memory for better accuracy.

To compress the summary information in conventional databases, in [1] Matias et al. introduce a new type of histograms, called wavelet-based histograms, based upon multidimensional wavelet decomposition. Wavelet decomposition is performed on the underlying data distribution, and most significant wavelet coefficients are chosen to compose the histogram. In other words, the data points are compressed into a set of numbers via a sophisticated multi-resolution transformation. Those coefficients constitute the final histogram. This approach can be extended very naturally to efficiently compress the joint distribution of multiple attribute.

3. MW Histogram

The construction procedure for MW histogram consists of the following three stages.

●S patial Partitioning Stage

Begin partitioning the entire space |Dx|*|Dy| in a low resolution, which is only 50% of the basic resolution r. If the required number of buckets are *b*, then *b-1* partitioning nodes $snode_i(0=i=b-1)$ generated by the MinSkew partitioning algorithm from the structure of binary split tree. Fig.1(b) shows the spatial partitioning of the Fig.1(a) in which r=8, and b=5. From these processes, the partitioning nodes with binary split tree pattern as shown in Fig.1(c) are generated. This process is explained in algorithm 1 and each partitioning node consists of the following structure :

$snode_i$ = <*axis, split position, spatial-skew, right split node, left split node*>

-. axis : an axis to partition the entire buckets,
-. split position : the split position on the axis for partitioning the entire buckets,
-. spatial-skew : the spatial skewness of the entire buckets,
-. right or left split node : the right or left split node of the subsequent partitioning.

Algorithm 1. Spatial Partitioning

Input : spatial data distribution *ST*, resolution *r*, required the number of buckets *b*
Output : binary split tree *BST*
$snode_i$: i^{th} split nodes,
B_i : i^{th} bucket, $B = \{B_0, B_1, B_2 \cdots, B_{b-1}\}$

Construct grid consisting of *0.5 r* X *0.5 r* uniform rectangles and then compute the spatial density or spatial frequency of grid cells c_{ij}.
Start with a single bucket B_0 consisting of all the regions.
While the size of $B < b$
For Each B_j in B Do
Compute a marginal frequency and marginal skew along each axis of B_j.
IF a marginal skew along any axis *split_axis* in B_k has the biggest marginal skew of all buckets in B Then
store *k* , *split_axis* and *spatial* of the bucket into temporal storage space
End IF
End For
For Each index *idx* along *split_axis* of B_k Do
Compute spatial skew of two areas split by *idx* position along *split_axis* and sum of two spatial skews.
IF the sum by *split_idx* is the most minimum sum of it by all *idx* in B_k Then
store *split_idx* into temporal storage space
End IF
End For
IF B_{t1}, B_{t2} are buckets split by *split_idx* position along *split_axis* in B_k Then
replace B_k with B_{t1} and add B_{t2} to B
End IF
IF *p_snode* and *p_direction* (*right* or *left*) are the parents node and direction pointing at B_k Then
IF *p_snode* is *NULL* Then
the split node is inserted into *BST* as root node
Else
create a split node which has *split_axis*, *split_idx*, *spatial_skew* as parameters and then the split node is inserted into *BST* as a *p_direction* child node of *p_snode*.

```
End IF
End IF
End While
```

- Wavelet Transformation Stage

Produce one or two buckets on the terminal node of the binary split tree. Generate wavelet summary data $W_A(0=i=b-1)$ after applying one dimensional Haar wavelet to the domain of each generated bucket B_i, i.e, B_j transforms into W_{Ai}.

Algorithm 2 is about algorithm for wavelet transformation of single bucket, and Algorithm 3 is an algorithm about wavelet decomposition.

Algorithm 2. Bucket Wavelet Transformation

```
Procedure Transformation (bucket Bi , split node pnodej, direction)

Each grid cell in Bi is divided into four regions so that original resolution r becomes and then each grid cell has spatial frequency calculated by a TF function.

A[j,k] ∈ range ri of bucket Bi.
N is Min (N / log2N> the number of grid cells in Bi).
level = log2.N -1 // Level is the height of a error tree.

Create a array WAi whose size is 2^N and S whose size is N and then two array initialize zero.

While exits next index m1,n1 and
following index m2,n2 ordered by H-mirror curve Do
// index range of average is from 0 to N/2-1
  S = (A[m1,n2]+ A[m2,n2])/2;
// index range of coefficients is from 2^level to N-1
  WAi = (A[m1,n1] - A[m2,n2])/2;
End While

Decomposition(S,WAi, level-1, N/2);
// replace bucket Bi with wavelet synopsis WAi.
Delete Bi ;
pnodej.direction_pointer = WAi
End procedure
```

Algorithm 3. Haar Wavelet Decomposition

```
Procedure Decomposition(Array A, Array WAi, level , N)
Create a array S whose size is N/2
For Each a1,a2 in A Do
// index range of average is from 0 to N/2-1
  S=(A[a1]+ A[a2])/2;
// index range of coefficients is from 2^level to 2^(level+1)-1
  WAi = (A[a1] - A[a2])/2;
End For
If N is 2 Then
  WAi[0] = S[0];
  Exit;
End If
Decomposition(S, WAi , level-1, N/2);
End procedure
```

- Coefficient Shrinking Stage

Reduce the number of coefficients to be kept in each wavelet summary data W_{Ai} until the limited storage space is completely filled.

$$B = \{spatial\text{-}skew,\ Wavelet\ Synopsis \in \{coefficient\ index,\ coefficient\}\}$$

Where, *Wavelet Synopsis* is a set of preserved wavelet coefficient and index.

If the required number of buckets is b, preserved average number of the wavelet coefficient in each bucket is Ws, and the factors of each structure are all same sized, then the size of the total storage space M is computed as follows:

$$M = 5(b-1) + b(1+2Ws) \quad (1)$$

4. Experiment and performance evaluation

In this section, we evaluate the accuracy with which the designed method estimates using actual data, alternating various factors. The experiments were conducted using Intel Pentium III 1.0GHz PC of the Window XP operating system and 384M main memory. The 11,000 building data in JoongGu, Seoul, Korea, was used as the experimental data. The spatial domain of the experimental data set is {(196500, 449 500),(202300, 452

100)}, and we set one lattice to have big territory, establishing 64 as the resolution. Also we changed storage space to 60~720 units as an randomly experimental parameter, and the size of the query to 5%~20% of the total space. Finally, we produced MW histogram that generates the required number of buckets and the size of the storage space.

MW0~MW2 is a histogram that makes preserved number of the buckets to be {0.3, 0.5, 0.7} * M/6. This evaluates the relationship between the number of buckets and the number of wavelets. We took the average value of 10 queries with equal size, and compared with the estimated result. Relative error(e_r), defined as follows, was used to estimate the accuracy of the estimation.

$$\cdot\ e_r = \frac{|d-\hat{d}|}{d} \times 100\% \qquad (2)$$

Where, d is actual size of the result, $\hat{d}$ is estimated size of the result

4.1 Evaluation of Relative Error

We obtain credible selectivity while preserving spatial distribution even in the small storage space. In order to experiment the validity of the proposed method, we compared MinSkew histogram with 360 storage space with MW histogram, MW0~MW2 with 180 storage space. As Fig.2 shows, MW0~MW2 demonstrates relatively lower or similar error than MinSkew histogram despite the storage space being half the size.

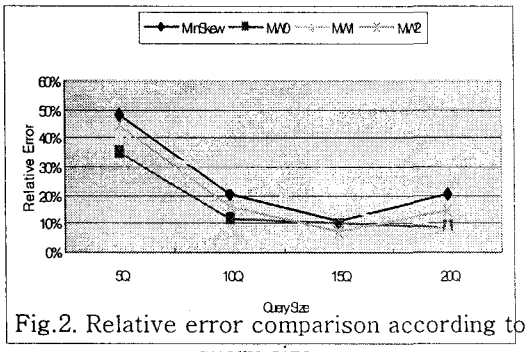


Fig.2. Relative error comparison according to query size

Also, we examined the change in the relative error of MW0~MW2 to the query in that picture, we can see that the size of the relative error to the large size of the query is similar in MW0through the experimental results. Also, in case of MW1, the larger the query, the lesser the relative error. On the other hand, MW2 has higher error than MW0 and MW1 in 20% of query. The smaller size of the query is included relatively easily in larger domain, and thus the probability of false counting is high, resulting in higher error. But, MW0, in which preserved wavelet coefficient is big, reduces the probability of false counting because it stores efficiently as compressing the distribution information inside each bucket into wavelet summary data. The big size query includes completely one bucket, or most buckets, so that it can get more accurate selectivity. But, MW2 has smaller number of buckets than MinSkew, and its preserved number of wavelet coefficients is small. Thus, the number of buckets included in large query is small, therefore making it difficult to reflect exactly the distribution information within the bucket. As a result, the relative error of the MW2 to the 20% query in Fig.2 is a little higher than other MW histograms, although it has lower error than MinSkew histogram. This result implies that the designed MW histogram can solve the problem of distributional skewness in a relatively large bucket space, even in the case where the spatial domain is big, or the restriction to the storage space is severe.

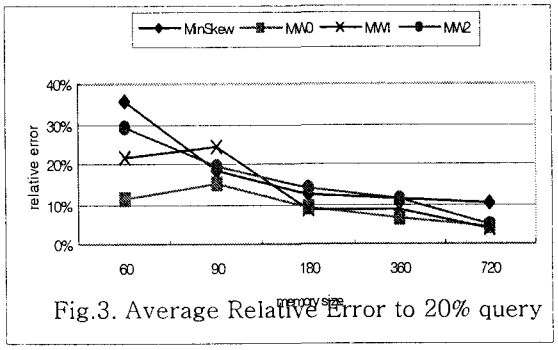


Fig.3. Average Relative Error to 20% query

Fig.3 shows relative error of MW0~MW2 in storage space of 20% query. The relative error curve in 20% query shows a big change as the size of

the storage space increases. Especially, MW histogram in 60 storage space has lower error than the relative error of the MinSkew histogram. MW0, in which number of wavelet coefficients preserved per bucket are big, has lower error than other histogram. That is, the bigger the number of preserved wavelet coefficients, the better selectivity we can get even with small storage space. In the case of the experimental data used as a result of this experimental evaluation, the number of MW0 of 0.3*M/6 is proved as the most efficient.

5. Conclusions

Selectivity estimation is used in query optimization and decision of optimal access path cardinally. Until now, several techniques of spatial selectivity estimation have been proposed. These techniques are focused on obtaining high accuracy and fast response time. However, they require very large memory space to maintain high accuracy of selectivity if spatial domain is also large. Therefore, we proposed a new method called MW histogram that could get reasonable selectivity with small memory size. MW histogram combined modified spatial split method with Haar Wavelet transformation so that we obtained maximum compression effects consequently. Based on our experimental analysis, we showed that the proposed technique which called MW histogram can obtain maximum compression effects and reasonable selectivity simultaneously. Especially, MW Histogram which the buckets and wavelet coefficients ratio is 0.3 is lower relative error than MinSkew Histogram about 5%~20% queries, demonstrates that MW histogram gets a good selectivity in little memory. The proposed MW histogram is useful in very large spatial domain.

In the future, we need to analyze our histogram to improve much experimental evaluation. We also will extend our histogram to do work easily about dynamic insertion.

References

[1] Yossi Matias, Jeffrey Scott Vitter, Min Wang, "Wavelet-Based Histograms for Selectivity Estimation", In Proc. ACM SIGMOD Int. Conf. on Management of Data, 1998, pp.448-459.

[2] Swarup Acharya, Viswanath Poosala, Sridhar Ramaswamy, "Selectivity estimation in spatial databases", In Proc. ACM SIGMOD Int. Conf. on Management of Data, 1999, pp.13-24.

[3] L. Getoor, B. Taskar, D. Koller, "Selectivity estimation using probabilistic models", In Proc. ACM SIGMOD Int. Conf. on Management of Data, 2001, pp.461-473

[4] C. Sun, D. Agrawal, A. El Abbadi, "Selectivity for spatial joins with geometric selections", Proc. of EDBT, 2002, pp.609-626

[5] Sun, C., Agrawal, D., El Abbadi, A., "Exploring spatial datasets with histograms (full version)", Technical Report, Computer Science Department, University of California, santa Barbara, 2001

[6] Minos G., Phillip B.G., "Wavelet Synopses with Error Guarantees", ACM SIGMOD 2002, Jine 4-5, Madison, Wisconsin, USA, pp.476-487.

[7] Jin, N. An, A. Sivasubramaniam, "Analyzing Range Queries on Spatial Data", In Proceedings of the IEEE International Conference on Data Engineering, 2000, pp. 525-534

[8] Ning An, Zhen-Yu Yang, Sivasubramaniam A., "Selectivity estimation for spatial joins", In Proceedings of the IEEE International Conference on Data Engineering, 2001, pp.175-196

[9] Barbara, D. et al., " The New Jersey Data Reduction Report", IEEE Data Engineering Bulletin 1997, Vol.20, No.4, pp.3-45

[10] Selinger, P. et al., "Access Path Selection in a Relational Databases Management System", ACM SIGMOD, 1979, pp.232-244.

[11] Whang, K., Kim, s, and Wiederhold, G., "Dynamic Maintenance of Data Distribution for Selectivity Estimation", the VLDB Journal 1994, Vol.3, No.1, pp.29-51