Short Paper: Euler++: An Improved Selectivity Estimation for Rectangular Spatial Records

A. B. Siddique
University of California, Riverside
msidd005@ucr.edu

Ahmed Eldawy
University of California, Riverside
eldawy@ucr.edu

Vagelis Hristidis
University of California, Riverside
vagelis@cs.ucr.edu

***Abstract*—Selectivity estimation is one of the common research problems for big spatial data, where the objective is to quickly estimate the number of records in a given query range. Euler histogram has been used to answer the selectivity estimation queries for objects with extents such as rectangles in constant time. However, it is only accurate when the query range is aligned with the histogram grid lines. In this paper, we improve the Euler histogram to accurately answer arbitrary queries, i.e., even if they do not align with the histogram grid lines. The improved histogram, called *Euler*++, has the same space and time complexity as the regular Euler histogram and provides a better accuracy for objects with extents. We use both real and synthetic datasets for extensive experiments, and show that the proposed technique, Euler++, consistently outperforms the existing ones, while still providing answer in constant time.**

***Index Terms*—Euler++, Spatial data synopsis, big spatial data, selectivity estimation, query optimization**

I. Introduction

Recently, there has been an exponential increase in the amount of big data, of which 60-80% is geo-referenced [1]. Big Spatial data is supporting many important applications, such as brain simulation [2], event detection [3], climate studies [4], and others. One of the most important research problems for big spatial data is *selectivity estimation*, which tries to estimate the number of records in a given query range. It has been employed in many applications including load balancing [5]–[7], indexing [8], [9], and query optimization [10].

Selectivity estimation for objects with extents, e.g., rectangles, is particularly an interesting research problem, as most of the real-world objects are not just points, but have extended boundaries. Euler histogram (EH) has been used to generate an efficient synopsis (i.e., summary of the data) for such objects, which can be used to answer selectivity queries in constant time. For example, AQWA [5] proposes a technique which estimates the number of the records which intersect a given spatial range query but it is only accurate when the query boundary is perfectly aligned to the histogram cells.

This paper builds on Euler histogram to provide a more accurate estimate for the spatial query by carefully considering partially covered histogram cells, and excluding the unwanted regions. The proposed solution modifies the traditional Euler histogram in two ways. First, during the offline histogram construction phase, we collects two additional aggregate numbers, average width and height for all the records in the input. Second, in the online query answering phase, we modify the equations of the Euler histogram to take into account partially overlapping cells with the help of the average record width and height. Similar to the original Euler histogram, the proposed formulas can be computed in constant time regardless of the size of the query range, and provide a much higher accuracy.

We run extensive experiments on real as well as synthetic datasets for a wide range of selectivity ratios, and compare the accuracy and query response time against well-known selectivity estimation techniques that use samples and histograms [5], [11]. The experimental results show that the proposed technique, Euler++, outperforms the existing ones, especially, when the input records have large extents.

The rest of the paper is organized as follow: Section II discusses the offline histogram construction phase, whereas the online selectivity estimation phase is described in Section III. The experimental evaluation is presented in Section IV, and Section V concludes the paper.

II. Histogram Construction

The proposed histogram is constructed in an offline phase that runs in two steps. The first step counts the number of records in each grid cell. The second step computes a two-dimensional prefix sum that allows the selectivity estimation query in constant time.

A. Euler Histogram

a) Background on Euler Histogram: Traditional synopsis techniques, such as sampling, and regular histograms convert objects with extents, e.g., rectangles, to a point by computing its centroid. Therefore, they cannot accurately answer the selectivity estimation query. Since rectangles can overlap multiple cells, counting them in one cell only or counting them in all overlapping cells results in underestimated or overestimated numbers, respectively. Euler histogram (Figure 1) keeps four counters, called C_1, C_2, C_3, and C_4 for each cell. C_1 represents the number of records in the dataset that partially or fully overlaps a cell. C_2 is the number of records which have their left edge intersecting with the cell. C_3 maintains the number of records which have their top edge intersecting with the cell. C_4 keeps track of the number of records which have their top-left corner inside the cell. Figure 2 presents how each record updates the Euler histogram, and Figure 1b shows the computed Euler histogram for the input dataset in Figure 1a.

(a) Input dataset

243 (c1)	243 (c2)	481	463	889 849 423 402
231 (c3)	231 (c4)	464	447	845 806 402 383
35	35	314	310	784 760 176 133
35	35	314	310	784 760 176 133

(b) Euler Histogram Synopsis

Fig. 1: A Spatial dataset and its Euler Histogram Synopsis.

0,0 0,0	1,1 1,1	1,0 1,0	1,0 1,0	1,0 1,0	1,0 1,0	0,0 0,0	0,0 0,0
0,0 0,0	1,1 0,0	1,0 0,0	1,0 0,0	1,0 0,0	1,0 0,0	0,0 0,0	0,0 0,0
0,0 0,0	1,1 0,0	1,0 0,0	1,0 0,0	1,0 0,0	1,0 0,0	0,0 0,0	0,0 0,0
0,0 0,0	0,0 0,0	0,0 0,0	0,0 0,0	0,0 0,0	0,0 0,0	0,0 0,0	0,0 0,0
0,0 0,0	0,0 0,0	0,0 0,0	0,0 0,0	0,0 0,0	0,0 0,0	0,0 0,0	0,0 0,0

(a) EH after first record

0,0 0,0	1,1 1,1	1,0 1,0	1,0 1,0	1,0 1,0	1,0 1,0	0,0 0,0	0,0 0,0
0,0 0,0	1,1 0,0	2,1 1,1	2,0 1,0	2,0 1,0	2,0 1,0	0,0 0,0	0,0 0,0
0,0 0,0	1,1 0,0	2,1 0,1	2,0 0,0	2,0 0,0	2,0 0,0	0,0 0,0	0,0 0,0
0,0 0,0	0,0 0,0	1,1 0,0	1,0 0,0	1,0 0,0	1,0 0,0	0,0 0,0	0,0 0,0
0,0 0,0	0,0 0,0	0,0 0,0	0,0 0,0	0,0 0,0	0,0 0,0	0,0 0,0	0,0 0,0

(b) EH after second record.

Fig. 2: How four counters are updated for EH.

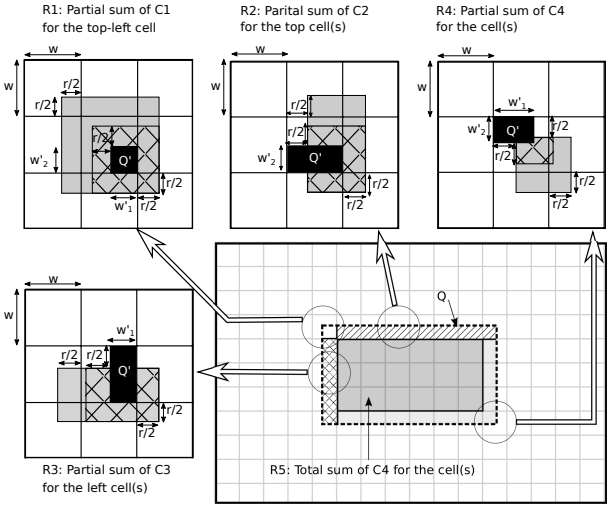


Fig. 3: Selectivity estimation using Euler++.

b) Efficient Construction of Euler Histogram: The input is the big spatial dataset, and number of cells, d, in the histogram. The histogram is represented by four $2d$ arrays. Assuming, square-shaped cells, each cell has an area of $Area(I)/d$ and a side length of $w = \sqrt{Area(I)/d}$ where $Area(I)$ is the area of the MBR of the input data. This results in a grid size of $C = \lfloor Width(I)/w \rfloor$ columns and $R = \lfloor Height(I)/w \rfloor$ rows. To compute the histogram, we scan the whole input in parallel, and for every record, find the top-left and bottom-right overlapping cells, and increment C_1 by 1 for all cell(s) from top-left through bottom-right cell(s), C_2 for left edge cell(s), C_3 for top edge cell(s), and C_4 for only top-left corner cell (Figure 2). Moreover, width and height of each record are also accumulated to compute the averages.

B. Prefix-Sum Technique

The prefix-sum technique [12] has been used to provide selectivity estimations based on the histograms in constant time. It aggregates the horizontal, and then vertical sum over the histogram. To answer a selectivity query, which has the top-left and bottom-right cells at (i_1, j_1) and (i_2, j_2) indices of the histogram respectively, the answer can be evaluated by adding the frequencies of (i_1–1, j_1–1) and (i_2, j_2) and subtracting the (i_2, j_1–1) and (i_1–1, j_2) indices. This technique is extended in [5] to work with the Euler histogram. For C_2 and C_3, horizontal prefix-sums for each row and vertical prefix-sums for each column are maintained respectively. For C_4, the values are aggregated horizontally and then vertically. Then, our proposed technique, Euler++, can be used to provide estimates for spatial range queries.

III. Selectivity Estimation using Euler++

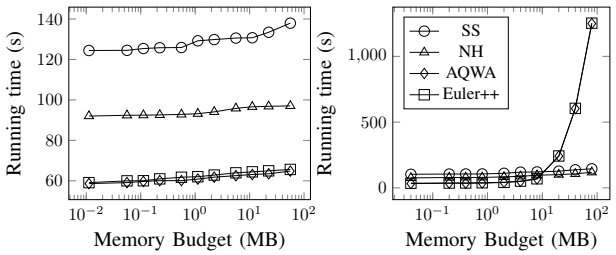
AQWA [5] always expands the query rectangle to align with grid boundaries which might result in an over-estimation. Moreover, taking a fraction of the cell, proportional to the area covered by the query, cannot apply here because the four counters have four different meanings and the data synopsis contains rectangles, not points. Therefore, the next part provides deeper analysis and proposes a novel technique for accurate selectivity estimation on Euler histogram which is shown in the experiments section to be superior to other types of samples/histograms.

To compute the value of partially covered cells accurately, we need to use the values of C_1, C_2, C_3, and C_4 counters partially for partially covered cells, as shown in Figure 3. To simplify our analysis, we assume square cells of side length w and that all data records are also squares of side length r with uniformly distributed centers. The analysis can be easily extended to records of arbitrary rectangular cells and data records. The overlap between the query rectangle Q and a cell is denoted Q'. The width and height of Q' are denoted w'_1 and w'_2, respectively. In the analysis below, please refer to the illustration in Figure 3.

Case R_1 (top-left corner): In this case, the counter C_1 is used which represents all data objects that overlap the cell. This means that the centers of the data records are evenly distributed in a buffer of size $r/2$ around the cell (denoted with a gray square). Out of these data records, we only want to count the records that overlap the portion of the cell that overlaps the query Q' (denoted with a solid black rectangle). The centers of the records that overlap the query rectangle are evenly distributed in a buffer of size $r/2$ around the black rectangle. To compute the estimated number of records that overlap Q', we define two random events, A_1 and B_1. A_1 is the event of a record overlapping Q'. B_1 is the event of a record overlapping the cell. The conditional probability $Pr\{A_1|B_1\}$ is equal to the cross-hatched area divided by the gray area as shown below.

$$Pr(A_1|B_1) = \frac{(w'_1 + r) \cdot (w'_2 + r)}{(w + r)^2} \quad (1)$$

Case R_2 (top edge): In this case, C_2 represents the number of records that have a left edge in the cell. Since the records are all squares with a side-length of r, their centers are evenly distributed in the gray area. Out of these records, we only



(a) edges dataset (b) synthetic dataset

Fig. 4: Synopsis generation time.

want to compute the records that have left edge in Q', the cross-hatched area. We define the random events A_2 and B_2 as the event of a record having the left edge in Q' and the event of a record with a left edge in the cell, respectively. The conditional probability is defined below.

$$Pr(A_2|B_2) = \frac{w_1' \times (w_2' + r)}{w \times (w + r)} \tag{2}$$

Case R_3 (left edge): In this case, the counter C_3 represents the records that have a top edge in the cell. Their centers are evenly distributed in the gray area. Out of these, the records that have top-edge in the Q' have centers in the hatched area. The conditional probability for this case is defined as:

$$Pr(A_3|B_3) = \frac{w_2' \times (w_1' + r)}{w \times (w + r)} \tag{3}$$

Case R_4 (other partial cells): In this case, the counter C_4 represents the records that have a top-left corner in the cell. Their centers are evenly distributed in the gray area. Out of these, the records that have the top-left corner in the Q' have centers in the hatched area. The conditional probability for this case is defined as:

$$Pr(A_4|B_4) = \frac{w_1' \times w_2'}{w^2} \tag{4}$$

Case R_5 (fully contained cells): Since these cells are fully contained, we add up their C_4 counters without any partial computations, i.e., the probability is 1.

The computed conditional probability is multiplied by the corresponding counter $C_1, \ldots, C_4$ to compute the estimated number of records. In practice, records might not consist of equi-sized squares. Therefore, as explained in Section II-A, the synopsis step is extended to compute the average side width and length of all objects. The computation of the averages can be easily piggybacked on the synopsis step using counters or accumulators (available in Hadoop and Spark) which add no overhead. The computed averages are attached to the synopsis to improve the accuracy for partially overlapping queries.

IV. Experimental Evaluation

A. Experimental Setup

Experiments for offline phase are run using 12-node cluster of Apache Spark with 12 cores, 64 GB RAM, and 10 TB disk storage. The selectivity queries are executed on a machine with 16 cores, 128 GB RAM, and 10 TB HDD. While any big data system can be used to implement this technique, we use Apache Spark.

Name	Size	Records	Description
edges	23 GB	70 million	Polygons
all-objects	92 GB	263 million	Mixed
synthetic	51 GB	250 million	Rectangles

TABLE I: Datasets

a) Datasets: We use both real [8] and synthetic datasets (Table I). The MBR of the synthetic dataset is $x_1 = -180, y_1 = -90, x_2 = 180, y_2 = 90$. In this MBR, uniformly random points are generated to be used as the center of the rectangles of width and height of ≈ 2, the rectangles close to the MBR boundaries can have width or height < 2 to keep the centers uniformly distributed and within the MBR.

b) Baselines: We compare our proposed technique, Euler++, against three well-known selectivity estimation techniques based on, stratified sampling (SS), non-uniform histogram (NH), and Euler histogram-based technique (AQWA) [5]. Following the benchmark in [11], we compare these techniques while varying the memory budget, B, which represents the amount of memory allotted for the synopsis, e.g., sample or histogram. We also run selectivity estimation queries for different selectivity ratios, i.e., ratio between the areas of the query and the input dataset.

B. Synopses Performance

For offline phase, Euler++ has comparable or better running time for edges (Figure 4a), and all-objects datasets. Whereas it is significantly slower than SS and NH for the synthetic dataset when $B >$ 10MB, (Figure 4b), as it contains records with relatively large extents, which can overlap many grid cells. The running time of the AQWA and Euler++ is affected by large extent records, when B is also big, which should not be a problem, as the this step is performed *only once* for a dataset, and all the future selectivity queries are answered based on the synopsis only.

C. Selectivity Estimation Performance

To prepare the query workload, we pick $1,000$ random points from the input dataset and use these as query centers. The queries are rectangles with an area of 10^{-4}, 10^{-3}, 10^{-2}, and 10^{-1} of the area of the MBR of the input dataset.

1) Quality Measure: We use average absolute-relative-accuracy as the quality measure, which is on-the-average how close is the estimate to the ground truth for all the queries. We also use full dataset, which always computes exact answer by scanning the whole input dataset in parallel, and filtering all the intersecting records (we consider it as ground truth) for the given query. For a query q, if the ground truth is t_q and the estimated value is e_q, we compute the accuracy of q as $\max\{0, 1 - |t_q - e_q|/t_q\}$. This gives a range of $[0, 1]$ for the accuracy. The average accuracy of all the $1,000$

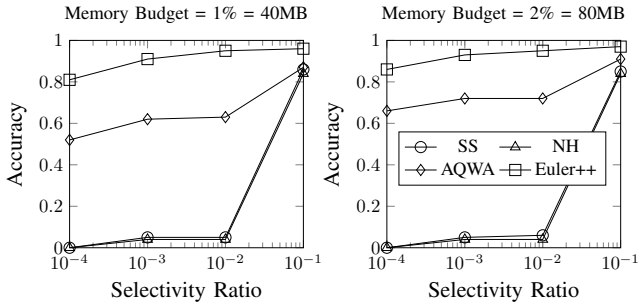
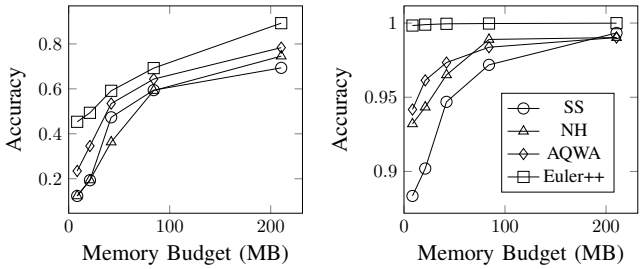
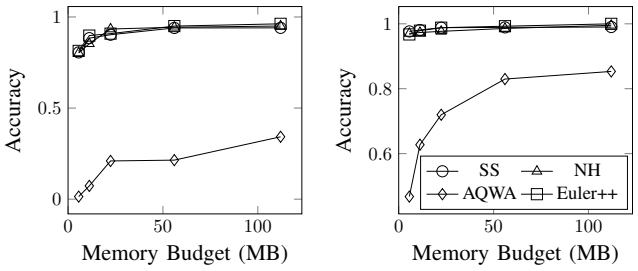


Fig. 5: `synthetic`: Average accuracy for selectivity queries.



(a) Selectivity Ratio = 10^{-4} (b) Selectivity Ratio = 10^{-1}

Fig. 6: `all-objects`: Average accuracy for queries.



(a) Selectivity Ratio = 10^{-3} (b) Selectivity Ratio = 10^{-2}

Fig. 7: `edges`: Average accuracy for selectivity queries.

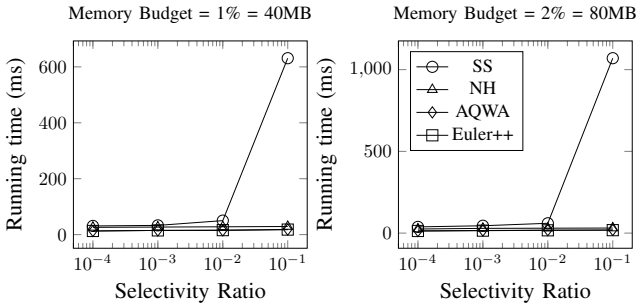


Fig. 8: `synthetic`: Average query response time.

queries for each selectivity ratio is used as the accuracy for the corresponding technique.

Figures 5, 6, and 7 present the accuracy of our proposed selectivity estimation technique, Euler++, compared against other sampling, and histogram-based techniques on the `synthetic`, `all-objects`, and `edges` datasets respectively for different selectivity ratios ranging from 10^{-4} to 10^{-1}, and memory budgets varying from 8MB to 80MB. Euler++ is consistently more accurate or very close to all other techniques. In particular, with the `synthetic` dataset, it is much better than other techniques as it handles the partially covered cell accurately, whereas AQWA [5] over-counts by expanding the query region to the histogram cell boundaries. The Euler++ has superior accuracy for very small to big selectivity query ranges, and for a wide range of memory budgets on real `all-objects` dataset (Figure 6), which has a wide range of objects. Similarly, in the real `edges` dataset, only a small fraction of the records are large, yet Euler++ has consistently better or comparable accuracy.

2) Performance Measure: To measure the performance, we use the average time to answer the estimation query (Figure 8). The histogram-based techniques are clear winners as they can answer any query in *constant time*, ascribed to the prefix-sum technique. Although we can get the exact answer using the full dataset in parallel by filtering the records in the input dataset, yet it takes significantly more time, e.g., 55 seconds to answer a single selectivity query for `synthetic` dataset.

V. Future work

In future we plan to provide error bounds and deep theoretical analysis for this work.

References

[1] C. Dempsey, "Where is the phrase "80% of data is geographic" from? ˜ gis lounge," https://www.gislounge.com/80-percent-data-is-geographic/, (Accessed on 01/10/2018).

[2] H. Markram, "The blue brain project," *Nature Reviews Neuroscience*, vol. 7, no. 2, p. 153, 2006.

[3] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling, "Twitterstand: news in tweets," in *Proceedings of the 17th acm sigspatial international conference on advances in geographic information systems*. ACM, 2009, pp. 42–51.

[4] J. H. Faghmous and V. Kumar, "Spatio-temporal data mining for climate data: Advances, challenges, and opportunities," in *Data mining and knowledge discovery for big data*. Springer, 2014, pp. 83–116.

[5] A. M. Aly, A. R. Mahmood, M. S. Hassan, W. G. Aref, M. Ouzzani, H. Elmeleegy, and T. Qadah, "Aqwa: adaptive query workload aware partitioning of big spatial data," *Proceedings of the VLDB Endowment*, vol. 8, no. 13, pp. 2062–2073, 2015.

[6] W. Lu, Y. Shen, S. Chen, and B. C. Ooi, "Efficient processing of k nearest neighbor joins using mapreduce," *Proceedings of the VLDB Endowment*, vol. 5, no. 10, pp. 1016–1027, 2012.

[7] R. T. Whitman, M. B. Park, S. M. Ambrose, and E. G. Hoel, "Spatial indexing and analytics on hadoop," in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2014, pp. 73–82.

[8] A. Eldawy and M. F. Mokbel, "Spatialhadoop: A mapreduce framework for spatial data," in *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*. IEEE, 2015, pp. 1352–1363.

[9] P. Lu, G. Chen, B. C. Ooi, H. T. Vo, and S. Wu, "Scalagist: scalable generalized search trees for mapreduce systems [innovative systems paper]," *Proceedings of the VLDB Endowment*, vol. 7, no. 14, pp. 1797–1808, 2014.

[10] A. Eldawy, L. Alarabi, and M. F. Mokbel, "Spatial partitioning techniques in spatial hadoop," *PVLDB*, vol. 8, no. 12, pp. 1602–1605, 2015.

[11] A. Siddique, A. Eldawy, and V. Hristidis, "Comparing synopsis techniques for approximate spatial data analysis," *Proceedings of the VLDB Endowment*, vol. 12, no. 11, pp. 1583–1596, 2019.

[12] C.-T. Ho, R. Agrawal, N. Megiddo, and R. Srikant, *Range queries in OLAP data cubes*. ACM, 1997, vol. 26, no. 2.