Selectivity Estimation of Range Queries in Data Streams using Micro-Clustering

Sudhanshu Gupta and Deepak Garg
Computer Science and Engineering Department, Thapar University, India

Abstract: Selectivity *estimation is an important task for query optimization. The common data mining techniques are not applicable on large, fast and continuous data streams as they require one pass processing of data. These requirements make Range Query Estimation (RQE) a challenging task. We propose a technique to perform RQE using micro-clustering. The technique maintains cluster statistics in terms of micro-clusters. These micro-clusters also maintain data distribution information of the cluster values using cosine coefficients. These cosine coefficients are used for estimating range queries. The estimation can be done over a range of data values spread over a number of clusters. The technique has been compared with cosine series technique for selectivity estimation. Experiments have been conducted on both synthetic and real datasets of varying sizes and results confirm that our technique offers substantial improvements in accuracy over other methods.*

Keywords: *Selectivity estimation, range query, data streams, micro-clustering*.

Received September 22, 2012; accepted December 24, 2013; published online August 22, 2015

1. Introduction

Advances in hardware technologies have resulted in a large continuous stream of data. Various applications such as ATM transactions, sensor networks, web clicks, telephone calls, network monitoring etc produce a large data which need to be processed to find the useful information such as fraud detection, network monitoring etc., these data streams cannot be stored in memory due to their unbounded nature and hence require online processing. Data stream algorithms are to be designed to scan the data only once. They also, have to deal with evolving nature of data stream. The processing time should be proportional to the size of a data stream and approximate answers should be within acceptable probabilistic guarantees.

Selectivity estimation is finding the fraction of values satisfying a predicate. Selectivity estimation is important in choosing the accurate query plan [4]. It saves the time and decreases the error propagation in the distributed environment. Selectivity estimation can be used in other applications like telephone call monitoring. The large size of data streams and their distributed nature increases the complexity of query processing. Selectivity estimation technique needs to have minimum computational cost with optimal accuracy. The technique should work independent of distribution type as it is difficult to predict the behaviour of data stream distributions. It should be able to work efficiently on complex queries, multi-dimensional data with minimum memory requirement.

Various synopsis techniques have been proposed for selectivity estimation in data streams. Sampling techniques [1, 14, 28, 29] scales down the data by randomly selecting some data values and is easy to maintain but does not work well with updates as well as with multi-dimensional data. Histogram techniques [17, 19, 25, 26] partition information into buckets. It does not work well with multi-dimensional data. Wavelet techniques [22, 23] decompose data into significant coefficients. It takes large space for calculation. Other techniques are kernel density estimation, Legendre polynomial and sketches.

In this paper, we propose an approach for Range Query Estimation (RQE) for evolving data streams using micro-clustering and cosine coefficients. The technique maintains micro-clusters with summary information about the data and its density distribution using cosine series. It estimates selectivity better than cosine series [30] method and also deals with evolving nature of data streams. It also, works well under updates. The rest of paper is organized as follows. Section 2 reviews various selectivity estimation methods for range query. Section 3 discusses various preliminary issues and section 4 contains the detailed explanation of the proposed technique. Result and comparison with cosine series technique has been shown in sections 5 and 6 discusses the implication of the method and section 7 concludes our study.

2. Literature Review

Selectivity estimation techniques can be divided into parametric and non-parametric types. Methods using parametric approach approximate by assuming the type of data distribution. This approach does not work without the prior knowledge of distribution. On the other hand non-parametric methods do not assume any distribution of data and employ various mathematical and statistical techniques to estimate selectivity.

Sampling techniques has been used for selectivity estimation in various works. Concise samples [14] maintained incrementally as <value, count> pairs. They overcome the space limitations of reservoir

sampling by keeping track of all occurrences of a value inserted into the relation. The count is increased for the sampled item. The Golden rule of sampling has been used for RQE [28]. It transforms non-uniform function to uniform distribution by using cumulative distribution function (cdf). The inverse of cdf function gives the values in the original domain. Query results are also estimated using adaptive sampling on cdf [29]. Although, easy to work with sampling methods, may not do well with unbounded and evolving data streams.

Spatial datasets are approximated using Min-Skew histograms. The buckets of the histogram are created by partitioning the array produced by the grid. The aim is to minimize the variance inside each bucket.

To minimize storage requirements information in the bucket is compressed using discrete cosine transformation [21]. Selectivity estimation is done by integrating cosine coefficients over the query range. ST holes method [7] estimate statistically independent data. Multi-dimension histograms are organized hierarchically as a tree. Each node of the tree is a bucket. Estimate is improved using query feedback. It also considers merging of buckets to solve the limited memory size problem.

The 4-level tree index [9] has been proposed to approximate cumulative frequencies for each bucket in 32bits. It stores the partial frequency sums at seven intervals inside the bucket and overall frequency sum of the bucket. nLT is another histogram used for RQE [8]. It is a binary tree having hierarchal decomposition of the original data distribution. Result of multi-dimensional range queries has been estimated using variable size buckets [17]. All the dense grids are turned into buckets so as to ensure uniform data distribution within buckets. Histograms are not efficient for large dimensional data. The information entropy has also been used to build histograms [27].

Matias *et al*. [22] have used wavelet based histogram for selectivity estimation which improves histogram methods proposed by Poosala *et al*. [26]. It uses multi-resolution wavelet decomposing for building histograms on the underlying distribution. The technique can be easily applied to multiple attributes but has problem in updating. It requires large space for calculation. Wavelets are also used by [10, 15, 16, 18] for query approximation.

The AGMS sketch [5, 6] proposed a way of generating summaries of data as a random variable. Whenever, a data item arrives, the sketch vector is updated. The inner product of the frequency vectors $X[i]=x_f[i].x_g[i]$ is an unbiased estimator. Sketch partitioning method [11] partitions the join attribute domains to improve the method. Reducing variance of the estimate but requires prior knowledge of data distribution for proper partitioning. Skimmed sketch technique [12] is used to reduce the variance to improve the accuracy of result. 2-level hash sketches [13] are used to estimate Join distinct queries. Sketches have been also used to estimate graph streams [24]. Sketches have a good updating mechanism but may not give good results for multi-dimensional data.

Cosine series methods [20, 30] maintain significant transform values as cosine energy of data is localized in small number of coefficients. The method works well for complex queries, takes less space. Other data density estimation technique such as kernel density estimation is also being used for selectivity estimation [32].

Micro-clustering technique has been used to predict estimation of a future query [2]. This technique is based on cluster feature vectors proposed by Zhang *et al*. [31]. It has also been used in clustering [3] and classification, it deal well with the evolving nature of data streams.

3. Preliminaries

3.1. Range Queries and their Estimation using Data Density Functions

Range queries are used to find number of data values falling in a particular range. These are normally of the form $a \leq X \leq b$, here X is an attribute in the range a to b, whereas a and b are constant values. When a and b are equal it gives estimation at a point e.g., find the number of persons having height greater than 160cm and less than 170cm.

Selectivity of range queries tells us how much percentage of total values satisfies a predicate. Data density functions can be used for estimating the selectivity of range queries. Consider random quantity X that has probability density function and then f gives natural description of X, using which probabilities associated with X can be found. Density estimation is construction of an estimate of density function from the observed data.

$$P(a < X < b) = \int_a^b f(d)dx \quad for\ all\ a < b \tag{1}$$

The above probability density function can be used to estimate how many data values lie in a particular range.

3.2. Normalization of Data

To make implementation easier data values are normalized to domain (0, 1) by considering a large maximum value *max* and minimum value *min*.

$$X = \begin{cases} 0 & x \leq min \\ \dfrac{x - min}{max - min} & min < x < max \\ 1 & x \geq max \end{cases} \tag{2}$$

3.3. Cosine Series as Orthonormal Basis

Orthonormal series estimators estimate the density f on the unit interval [0, 1] by estimating the coefficient of the expansion. For example f can be represented as the Fourier series $\sum_{i=0}^{\infty} fi\,\varphi i$

$$fi = \int_0^1 f(x)\varphi i(x)dx \quad for\ each\ i \geq 0 \tag{3}$$

Where sequence ϕ_i for r= 1, 2, ...

$$\varphi_0(x) = 1 \tag{4}$$

$$\varphi_{2r-1}(x) = \sqrt{2}\cos\pi rx \tag{5}$$

$$\varphi_{2r+1}(x) = \sqrt{2}\sin\pi rx \tag{6}$$

And estimator of f_i for $X_1, X_2, \ldots, X_n$

$$\hat{fi} = \frac{1}{n}\sum_{j=1}^{n}\varphi i(Xj) \tag{7}$$

Density estimate is given by:

$$\hat{f}(x) = \sum_{i=0}^{k} fi\,\hat{\varphi i}(x) \tag{8}$$

Cosine series has very good compaction property as most of the signal information is concentrated in low frequency components. It can be updated easily. Cosine series have infinite functions.

$$1, \sqrt{2}\cos\pi x, \sqrt{2}\cos 2\pi x, \sqrt{2}\cos 3\pi x, ...$$

These functions work as orthonormal basis. They can be used for distribution selectivity, as the selectivity of all the data values i.e., 0 to 1 is guaranteed equal to 1. Let n is the length of the input sequence; coefficients of data density estimator can be calculated and updated easily.

$$\hat{\beta}0 = 1 \tag{9}$$

$$\hat{\beta}1 = \frac{1}{n}\sum_{i=0}^{n}\sqrt{2}\cos\pi i \tag{10}$$

$$\hat{\beta}2 = \frac{1}{n}\sum_{i=0}^{n}\sqrt{2}\cos 2\pi i \tag{11}$$

$$\hat{\beta}m = \frac{1}{n}\sum_{i=0}^{n}\sqrt{2}\cos m\pi i \tag{12}$$

Estimator of data density function can be given as:

$$f(x) = 1 + \hat{\beta}1\sqrt{2}\cos\pi i + \hat{\beta}2\sqrt{2}\cos 2\pi i + \hat{\beta}3\sqrt{2}\cos 3\pi i + ... \tag{13}$$

Integration of this function gives the estimation.

$$\hat{\sigma}sel = \int_a^b f(x)dx \tag{14}$$

On insertion of an element x, $\hat{\beta}_i$ is updated by calculating average coefficients for n+1 data values.

$$\hat{\beta}i = \frac{\hat{\beta}i * n + \sqrt{2}\cos\pi ix}{n+1} \tag{15}$$

On deletion of an element x, $\hat{\beta}_i$ is updated by calculating average coefficients for n-1 data values.

$$\hat{\beta}i = \frac{\hat{\beta}i * n - \sqrt{2}\cos\pi ix}{n-1} \tag{16}$$

4. Proposed Micro-Cluster Based Technique

The given data sequence is represented in the compact form using clustering. These clusters are used for selectivity estimation. Statistical information about the data locality is maintained in terms of micro-clusters, which are defined in similar way as cluster feature vector [31].

4.1. Definition of Micro-Cluster

A data stream is viewed as an infinite process consisting of data which continuously evolves with time. Let us assume that the data stream consists of a sequence of data values $X_1, \ldots, X_k$, arriving at time stamps $T_1, \ldots, T_k$. Then, a micro-cluster MC_i stores the statistical information about the all the data values of a cluster. This information can be used to maintain the cluster and calculate the data density estimation over the data values of that cluster. A MC_i contains the following information:

1. *S*: Sum of all the data values in a cluster is used to calculate centroid, adding or deleting values from a cluster.
2. *SS*: Square Sum of all the data values in a cluster is used to find standard deviation of data values.
3. *N*: Number of data values in the cluster
4. *mCC*: *m* number of Cosine Coefficients which are used to generate data density estimator for a particular cluster.
5. *STS*: Sum of Time Stamps of data values is used to find the mean arrival time of data values in the cluster. It tells how old that cluster is hence; make it possible to delete the micro-cluster with the least recent time-stamps.
6. *SSTS*: Sum of Squares of Time Stamps of data items is stored to find the standard deviation. Approximate the average time stamp of the last *m* data values of the cluster. When the least relevance stamp of any micro-cluster is below a user-defined threshold, it can be deleted.

4.2. Maintenance of Micro-Clusters

As fast and continuous data values arrive, they are assigned to nearest cluster. Nearness to a cluster is calculated as distance of origin from the centroid of that cluster. If the data value is not in the Mahalanobis radius of nearest cluster and total number of micro-cluster has not reached maximum, new cluster is created and the statistics is stored. If number of clusters has reached maximum number then cluster with oldest time stamp is deleted. If there is no cluster old enough to be deleted then two nearest clusters are merged, followed by creation of new micro-cluster.

Algorithm 1: MaintainMicroClusters(*DS, list, MC$_i$, Max*).

 # DS: $x_1, x_2, \ldots, x_n$ i.e., stream of data values normalized to #(0, 1).
 # Max: Maximum number of clusters
 # S_i(S, SS, N, CC, STS, SSTS): Statistics of i^{th} micro-cluster
 # list: List of micro-clusters $MC_1, MC_2, \ldots, MC_m$ in the
 # increasing order of distance from centroid.
If (*list = NULL*)
{

```
    MC_new = new_cluster(S_i)
  }
else
{
  # Traverse the list to find the micro-clusters nearest to x
  MC_nearest with dist_ix=MIN(dist_1x,dist_2x...dist_mx)
 }
If ( (X − μ)/σ < 4 * σ )
  {
    allot( X, MC_nearest)
    update(S_nearest)
  }
else if ( no_of_clusters>M)
       {
         MC_new= new_cluster(S_i)
         insert(MC_new,list)
       }
else
       {
         MC_oldest= search_oldest_cluster(list)
         delete(MC_oldest, list)
         MC_new= new_cluster(S_i)
        insert(MC_new,list)
       }
If (no old cluster)
  {
    find two nearest clusters MC_i and MC_i+1
    MC_merged= MC_i + MC_i+1
    insert(MC_merged, list)
  }
return(list)
```

Here, in the Algorithm1 we search MC_{oldest} searched in the *list* using the time stamps stored in micro-clusters. Statistics of two micro-clusters S_i and S_{i+1} to be merged can be added easily. *Sum*, *SquareSum*, *N*, *STS* and *SSTS* can be added directly while new average of coefficient is calculated for all the m coefficients. $\frac{X-\mu}{\sigma}$

Gives the Mahalanobis radius of X from a cluster, where μ is mean of all the data items σ is standard distribution and μ can be easily calculated using the *S*, sum of data values and *SS* square sum of data values stored as micro-cluster statistics.

4.3. Selectivity Estimation Using Micro-Cluster

Selectivity is estimated using the information stored in the micro-clusters. To estimate range query the micro-clusters within query range are selected and average of cosine coefficients are calculated. These coefficients are used for generating data distribution function, which is used to calculate the number of values lying in the range query. Algorithm 2 proposed a technique for the RQE:

Algorithm 2: RQE(*list, a, b*).

```
# list: List of micro-clusters MC_1, MC_2, ..., MC_m in the
# increasing order of distance from the origin.
# a, b: Range of the query  a ≤ b
# result: Number of data values in the given range (a, b)
# Traverse the list and find cluster nearest to 'a'
  MC_a= Find_nearest_cluster(list, a)
  store_ cosine_coefficient(a, coa[1], coa[2], …, coa[m])
  store_ cosine_coefficient(b, cob[1], cob[2], ..., cob[m])
   n1=n2=N
if (MC_a≠ nearest to 'b')
  {
     While (MC_i ≠ MC_b)
      {
        update_ coefficients(cob[1...m])
        n2=n2+N
        jump to next cluster in the list
      }
      for(i=1 to m)
        {
                    fa=fa+coa[i]*√2 sin πia / πi
                    fb=fb+cob[i]*√2 sin πib / πi
        }
    }
 if (fa<0)
  {
     fa=0
  }
 if (fb<0)
  {
     fb=fa=0
  }
 result=fb*n2 –fa *n1
 return(result)
```

Here, we search the linked list for the micro-cluster nearest to left value of the range a i.e., MC_a then store the m cosine coefficients in the *coa*[1, ..., *m*] and *cob*[1, ..., *m*] where *coa*[1, ..., *m*] represent the cosine coefficient in micro-cluster MC_a and *cob*[1, ..., *m*] is the average of cosine coefficients of micro-clusters from MC_a to MC_b. Now, find micro-cluster MC_b nearest to right value of the range i.e., *b*. The *coa*[1, ..., *m*] represent the cosine coefficient in micro-cluster MC_a and *cob*[1, ..., *m*] is the average of cosine coefficients of micro-clusters from MC_a to MC_b. In case MC_a is not same as MC_b then traverse the list up to micro-cluster MC_b nearest to *b* is found and keep averaging *cob*[1, ..., *m*]. *fa* gives selectivity of a in MC_a and *fb* gives selectivity of *b* from MC_a to MC_b. *n*1 gives number of data items in MC_a and *n*2 gives number of data items from MC_a to MC_b.

5. Experimental Results

In this section we report the experiments result of the RQE method and cosine series technique used in the literature. The analysis has been done using C language. Experiments were performed on a PC with 2.67GigaHtz processor CPU and 1GB memory. Experiments were done on both real and synthetic datasets. We compare the accuracy of selectivity estimation. The technique has been tested for various range queries such as $a \leq X \leq b$. Result has been verified and analyzed for varying number of coefficients and clusters. To perform experiment data sets are normalized to (0, 1). Error was calculated as:

$$Error = \frac{(Estimated\ value - Actual\ value)}{Actual\ value} \quad (17)$$

Table 1 presents results of experiments performed on dataset 1 having 3769 data values and experiment was

conducted using 16 micro-clusters. The results of experiments conducted on dataset 2 (synthetic.control.data) have been shown in Table 2. Experiments results for dataset 3 (docbyterm.tfidf.txt) are shown in Table 3 and results for dataset 4 (ECG dataset mitdbx_mitdbx_108) are compared in Figure 1. Experiments were conducted on dataset5 (taken from advertising dataset who_rated_what_2006.txt) for different number of clusters and results are shown in Table 4. The result of experiments conducted on dataset 6 (ann_gun_centroidA.txt) are being shown in Table 5, Experiments were also conducted on dataset 7 (ECG dataset chfdbchf15.txt) and dataset 8 (Respiration dataset nprs43.txt). Table 6 presents the comparison of various datasets for the 12 number of clusters.

Table 1. Percentage of queries for given error range and number of coefficients for synthetic dataset1.

% Error	Number of Coefficients															
	RQE								Cosine Series Method							
	100	200	300	400	500	600	700	800	100	200	300	400	500	600	700	800
0-4	25	20	20	25	20	25	25	20	16	16	20	20	16	25	20	33
0-8	33	25	29	29	25	29	29	29	20	25	20	25	25	37	29	37
0-12	33	25	33	33	29	29	29	29	20	29	29	29	33	41	33	37
0-16	37	37	37	37	37	37	37	37	20	29	29	45	33	41	41	41
0-20	37	37	41	41	37	37	41	41	20	37	33	45	33	50	41	45
0-24	41	45	45	45	41	45	45	45	20	37	33	50	37	54	45	54

Table 2. Percentage of queries for given error range and number of coefficients for synthetic dataset2.

% Error	Number of Coefficients															
	RQE								Cosine Series Method							
	100	200	300	400	500	600	700	800	100	200	300	400	500	600	700	800
0-4	41	41	50	50	50	50	50	50	0	0	0	0	0	0	0	0
0-8	58	50	58	62	58	58	58	62	0	0	0	0	0	0	0	0
0-12	70	66	70	70	66	70	70	70	0	0	0	0	0	0	0	0
0-16	75	70	70	75	75	75	75	75	0	0	0	0	0	0	0	0
0-20	79	75	75	79	79	79	79	79	0	0	0	0	0	0	0	0
0-24	79	75	75	79	79	79	79	79	0	0	0	0	0	0	0	0

Table 3. Percentage of queries for given error range and number of coefficients for dataset3.

% Error	Number of Coefficients															
	RQE						Cosine Series Method									
	100	200	300	400	500	600	700	800	100	200	300	400	500	600	700	800
0-4	45	50	58	58	58	58	58	62	0	0	4	8	0	8	4	8
0-8	50	58	62	62	58	66	62	62	0	4	4	8	4	8	4	8
0-12	70	62	66	70	70	70	70	70	0	8	8	8	8	8	8	8
0-16	79	70	75	79	75	75	75	75	0	12	8	8	12	8	12	8
0-20	79	75	75	79	79	79	79	79	0	12	8	8	12	8	12	8
0-24	79	75	75	79	79	79	79	79	0	12	12	8	16	8	12	8

Table 4. Percentage of queries for given error range and number of clusters for dataset5.

% Error	Number of Clusters															
	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36
0-4	45	37	45	29	45	45	45	50	20	20	33	29	33	29	29	62	62
0-8	66	62	70	50	54	50	58	62	41	41	45	45	45	45	45	66	66
0-12	66	62	70	54	58	58	62	62	58	62	66	66	62	62	62	66	66
0-16	75	70	79	66	62	66	70	75	62	62	66	66	62	66	66	75	75
0-20	79	70	83	79	66	70	79	75	62	66	66	66	66	66	66	79	79
0-24	79	70	83	79	66	70	79	75	75	75	70	70	70	70	70	83	83

Table 5. Percentage of queries for given error range and number of coefficients for dataset6.

% Error	Number of Coefficients															
	RQE								Cosine Series Method							
	100	200	300	400	500	600	700	800	100	200	300	400	500	600	700	800
0-4	25	25	29	29	20	25	33	33	4	4	4	4	4	4	4	4
0-8	37	41	45	45	45	41	41	41	4	4	4	4	4	4	4	4
0-12	45	50	58	58	54	54	58	54	4	4	4	4	4	4	4	4
0-16	50	54	58	58	58	58	58	58	4	4	4	4	4	4	4	4
0-20	54	54	58	58	58	58	58	58	4	4	4	4	4	4	4	4
0-24	58	62	62	62	62	62	66	62	8	8	8	8	8	8	8	8

Table 6. Comparison of percentage of queries for given error range and number of values for 12 clusters.

% Error	Dataset1	Dataset 7	Dataset	Dataset 6	Dataset 4	Dataset 5	Dataset 3
	3793	15024	18077	22527	64825	200025	741502
0-4	20	25	16	25	37	45	54
0-8	25	41	16	41	50	54	62
0-12	29	50	29	50	54	58	66
0-16	33	54	37	54	62	62	70
0-20	37	58	37	54	70	66	75
0-24	45	58	41	58	75	66	75

6. Discussion

The proposed technique improves the selectivity estimation of range query over data streams. Results of experiments in Tables 2 and 3 shows that there is a significant improvement over cosine series method for different datasets. The result of synthetic dataset1 has got comparable results with existing technique. All the data values are represented compactly in the form of micro-clusters. The spread of data values covered in a micro-cluster is less and the cosine coefficients are used for selectivity estimation. Cosine series method covers the whole range of data values and RQE method works over a narrow range of data values. Hence, for narrow range queries RQE technique works better than cosine series technique proposed by Yan *et al.* [30]. The results for RQE and cosine series techniques are comparable if the range of the query covers whole range of data values.

RQE technique performs well for normal, moderately skewed and highly skewed data distributions. Dataset4 and dataset7 are highly skewed with values 1.66 and -2.41 respectively. Dataset 1, dataset 3 and dataset 5 are moderately skewed with values 0.66, 0.879 and 0.938 respectively. Dataset 2 and dataset 6 are approximately symmetric. Figures 1 and 2 shows that the results for all these datasets are comparable.

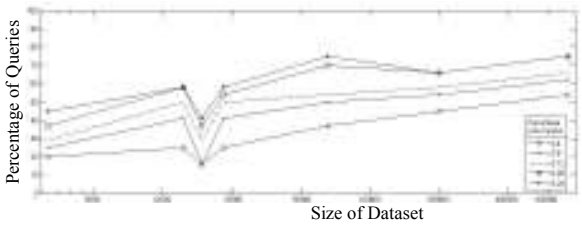

Figure 1. Comparison of results for increasing size of datasets (12 clusters, 200 coefficients).

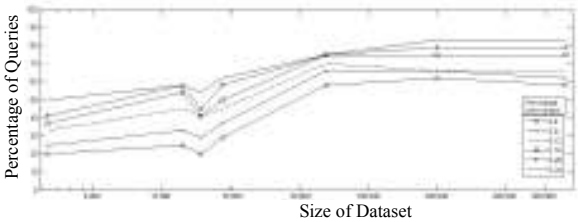

Figure 2. Comparison of results for increasing size of datasets (36 clusters, 200 coefficients).

The technique works well with datasets of different sizes. Figures 1 and 2 show that the accuracy of result improves with the increase in the size of dataset. For

example data set with size 3793, 22527, 64825 and 741502, percentage of queries within 4% error range are 20%, 25%, 37% and 54% respectively. This means that highest size dataset has highest number of queries in the lowest error range. Figure 3 shows that the RQE method performs better than the cosine series method.

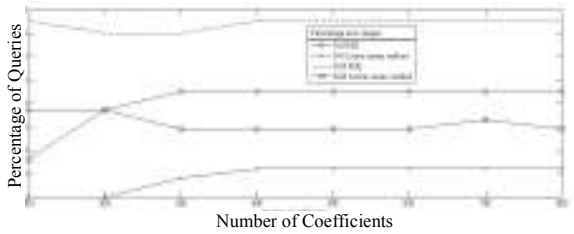

Figure 3. Comparison of results for dataset4 for 16 clusters.

Cosine series has infinite number of coefficients. It stores significant information in few low frequency coefficients. Accuracy of results improves with the increase in number of stored cosine coefficients. Larger numbers of coefficient are required for multi-dimensional data.

RQE stores cosine coefficients separately for each micro-cluster, so with increase in the number of micro-clusters the requirement of is also increased. On the other hand cosine series require memory to store only one set of cosine coefficients.

Processing time of RQE method is more as compared to cosine series method. It has to maintain micro-clusters and search the nearest micro-clusters for selectivity estimation. As the number of clusters increases the more processing time is required for searching the nearest cluster. It has been observed that the small number of clusters also gives comparable result to large no of clusters.

The proposed method requires an efficient and accurate clustering algorithm. Better clustering algorithm can improve the efficiency and accuracy of results.

7. Conclusions

The paper studied the use of micro-clusters for estimating result size of range queries. We have demonstrated an efficient approach for RQE using cosine series and micro-clustering. Proposed technique is compared with cosine series technique. The accuracy of result has been proved for different datasets. The technique performs well for normal, moderately skewed and highly skewed data distributions. Accuracy of results increases with the increase in size of data and number of stored cosine coefficients. Future work will be to generalize the technique for multi-dimensional data.

References

[1] Acharya S., Gibbons P., Poosala V., and Ramaswamy S., "Join Synopses for Approximate Query Answering," *in Proceedings of ACM SIGMOD International Conference on Management of Data*, Philadelphia, Pennsylvania, pp. 275-286, 1999.

[2] Aggarwal C., "On Futurisitic Query Processing in Data Streams," *in Proceedings of the 10th International Conference on Advances in Database Technology*, Germany, pp. 41-58, 2006.

[3] Aggarwal C., Han J., Wang J., and Yu P., "A Framework for Clustering Evolving Data Streams," *in Proceedings of the 29th VLDB Conference*, Berlin, Germany, pp. 81-92, 2003.

[4] Aljanaby A., Abuelrub E., and Odeh M., "A Survey of Distributed Query Optimization," *the International Arab Journal of Information Technology*, vol. 2, no. 1, pp. 48-57, 2005.

[5] Alon N., Gibbons P., Matias Y., and Szegedy M., "Tracking Join and Self-join Sizes in Limited Storage," *the Journal of Computer and System Sciences*, vol. 64, no. 3, pp. 719-747, 2002.

[6] Alon N., Matias Y., and Szegedy M., "The Space Complexity of Approximation the Frequency Moments," *in Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, Philadelphia, Pennsylvania, pp. 20-29, 1996.

[7] Bruno N., Chaudhuri S., and Gravano L., "STHoles: A Multidimensional Workload-Aware Histogram," *in Proceedings of ACM SIGMOD International Conference on Management of Data*, Santa Barbara, pp. 211-222, 2001.

[8] Buccafurri F., Lax G, "Fast Range Query Estimation by N-level Tree Histogram," *Data and Knowledge Engineering*, vol. 51, no. 2, pp. 257-275, 2004.

[9] Buccafurri F., Pontieri L., Rosaci D., and Sacca D., "Improving Range Query Estimation on Histograms," *in Proceedings of the 18th International Conference on Data Engineering*, San Jose, USA, pp. 628-638, 2002.

[10] Chakrabarti K., Garofalakis M., Rastogi R., and Shim K., "Approximate Query Processing using Wavelets," *in Proceedings of the 26th International Conference on Very Large Data Bases*, pp. 111-122, 2000.

[11] Dobra A., Garofalakis M., Gehrke J., and Rastogi R., "Processing Complex Aggregate Queries over Data Streams," *in Proceedings of ACM SIGMOD International Conference on Management of Data*, Wisconsin, pp. 61-72, 2002.

[12] Ganguly S., Garofalakis M., and Rastogi R., "Processing Data-Stream Join Aggregates Using Skimmed Sketches," *in Proceedings of the 9th International Conference on Extending Database Technology*, Crete, Greece, pp. 569-586, 2004.

[13] Ganguly S., Garofalakis M., Kumar A., and Rastogi R., "Join-Distinct Aggregate Estimation over Update Streams," *in Proceedings of 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, Maryland, USA, pp. 259-270, 2005.

[14] Gibbons P. and Matias Y., "New Sampling Based Summary Statistics for Improving Approximate Query Answers," *in Proceedings of ACM SIGMOD International Conference on Management of Data*, Seattle, Washington, pp. 331-342, 1998.

[15] Gilbert A., Kotidis Y., Muthukrishnan S., and Strauss M., "One-Pass Wavelet Decompositions of Data Streams," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 3, pp. 541-554, 2003.

[16] Gilbert A., Kotidis Y., Muthurkrishan S., and Strauss M., "Surfing wavelets on streams: One-Pass Summaries for Approximate Aggregate Queries," *in Proceedings of the 27th International Conference on Very Large Data Bases*, Roma, Italy, pp. 79-88, 2001.

[17] Gunopulos D., Kollios G., Tsotras V., and Domeniconi C., "Selectivity Estimators for Multidimensional Range Queries Over Real Attributes," *the VLDB Journal*, vol. 14, no. 2, pp. 137-154, 2004.

[18] Ilic S. and Spalevic P., "Using Wavelet Packets for Selectivity Estimation," *the Computer Journal*, vol. 56, no. 7, pp. 827-842. 2012.

[19] Jagadish H., Koudas N., Muthukrishnan S., Poosala V., Sevcik K., and Suel T., "Optimal Histograms with Quality Guarantees," *in Proceedings of the 24th International Conference on Very Large Data Bases*, New York, pp. 275-286, 1998.

[20] Jiang Z., Luo C., Hou W., Yan F., Zhu Q., and Wang C., "Join Size Estimation Over Data Streams using Cosine Series," *International Journal of Information Technology*, vol. 13, no. 1, pp. 27-46, 2007.

[21] Lee J., Kim D., and Chung C., "Multi-Dimensional Selectivity Estimation using Compressed Histogram Information," *in Proceedings of ACM SIGMOD Conference on Management of Data*, Philadelphia, Pennsylvania pp. 205-214, 1999.

[22] Matias Y., Vitter J., and Wang M., "Dynamic Maintenance of Wavelet-based Histograms," *in Proceedings of the 26th International Conference on Very Large Data Bases*, Cairo, Egypt, pp. 101-110, 2000.

[23] Matias Y., Vitter J., and Wang M., "Wavelet-Based Histograms for Selectivity Estimation," *in Proceedings of ACM SIGMOD International Conference on Management of Data*, Washington, USA, pp. 448-459, 1998.

[24] Peixiang Z., Aggarwal C., and Wang M., "gSketch: On Query Estimation in Graph Streams," *in Proceedings of the 38th International Conference on Very Large Databases*, Istanbul, Turkey, pp. 193-204, 2012.

[25] Poosala V. and Ioannidis Y., "Selectivity Estimation Without the Attribute Value Independence Assumption," *in Proceedings of the 23rd International Conference on Very Large Data Bases*, pp. 486-495, 1997.

[26] Poosala V., Ioannidis Y., Haas P., and Shekita E., "Improved Histogram for Selectivity Estimation of Range Predicates," *in Proceedings of ACM SIGMOD Conference*, pp. 294-305, 1996.

[27] To H., Chiang K., and Shahabi C., "Entropy-based Histograms for Selectivity Estimation," available at: http://infolab.usc.edu/DocsDemos/to-CIKM13.pdf, last visited 2013.

[28] Wu Y., Agrawal D., and Abbadi A., "Applying the Golden Rule of Sampling for Query Estimation," *in Proceedings of ACM SIGMOD International Conference on Management of Data*, Santa Barbara, pp. 449-460, 2001.

[29] Wu Y., Agrawal D., and Abbadi A., "Query Estimation By Adaptive Sampling," *in Proceedings of the 18th International Conference on Data Engineering*, San Jose, USA, pp. 639-648, 2002.

[30] Yan F., Hou W., Jiang Z., Luo C., and Zhu Q., "Selectivity Estimation of Range Queries based on Data Density Approximation via Cosine Series," *Data and Knowledge Engineering*, vol. 63, no. 3, pp. 855-878, 2007.

[31] Zhang T., Ramakrishnan R., and Livny M., "BIRCH: An Efficient Data Clustering Method for Very Large Databases," *in Proceedings of ACM SIGMOD International Conference on Management of Data*, Quebec, Canada, pp. 103-114, 1996.

[32] Zhou A., Cai Z., Wei L., and Qian W., "M-Kernel Merging: Towards Density Estimation over Data Streams," *in Proceedings of 8th International Conference on Database Systems for Advanced Applications*, Kyoto, Japan, pp. 285-292, 2003.

Deepak Garg is Chair, Computer Science and Engineering Department. He is Chair, IEEE Computer Society, India Council as wellas Chair, ACM SIGACT, North India. He has more than 100 publications to his credit.

Sudhanshu Gupta is a research scholar in Computer Science and Engineering Department, Thapar University, India. He has done his Master degree in Computer Applications. His main research interests are data stream mining and big data.