Selectivity Estimation over Multiple Data Streams using Micro-clustering

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

Abstract:

Selectivity *estimation is an important task for query optimization. We propose a technique to perform range query estimation over multiple data streams using micro-clustering. The technique maintains cluster statistics in terms of micro-clusters and cosine series for all streams. These micro-clusters maintain data distribution information about the stream 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 streams.*

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

1. Introduction

A large continuous stream of data is generated by various applications such as ATM transactions, sensor networks, web clicks, telephone calls, network monitoring etc.We analyze these data streams to analyze useful information. Data streams are unbounded and cannot be stored in memory for processing. We design algorithms which can scan the online data in one pass for answering queries.

Selectivity isdefined as fraction of values satisfying a predicate. It is important in various applications such as choosing the accurate query plan, telephone call monitoring. We design selectivity estimation techniques to have accurate result in minimum space and time. These techniques should be able to work efficiently on multi-dimensional data.

Let us assume that there exists N number of sources generating stream of data. Selectivity of range queries over these multiple data streams can be useful in analyzing the cumulative trends in data generated by different sources.

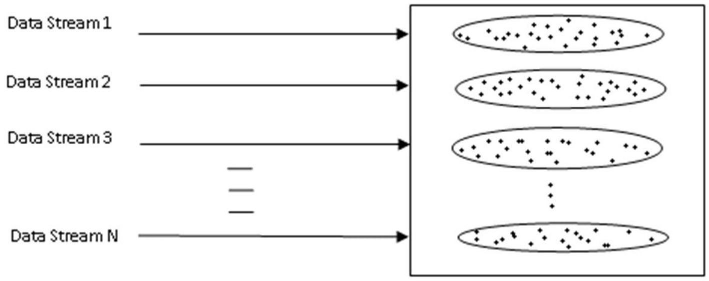

Fig 1: Multiple data streams

Various techniques have been proposed for selectivity estimation over data streams. These techniques use different synopsis techniques to store summery statistics of the data stream. Sampling based techniquesscale down the data by randomly selecting some data values. Histogram techniques partition data distribution into data ranges called buckets. Wavelet techniques decompose data into significant coefficients. Other techniques are kernel density estimation, cosine series and sketches.

In this paper we propose an approach for range query estimation over multiple data streams using micro-clustering. The technique maintains micro-clusters with summary information about the data and its density distribution using cosine series for all the data streams. It deals with evolving nature of data stream. 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. Section 6 discusses the implication of the method and section 5 concludes our study.

2. Literature Review

Various selectivity estimation techniques have been proposed in the literature. These techniques maintain the summery statistics to estimate the selectivity over data streams.

We can use reservoir sampling technique togenerate data stream summery in one pass. The concise sampling method [9]overcome the constraints posed due space limitation by incrementally maintaining <value,count> pairs. The Golden rule of sampling has been used for range query estimation[15] by transforming non-uniform function to uniform distribution by using cumulative distribution function(cdf). The inverse of cdf function gives the values in the original domain. Adaptive sampling [16]is another technique used for estimating queries. Sampling methods are easy to work with but may not do well with unbounded and evolving data streams.

Histogram based methods partition the data distribution in a set of ranges called bucket. We use count of the values in the bucket range for the query estimation. Min-skew histograms are useful in approximating queries in spatial datasets. We create buckets of the histogram by partitioning the array produced by the grid. This minimizes the variance within each bucket.

The STholes method estimates selectivity using a tree of histograms. We treat each node of the tree is treated as bucket and improve estimate using query feedback.The merging of buckets solves the memory limitation problem. The storage requirements information in the bucket can be compressed using discrete cosine transformation[12]. Here we estimate selectivity by integrating cosine coefficients over the query range.

The 4-level tree index[5] approximates cumulative frequencies for each bucket. The technique stores the partial frequency sums at seven intervals inside the bucket and overall frequency sum of the bucket. nLT[4] estimates range queries using abinary tree with hierarchal decomposition of the original data distribution. Gunopulos et al. estimate multi-dimensional range queries using variable size buckets[10]. The technique turns dense grids into buckets so as to ensure uniform data distribution within buckets. Histograms can be built using information entropy [14].Matias et al.[13] use wavelet based histogram for selectivity estimation. They use multi-resolution wavelet decomposition for building histograms. However it requires large space for calculation. Chakarbarti et al. use multi-dimensional wavelets for query estimation.

Sketch is a psedu random linear projection on data. Sketches has been used in various applications such as L2 distance, second moment etc. The AGMS sketch[2,3] proposed a way of generating summaries of data as a random variable. Sketch partitioning method [6] partitions the attribute domains and maintains sketches over all the partitions. It reduces the variance of the estimatebut requires proper partitioning of data distribution. Skimmed sketch technique[8] reduces the variance to improve the accuracy of result. 2-level hash sketches[7] estimates Join distinct queries. Kernel density methods approximate the probability distribution. M-kernel method estimates the data density merging the similar kernels to deal with memory constraint.

Cosine series methods [11,17] use cosine energy of data to estimate complex queries.Cluster based methods are also useful for selectivity estimation. Micro-clustering technique has been used to predict estimation of a future queries [1]. This technique is based on cluster feature vectors proposed in [18].It deals well with the evolving nature of data streams.

3.Preliminaries

3.1 Range Queries and Data Density Functions

Range queries generates the data values in a particular range. It is of the form$a \le X \le b$, here X is an attribute in the range $a$to$b$. Selectivity of range queries is given as the percentage of total values satisfies a predicate. We use data density functions to estimate the selectivity of range queries. Consider random quantity Xthat has probability density function, and then f gives probabilities associated with X.

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

We use the probability density function to find the estimate of data values lie in a particular range.

3.3 Cosine series

To make implementation easier we normalize data values to domain (0,1)by considering a large maximum value *max* and minimum value *min*.Orthonormal series estimators estimate the density $f$on the unit interval [0,1] by estimating the coefficient of the expansion.

Cosine series concentrates most of the signal information in low frequency components, hence has very good compaction property. Cosine series has infinite functions which work as orthonormal basis and 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.

$$\hat{\beta}_0 = 1 \qquad (2)$$

$$\hat{\beta}_1 = \frac{1}{n}\sum_{i=0}^{n}\sqrt{2}\cos(\pi i) \qquad (3)$$

$$\hat{\beta}_2 = \frac{1}{n}\sum_{i=0}^{n}\sqrt{2}\cos(2\pi i) \qquad (4)$$

… … ..

$$\hat{\beta}_m = \frac{1}{n}\sum_{i=0}^{n}\sqrt{2}\cos(m\pi i) \qquad (5)$$

Estimator of data density function can be given as

$$f(x) = 1 + \hat{\beta}_1\sqrt{2}\cos\,\pi i + \hat{\beta}_2\sqrt{2}\cos\,2\pi i + \hat{\beta}_3\sqrt{2}\cos\,3\pi i + ... \qquad (6)$$

Integration of this function gives the estimation

$$\hat{\sigma}_{sel} = \int_a^b f(x)dx \qquad (7)$$

We update $\hat{\beta}_i$ on insertion of an element *x*, by calculating average coefficients for *n+1* data values.

$$\hat{\beta}_i = \frac{\hat{\beta}_i * n + \sqrt{2}\cos\pi i x}{n+1} \qquad (8)$$

We update $\hat{\beta}_i$ on deletion of an element *x*, calculating average coefficients for *n-1* data values.

$$\hat{\beta}_i = \frac{\hat{\beta}_i * n - \sqrt{2}\cos\pi i x}{n-1} \qquad (9)$$

4. Proposed Technique

In the proposed technique we maintain micro-clusters for different data streams. We generate aggregate statistics using the information stored in micro-clusters of different data streams. This aggregate statistics is used to estimate the selectivity of range queries.

4.1 Definition of Micro-cluster

A data stream is an infinite processconsisting 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 various time stamps. Then a micro-cluster stores the information about the all the data values of a cluster. This information is sufficient to maintain the cluster and to find the data density estimation over the multiple streams. A micro-cluster contains the following information

1. ***S*:** Sumof all the data values in a cluster is useful in calculating centroid of the cluster
2. ***SS*:** Square Sum of all the data values in a cluster is useful to find standard deviation of data values.
3. ***N*:**Number of data values in the cluster
4. ***mCC*:** m number of Cosine coefficients for generating data density estimator for a particular cluster.

4.2 *Maintenance of Micro-clusters*

As the data value arrives, we measure its distance from the centroid of different clusters. The data value is assigned to the nearest micro-cluster. If the data value is not in the Mahalanobis radius of nearest cluster, a new cluster is created and the statistics is stored. We consider merging of clusters when they fall within the Mahalanobis radius. Algorithm1 maintains micro-clusters over a data stream and is applied to all the data streams.

```
Algorithm1 MicroClusters(DS,list,$MC_i$ ,Max)
DS: stream of data values normalized to (0,1).
$S_i$(S,SS,N,CC): Statistics of $i^{th}$ micro-cluster
list: List of micro-clusters $MC_1$, $MC_2$…$MC_m$
If( list is empty)then
  Create micro-clusters using K-means algorithm and add statistics to them
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  point falls in Mahalanobis radius then
Allot X to nearest cluster $MC_{nearest}$
else
create new cluster and store summary statistics
endif
Find the two clusters fall in the Mahalaobis radius
$MC_{merged} = MC_n + MC_m$
endif
return(list)
```

Any two micro-clusters can be merged easily by adding their summery statistics. *Sum, SquareSum and N*can be added directly while new average of coefficient is calculated for all the m cosine series coefficients. The Mahalanobis radius of *X* from a clusteris given as $\frac{X-\mu}{\sigma}$. Where μ is mean of all the data items σ is standard distribution. Mean and variance are calculated using the *S andSS* stored as micro-cluster statistics.

4.3 Selectivity Estimation over multiple data streams

We estimate the selectivity using the information stored in the micro-clusters. We compute the selectivity over multiple streams. Firstly we select the set of streams on which we want to apply the range query. In the next step we search all the micro-clusters of selected stream which fall in the range of the query. These selected clusters are used to generate aggregate statistics as a new micro-cluster. This new micro-cluster having aggregated statistics is used for range query estimation.

We calculate the average of cosine coefficients over all the aggregated clusters. These coefficients are used for generating data distribution function, which is used to calculate the number of values lying in the range query.

```
Algorithm2. ($list_{1..N}$,a,b)
 $List_{1..N}$: List of micro-clusters for N data streams.
a,b: range of the query $a \le b$
result : Number of data values in the given range $(a,b)$
Select the set of streams for selectivity estimation from the $list_{1..N}$
For all the selected lists
Traverse the list and find clusters in the range ‘a’ to’ b’
```

```
    Add the summery statistics of all the selected clusters to $MC_{new}$
End
For all the cosine coefficients in $MC_{new}$
  $result = result + \frac{cc[i] * \sqrt{2}\sin \pi ia}{\pi i}$
End
If result is less than 0
  result=0
End if
return(result)
```

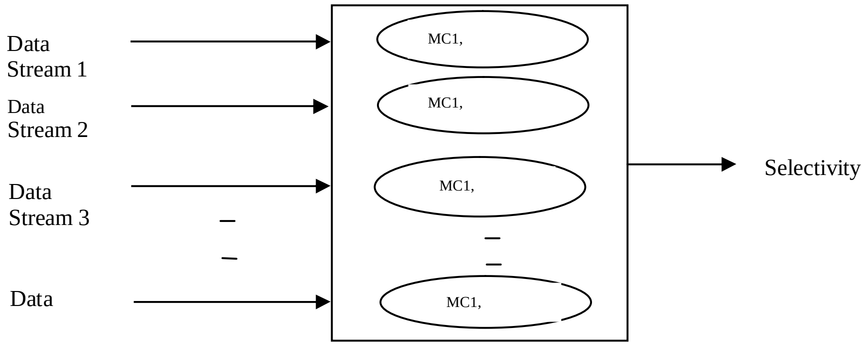

Fig 2: Selectivity estimation over multiple data streams

In the algorithm2 we select a set of streams for the selectivity estimation. We store these streams as list of micro-clusters using algorithm1. All the selected lists are processed to find the micro-clusters in the range of query. We add the summery statistics of all the micro-cluster in a new micro-cluster MC_{new}. We use cosine coefficient stored in MC_{new} to estimate the selectivity of range query $a \le X \le b$.

5. Conclusion

The paper studied the use of micro-clusters for estimating range queries over multiple data streams. We have proposed atechnique for range query estimation using cosine series and micro-clustering. Future work will be to generalize the technique for multi-dimensional data and use binary tree of micro-clusters for selectivity of continuous queries.

References

[1] Aggarwal C.C., "*On futurisitic query processing in Data streams*," Proceedings of the 10th international conference on Advances in Database Technology, pp. 41-58, 2006.

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

[3] Alon N., Matias Y., and Szegedy M., "*The space complexity of approximation the frequency moments*," Proceedings of 28th Annual ACM Symposium on the Theory of Computing, pp. 20-29, 1996.

[4] Buccafurri F., Lax, and Gianluca, "*Fast range query estimation by N-level tree histogram*," Data & Knowledge Engineering, Vol. 51 , no.2, pp. 257-275, 2004.

[5] Buccafurri F., Pontieri L., Rosaci D., and Sacca D., "*Improving Range Query Estimation on Histograms*," Proceedings of the 18th International Conference on Data Engineering (ICDE.02), Vol. 25, no.2, pp. 628-638, 2002.

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

[7] Ganguly S., Garofalakis M., Kumar A., and Rastogi R., "*Join-Distinct Aggregate Estimation over Update Streams*," Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 259-270, 2005.

[8] Ganguly S., Garofalakis M., and Rastogi R., "*Processing Data-Stream Join Aggregates Using Skimmed Sketches*," EDBT 2004, 9th International Conference on Extending Database Technology, pp. 569-586, 2004.

[9] Gibbons P.B. and Matias Y., "*New sampling based summary statistics for improving approximate query answers*," SIGMOD '98 Proceedings of the 1998 ACM SIGMOD international conference on Management of data, Vol. 27, no.2, pp. 331-342, 1998.

[10] Gunopulos D., Kollios G., Tsotras V.J., and Domeniconi C., "*Selectivity Estimators for Multidimensional Range queries over real attributes*," The VLDB Journal, Vol. 14, no.2, pp. 137-154, 2004.

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

[12] Lee J.-H., Kim D.-H., and Chung C., "*Multi-dimensional Selectivity Estimation Using Compressed Histogram information*," Proceedings of the ACM SIGMOD Conference on management of data, Vol. 28, no.2, pp. 205-214, 1999.

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

[14] To H., Chiang K., and Shahabi C., "*Entropy-based Histograms for Selectivity Estimation*," Proceedings of the 22nd ACM international conference on information & knowledge management (CIKM'13), pp. 1939-1948, 2013.

[15] Wu Y.-L., Agrawal D., and Abbadi A.E., "*Applying the Golden Rule of Sampling for Query Estimation*," Proceeding of ACM SIGMOD international conference on management of data, pp. 449-460, 2001.

[16] Wu Y.-L., Agrawal D., and Abbadi A.E., "*Query Estimation By Adaptive Sampling*," Proceedings of the 18th International Conference on data engineering, pp. 639-648, 2002.

[17] Yan F., Hou W.C., Jiang Z., Luo C., and Zhu Q., "*Selectivity Estimation of Range queries based on data density approximation via cosine series*," Data & Knowledge Engineering, Vol. 63, no., pp. 855-878, 2007.

[18] Zhang T., Ramakrishnan R., and Livny M., "*BIRCH:An efficient data clustering method for very large databases*," Proceeding of ACM SIGMOD international conference on management of data, pp. 103-114, 1996.