Proceedings of the World Congress on Engineering and Computer Science 2012 Vol I
WCECS 2012, October 24-26, 2012, San Francisco, USA

V-Optimal Filters for Data Approximation in Continuous Data Streams

Joseph Gomes, Wenhao Chen, and Pushkar Dahal

***Abstract*—Monitoring data streams in real time over distributed streaming environments plays a large role in maintaining situation awareness, which is a great challenge due to huge data volumes and bandwidth limitations. In this monitoring process, transmission cost and value accuracy are two very important but conflicting factors in measuring the efficacy of the system. On one hand, increasing value accuracy increases transmission cost. On the other hand, reducing transmission cost, which can be accomplished by smoothing the data, will reduce value accuracy. In this paper, we use V-Optimal histograms to approximate the data distribution at the data sources. The V-Optimal algorithm is used for computing optimal number of buckets and bucket boundaries given a certain error bound, which are then used to approximate and communicate data between the source and the server. We introduce the notion of a soft precision constraint (PC) and use two additional metrics namely, weighted average error and PC violation rate or error rate to control the quality of approximation. We show through extensive experimentation that our approach performs very well in maintaining data quality as well as reducing communication cost.**

***Index Terms*—V-Optimal, Histogram, Filter, Continuous Query, Data Approximation.**

I. Introduction

CONTINUOUS aggregate queries are used in many data streaming applications such as Internet applications, and sensor networks for monitoring purposes [3], [13]. For example, in Internet applications, by monitoring logs generated by high-speed routers for hot IP addresses, Denial Of Service (DoS) attacks can be detected. Many environment-aware technologies need to monitor temperature, light and moisture. Highway-traffic monitoring systems have to constantly monitor number of cars, their speed and average distance between cars to estimate traffic conditions and real-time travel duration. These smart infrastructures must continuously maintain an accurate representation of the current environment to be able to perform their jobs. Keeping an accurate representative view of present conditions is crucial for optimal decision making. Typically, data is gathered from hundreds of data sources and sent to a central server for further analysis. Continuous aggregate queries and analysis tools are registered at the central coordinator to process the incoming data. In a naïve approach, remote sources forward all data to the central server. However, the sheer volume and speed of the generated data along with energy constraints and wireless bandwidth limitations make this approach very inefficient. This approach is also wasteful as many applications can tolerate inaccuracy to a certain extent. Other available mechanisms that sacrifice accuracy for transmission cost use periodic sampling to estimate current conditions. In cases where the measurements are stable([1], [2], [4]), such sampling measures are sufficient to make real-time decisions. However, when data is characterized by high variability and abrupt trend changes and the applications require a certain level of accuracy, such schemes fail to deliver. In light of this, we propose to deploy error-bounded histogram based filters at the remote data sources that would significantly reduce the volume of data that are transmitted while maintaining a certain level of data accuracy. Histograms are commonly used in database systems for approximating data distribution and providing approximate answers to queries. However, no prior work has used histograms for approximate data communication. Although, data filtering will reduce communication cost, it would also result in a reduced level of accuracy in query results at the central server which may not be very desirable. The goal of our research is to balance between communication cost and value accuracy in a way that minimizes cost without sacrificing accuracy beyond the threshold demanded by the applications. The basic idea is to use the average value of a histogram bucket to represent each value in the bucket. When a new value is produced at the data source, the bucket number is communicated to the server only if the previous value did not belong in the same bucket. The server uses the average value of the bucket to approximate the new value at the data source. When the new value is still in the same bucket no transmission is required and the central server continues to use the same approximation. The difference between the new value at the data source and its approximation at the server represents the approximation error, which must follow any predefined constraint set by the application or user. In this paper, we describe our histogram based filtering approach, show the relationships among different parameters which affect transfer cost and value accuracy, and demonstrate the efficacy of our approach through extensive experimentation.

II. Problem Description and Overview of Approach

A distributed data streaming environment constitutes a central server and several remote data sources. Each source S_j produces a stream of data values $v_{j,i}$. The applications and the aggregate queries installed at the server use the data streams for various purposes such as producing aggregate results. For many of these queries or applications, complete accuracy may not be necessary, which allows for the actual value at the source and the received or estimated value at the server to be different. We use a predefined parameter called *Precision Constraint* (PC) to represent this flexibility. In order to take advantage of this flexibility a filter can be installed at each remote data source. The key performance

Manuscript received July 23, 2012.

J. Gomes, W. Chen and Pushkar Dahal are with the Department of Computer Science, Bowie State University, Bowie, MD, 20715 USA e-mail: (joe.sgomes@gmail.edu, winstonwchen@gmail.com, dahal.pushkar@gmail.com).

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

metrics in this environment are communication cost and accuracy.

In this paper, we will mainly focus on one data source without loss of generality as the same idea can be replicated across all data sources. From here on, we will use v_i to represent the i^{th} value from a data source. A precision constraint PC is also specified by the application or user which determines the window width within which the approximate value must fall. Given v_i and the corresponding approximation a_i, the approximation error e_i is computed as $|v_i - a_i|$, which should be less than $PC/2$. It is possible that even the precision constraint may be allowed to be violated, especially for aggregate queries as long as the average error stays within a certain limit. Therefore, we consider the precision constraint to be a loose constraint, not a strict one. Our goal is to reduce both the number of precision constraint violations and the average error while keeping communication cost down. Eventually, we would like to come up with a parameter or a set of parameters that could be used as a dial to control the number of PC violations, average error, and communication cost for a given application or scenario. We measure communication cost as the percentage of data values at the source that need to be transmitted to the server, which we call the *New Value Refresh Rate* (NVRR). The installed filters at the data source will reduce NVRR and thus communication cost; however, it will also result in a reduced level of data accuracy at the central server which will lead to precision constraint violations and an increase in average error. Our goal is to minimize refresh rate without sacrificing accuracy beyond the threshold desired by the application.

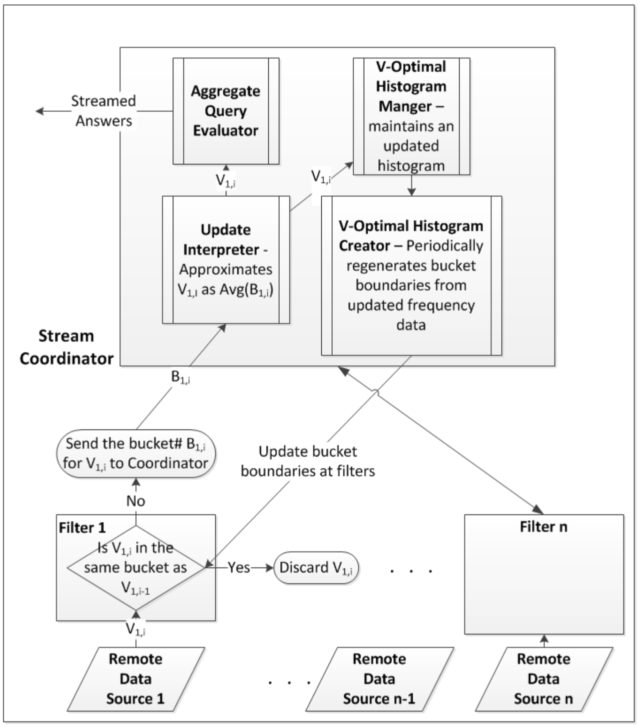


Fig. 1. System Architecture

Below, in Fig. 1 we show a flow diagram describing the process and the architecture of our approach. During the training period, the data sources forward data to the stream coordinator without performing any filtering. The histogram manager at the coordinator site uses this training data to create the histograms. In this paper, we use the V-Optimal Histogram algorithm, which we discuss in section III, to determine the number of buckets and bucket-boundaries that would keep the average estimation error below the desired level while minimizing the number of buckets. The bucket-boundaries are then communicated to the source to be used in the filter, which we call the V-Optimal filter. V-Optimal bucket boundaries are supposed to demarcate the data in a way where each bucket contains similar elements. In other words, buckets represent trend sequences or clusters. Note that, in this work we consider data sequences that follow a trend but can make abrupt directional changes, e.g. weather temperature, humidity and stock prices. Once the bucket-boundaries and bucket-means are computed, we use the bucket-means as approximations for all the bucket values, i.e. at the coordinator the value is approximated by the bucket-mean. Once the filters are in place, any new value generated at the source has to go through the filter before being sent to the coordinator. If the current value, unless its the first value, falls in the same bucket as the previous value nothing needs to be transmitted, thus reducing cost. On the other hand, if it is different, the bucket number of the new value is communicated to the coordinator. The coordinator uses the bucket-mean as an estimation of the value. The difference between the real value and the estimated value is the error. Let $V = \{v_i | v_i \in V, i \in \mathbb{N}\}$ be the set of data generated at the remote source. Let $B = \{b_j | b_j \in B, j \in \mathbb{N}\}$ be the set of all buckets calculated during the training period. Let m_j be the mean value of bucket b_j. When the remote source gets $v_i \in b_j$, it sends the value b_j to the central server. The server interprets this value as m_j, i.e. $a_i = m_j$. This gives an error of $|v_i - m_j|$. PC violation occurs when this error exceeds $\frac{PC}{2}$, i.e. the following constraint must hold:

$$|v_i - m_j| \leq \frac{PC}{2} \tag{1}$$

The number of buckets, in effect, indirectly controls the transfer/update costs and directly controls the average error. New data values tend to stay close to previous ones and as a result tend to fall in the same bucket, thus not requiring any updates. Therefore, when fewer buckets are used their value intervals are wider, and consequently, fewer updates are needed. However, this causes the average error to rise since more values that are further away from the mean are being approximated by the mean. Similarly, a higher bucket count results in narrower intervals and consequently, cause the average error to go down and NVRR to go up. The goal of the V-Optimal algorithm is to keep the average error between the actual values and their approximations within a certain bound while minimizing the required number of buckets.

Fig. 2 gives a concrete example to further illustrate the process. The training data consists of values between 1 and 18 with various frequencies, e.g. 2 occurred 3 times. The optimal bucket boundaries using at most 3 buckets that minimizes average error for the given training data is shown in the figure along with the corresponding bucket-means. The average error for the training data is 2.05 in this case. If the requirement were to come up with a histogram with the minimum number of buckets that keeps the average error to at most 2.05 then the same histogram would be produced by

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

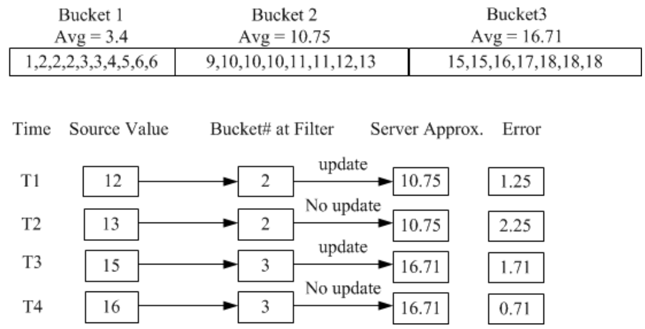


Fig. 2. An example

the V-Optimal algorithm. After the filter is installed at the source, the value 12 is produced at the data source. Since it is the first value, the corresponding bucket number, 2, is sent to the server or stream coordinator. The server uses the bucket-mean, 10.75, as an estimation of the value thus incurring an error of 1.25. The next value at the source is 13, which is still in the same bucket and does not require any update. Over the first four values, the average error is 1.48 and new value refresh rate is 0.5. If the precision constraint was set to 4, then there would be only one precision constraint violation (i.e. error >2) out of 4 instances, resulting in a violation rate of 0.25.

III. Related Work and Background

Precision constraints for aggregate queries were first introduced in [15], [16], where only one-time ad-hoc queries were explored. A smooth trade off between precision and performance was offered in [14] for distributed continuous queries using adaptive filters . They describe a way to maintain accuracy while reducing cost by timely adjusting the width boundaries for continuous queries. However, their weakness is that although filter widths are periodically reset, the quality of the approximations cannot be guaranteed. Reference [11] proposes an algorithm to efficiently dispatch precision windows to reduce the total communication overhead while keeping the quality of answers to the registered aggregate queries. However, they only consider strict precision constraint. In this paper, we allow precision constraints to be violated by specifying the allowable error rate. This flexibility allows us to achieve higher savings in communication overhead. We also have an additional metric, the weighted average error, to maintain the quality of the approximations. Our approach employs the V-Optimal algorithm to compute error-bounded histograms which are then used to communicate approximate answers between data source and server.

As pointed out in [5], histograms were first proposed in Kooi's PhD thesis [12] for approximating data distributions. These were the *equi-width* histograms that divide the value range into buckets of equal width. Then came the *equi-depth* histograms [17]. In [7], it was shown that errors in query size estimates can grow exponentially, in the worst case. This led to the effort in [8] that finds methods to minimize error propagation. This was followed by other optimality results for histograms such as [6], [9]. Reference [9], which is of particular interest to us, showed that *V-Optimal* histograms using a new kind of partition constraint can minimize the error by minimizing the cumulative variance of source parameter values of all the buckets.

Given a set of n numbers $X = x_1, \cdots, x_n$ the *V- Optimal* histogram construction problem searches for a piecewise constant representation (function) H that approximates X with at most B non-overlapping intervals such that $e(H) = ||X - H||_2^2 = \sum_i (x_i - H(i))^2$ is minimized. Each piece is a bucket and defines a subrange $[p, q]$ of the range $[1, n]$. The dual problem, which is called the *error-bounded histogram problem*, searches for H with the minimum number of buckets for which $e(H)$ is less than a certain bound ε. Jagadish et. al. show in [10] that the best representative value v of a bucket or range $[j+1, \cdots, i]$ which minimizes the *Sum Squared Error* $e(j+1, i) = \sum_{r=j+1}^{i} (x_r - v)^2$ is the mean $\frac{\sum_{r=j+1}^{i} x_r}{j-i}$. They gave an $O(n^2 B)$ time algorithm that find the optimum histogram boundaries using $O(nB)$ space. They also gave an alternative approach that reduces the space requirement to $O(n)$ at the expense of increasing the running time to $O(n^2 B^2)$. Their Dynamic Programming algorithm, which is the optimal Polynomial time algorithm for the V-Optimal partition constraint, is given in Algorithm 1. Error $e(j+1, i)$ can be computed in $O(1)$ time from the arrays $Sum[i] = \sum_{r=1}^{i} x_i$ and $SumSquare[i] = \sum_{r=1}^{i} x_i^2$ as $e(j+1, i) = SumSquare[i] - SumSquare[j] - (Sum[i] - Sum[j])^2/(i-j)$

Algorithm 1 V-Optimal

Let$E[i, b]$ = min. error b bucket histogram for $[1, \cdots, i]$.
Initially $E[i, 1] = e(1, i)$ for all $1 \leq i \leq n$
for $b = 2$ to B do **do**
  for $i = 2$ to n do **do**
    $E[i, b] = \infty$
    for $j = i - 1$ downto 1 do **do**
      $E[i, b] = min\{E[i, b], E[j, b-1] + e(j+1, i)\}$
      /* perform book-keeping if minimum is changed */
    end for
  end for
end for

Using this algorithm to solve the error-bounded histogram problem is quite simple. We can run the above algorithm and terminate once we compute an $E[N, k]$ that is at most ε. We can also use the dual approach shown in [10].

IV. Experimental Results

We conducted a series of experiments to assess the effectiveness of our approach and the impact of various parameters on the transfer cost and approximation accuracy at the server. In our experiments, we use the one-dimensional random walk model to generate data at the data source. The random walk model is a well-known and often used model to simulate scalar value changes [14], [15]. In this model, given the step parameter SS and a current scalar value V_i, the next value V_{i+1} is randomly chosen from the interval $[V_i - SS, V_i + SS]$. For our experiments, the lower and upper bounds of the data are respectively 60 and 90, i.e. when $V_i - SS < 60$, it is set to 60. Similarly, if $V_i - SS > 90$, it is set to 90. The data generated during the training period was used to calculate the buckets and their boundaries using the V-Optimal algorithm for the error-bounded histogram problem,

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

based on a given precision constraint, PC. Note that, PC and the error-bound ε used in V-Optimal are not the same. To this end, we introduce a parameter α and compute ε as

$$\varepsilon = \mathrm{PC} * \alpha \qquad (2)$$

$\varepsilon = \mathrm{PC} * \alpha$. Thus, the three parameters that have an impact on performance are PC, SS and α. The three performance metrics that we consider are new value refresh rate (NVRR), weighted average error (WAE) and error rate (ER). NVRR has already been discussed in section 1. WAE is computed as $\frac{\sum_i |v_i - a_i|}{N}$, where a_i is the approximation for v_i and N is the total number of values. Error rate is defined by the fraction of values that cause the precision constraint to be violated.

In the following subsections we present the impact of PC, SS and α on NVRR, WAE and ER.

A. Impact of Step Size

In a real data stream, step size is the maximum difference two consecutive data values can have between them, which is dictated by the underlying phenomenon the data stream represents. E.g. if the data stream is for daily temperature and the samples are being taken every second then step size is going to be very small. On the other hand, if the samples are being taken every hour then the step size will be much larger. In our experiments, the step size is being used by the random walk model for generating the data values. The purpose of step size is to control the variability among consecutive values of the data stream. If the variability in data is to the extent where two consecutive values are always very distant from each other then consecutive values will always fall in different buckets thus nullifying any utility of approximation. In this paper, we are targeting data streams where the underlying variability between consecutive values is much more controlled, although the actual data pattern can be quite unpredictable, e.g. stock prices, number of packets going through routers, highway traffic.

In these experiments, step sizes were varied among $\frac{PC}{2}, \frac{PC}{4}$ and $\frac{PC}{8}$ to vary the data characteristics and α was fixed at 0.13. Fig. 3(a) shows the relationship between NVRR and step size. With step size decreasing, the probability of the new value jumping to the next bucket decreases and consequently, NVRR decreases regardless of the PC value. In all cases, NVRR stays below 0.05, i.e. values have to be communicated less than 5% of the time. Fig. 3(b) shows that the step size does not have any significant impact on WAE. Weighted average error remains the same even when SS is increased. The impact of step size on error rate can be seen in Fig. 3(c). For all precision constraints, as step size is increased the probability of a precision constraint violation or error rate increases. These violations occur when bucket width is larger than the precision constraint. Such violations are unavoidable since the goal of error-bounded V-Optimal algorithm is to minimize the bucket count while keeping the average error below the error bound, ε. While doing so it may choose to keep some of the bucket widths large where the data values are sparse. This allows the algorithm to use more buckets where the data density is higher. This dense area contributes more to the overall weighted average error and thus requiring higher accuracy and narrower bucket widths.

B. Impact of Precision Constraint

Precision constraint, PC, is the maximum width of the error window for any value and is an user defined input. For any pair of (v_i, a_i), where a_i is the approximation of v_i, the constraint in equation 1 must hold. We use the PC to compute ε, which is the error bound for the V-Optimal algorithm, using equation 2. Consequently, PC determines the number of buckets in the V-Optimal histogram. When the value of PC is high the histogram can tolerate high error which in turn reduces the number of buckets. This increases the width of the buckets and as a result, decreases NVRR. This is evident in Fig. 4(a), especially when SS is fixed, i.e. SS $\in \{1, 2, 3, 4, 5\}$. When SS is fixed, variability among consecutive data points stay the same; however, with increasing PC, the buckets become wider, which reduces NVRR. However, when SS increases with increasing PC (i.e. SS $\in \{\frac{PC}{2}, \frac{PC}{4}, \frac{PC}{8}\}$ in Fig. 4(a)), the value of PC does not have much effect on NVRR. Fig. 4(b) shows that PC and WAE have a positive correlation, which is quite intuitive. Increasing PC also reduces error rate as it becomes easier to maintain the PC when it is higher. In any case, our method performs quite well in all scenarios as both NVRR are error rate stayed relatively low. NVRR was below 11% and ER stayed below 1%. For these experiment, α was fixed at 0.13.

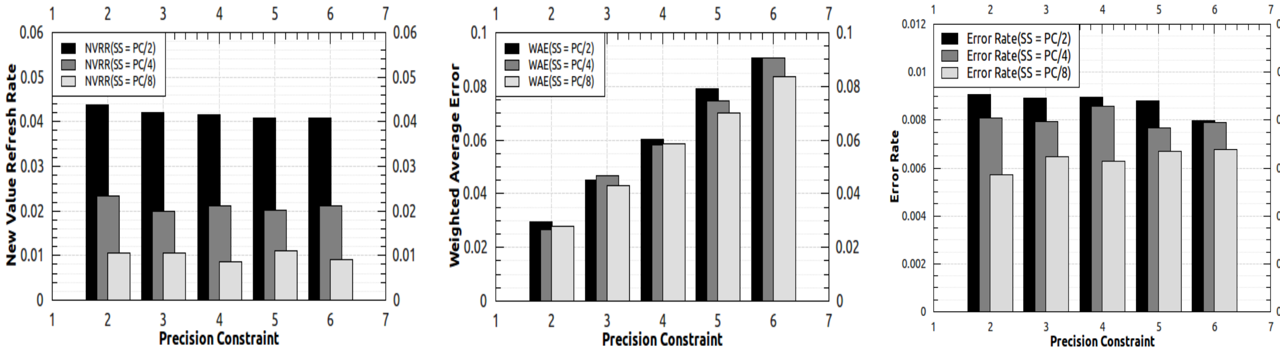
C. Impact of α

For these experiments, we varied α between 0.1 and 0.2 for different precision constraints while step size was fixed at 4. α is the parameter which relates PC to error bound, ε, as given in equation 2. For a given PC, increasing α will increase ε, which in turn will make the V-Optimal algorithm to produce fewer and wider buckets. Wider buckets will increase the WAE and decrease NVRR, which can be seen in figures 5(a) and 5(b). The positive correlation between α and WAE, and the negative correlation between α and NVRR holds for all values of PC. However, we can see that for a fixed α as PC increases WAE increases while NVRR stays unaffected just as expected. This trend was also deduced in the previous subsection. Also, for almost all the data points WAE has a value close to ε which is the goal of V-Optimal algorithm. Fig. 5(c) shows that error rate has a positive correlation with α because when α is increased while keeping PC unchanged, ε becomes larger and as a result the number of buckets that are wider than PC increases. This causes more approximations to violate the precision constraint and error rate to go up.

D. Impact of Training Period

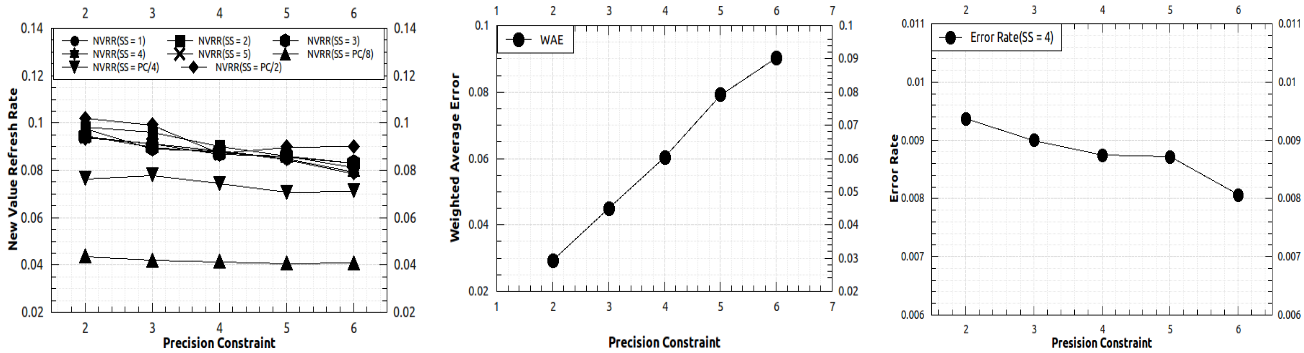
In this set of experiments, our goal was to find out if the size of the training data set has an effect on the performance of our scheme. Step size and α where fixed at 4 and 0.13 as usual. It is quite obvious that the training set has to be large enough to cover the whole data range (in our case 60-90). However, it is not obvious whether the performance will deteriorate if we keep applying the same histogram without retraining. To this end, we increased $\frac{TeVC}{TrVC}$ and noticed that NVRR stays unaffected (Fig. 6), where $TeVC$ and $TrVC$ are respectively, the number of test values and the number of training values. Fig. 7 shows that for the same PC values, WAE stays unchanged when the ratio is increased.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



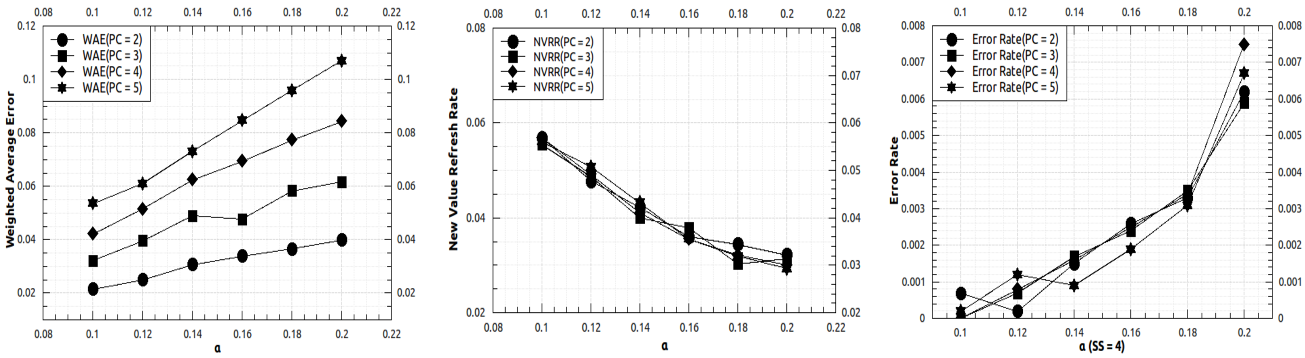
(a) Impact on New Value Refresh Rate (b) Impact on Weighted Average Error (c) Impact on Error Rate

Fig. 3. Impact of temporal variability in data



(a) Impact on New Value Refresh Rate (b) Impact on Weighted Average Error (c) Impact on Error Rate

Fig. 4. Impact of Precision Constraint (PC)



(a) Impact on Weighted Average Error (b) Impact on New Value Refresh Rate (c) Impact on Error Rate

Fig. 5. Impact of Alpha (α)

E. How to choose α

α is a parameter that can be tuned to achieve desired outcome. However, if desired outcome is unrealistic, any fine tuning may be futile. E.g. if PC is too low compared to the underlying step size of the data, then both NVRR and ER will be high. Moreover, we cannot have very low WAE relative to PC because of the positive correlation between them. We can also notice from figures 5(b) and 5(c) that there has to be a trade-off between NVRR and ER due to the inverse correlation between them. Therefore, we cannot desire unrealistically low values for both NVRR and ER. So, the question is how can we fine-tune α to achieve good but realistic performance goals or how do we even set realistic performance goals. Our suggestion is that first we perform statistical analysis (e.g. examine mean, max, variance) on consecutive value differences to determine the underlying SS. Once SS is known, we suggest that PC be set to $SS \times 2$ or higher. If we have a desired WAE upper-bound ε, we can compute the upper bound for α as ε/PC. If we have a desired NVRR upper-bound β and an ER upper bound η, then we can use graphs similar to 5(b) and 5(c) to lower α until we find an α for which NVRR and ER are lower than β and η.

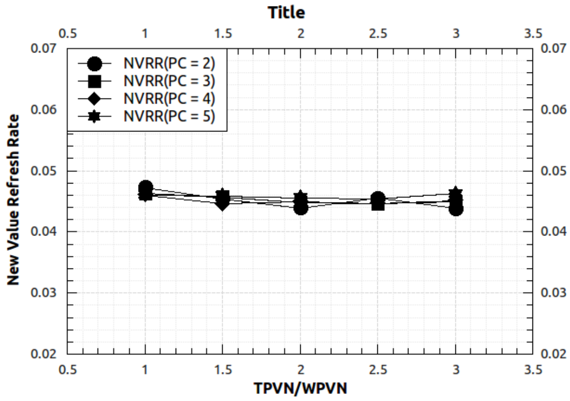


Fig. 6. Effect of $\frac{TeVC}{TrVC}$ on NVRR

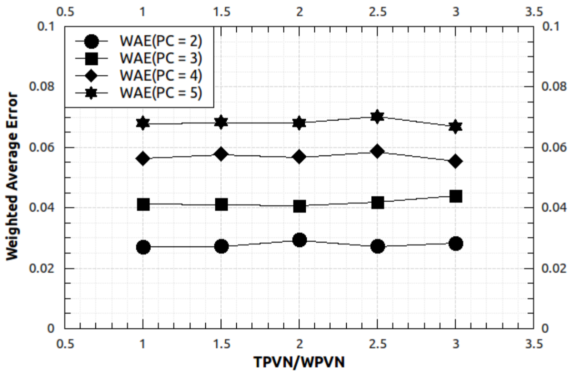


Fig. 7. Effect of $\frac{TeVC}{TrVC}$ on WAE

V. Conclusion

In this paper, we have proposed a novel approach based on V-Optimal histograms for approximating and communicating data values between a source and coordinator. In our approach, a V-Optimal filter is installed at each remote source to filter data that are still in the same histogram bucket as previous values. The remote source and the server are in sync as to what the bucket boundaries and means are. We introduced the notion of soft precision constraints as opposed to strict ones. Experimental results show that our approach can significantly reduce data transfer cost as well as maintain data quality not only in terms of precision constraint but also in terms of weighted average error and PC violation rate.

References

[1] T. F. Abdelzaher, K. G. Shin, and N. Bhatti. Performance guarantees for web server end-systems: A control-theoretical approach. *Ieee Trans. Parallel Distrib. Syst.*, 13(1):80–96, 2002.

[2] Y. Baryshnikov, E. Coffman, G. Pierre, D. Rubenstein, M. Squillante, and T. Yimwadsana. Predictability of web-server traffic congestion. In *Proceedings Of The 10Th International Workshop On Web Content Caching And Distribution*, pages 97–103, Washington, Dc, Usa, 2005. Ieee Computer Society.

[3] J. Carle and D. S. Ryl. Energy-Efficient Area Monitoring for Sensor Networks. *Computer*, 37(2):40–46, 2004.

[4] X. Chen and J. Heidemann. Flash crowd mitigation via adaptive admission control based on application-level observations. *Acm Trans. Interet Technol.*, 5(3):532–569, 2005.

[5] Y. Ioannidis. The history of histograms (abridged). In *Proc. of VLDB Conference*, 2003.

[6] Y. E. Ioannidis. Universality of serial histograms. In *VLDB '93: Proceedings of the 19th International Conference on Very Large Data Bases*, pages 256–267, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[7] Y. E. Ioannidis and S. Christodoulakis. On the propagation of errors in the size of join results. *SIGMOD Rec.*, 20(2):268–277, 1991.

[8] Y. E. Ioannidis and S. Christodoulakis. Optimal histograms for limiting worst-case error propagation in the size of join results. *ACM Trans. Database Syst.*, 18(4):709–748, 1993.

[9] Y. E. Ioannidis and V. Poosala. Balancing histogram optimality and practicality for query result size estimation. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pages 233–244, New York, NY, USA, 1995. ACM.

[10] H. V. Jagadish, N. Koudas, S. Muthukrishnan, V. Poosala, K. C. Sevcik, and T. Suel. Optimal histograms with quality guarantees. In *VLDB '98: Proceedings of the 24rd International Conference on Very Large Data Bases*, pages 275–286, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[11] C. Jin, B. Ding, and J. X. Yu. Making filters smart in distributed data stream environments. *Inf. Sci.*, 179(9):1348–1361, 2009.

[12] R. P. Kooi. *The optimization of queries in relational databases*. PhD thesis, Cleveland, OH, USA, 1980.

[13] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97, New York, NY, USA, 2002. ACM.

[14] C. Olston, J. Jiang, and J. Widom. Adaptive filters for continuous queries over distributed data streams. In *SIGMOD Conference*, pages 563–574. ACM, 2003.

[15] C. Olston, B. T. Loo, and J. Widom. Adaptive precision setting for cached approximate values. In *SIGMOD Conference*, pages 355–366, 2001.

[16] C. Olston and J. Widom. Offering a precision-performance tradeoff for aggregation queries over replicated data. In *VLDB*, pages 144–155. Morgan Kaufmann, 2000.

[17] G. Piatetsky-Shapiro and C. Connell. Accurate estimation of the number of tuples satisfying a condition. In *SIGMOD Conf.*, pages 256–276, 1984.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)