

Building Data Synopses within a Known Maximum Error Bound

Chaoyi Pang, Qing Zhang, David Hansen, and Anthony Maeder

eHealth Research Centre, ICT CSIRO, Australia
{chaoyi.pang; qing.zhang; david.hansen; anthony.maeder}@csiro.au

Abstract. The constructions of Haar wavelet synopses for large data sets have proven to be useful tools for data approximation. Recently, research on constructing wavelet synopses with a guaranteed maximum error has gained attention. Two relevant problems have been proposed: One is the *size bounded* problem that requires the construction of a synopsis of a given size to minimize the maximum error. Another is the *error bounded* problem that requires a minimum sized synopsis be built to satisfy a given error bound. The optimum algorithms for these two problems take $O(N^2)$ time complexity. In this paper, we provide new algorithms for building error-bounded synopses. We first provide several property-based pruning techniques, which can greatly improve the performance of optimal error bounded synopses construction. We then demonstrate the efficiencies and effectiveness of our techniques through extensive experiments.

1 Introduction

Approximate Query Processing (AQP) has been extensively studied and used to deal with massive data sets in decision support and data exploration applications. As AQP usually relies on the pre-computed data synopses to compute the approximate results of queries over original data, research on improving the accuracy of approximate results inevitably focuses on finding good data synopses construction methods. Many techniques have been proposed on constructing data synopses [4, 1]. Among them, the wavelet technique has been considered very promising as it was first adopted by Matias et al. to process range query approximation in relational database [11]. The basic idea of constructing a wavelet synopsis of a data vector, with size N , is to first transform the data vector into a representation with respect to a wavelet basis. Then it is approximated by retaining M coefficients as the wavelet synopsis and setting remains to 0 implicitly. The procedure of choosing M coefficients is called *coefficients thresholding*. A conventional approach is to find M coefficients to minimize the overall mean squared error [13]. This can be easily solved by applying the Parseval's theorem. However, the main drawback of this synopsis is that users have no method in which they can control the approximation error of individual elements in the data vector. This severely impedes further applications of the wavelet approximation. To alleviate this, researches have made efforts on constructing wavelet synopses with error guarantee [2]. Two dual approaches have been taken: one is to construct size bounded synopses which would minimize the maximum approximation error of single data elements [3] whilst the other is to construct the

smallest size of synopses such that the maximum approximation error does not exceed a given error bound [12]. The optimal synopses construction for both approaches has $O(N^2)$ time complexity. Although several methods have been proposed to improve the performance of constructing size bounded synopses [5, 7, 9], there are no investigations in the literature on improving the performance of constructing error bounded synopses. The approximate algorithm for size bounded synopses construction [9] can be easily extended to approximately solve the construction of the error bounded synopses but it may incur large approximation error in some situations as we will illustrate later on. Indeed, there exist some nice features that can greatly improve the performance of error bounded synopses's construction, which are not very obvious applicable on the size bounded synopses's construction. Motivated by this, in this paper we develop fast wavelet synopses construction method, which aims at minimizing synopses size under a given error bound.

Our contributions can be summarized as follows: We have obtained nice features based on the error tree structure used in Haar wavelet transformation. We propose pruning strategies that can greatly improve the performance of the original optimal algorithm. With these properties, we give a nontrivial low bound on the size of an optimal synopsis in linear time.

The rest of the paper is organized as follows. Section 2 defines the problem and enumerates related works. Section 3 investigates the error tree structure and proposes our pruning strategy to improve the original optimal algorithm. Section 4 reports our experiment results on applying our pruning strategy. Section 5 concludes this paper.

2 Background Information

In this section, we first introduce Haar wavelet transformation and coefficients thresholding. Then we present the two types of synopses and review related techniques. In Table 1 we summarize the math notation used throughout the paper.

Symbol	Description
$i \in \{0..N - 1\}$	
$D, [d_0, \dots, d_{N-1}]$	Original data vector
W_D	Haar wavelet transformation on D
c_i	(coefficient) node
d_i, \hat{d}_i	(leaf, data) node and its reconstruction
$path(u)$	All ancestors of node u in the error tree
$T, T(c)$	Error tree and its subtree rooted at c
$T_L(c)/T_R(c)$	The subtree rooted at c 's left/right child
Δ	A given error bound

Table 1. Notations

2.1 Haar Wavelet Transformation and Thresholding

Approximate query processing using Haar wavelet is first introduced in [11] by Matias et al.. The basic idea of Haar wavelet transformation is to recursively find the average and difference of two adjacent data of the data vector D . The final average value, i.e. the average of all data in D , together with those differences form a new vector W_D . The data elements of W_D are named wavelet coefficients. We use the following example to illustrate details. Given $D = [12, 6, 4, 2, 5, 1, 2, 0]$, we transform D to $W_D = [4, 2, 3, 1, 3, 1, 2, 1]$. Figure 1(i) shows details.

Each internal node c_i represents a wavelet coefficient whilst each leaf node d_i represents an original data item. l represents the corresponding resolution listed in Figure 1(i). Given a node u (internal or leaf), we define $path(u)$ as the set of nodes that lie on the path from the root to u (excluding u); $T(u)$ as the subtree rooted at u ; $T_L(u)$ as the subtree rooted at the left child of u , if it exists; and $T_R(u)$ as the subtree rooted at the right child of u , if it exists. To reconstruct any leaf node d_i through the error tree T , we only need to compute the summation of nodes belong to $path(d_i)$. That is, $d_i = \sum_{c_j \in path(d_i)} \delta_{ij} c_j$, where $\delta_{ij} = +1$ if $d_i \in T_L(c_j)$ and $\delta_{ij} = -1$ if $d_i \in T_R(c_j)$. For example, d_2 can be reconstructed through the nodes of $path(d_2)$, i.e. $d_2 = 4 + 2 + (-3) + 1$. It is easy to see that reconstructing any original value of an error tree with N internal nodes, requires $O(\log N)$ time complexity.

The idea of wavelet synopses construction is to only keep a certain number of exact coefficients of W_D , while setting values of the remains as a constant number - zero is the normally implicit one. The goal is to find an optimal synopses that minimize the approximation error under certain metrics. Two commonly adopted error metrics ones are the mean squared error (L_2) and the maximum absolute error (L_∞). More formally, let \hat{d}_i be the approximate value of d_i . Minimizing L_2 is to minimize $\sqrt{\frac{1}{N} \sum_i (\hat{d}_i - d_i)^2}$. Finding an optimal solution to minimize L_2 leads to a simple graceful algorithm due to the energy preserving property of wavelet transformations [13]. However this error metric is arguably not the best choice for approximate query processing [2]. One of the main drawbacks of this error metric is that users have no way of knowing the accuracy of any individual value approximation. Thus techniques on minimizing L_∞ , i.e. $\max_i \{|\hat{d}_i - d_i|\}$, have been developed in recent years.

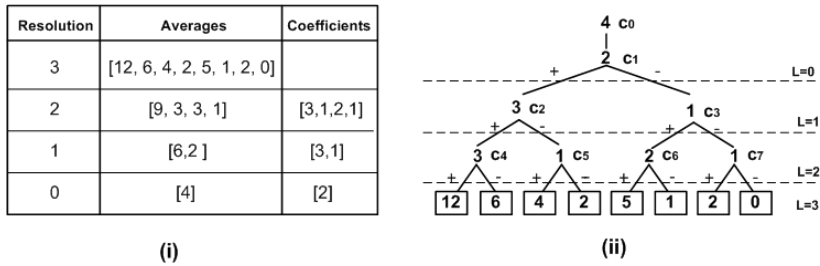


Fig. 1. Haar Wavelet Decomposition and Error Tree

One approach is to minimize L_∞ under a fixed number of coefficients to construct the *size bounded* synopses, also called B -bound. The first solution were proposed in [2]. This probabilistic method, however, has flaws due to its questionable expectation guarantees[8]. In [3], Garofalakis and Kumar propose a deterministic solution for constructing B -bound synopses. Given a data vector D with N elements, their algorithm takes $O(N^2B \log B)$ time complexity and $O(N^2B)$ space complexity. Guha improved the space requirements of this deterministic algorithm to $O(N)$ with a divide and conquer idea in [5]. To make the construction of B -bound synopses applicable in a data stream environment, Karras and Mamoulis propose a greedy algorithm [9]. Meanwhile, Guha extended this problem from the original restricted version to unrestricted version, where the stored set B could be any set of real numbers without being limited to the wavelet coefficients [7, 6]. The details are out of the scope of this paper.

The other approach aims at minimizing the number of necessary coefficients under an error bound (Δ) to construct the *error bounded* synopses. It is also called Δ -bound. Instead of fixing the wavelet synopsis size, it fixes the error tolerance through constructing a synopsis that satisfies $L_\infty < \Delta$. The goal is to find a synopsis with the smallest set of coefficients B among all possible solutions that would satisfy the Δ bound. This model is also very important and promising on providing approximate answers with good quality. Interestingly, it was only mentioned recently by Muthukrishnan and Guha in [12, 5]. They proposed an optimal solution which however takes $O(N^2)$ time complexity.

In the next section, we will provide several important properties that can greatly improve the optimal error bounded synopses construction of [12, 5].

3 Path Traverse Pruning for Synopses Construction

We start this section by first reviewing the existing optimal wavelet synopses construction algorithm, we then introduce our pruning techniques to accelerate the synopses construction for Δ -bounded approximation.

The optimal algorithm for Δ -bounded approximation has been proposed in [9]. Briefly, its idea can be described as follows. Assume there is a subtree $T(v)$ rooted at node v and set S of retained nodes on $path(v)$. Let $B(T(v), \Delta, S)$ denote the least number of retained wavelet coefficients of $T(v)$ that satisfies Δ -bound for the approximation. The algorithm of [9] uses the following two equations to compute $B(T(v), \Delta, S)$: use Equation (1) if node v is to be retained, otherwise, use Equation (2).

$$B(T(v), \Delta, S) = B(T_L(v), \Delta, S \cup v) + B(T_R(v), \Delta, S \cup v) + 1 \quad (1)$$

$$B(T(v), \Delta, S) = B(T_L(v), \Delta, S) + B(T_R(v), \Delta, S) \quad (2)$$

Therefore, the final $B(T(v), \Delta, S)$ is the minimum of the above two possibilities. This method shares the same dynamic programming idea as the one published in [3], where an optimal algorithm of synopses construction for B -bound approximation was proposed.

However, due to special characteristics of the Δ -bounded problem, we can exploit typical Δ related features to improve the performance of the optimal

Algorithm 1 - $\text{minMax}(c_r, v_k, v_d)$

Input:

c_r is the root node of a subtree; v_k is the summation of kept nodes on $\text{path}(c_r)$; v_d is the summation of discarded nodes on $\text{path}(c_r)$

Output:

The optimal set of kept coefficients in T_{c_r} that satisfies Δ bound

Description:

```
1: initialize  $OPT$ , the optimum results set
2: if  $c_r$  is leaf node then
3:   if  $|v_k - c_r| < \Delta$  then
4:      $OPT.bucketNumber = 0$ 
5:   end if
6: else
7:    $OPT.bucketNumber = +\infty$ ; //indicate the retained set no valid
8: end if
9: if  $c_r$  is an inner node then
10:   $L(c_r)(R(c_r)) =$  left (right) child of  $c_r$ 
11:   $L1_{OPT}, L2_{OPT} (R1_{OPT}, R2_{OPT})$ , left (right) subtree's optimum result
12:  //pruning criteria, if not satisfied,  $c_r$  must be kept
13:  if  $|c_r| + |v_d| < \Delta$  then
14:     $L1_{OPT} = \text{minMax}(L(c_r), v_k, v_d + c_r)$ 
15:     $R1_{OPT} = \text{minMax}(R(c_r), v_k, v_d - c_r)$ 
16:  end if
17:   $b_1 =$  left + right //bucket numbers of not keeping  $c_r$ 
18:   $L2_{OPT} = \text{minMax}(L(c_r), v_k + c_r, v_d)$ 
19:   $R2_{OPT} = \text{minMax}(R(c_r), v_k - c_r, v_d)$ 
20:   $b_2 =$  left + right + 1 //bucket numbers of keeping  $c_r$ 
21:  find  $\text{min}(b_1, b_2)$  and combine subtree results to get  $OPT$ , accordingly.
22:  return  $OPT$ 
23: end if
```

algorithm in some cases. For example, in Equation (1) and (2), set S can be constrained to satisfy certain conditions rather than being an arbitrary subset of nodes on $\text{path}(v)$. In the following, we will describe some properties of Δ -bounded synopses which will be used in our algorithm.

Let M_∞ be an optimal Δ -bounded synopsis on error tree T and denote $\text{diff}(d_i)$ as $\sum_{c_j \in \text{path}(d_i) - M_\infty} \delta_{ij} c_j$.

By definition, we know that for any viable solution that satisfies the Δ bound, the summation of deleted nodes along any path (from root to a leaf node) of the error tree is less than Δ . That is, $|\text{diff}(d_i)| < \Delta$ holds for $i = 0, 1, \dots, N - 1$.

For coefficient $c_j \in \text{path}(d_i)$, we define $\text{diff}(c_j)$ to be the summation of deleted ancestor nodes along $\text{path}(d_i)$, which is

$$\text{diff}(c_j) = \sum_{c_k \in \text{path}(c_j) - M_\infty} \delta_{ik} c_k.$$

Based on these formulae, we develop the following three properties:

Property 1. Let T be the error tree on $D = [d_0, d_1, \dots, d_{N-1}]$ and M_∞ be an optimal Δ -bounded synopsis on T . Suppose $c_j \in \text{path}(d_i)$. Then

- (i) $|\text{diff}(c_j)| < \Delta$;
- (ii) $|\text{diff}(c_j)| + |c_j| < \Delta$ if $c_j \notin M_\infty$;
- (iii) For any node $c_k \in T$, $c_k \in M_\infty$ if $|c_k| \geq \Delta$.

Proof. Suppose there are l leaf nodes in $T(c_j)$, ranging from d_h to d_{h+l-1} .

The proof of (i): It is easy to verify that discarding any inner node of $T(c_j)$ will not change the summation of the difference of the l leaf nodes [10]. That is:

$$\sum_{i=h}^{h+l-1} \text{diff}(d_i) = l \times \text{diff}(c_i)$$

Since $|\text{diff}(d_i)| < \Delta$, we have: $l \times |\text{diff}(c_i)| < l \times \Delta$. Thus $|\text{diff}(c_i)| < \Delta$ is proven.

The proof of (ii): From (i), we have

$$|\text{diff}(c_j) + c_j| < \Delta \quad \text{and} \quad |\text{diff}(c_j) - c_j| < \Delta,$$

which implies (ii).

The proof of (iii): A contradiction will be derived from (ii) if $c_j \notin M_\infty$ and $c_j \geq \Delta$ are assumed. From (ii) can be derived straightforwardly from the above formulas.

We propose an optimal algorithm with $\text{minMax}(c_r, v_r, v_d)$ as the key procedure (Algorithm 1). $\text{minMax}(c_r, v_r, v_d)$ has three parameters: c_r is the currently considered node; v_r is the summation of the retained nodes that are on the path from the root node to node c_r (excluding c_r); and v_d is the summation of discarded nodes that are on the path from the root node to node c_r (excluding c_r). The function returns the set of coefficients that represents an optimal Δ -bounded synopsis in the subtree T_{c_r} under the two given values v_r and v_d .

Property 1(i) and (ii) can be used to check the Δ condition dynamically. Property 1(ii) is used at Steps 15-18 of $\text{minMax}(c_r, v_r, v_d)$ to prune unnecessary data expansion: node c_r can not be discarded if $|v_d| + |c_r| \geq \Delta$.

While the time complexity of our property-based algorithm is still of $O(N^2)$ in theory, the extensive experiments, as described in Section 4, indicate that this algorithm is more efficient than the existing algorithms in many situations and no worse in others. Refer to Section 4 for details.

Additionally, Property 1(iii) also gives a lower bound on M_∞ which can be used for a rough estimation on the size of M_∞ . That is,

Corollary 1. *Let T be the error tree on $D = [d_0, d_1, \dots, d_{N-1}]$ and M_∞ be an optimal Δ -bounded synopsis on T . Then*

$$|M_\infty| \geq |\{c_i | (c_i \in T) \wedge (|c_i| \geq \Delta)\}|.$$

Clearly, $\{c_i | (c_i \in T) \wedge (|c_i| \geq \Delta)\}$ can be obtained in $O(|T|)$ time.

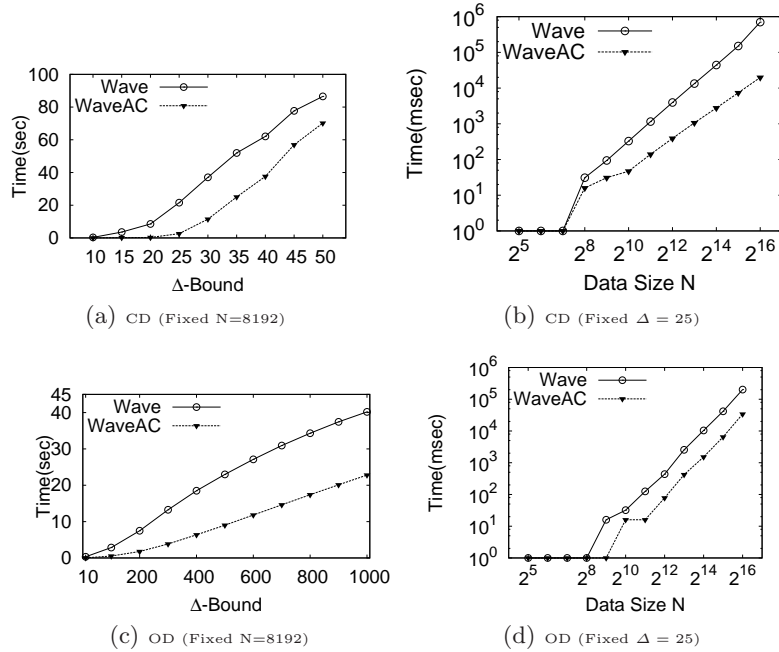


Fig. 2. The comparisons of WaveAC and WaveAC

4 Experimental Evaluation

In this section, we evaluate the effectiveness of our pruning techniques. We implement our algorithms through VC++ .NET. All the experiments were performed on a Pentium IV 3.6GHZ machine with 2 GB memory.

Two types of synthetic data sets are generated for our experiments on constructing Δ -bounded synopses: the coefficient-data set (CD) and the original-data set (OD). The CD data set contains data uniformly selected from $[10, 20]$ as a set of wavelet coefficients (W_D). It is actually an error tree and we can directly construct the Δ -bounded synopses from the CD. The OD data set contains data uniformly selected from $[0, 10000]$ as a vector data (D). It is the original data set and we need to apply the Haar wavelet transformation on it before we can construct the synopses.

We conducted experiments to evaluate the efficiency of the two algorithms in generating optimal Δ -bounded synopses, one with our pruning techniques (named as WaveAC) and one without it (named as Wave), i.e., the original algorithm mentioned at [12]. Their comparisons on computation time are depicted in Figure 2.

In Figure 2(1) and (2), the experimental results were on CD data. Figure 2(1) is on a fixed size CD data ($|D| = 8192$) under varied Δ ranging from 10 to 50. Figure 2(2) is on a fixed Δ ($\Delta = 25$) under varied data size D ranging from 32 to 65536 nodes. The experiments on OD data under the same scenario are given in Figure 2(3) and (4).

From the experiments, we have the following observations. On a fixed size data set, as indicated in Figure 2(1), the pruning technique (WaveAC) can improve the speed up to 25 times faster when Δ is between 10 and 20, which is the range of the values of the coefficients. The increase of speed will drop to 1.5 times as Δ increases. Figure 2(2) shows a comparison of varied data size for a fixed Δ . The improvement caused by the pruning techniques increased the speed to up to 35 times faster. These facts are further supported with the results of Figure 2(3) and (4).

5 Conclusion

In this paper, we have proposed new algorithms on the construction of Δ -bounded synopses to minimize maximum error. Our approach is based on the intrinsic properties of W_D upon a Δ error bound.

Our future work is to improve and extend this work in several ways: to apply the obtained properties in different ways to derive better results; to investigate more properties that can lead more efficient algorithms on construction of Δ -bounded synopses and to support streaming data processing and applications.

References

1. S. Chaudhuri, R. Motwani, and V. Narasayya, *Random sampling for histogram construction: How much is enough?*, ACM SIGMOD'98, pp. 436–447.
2. M. Garofalakis and P. B. Gibbons, *Wavelet synopses with error guarantees*, ACM SIGMOD'02, pp. 476–487.
3. M. Garofalakis and A. Kumar, *Deterministic wavelet thresholding for maximum-error metrics*, ACM PODS'04, pp. 166–176.
4. A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. Strauss, *Optimal and approximate computation of summary statistics for range aggregates*, ACM PODS'01, pp. 227–236.
5. S. Guha, *Space efficiency in synopsis construction algorithms*, VLDB'05, pp. 409–420.
6. S. Guha and B. Harb, *Approximation algorithms for wavelet transform coding of data streams*, SODA, 2006, pp. 698–707.
7. S. Guha and B. Harb, *Wavelet synopsis for data streams: minimizing non-euclidean error*, ACM SIGKDD, 2005, pp. 88–97.
8. S. Guha, K. Shim, and J. Woo, *Rehist: Relative error histogram construction algorithms.*, VLDB'04, pp. 300–311.
9. P. Karras and N. Mamoulis, *One-pass wavelet synopses for maximum-error metrics*, VLDB'05, pp. 421–432.
10. Y. Matias and D. Urieli, *Inner-product based wavelet synopses for range-sum queries.*, ESA, 2006, pp. 504–515.
11. Y. Matias, J. S. Vitter, and M. Wang, *Wavelet-based histograms for selectivity estimation*, ACM SIGMOD'98, pp. 448–459.
12. S. Muthukrishnan, *Subquadratic algorithms for workload-aware haar wavelet synopses.*, FSTTCS, 2005, pp. 285–296.
13. E. J. Stollnitz, T. D. Derose, and D. H. Salesin, *Wavelets for computer graphics: theory and applications*, Morgan Kaufmann Publishers Inc., 1996.