Visualization-aware Timeseries Min-Max Caching with Error Bound Guarantees: Supplementary Material

I. Introduction

This document presents supplementary material supporting our research paper, "Visualization-aware Time series Min-Max Caching with Error Bound Guarantees". Due to space constraints in the main paper, some details were omitted and are included here for comprehensive understanding.

Specifically, we provide detailed algorithms for the error bound calculation, as well as for query evaluation.

For any queries or further information, feel free to reach out to us at the following emails:

stavmars@athenarc.gr(Stavros Maroulis); bstam@athenarc.gr (Vassilis Stamatopoulos); gpapas@athenarc.gr (George Papastefanatos); mter@athenarc.gr (Manolis Terrovitis);

II. Algorithms

A. Upper Error Bound Evaluation

By considering the definition of the error bound for the visualization and the two theorems related to inner- and inter-column errors, we can calculate the maximum error in a line chart visualization of a variable y within a time series T when it is generated using a grouping $\mathcal{G}^k(T)$. This analysis is conducted through an iterative examination of each pixel column, where we determine its potential inner-column and inter-column pixel errors. The details of this process are presented in Algorithm 1.

Initially, we iterate over the groups within $\mathcal{G}^k(T)$ and determine, for each group B_j^k, whether it overlaps with up to two pixel columns (line 4). We also ascertain whether the group is fully-contained within a single pixel column p_i or partially contained across two adjacent columns, p_i and p_{i+1}. In the case of full containment, we consider the min-max y range of the group and update the corresponding inner-column pixel range P_i with values that we can confidently determine (line 3). For partially contained groups, we adjust the pixel ranges P_r^i and P_l^{i+1} to account for the potential foreground pixels contributed by B_j^k to columns p_i and p_{i+1}, respectively (lines 9 - 10).

After establishing both the sets of correctly rendered inner-column pixels and the potential pixels that could be affected by the left and right partially overlapping groups in each pixel column, we proceed to iterate over each pixel column p_i. Within each of these columns, we determine the potential inner-column pixel errors (line 11) and inter-column pixel errors (lines 13 to 18). Specifically concerning the inter-column errors, for each pixel column and at each of its boundaries with neighboring columns, we rasterize the potentially false inter-column lines that would be rendered using $\mathcal{G}^k(T)$ across the $i-j$ intersection (line 16). Additionally, we calculate the maximum set of pixels that may be missing due to the absence of rendering the correct inter-column lines (line 17). By combining these sets of inner and inter-column errors, we can identify the maximum potential errors within the visualization, enabling us to compute the error bound ϵ (line 22).

The algorithm has a computational complexity of $O(k+w)$, since it iterates over k groups in $\mathcal{G}^k(T)$ and w pixel columns in the visualization.

Algorithm 1: Determining the Upper Error Bound in a Line Chart Visualization

Input: $\mathcal{G}^k(T)$, w, h
Output: Error bound ϵ

$E_{max} \leftarrow 0$
for $i \leftarrow 1$ **to** w **do**
  $P_i \leftarrow \emptyset$ // Initialize the foreground pixel range that can be accurately determined by the fully-contained groups in pixel column p_i
for $B_j^k \in \mathcal{G}^k(T)$ **do**
  Determine the leftmost pixel column p_i that group B_j^k spans
  // Check if B_j^k is fully contained in p_i
  if $B_j^k \subseteq B_i^w$ **then**
    $P_i \leftarrow P_i \cup [p_y(B_j^{min}), p_y(B_j^{max})]$
  else
    // For partial containment, set p_i's right partially overlapping group and the left of its immediate right neighbor
    $P_r^i \leftarrow [p_y(B_j^{min}), p_y(B_j^{max})]$
    $P_l^{i+1} \leftarrow [p_y(B_j^{min}), p_y(B_j^{max})]$
for $i \leftarrow 1$ **to** w **do**
  // Calculate inner-column errors for column p_i
  $E_{inner}^i \leftarrow (P_l^i \cup P_r^i) \setminus P_i$
  // Initialize set of potential inter-column errors
  $E_{inter}^i \leftarrow \emptyset$
  // Calculate inter-column errors for column p_i
  for *each neighbor* $p_j \in \{p_{i-1}, p_{i+1}\}$ **do**
    if p_j *exists* **then**
      Rasterize the line segment between the min or max values of $\mathcal{G}^k(T)$ before and after the intersection between the $i-j$ columns to obtain the set $F_{i,j}$
      Determine the maximum set of pixels that the actual missing correct line across $i-j$ would render to obtain the set $M_{i,j}$
      $E_{inter}^i \leftarrow E_{inter}^i \cup F_{i,j} \cup M_{i,j}$
  // Combine inner- and inter-column errors for column p_i
  $E_{inter}^i \leftarrow E_{inter}^i \setminus P_i$
  $E^i \leftarrow E_{inner}^i \cup E_{inter}^i$
  $E_{max} \leftarrow E_{max} + |E^i|$
$\epsilon \leftarrow \frac{E_{max}}{w \times h}$
return ϵ

B. Query Processing

Next, we provide the detailed query processing algorithm of MinMaxCache:

Algorithm 2: Query Evaluation in MinMaxCache

Input: $Q = (I_Q, Y_Q, w, h)$: visualization query; ϵ: error bound
Output: R: query results

Initialize w pixel columns, each maintaining left and right overlapped groups and fully contained min-max range for every $y \in Y_Q$
foreach $y \in Y_Q$ **do**
    Identify the interval tree IT_y based on $\mathcal{T}_{id}$ and y
    if IT_y *does not exist* **then**
        Initialize a new interval tree IT_y;
    Find groupings $G^y_{overlapping}$ in IT_y that overlap with I_Q;
if $\forall y \in Y_Q, G^y_{overlapping} = \emptyset$ **then**
    `// Complete cache miss`
    Fetch data from database for the entire interval I_Q using the initial default value for AF
else
    foreach $y \in Y_Q$ **do**
        foreach $G \in G^y_{overlapping}$ **do**
            Update pixel columns based on G
        Calculate partial error ϵ'_y based on updated pixel columns for y
        Compute a weighted mean aggregation factor AF_y, based on the coverage percentage of each grouping within I_Q
        $I^y_{missing} \leftarrow$ intervals in I_Q not covered by $G^y_{overlapping}$
        if $\epsilon'_y \leq \epsilon$ **then**
            if $I^y_{missing} \neq \emptyset$ **then**
                `// Partial cache hit for variable` y
                Fetch $I^y_{missing}$ from the database with AF_y
        else
            `// Cache miss for variable` y `because error exceeds bounds`
            Fetch data from database for the entire interval I_Q for y with $AF'_y = 2 \times AF_y$ or use raw data if $\frac{\tau_{agg}}{\tau_s} < 4$
if *data was fetched from the database* **then**
    foreach $y \in Y_Q$ **do**
        Update pixel columns and IT_y with new groupings from fetched data for y
        Reevaluate ϵ'_y
if $\exists y \in Y_Q, \epsilon'_y > \epsilon$ **then**
    Issue an M4 query for I_Q for all $y \in Y_Q$ with $\epsilon'_y > \epsilon$
    return *M4 query results for visualization*
else
    Prepare R as a set of points containing the min-max values and the ones with the first and last timestamps from each pixel column for each variable $y \in Y_Q$
    return R