A COMPLEXITY ANALYSIS

Next, we analyze the time and space complexity of the eBUG algorithm. Its time complexity ranges between log-linear and quadratic, while its space complexity is linear.

Time Complexity. We first analyze the time complexity of the eBUG sampling (Algorithm 3), which mainly consists of two parts: the complexity of significance calculation and the complexity of maintaining the min-heap structure. Then, we provide the time complexity of the eBUG precomputation (Algorithm 4).

Given the input data size as n, and the target number of sampled points as m, eBUG requires a bottom-up cumulative elimination of $n-m$ points. Let k ($1 \leq k \leq n-m$) denote the number of points eliminated in each round. In each round, the eliminated points are logically removed through marking, requiring no adjustment to the heap structure, resulting in a time complexity of $O(1)$. When updating the significance of the left (or right) neighboring points, the total time complexity complexity$_{Sig}$ for calculating point significance across all rounds is a piecewise function of the algorithm parameter e, as shown below.

$$
\begin{aligned}
\text{complexity}_{\text{Sig}} &= \begin{cases} 3(n-m), \text{ if } e=0, \\ 2\left(\sum_{k=1}^{e}(k+3)+\sum_{e+1}^{n-m}(e+3)\right), \text{ if } 1 \leq e < n-m, \\ 2\sum_{k=1}^{n-m}(k+3), \text{ if } e \geq n-m. \end{cases} \\
&= \begin{cases} 3(n-m), \text{ if } e=0, \\ -e^2+e+2(n-m)(e+3), \text{ if } 1 \leq e < n-m, \\ (n-m)^2+7(n-m), \text{ if } e \geq n-m. \end{cases}
\end{aligned}
$$

The above result is because (1) When $e=0$, each round only requires calculating the area of a triangle, involving computations for three points. (2) When $1 \leq e < n-m$, the range of k is divided into two parts: $[1, e]$ and $[e+1, n-m]$. For the former, since $k \leq e$, each round involves computations for at most $k+3$ points; for the latter, since $k > e$, each round involves computations for at most $e+3$ points. (3) When $e \geq n-m$, the range of k is $[1, n-m]$, and since $k \leq e$, each round involves computations for at most $k+3$ points. Finally, when $e \geq 1$, calculating the areal difference requires finding the intersection point and computing the polygon area, involving two traversal processes. Thus, the time complexity is multiplied by a constant factor of 2.

The other steps mainly involve reinserting the updated points into the min-heap and maintaining the heap structure, with this part of the total time complexity being $O((n-m)\log n)$. In summary, the time complexity of the eBUG algorithm for sampling m points from n points is $O((n-m)\cdot(\min(e, n-m)+\log n))$. It can be observed that the algorithm complexity is primarily determined by the algorithm parameter e, where smaller values of e keep the algorithm within the log-linear complexity range, while larger values of e cause the algorithm to degrade to quadratic complexity. Additionally, since the algorithm employs a bottom-up elimination strategy, the larger the number of sampled points m, the fewer the number of points that need to be eliminated (i.e., $n-m$), resulting in higher algorithm efficiency.

In the precomputation mode, it is equivalent to sampling $m=2$ points, meaning eBUG eliminates $n-2$ points from the bottom up until only the global start and end points remain. In this case, the time complexity is $O(n \cdot (\min(e, n)+\log n))$, which simplifies to $O(n(e+\log n))$ when e is small, and degrades to $O(n^2)$ when e is large.

Space Complexity. During the online sampling or precomputation process of eBUG, for each point, we maintain information such as its significance and pointers to its two adjacent non-eliminated points. Therefore, the space complexity is $O(n)$. After the precomputation process is completed, the space complexity required to store the precomputed time series is also $O(n)$.