
Algorithm 1 FilteredVamana Indexing Algorithm

Require: Database P with n points where the i -th point has coordinates x_i , parameters α, L, R .

Ensure: Directed graph G over P with out-degree $\leq R$.

- 1: Initialize G to an empty graph
 - 2: Let s denote the medoid of P
 - 3: Let $\text{st}(f)$ denote the start node for filter label f for every $f \in F$
 - 4: Let σ be a random permutation of $[n]$
 - 5: Let F_x be the label-set for every $x \in P$
 - 6: **for** $i \in [n]$ **do**
 - 7: Let $S_{F_{x_{\sigma(i)}}} = \{\text{st}(f) : f \in F_{x_{\sigma(i)}}\}$
 - 8: Let $[\emptyset, \mathcal{V}_{F_{x_{\sigma(i)}}}] \leftarrow \text{FilteredGreedySearch}(S_{F_{x_{\sigma(i)}}}, x_{\sigma(i)}, 0, L, F_{x_{\sigma(i)}})$
 - 9: Let $[\emptyset, \mathcal{V}] \leftarrow \text{GreedySearch}(s, x_{\sigma(i)}, 0, L)$
 - 10: $\mathcal{V} \leftarrow \mathcal{V} \cup \mathcal{V}_{F_{x_{\sigma(i)}}}$
 - 11: Run $\text{FilteredRobustPrune}(\sigma(i), \mathcal{V}, \alpha, R)$ to update out-neighbors of $\sigma(i)$
 - 12: **for** $j \in N_{\text{out}}(\sigma(i))$ **do**
 - 13: Update $N_{\text{out}}(j) \leftarrow N_{\text{out}}(j) \cup \{\sigma(i)\}$
 - 14: **if** $|N_{\text{out}}(j)| > R$ **then**
 - 15: Run $\text{FilteredRobustPrune}(j, N_{\text{out}}(j), \alpha, R)$ to update out-neighbors of j
-

Algorithm 2 FilteredVamana Indexing Algorithm - Production Stage

Require: Database P with n points where the i -th point has coordinates x_i , parameters α, L, R .

Ensure: Directed graph G over P with out-degree $\leq R$.

- 1: $G \leftarrow$ FilteredVamana Graph obtained from Training Phase
 - 2: Let s denote the medoid of P
 - 3: Let $\text{st}(f)$ denote the start node for filter label f for every $f \in F$
 - 4: Let σ be a random permutation of $[n]$
 - 5: Let F_x be the label-set for every $x \in P$
 - 6: **for** $i \in [n]$ **do**
 - 7: Let $S = \{\text{st}(\cdot) \cup s\}$
 - 8: Let $[\emptyset, \mathcal{V}] \leftarrow \text{GreedySearch}(S, x_{\sigma(i)}, 0, L)$
 - 9: Run $\text{RobustPrune}(\sigma(i), \mathcal{V}, \alpha, R)$ to update out-neighbors of $\sigma(i)$
 - 10: **for** $j \in N_{\text{out}}(\sigma(i))$ **do**
 - 11: Update $N_{\text{out}}(j) \leftarrow N_{\text{out}}(j) \cup \{\sigma(i)\}$
 - 12: **if** $|N_{\text{out}}(j)| > R$ **then**
 - 13: Run $\text{RobustPrune}(j, N_{\text{out}}(j), \alpha, R)$ to update out-neighbors of j
-