Range generation algorithm

Aircloak

October 10, 2013

This document gives an overview of the algorithm used to generate the reported ranges for a given property label/string pair. The basic idea is a sweep-line algorithm where we sweep over the different value points.

Input: List of pair $((v_1, c_1), (v_2, c_2), \ldots, (v_n, c_n))$ where v_i represents the value and c_i the amount of users that reported this value. The sequence has to be sorted, such that $\forall i = 1, 2, \ldots, n-1 : v_i < v_{i+1}$.

Output: List of ranges where each range is given in the form $(\min_i, \max_i, c_i, c_i')$ which represents the interval $[\min_i, \max_i)$ containing c_i users. c_i' is the noisy count in the interval, thus applying δ_{T_1} (see the TÜV document describing the anonymization function).

Steps:

- Find the minimum and maximum values $\min, \max$ of all values $v_1, v_2, \ldots, v_n$. These are $\min = v_1$ and $\max = v_n$.
- Find the smallest $k \in \mathbb{N}$ such that $[\min, \max] \subseteq [-2^k, 2^k)$.
- Generate the initial set of working ranges. This is stored as a list of the form

  $$((2^0, -2^k, -2^k + 2^0, 0), (2^1, -2^k, -2^k + 2^1, 0), \ldots, (2^(k+1), -2^k, 2^k, 0)).$$

  Each element is represented by its size, the range's lower and upper bounds, and the currently known number of users belonging to that range.
- Loop through the values from 1 to n $(i = 1, 2, \ldots, n)$. Each value is an event for the sweep-line algorithm. For each event go through the list of working ranges beginning with the smallest range. We have the following kind of events for each range $(s_r, \min_r, \max_r, C_r)$:
  1. $v_i < \min_r$: do nothing. The value is in front of the range.
  2. $\min_r \leq v_i < \max_r$: update $C_r \leftarrow C_r + c_i$ (the value is in the range, so update their values).
  3. $\max_r \leq v_i$: no new value/count pair may be found that is in that range. We can check if we want to report that range. Test if $\delta_{T_1}(C_r) > T_1$.
     - *yes*: report $(\min_r, \max_r, C_r, \delta_{T_1}(C_r))$ and move this range to the next position like in the "*no*"-case. Update all working ranges including the range $[\min_r, \max_r)$ (before the position update) to a new position as in the "*no*"-case. These working ranges have to be moved at least by one position because they include the reported range.
     - *no*: move the range by $l \cdot s_r$ $(l \in \mathbb{N})$ positions to the right, such that v_i is in that range: $\min_r \leftarrow \min_r + l \cdot s_r$, $\max_r \leftarrow \max_r + l$, $\min_r \leq v_i < \max_r$

The order of the working ranges is important. We need to process the small ranges first as we want to report the smallest possible ranges with the corresponding property $(\delta_{T_1}(C_r) > T_1)$.

Optimization: If a working range's start is greater than the maximum value v_n, we can remove that working range from the list as we will never have an event of type 2 or 3.

Conjecture: If a range r of size s_r is moved such that $v_n < \min_r$ (past the maximum value), then for all ranges r' with $s_{r'}$

$$v_n < \min_{r'} .$$

This would allow a further optimization.