APPENDIX

A PROOFS

Lemma 1 *(Correctness) The* Group-Coverage *algorithm successfully identifies if a group* $\mathbf{g}$ *is covered or not, i.e., if there are at least* τ *instances of* $\mathbf{g}$ *in the data set* $\mathcal{D}$.

Proof. Each set query with a yes answer contains at least one object belonging to the group $\mathbf{g}$. Using this, the algorithm maintains a lower bound cnt on the $|\mathbf{g}|$ in $\mathcal{D}$. That is, $cnt \leq |\mathbf{g}|$. The algorithm returns `true` when $cnt = \tau$. When $cnt = \tau$, $\mathbf{g}$ is covered, because $\tau = cnt \leq |\mathbf{g}|$. The algorithm returns false when the queue is empty and $cnt < \tau$. For sets with yes answers, the algorithm divides the set in two halves, unless the set size is 1. As a result, when the queue, all the set questions with yes answers have had size 1, otherwise the queue would have not empty. Therefore, $|\mathbf{g}| = cnt < \tau$, meaning that $\mathbf{g}$ is uncovered. □

B PSEUDO-CODES

Algorithm 5 Partition & Label

function Partition($\mathcal{D}, \mathcal{G}, n$)
  Let Q = an empty queue
  for $i \leftarrow 0$ to N with step size n **do**:
    $t \leftarrow \{t_i, \cdots, t_j\}$
    $Q.add(t)$
  Let S = an empty set
  while Q is not empty **do**
    $T \leftarrow Q.del\_top()$
    $(i, j) \leftarrow (T.b\_index, T.e\_index)$
    ans ←AskQuestion($\{t_i, \cdots, t_j\}, \mathbf{g}'$)
    if ans=no **then**
      S.add($\{t_i, \cdots, t_j\}$)
    else
      if $j > i$ `/*if setsize>1*/` **then**
        $T_1 \leftarrow \{t_i, t_{\lfloor \frac{i+j}{2} \rfloor}\}$
        $T_2 \leftarrow (t_{\lfloor \frac{i+j}{2} \rfloor + 1}, t_j)$
        $Q.add(T_1)$; $Q.add(T_2)$
  return S
function Label($\mathcal{D}, \mathcal{G}, \tau$)
  $cnt \leftarrow 0$
  for $t \in \mathcal{G}$ **do**
    $l \leftarrow$PointQuery(t)
    if $l \neq \mathbf{g}$ **then** $\mathcal{G}$.Remove(t)
    else $cnt \leftarrow cnt + 1$
    if $cnt \geq \tau$ **then**
      break
  return $\mathcal{G}$

Algorithm 6 Label Samples & Aggregate

function LabelSamples($\mathcal{D}, \tau, c = 2$)
  for $c\tau$ random samples t from $\mathcal{D}$ **do**
    $l \leftarrow$PointQuery(t)
    $\mathcal{L}.add(\langle t, l \rangle)$; $\mathcal{D}.remove(t)$
  return $\mathcal{D}$, $\mathcal{L}$
function Aggregate($\mathcal{L}, N, \tau, \mathbb{G}, mult = false$)
  sort $\mathbb{G}$ based on $\mathcal{L}$.count($\mathbf{g}$), $\mathbf{g} \in \mathbb{G}$, ascending
  $sum \leftarrow 0$; $\mathcal{G} \leftarrow \{\}$
  for $\mathbf{g} \in \mathbb{G}$ **do**
    $E_{\mathbf{g}} \leftarrow \frac{\mathcal{L}.\text{count}(\mathbf{g})}{|\mathcal{L}|} \times N$
    if: $mult$ = true **then** $\mathcal{G} \leftarrow \mathcal{G} \mid \mathcal{G}$.parent= $\mathbf{g}$.parent
    if:$sum + E_{\mathbf{g}} < \tau$ **then** $\mathcal{G}$.add($\mathbf{g}$); $sum \leftarrow sum + E_{\mathbf{g}}$
    else: $\mathbb{G}_{agg}$.add($\mathcal{G}$) ; $\mathcal{G} \leftarrow \{\mathbf{g}\}$; $sum \leftarrow E_{\mathbf{g}}$
  return $\mathbb{G}_{agg}$.add($\mathcal{G}$)

Algorithm 7 Base-Coverage($\mathcal{D}, \tau, \mathbf{g}$)

Input: Dataset $\mathcal{D}$, coverage threshold τ, and target group $\mathbf{g}$
Output: Coverage of group $\mathbf{g}$
$cnt \leftarrow 0$
for $\forall t \in \mathcal{D}$ **do**
  ans ←AskQuestion($t, \mathbf{g}$)
  if ans =**true then** $cnt \leftarrow cnt + 1$
  if $cnt = \tau$ **then return true** `// covered`
return false `//uncovered`