Online negotiation for privacy preserving data publishing

A. Pilalidou, P. Vassiliadis
Dept. of Computer Science
Univ. Ioannina
Roadmap

• Introduction
• Study of the relationship between suppression, generalization and privacy
• User-time anonymization with an exhaustive, off-line pre-processing
• User-time anonymization with user-time pre-processing
• Conclusions
Roadmap

• **Introduction**
  – Background, motivation & terminology
  – Problem definition & outline of approach

• Study of the relationship between suppression, generalization and privacy

• User-time anonymization with an exhaustive, off-line pre-processing

• User-time anonymization with user-time pre-processing

• Conclusions
Problem in the real world

- **Organizations** (hospitals, ministries, internet providers, ...) *publicly release data* concerning individual records (internet searches, medical records, ...)

- Although data were stripped from identity-revealing attributes, it is still *possible to identify individuals* via various forms of attacks

<table>
<thead>
<tr>
<th>Voter Registration Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Andre</td>
</tr>
<tr>
<td>Beth</td>
</tr>
<tr>
<td>Carol</td>
</tr>
<tr>
<td>Dan</td>
</tr>
<tr>
<td>Ellen</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hospital Patient Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birthdate</td>
</tr>
<tr>
<td>1/21/76</td>
</tr>
<tr>
<td>4/13/86</td>
</tr>
<tr>
<td>2/28/76</td>
</tr>
<tr>
<td>1/21/76</td>
</tr>
<tr>
<td>4/13/86</td>
</tr>
<tr>
<td>2/28/76</td>
</tr>
</tbody>
</table>
The context of privacy-preserving data publishing

Deborah, a star DBA & a TRUSTED data publisher

Bob (the victim) to be hidden

Ben, the benevolent, data miner

Alice, the external attacker
Anonymization

• To retain privacy one must:
  – Remove the attributes that directly identify individuals (name, SSN, ...)
  – Organize the tuples and the cell values of the data set in such a way that:
    • The statistical properties of the data set are retained
    • The attacker cannot guess to which individual a tuple corresponds with statistical meaningful guarantee
**Fundamentals**

- **Identifier(s):** attribute(s) that explicitly reveal the identity of a person (name, SSN, ...). These attributes are removed from the public data set.

- **Quasi identifier:** attribute(s) that if joined with external data can reveal sensitive information (zip code, birth date, sex,...)
  - Typically accompanied by “generalization hierarchies”

- **Sensitive attribute:** containing the values that should be kept private (disease, salary,...)
General methods for Anonymization

• “Hide tuples in the crowd”
  – Generalization
  – Anatomization

• “Lies to the attacker, truth to the statistician”
  – Noise injection
  – Value perturbation
A relation $T$ is **$k$-anonymous** when every tuple of the relation is identical to $k-1$ other tuples with respect to their **Quasi-Identifier** set of attributes.

### $k$-anonymity

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Work_class</th>
<th>Education</th>
<th>Hours/week</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thales</td>
<td>39</td>
<td>Private</td>
<td>Hs-grad</td>
<td>40</td>
</tr>
<tr>
<td>Anaximander</td>
<td>38</td>
<td>Private</td>
<td>Hs-grad</td>
<td>50</td>
</tr>
<tr>
<td>Anaximenes</td>
<td>37</td>
<td>Private</td>
<td>Hs-grad</td>
<td>40</td>
</tr>
<tr>
<td>Pythagoras</td>
<td>38</td>
<td>Private</td>
<td>11th</td>
<td>45</td>
</tr>
<tr>
<td>Gorgias</td>
<td>28</td>
<td>Loc-gov</td>
<td>Bachelors</td>
<td>30</td>
</tr>
<tr>
<td>Heraclitus</td>
<td>31</td>
<td>Federal-gov</td>
<td>Master</td>
<td>50</td>
</tr>
<tr>
<td>Empedocles</td>
<td>30</td>
<td>State-gov</td>
<td>Bachelors</td>
<td>60</td>
</tr>
<tr>
<td>Leucippus</td>
<td>32</td>
<td>Self-emp-not-inc</td>
<td>Bachelors</td>
<td>50</td>
</tr>
<tr>
<td>Democritus</td>
<td>35</td>
<td>Self-emp-inc</td>
<td>Prof-school</td>
<td>54</td>
</tr>
<tr>
<td>Protagoras</td>
<td>33</td>
<td>Self-emp-inc</td>
<td>Assoc-acd</td>
<td>40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Age</th>
<th>Work_class</th>
<th>Education</th>
<th>Hours/week</th>
</tr>
</thead>
<tbody>
<tr>
<td>37-41</td>
<td>Private</td>
<td>Without-post-secondary</td>
<td>40</td>
</tr>
<tr>
<td>37-41</td>
<td>Private</td>
<td>Without-post-secondary</td>
<td>50</td>
</tr>
<tr>
<td>37-41</td>
<td>Private</td>
<td>Without-post-secondary</td>
<td>40</td>
</tr>
<tr>
<td>37-41</td>
<td>Private</td>
<td>Without-post-secondary</td>
<td>45</td>
</tr>
<tr>
<td>27-31</td>
<td>Gov</td>
<td>Post-secondary</td>
<td>30</td>
</tr>
<tr>
<td>27-31</td>
<td>Gov</td>
<td>Post-secondary</td>
<td>50</td>
</tr>
<tr>
<td>27-31</td>
<td>Gov</td>
<td>Post-secondary</td>
<td>60</td>
</tr>
<tr>
<td>32-36</td>
<td>Self-emp</td>
<td>Post-secondary</td>
<td>50</td>
</tr>
<tr>
<td>32-36</td>
<td>Self-emp</td>
<td>Post-secondary</td>
<td>54</td>
</tr>
<tr>
<td>32-36</td>
<td>Self-emp</td>
<td>Post-secondary</td>
<td>40</td>
</tr>
</tbody>
</table>
A relation $T$ satisfies the
naïve $l$-diversity
property whenever every group of the
relation contains at least $l$ different values
in its sensitive attributes.
Information utility

• Must prevent the attackers, by satisfying the privacy criterion (k for k-anonymity, l for l-diversity)
  – Fundamental anonymization technique: hide individual in groups of identical QI values!!

• Must serve the well-meaning users, by maximizing information utility i.e., by minimizing
  • The tuples we remove (see next)
  • the amount of generalization that we apply to the QI attributes.
This anonymization suppressed no tuples, and guarantees 3-anonymity.

What if we want 4-anonymity?
Generalization vs suppression

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Work_class</th>
<th>Education</th>
<th>Hours/week</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thales</td>
<td>37-41</td>
<td>Private</td>
<td>Without-post-secondary</td>
<td>40</td>
</tr>
<tr>
<td>Anaximander</td>
<td>37-41</td>
<td>Private</td>
<td>Without-post-secondary</td>
<td>50</td>
</tr>
<tr>
<td>Anaximenes</td>
<td>37-41</td>
<td>Private</td>
<td>Without-post-secondary</td>
<td>40</td>
</tr>
<tr>
<td>Pythagoras</td>
<td>37-41</td>
<td>Private</td>
<td>Without-post-secondary</td>
<td>45</td>
</tr>
</tbody>
</table>

Low height, 6 tuples suppressed

Higher height, no tuples suppressed

//the difference is in the work_class field
Problem parameters

• The problem has 3 important parameters...
  – **Generalization**: how much information is lost by generalizing the data to a certain level of generalization
  – **Suppression**: how many tuples are moved from the data set during anonymization, in order to quarantine outliers that will drive generalization to large heights
  – **Anonymity**: what is the minimum tolerable value for the privacy criterion – e.g., minimum tolerable k group size or minimum tolerable l for distinct sensitive values in any group
• ... which are antagonistic to the amount of useful information I present to the well-meaning end-users
State-of-the-art

• All the related bibliography is based on the assumption that we have plenty of off-line time to process the data set

• The emphasis has been placed
  – To different privacy criteria and the corresponding attacks they prevent
  – To fast algorithms for exact solutions to the problem of optimal anonymization (wrt to a utility function)
    • Still: not fast enough for user-time (in the order of minutes / hours / ...)

17
Research questions

• Can we help the data curator *negotiate* different configurations of privacy, generalization and suppression and decide what is best without resorting to some non-intuitive utility function?
  – e.g., by paying the price for less privacy (lower k) to attain a better value of suppression (less removed tuples) and, thus, higher information utility?

• *Can the system* guide the search by *suggesting alternatives* – esp., when tested configurations are impossible to attain?

• Can we do it in *user time*?
Our first research quest is to study the relationship of the three parameters suppression, generalization and privacy criterion (which, strangely, has not been studied in the past).

– Our findings suggest that the problem is valid and worth exploring

In contrast to the related literature, we present a mechanism to answer anonymization requests (expressed over the above 3 parameters) in user time.

If the user request cannot be satisfied by the data set, we suggest approximations to the user that are close to his original request.

The combination of user time and guidance via approximate answers allows the user to negotiate in user time the anonymization scheme for the published data set.
Preview of our solution

• We pre-compute off-line
  – All the possible combinations of levels for the QI attributes – organized in a lattice of anonymization schemes
  – The suppression histogram of each such combination (for a specific privacy criterion) – i.e., for every combination we know the amount of tuples that have to be suppressed for a specific value of the privacy criterion
• The user specifies a request with 3 parameters as constraints (max height per hierarchy, max tolerable suppression, min tolerable $k$ or $l$).
  – If a solution for this value combination exists
    • Among all the solutions that satisfy the request, we present the solution that is located at the lowest generalization height
  – If no such solution exists
    • we provide the user with 3 suggestions (i.e., approximate answers), each relaxing one of the 3 abovementioned constraints
Roadmap

• Introduction
• Study of the relationship between suppression, generalization and privacy
• User-time anonymization with an exhaustive, off-line pre-processing
• User-time anonymization with user-time pre-processing
• Conclusions
Problem Validity

• Is the problem valid in the first place? We look for answers to the following research questions:
  – Is the amount of suppressed (removed) tuples significant in all / some generalization levels?
    • If not, then the problem is not worth researching
  – How are the privacy criterion and the generalization height related to the amount of suppression?
    • Strangely, nobody had reported any findings in the past
**Employed Data Sets**

- **Adult** from the UC Irvine Machine learning repository with 30,162 tuples

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Distinct Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>72</td>
</tr>
<tr>
<td>Gender</td>
<td>2</td>
</tr>
<tr>
<td>Race</td>
<td>2</td>
</tr>
<tr>
<td>Marital Status</td>
<td>7</td>
</tr>
<tr>
<td>Education</td>
<td>16</td>
</tr>
<tr>
<td>Native country</td>
<td>41</td>
</tr>
<tr>
<td>Work Class</td>
<td>7</td>
</tr>
<tr>
<td>Occupation</td>
<td>14</td>
</tr>
<tr>
<td>Hours per week</td>
<td>94</td>
</tr>
<tr>
<td>Salary</td>
<td>2</td>
</tr>
</tbody>
</table>

- **IPUMS** with 600,000 tuples

<table>
<thead>
<tr>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
</tr>
<tr>
<td>Birthplace</td>
</tr>
<tr>
<td>Education</td>
</tr>
<tr>
<td>Gender</td>
</tr>
<tr>
<td>Occupation</td>
</tr>
</tbody>
</table>
Generalization Hierarchies

Race.
- White
- Colored
  - Asian – Pac-Islander
  - Amer - Eskimo
  - Black
  - Other

Marital.
- Married
  - Partner Present
  - Partner absent
    - Divorced
    - Widowed
    - Separated
    - Spouse absent

Workclass.
- With pay
  - Private
    - not-inc
    - inc
  - Self-emp
  - Gov
    - Federal
    - Local
    - State

- W/O pay

L3
L2
L1
L0
First Steps

• We construct the **lattice of all the possible combinations**, for all the hierarchy levels of all the quasi-identifier attributes.
  - Every such **combination** (aka anonymization scheme) is a **node** in the lattice
  - A node $v_{\text{child}}$ has an **edge** to a node $v_{\text{father}}$ if
    • (a) for all but one of the quasi-identifier attributes, both nodes have the same value, and,
    • (b) there exists exactly one attribute where $v_{\text{father}}$ is exactly one level higher than $v_{\text{child}}$
• For every lattice node, we construct the **suppression histogram** for a given criterion (anonymity or diversity).
Lattice & induced lattice

QI=3 – full lattice
(Age, Race, Work_class)

QI=5 – induced lattice of 11111 (Age, Race, Work_class, Occupation, Education)
Histograms

- Histograms allow us to compute the amount of suppression for a given value of $k$ (equiv. $l$).
- E.g., to achieve 3-anonymity in level A1W1R1 we must suppress groups with size 1 or 2 => 17 tuples ($17=1*11+2*3$).
Histograms Construction

• For each node we need an auxiliary view and a query defined over it
  – k-anonymity
    
    ```sql
    CREATE VIEW test(a,w,r,plithos) AS (
        SELECT Age.level0,Work_class.level2,Race.level0, count(*)
        FROM Adult, Age,Race,Work_class
        WHERE Adult.age=age.level0 and Adult.race=race.level0 and Adult.Work_class=Work_class.level0
        GROUP BY Age.level0,Work_class.level2,Race.level0)
    
    SELECT plithos, count(*) FROM test
    ```

  – l-diversity
    
    ```sql
    CREATE VIEW test(a,w,r,l,plithos) AS (
        SELECT Age.level0,Work_class.level2,Race.level0,  distinct  hourse_per_week,count(*)
        FROM Adult, Age,Race,Work_class
        WHERE Adult.age=age.level0 and Adult.race=race.level0 and Adult.Work_class=Work_class.level0
        GROUP BY Age.level0,Work_class.level2,Race.level0)
    
    SELECT plithos, count(*) FROM test
    ```
QI size is the most important factor for suppression

• As the size of the quasi identifier set increases, suppression increases too – sometimes drastically.

• Given a specific height and k, an increase in QI size by one increases the suppression by a factor of 2 – 3

• To attain the same suppression threshold, an increase in QI size by one, requires ascending 1-2 levels for k-anonymity and 2-3 levels for l-diversity.
  – For higher levels, even larger scale factors (~3-4)
Effect of height and privacy to suppression

- As the height increases:
  - suppression drops quickly at small heights
  - the drop in suppression is less important in higher heights, where the number of suppressed tuples becomes statistically small and drops slowly.

- As the value for the privacy criterion (e.g., \( k \) in \( k \)-anonymity) increases, the suppression increases too.
  - Especially important in lower heights

- Lower heights are important due both to their information utility and demonstrate high volumes of suppression.
- The overall trend for the decrease of suppression is practically the same for different values of \( k \) or \( l \).
Not all attributes/nodes are equal

- Not all attributes, generalization levels and, consequently, generalization schemes have the same effect to suppression.
- Within the same height, the minimum possible suppression is approximately 2.5 times lower than the average for k-anonymity and 3 times lowers for l-diversity.
  - This is especially evident in cases where the suppression has high values or values that cannot really be tolerated;
  - On the other hand, for too large values of suppression (e.g., too large QIs or $k$) the relationship between average and minimum value does not follow this rule.
Answers to the original questions

• **Q:** Is the amount of suppressed tuples significant? What is the relationship between suppression, generalization and privacy?

• **A:** large amounts of suppressed data, quite possibly much higher as compared to more careful choices concerning the generalization scheme can occur in areas of the problem space that matter:
  – Low generalization heights (that are of more interest to us due to their information utility), or
  – Large values for the privacy criterion (which is of more interest to us due to the increased privacy it offers to individuals), or
  – Erroneous choice of generalization scheme

• All the above findings are consistent with both k-anonymity and l-diversity over two data sets
  – As a side remark, k-anonymity is a good estimator for naïve l-diversity
Roadmap

• Introduction
• Study of the relationship between suppression, generalization and privacy
  • User-time anonymization with an exhaustive, off-line pre-processing
  • User-time anonymization with user-time pre-processing
• Conclusions
Our method

• **Offline** phase
  – Lattice construction
  – Histogram construction for all nodes of the lattice

• **Online** negotiation
  – User submits request with parameters (κόμβος vmax)
    • MaxSupp (maximum tolerable no. of suppressed tuples)
    • h=[h₁, ..., hₙ] (maximum tolerable height in the hierarchy of each QI => also represented by a single node v_max)
    • k or l (minimum tolerable value for the privacy criterion)
  – **Algorithm returns** (a) exact solution respecting the above constraints or (b) 3 approximations –see next
Approximate Solutions

• Solution #1: Keep $k$, $h$ fixed and relax $\text{maxSupp}$ – i.e., search for the minimum amount of suppressed tuples while respecting $k$, $h$

• Solution #2: Keep $k$, $\text{maxSupp}$ fixed and relax $h$ – i.e., search for the minimum height while respecting $k$, $\text{maxSupp}$

• Solution #3: Keep $h$, $\text{maxSupp}$ fixed and relax $k$ – i.e., search for the maximum possible $k$ which respects $h$, $\text{maxSupp}$
This is the lattice for \( QI=3 \)
This is the lattice for $QI=3$ annotated with the number of suppressed tuples for $k=3$
Assume the user requests:
\( h = 121 \)
\( K = 3 \)
\( \text{MaxSupp} = 20 \)

Observe \( \text{vmax} 121 \)
Assume the user requests:
\[ h = 121 \]
\[ K = 3 \]
\[ \text{MaxSupp} = 20 \]

The exact solution is \textbf{111} with \#supp. = 17
Assume the user requests:
\[ h = 121 \]
\[ K = 3 \]
\[ \text{MaxSupp} = 8 \]

Observe \( v_{\text{max}} \) 121: it fails to meet all three constraints.
Assume the user requests:
\( h = 121 \)
\( K = 3 \)
\( \text{MaxSupp} = 8 \)

Suggestions:

**Closest k:**
Node 121, k=2

**Closest height:**
Node 400, h=4

**Closest maxSupp:**
Node 121, maxsupp=11
Algorithm at a glance

Input:
- Input relation $R$ + hierarchies $H$ + lattice with histograms
- A user request $(k, h, \text{maxSupp})$ with the user constraints

1. Identify top-acceptable node $v_{\text{max}}$
2. If $v_{\text{max}}$ answers the $(k, h, \text{maxSupp})$
   - Search within the sublattice of $v_{\text{max}}$ for the lowest possible node that also answers $(k, h, \text{maxSupp})$
3. Else
   - Relax MaxSupp: stay at $v_{\text{max}}$ (respect $h$) and find the suppression value for $k$ (respect $k$)
   - Relax $k$: stay at $v_{\text{max}}$ (respect $h$) and find the largest $k$ that suppresses less than maxSupp (respect maxSupp)
   - Relax $h$ (retain $k$, maxSupp) and answer outside the sublattice:
     - Binary search between $v_{\text{max}}$ and lattice’s top
     - Exhaust all nodes of a level: if nobody answers, binary search between top and this level; else, whenever a node answers, perform binary downwards
     - Stop when it is impossible to descend and the last level is exhaustively tested
Algorithm Simple Anonymity Negotiation

\[ \text{Algorithm SimpleAnonymityNegotiation}(L, k, h, \text{MaxSupp}) \]
\[ \text{In: Lattice } L \text{ with the histograms for } R, H, \text{ constraints for } k, h, \text{ MaxSupp} \]
\[ \text{Out: an exact solution } s = [v, k, h, \text{supp} \_i] \text{ or } s_1, s_2, s_3, s_4 = [v, i, k, i, h, i, \text{supp} \_i] \]
\[ \text{Var: a 2D vector of candidate solutions } \text{Candidates}[\text{hmax}][] \]

\[ \text{Begin} \]
1. Let \( v_\text{max} \) be the node that corresponds to the constraint \( h \);
2. if \( v_\text{max} \) is visited then exit;
3. mark \( v_\text{max} \) as visited;
4. if (checkExactSolution(\( v_\text{max}, L, k, h, \text{MaxSupp} \)) == true) {
5. \hspace{1em} \text{Candidates}[\text{height}(v_\text{max})] = \text{Candidates}[\text{height}(v_\text{max})] \cup \{v_\text{max}\};
6. \hspace{1em} \text{for all } v \text{ in lower}(v_\text{max})
7. \hspace{2em} \text{ExactSublatticeSearch}(v_c, L, k, h, \text{MaxSupp}, \text{Candidates});
8. \hspace{1em} \text{// when the recursion is over, the Candidates has the full list of nodes}
9. \hspace{1em} \text{// that can serve as candidate solutions}
10. \hspace{1em} \text{minHeight = minimum height having Candidates[\text{minHeight}] != \{\};}
11. \hspace{1em} \text{v_win = v in Candidates[\text{minHeight}] with the lowest possible suppression for}
12. \hspace{2em} \text{k;}
13. \hspace{1em} \text{return (v_win, k, minHeight, suppressed(v_win, k));}
14. } \]
15. else{
16. \hspace{1em} \text{approxSol}_1 = \text{ApproximateMaxSupp}(L, v_\text{max}, k, h, \text{MaxSupp});
17. \hspace{1em} \text{approxSol}_2 = \text{ApproximateH}(L, v_\text{max}, \text{height}(v_\text{max}), \text{height}(top), k, h, \text{MaxSupp});
18. \hspace{1em} \text{approxSol}_3 = \text{ApproximateK}(L, v_\text{max}, k, h, \text{MaxSupp});
19. \hspace{1em} \text{return approxSol}_1, \text{approxSol}_2, \text{approxSol}_3;\]
20. } \]
\[ \text{End.} \]
Exact Solution

```
ExactSublatticeSearch(v,L,k,h,MaxSupp,Candidates){
    if v is visited then exit;
    mark v as visited;
    if (checkExactSolution(v,L,k,h,MaxSupp) == true){
        Candidates[height(v)] = Candidates[height(v)] U {v};
        for all v_c in lower(v)
            ExactSublatticeSearch(v_c,L,k,h,MaxSupp,Candidates);
    }
}
```

```
checkExactSolution(v,L,k,h,MaxSupp){
    lookup histogram of v in L;
    if suppressed(v,k) <= MaxSupp && height(v) <= h
        return true;
    else return false;
}
```
Approximate answers (relaxing MaxSupp and k)

```
ApproximateMaxSupp(L,v,k,h,MaxSupp){
    find the minimum amount of suppressed tuples, approxSupp, s.t.
    checkExactSolution(v,L,k,h,approxSupp) returns true;
    if no such value exists, return {};
    else{
        for all v_c in sublattice(v) (recursively){
            checkExactSolution(v_c,L,k,h,approxSupp)
            break when a whole level fails to produce a solution;
        }
        let v_win be the node with the lowest height that satisfies k,h,approxSupp
        (with arbitrary tie resolution)
        return v_win,k,h,approxSupp;
    }
}

ApproximateK(L,v,k,h,MaxSupp){
    find the maximum value of k, approxK, s.t. checkExactSolution(v,L,approxK,h,maxSupp)
    returns true;
    if no such value exists, return {};
    else{
        for all v_c in sublattice(v) (recursively){
            checkExactSolution(v_c,L,approxK,h,maxSupp)
            break when a whole level fails to produce a solution;
        }
        let v_win be the node with the lowest height that satisfies approxK,h,maxSupp
        (with arbitrary tie resolution)
        return v_win,approxK,h,maxSupp;
    }
}
```
Approximate answers (relaxing H)

```plaintext
ApproximateH(L,v,h_low,h_high,k,h,MaxSupp){
    while(h_low <= h_high){
        h_current = middle between h_low and h_high;
        flag = checkIfNoSolutionInCurrentHeight(L,h_current,k,MaxSupp);
        if(flag == true){
            low = current + 1;
        }
        else{
            currentMinHeight = current;
            high = current - 1;
        }
    }

    for all v_c in currentMinHeight, find the one v_win, with the minimum
    suppressed(v_c,k);
    //exception: this fails only if k > |R|, else top of the lattice always answers
    return v_win,k,height(v_win),MaxSupp;
}

checkIfNoSolutionInCurrentHeight(L,h_current,k,MaxSupp){
    for all v_c in h_current
        if suppressed(v_c,k) <= MaxSupp return false;
    return true;
}
```
Crux of the approach

• k-anonymity and naïve l-diversity support a monotonicity property between an ancestor and a descendant node:
  • Every group $\gamma$ of the ancestor is the union of one or more groups of the descendant
    – Specifically, the ones whose ancestor QI values map to the ones of $\gamma$
    – Practically, the ancestor node creates equivalence classes to the groups of the descendant
Crux of the approach

• Therefore, two properties hold:
  – An ancestor node has less (or equal) groups than any of its descendants
  – These groups are larger (or equal) than the ones of the descendant

• Since groups are larger at the ancestors:
  – If the ancestor fails to satisfy $k$, all its descendants fail too
  – If a descendant satisfies $k$, then all its ancestors satisfy it too

• Based on this, we can prove that both the exact and the approximate search are correct
Theoretical results

• If the user request is satisfied by $v_{\text{max}}$ (exact search)
  – For any value $\alpha$, the cumulative histogram for $v_{\text{max}}$ has a smaller or equal value than the cumulative histogram for any node $v$ in the descendants of $v_{\text{max}}$. This holds both for $k$-anonymity and $l$-diversity.
  – Once a node respects the three criteria posed by the user, we need to search its descendants for the lowest possible node that returns an answer, too.
    • i.e., descend the sublattice, until all nodes of a level fail

• If the user request is not satisfied by $v_{\text{max}}$
  – None of the nodes of the sublattice induced by $v_{\text{max}}$ has a value $k^*$ which is larger than $v_{\text{max}} \cdot k$ and suppresses the same amount of tuples (in other words, nobody can provide better $k$-anonymity with the sacrifice of the same amount of tuples).
    • i.e., maxSupp and $k$ relaxations have to be answered at $v_{\text{max}}$
    • Also remember that the relaxation of $h$ explores all the lattice’s levels
Theorem 2  Given a node $v_{\text{max}}$, the sublattice it induces $L(v_{\text{max}})$, and an integer $\alpha$, the following hold:
\[
\begin{align*}
\text{cumKA}(v_{\text{max}} | \alpha) & \leq \text{cumKA}(v | \alpha), \ v \in L(v_{\text{max}}) \\
\text{cumSLD}(v_{\text{max}} | \alpha) & \leq \text{cumSLD}(v | \alpha), \ v \in L(v_{\text{max}})
\end{align*}
\]

Theorem 3  Assume a user request $q = [k, h, \text{maxSupp}]$ over a lattice $L$ annotated with the cumulative histograms for a data set $D$. Assume the top-acceptable node $v_{\text{max}}$ that has $h$ as its generalization scheme. If $v_{\text{max}}$ respects $q$, then the node with the lowest height that respects $q$ is in the lattice induced by $v_{\text{max}}$, $L(v_{\text{max}})$.

Theorem 4  Assume a user request $q = [k, h, \text{maxSupp}]$ over a lattice $L$ annotated with the cumulative histograms for a data set $D$. Assume that the top-acceptable node $v_{\text{max}}$ which has $h$ as its generalization scheme fails to respect $q$. Assume the largest value $k_r$, $k_r < k$, such that $v_{\text{max}}$ respects $q_r = [k_r, h, \text{maxSupp}]$. Then, there is no node $v$, $v \neq v_{\text{max}}$, $v \in L(v_{\text{max}})$, such that $q^* = [k^*, h, \text{maxSupp}]$, $k^* > k_r$, is respected at $v$. 

Adult data set Experiments

- We assess execution time and no. visited nodes for 3 cases: variant $k$, variant $v_{max}$, and, variant $maxSupp$

| Generalization constraints | $|QI| = 3$ | $|QI| = 4$ | $|QI| = 5$ | $|QI| = 6$ |
|----------------------------|-----------|-----------|-----------|-----------|
| level                      | 101, 211  | 1001, 2011| 11001, 21012| 111001, 211012|
| constraints                | (default), 212 | (default), 2112 | (default), 22112 | (default), 222112 |

For all QI’s, we have used three configurations: (a) a low one, with all levels constrained low in their hierarchies, (b) a middle-low (default) with some constraints placed on levels in the middle of their hierarchies and (c) middle, with all levels constrained at the middle in their hierarchies

- $k$ 3, 10 (default), 50

- MaxSupp 32, 321 (default), 3216 (approx. 0.1%, 1%, 10% of the data set)
## Adult data set: \#nodes visited

<table>
<thead>
<tr>
<th>Variant</th>
<th>QI = 3</th>
<th>QI = 4</th>
<th>QI = 5</th>
<th>QI = 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</td>
<td><img src="image1.png" alt="Graph" /></td>
<td><img src="image2.png" alt="Graph" /></td>
<td><img src="image3.png" alt="Graph" /></td>
<td><img src="image4.png" alt="Graph" /></td>
</tr>
<tr>
<td>$h$</td>
<td><img src="image5.png" alt="Graph" /></td>
<td><img src="image6.png" alt="Graph" /></td>
<td><img src="image7.png" alt="Graph" /></td>
<td><img src="image8.png" alt="Graph" /></td>
</tr>
<tr>
<td>$MaxSupp$</td>
<td><img src="image9.png" alt="Graph" /></td>
<td><img src="image10.png" alt="Graph" /></td>
<td><img src="image11.png" alt="Graph" /></td>
<td><img src="image12.png" alt="Graph" /></td>
</tr>
</tbody>
</table>

**Pink:** exact answer  
**Blue:** approximations
Results in a nutshell (1)

• When increasing the privacy criterion
  – When QI is small: exact answer + search is directed towards lower heights. Consequently, as k increases the solution is found earlier.
  – For larger QIs: need to resort to relaxations + the increase of k sublinearly increases the search space.

• When increasing the max. tolerable node
  – When QI size is small: we can have exact solutions and the height increase increases the search space.
  – Larger QIs: need to resort to relaxations + the higher $v_{max}$ is placed by the query, the less nodes we visit to find the approximate answer.

• When increasing the max. tolerable #suppressed tuples:
  – When QI is small: exact answers are possible + as maxSupp increases the number of visited nodes increases too
  – Larger QIs: the higher the constraint is set, the faster an approximate solution is found
Results in a nutshell (2)

- In all cases, the time needed to detect the solutions ranges in 1-8 milliseconds!
- In all experiments, it is clear that the costs are dominated by the QI size.
- Results present the same behavior
  - for l-diversity and k-anonymity
  - For the Adult and the IPUMS data set
Lattice (in fact: Histogram)

**Construction time**

- Adult data set (30162 tuples)

  ![Graph for Adult data set](image)
  
  - $k$-anonymity
  - $l$-diversity

- IPUMS data set (600000 tuples)

  ![Graph for IPUMS data set](image)
  
  - $k$-anonymity
  - $l$-diversity

<table>
<thead>
<tr>
<th></th>
<th>$k$-anonymity</th>
<th>$l$-diversity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average time</strong></td>
<td>10,5</td>
<td>38,087</td>
</tr>
<tr>
<td><strong>Histo size</strong></td>
<td>530,688</td>
<td>58,424</td>
</tr>
</tbody>
</table>
Roadmap

• Introduction
• Study of the relationship between suppression, generalization and privacy
• User-time anonymization with an exhaustive, off-line pre-processing
• User-time anonymization with user-time pre-processing
• Conclusions
Why a Partial lattice?

• The disadvantages of computing histograms for the full lattice
  – Too much time to compute the histograms (even if it is done only once)
  – The construction time increases exponentially with the size of the QI.

• What if we want user-time preprocessing?
  – Resort to constructing histograms for the partial lattice
  – Online lattice construction (mainly future work)
Partial lattice construction with choice of materialized nodes

• The goal is to compute the histograms only for a subset of the lattice’s nodes (say p% of the nodes)

• Problems:
  – Which nodes will be selected and how?
  – Will this save time adequately (for the user time requirement)?
  – What is the effect to the quality of the solution?
Results in a nutshell (1)

✓ Which nodes will be selected and how?
  ✓ Ultimately, it appears that we have an estimator metric $\Lambda(v) = \sum (\log(\gamma_i) \cdot \mu_i)$ that successfully "predicts" node importance

✓ Will this save time adequately (for the user time requirement)?
  ✓ Yes, we believe it did (~1 min preparation time for the QI=6 vs 20min of full lattice)

✓ What is the effect to the quality of the solution?
Results in a nutshell (2)

• Partial lattice preprocessing:
  – Works well with exact answers and some approximations
  – Misses approximations local to $v_{\text{max}}$; typically results in lower heights and higher suppressions
  – 4 sec – 1 min preprocessing time
  – Answers faster than full lattice (0.3 – 2msec)

• Partial lattice with $v_{\text{max}}$ histogram at runtime:
  – Small improvement for exact answer; identical behavior to full lattice for approximations local to $v_{\text{max}}$
  – 0.1 – 0.3 sec overhead
Our method for partial lattice materialization

**Offline preprocessing**

- Rank nodes according to an “importance” metric
  - Must compute the effect of different levels of different attributes to the subsequent suppression.
  - For every node, compute a score by combining the scores of the levels for each of its attributes
- Compute the histograms only for the top p% of the nodes.

**Online processing**

- The algorithm applied over the partial lattice is practically the one of the full lattice, for both the exact and the approximate answers.
Research challenges & solutions for the case of the partial lattice

• **Which nodes will be selected and how?**
• Will this save time adequately (for the user time requirement)?
• What is the effect to the quality of the solution?
Crux & main techniques

• Crux: The larger the groups, the lower the suppression

• Two fundamental metrics to assess the grouping power of an attribute’s level:
  – Average group size: to estimate the interplay of the level $h$ of an attribute $A_k$, with the other attributes, we group by $A_1@0, \ldots, A_k@h, \ldots, A_n@0$ and measure avg group size
  – Relative importance: divide avg group size of a level with the avg group size of its immediately lower level

• Attn: followed estimation by level (and not by node) as it computes fast ($|\text{levels}| << |\text{nodes}|$)
Why not only avg group size?

- Certain attributes dominate the ranking of levels, as they have small domains and larger groups
- Here: age at level 4,3,2 dominates the ranking
Why not only avg group size?

• On the other hand, relative importance “normalizes” this behavior
Important nodes (1)

• We have experimented with four metrics to assess the importance of every quasi-identifier level (i.e., for each level of each QI)
  – Average group size ($\gamma$)
  – Relative importance of a level ($\mu$)
  – The product $\gamma \times \mu$
  – The product $\log_2(\gamma) \times \mu$

\[
relImp(A^h) = \begin{cases} 
\frac{\text{avgGroupSize}(A^h)}{\text{avgGroupSize}(A^{h-1})}, & \text{for all heights } h \text{ in } A^0, \ldots, 1, \text{ or} \\
\frac{1}{\text{power}(A^1)}, & \text{for } h=0
\end{cases}
\]
Important nodes (2)

• Then, the importance of a node is the sum of the scores of its levels:
  
  – $\Gamma(v) = \Sigma(\gamma_i)$ //i ranging over all levels of a node
  – $M(v) = \Sigma(\mu_i)$
  – $\Gamma M(v) = \Sigma(\gamma_i \mu_i)$
  – $\Lambda(v) = \Sigma(\log(\gamma_i) \mu_i)$

• Tried also the product of the individual scores, but sum works better
Which is the best metric then?

• To assess which is the best metric to use, we have generated a partial lattice with the 5% of the lattice’s nodes as dictated by each metric.

• For all possible QI’s and k in {3, 10, 50},
  – For every height in the lattice, we compared
    • The node with the least suppression of the partial lattice, vs.,  
    • The node with the least suppression in the full lattice

• For every metric, we have computed
  – The number of misses
  – The avg deviation wrt the actual suppression

• Special care taken for cases where the actual suppression was zero:
  – a penalty of $|R|^{-1}$ was assigned to every extra tuple of the partial lattice’s best solution.
  – Still, in very low suppressions (here, low: < 30 tuples, 1‰$|R|$) small differences result in big deviations. So, two cases:
    • Ignore these deviations
    • Keep them in the overall results
### Average group size VS product log$_2(\gamma)$*$\mu$

#### Table 1

<table>
<thead>
<tr>
<th>$k=3$</th>
<th>Till 30 tuples</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#dev($\Gamma$)</td>
<td>Err($\Gamma$)</td>
</tr>
<tr>
<td>$QI=3$</td>
<td>0</td>
<td>0,00%</td>
</tr>
<tr>
<td>$QI=4$</td>
<td>2</td>
<td>21,96%</td>
</tr>
<tr>
<td>$QI=5$</td>
<td>2</td>
<td>2,86%</td>
</tr>
<tr>
<td>$QI=6$</td>
<td>1</td>
<td>0,60%</td>
</tr>
</tbody>
</table>

#### Table 2

<table>
<thead>
<tr>
<th>$k=10$</th>
<th>Till 30 tuples</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#dev($\Gamma$)</td>
<td>Err($\Gamma$)</td>
</tr>
<tr>
<td>$QI=3$</td>
<td>2</td>
<td>52,99%</td>
</tr>
<tr>
<td>$QI=4$</td>
<td>3</td>
<td>57,87%</td>
</tr>
<tr>
<td>$QI=5$</td>
<td>3</td>
<td>6,84%</td>
</tr>
<tr>
<td>$QI=6$</td>
<td>0</td>
<td>0,00%</td>
</tr>
</tbody>
</table>

#### Table 3

<table>
<thead>
<tr>
<th>$k=50$</th>
<th>Till 30 tuples</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#dev($\Gamma$)</td>
<td>Err($\Gamma$)</td>
</tr>
<tr>
<td>$QI=3$</td>
<td>2</td>
<td>33,20%</td>
</tr>
<tr>
<td>$QI=4$</td>
<td>4</td>
<td>38,38%</td>
</tr>
<tr>
<td>$QI=5$</td>
<td>1</td>
<td>0,05%</td>
</tr>
<tr>
<td>$QI=6$</td>
<td>0</td>
<td>0,00%</td>
</tr>
</tbody>
</table>
Eventually ...

- Given $\Gamma(v) = \sum(\gamma_i)$, and, $\Lambda(v) = \sum(\log(\gamma_i) * \mu_i)$
- In the “full” case, it appears that
  - wins are ~ equally split
  - when $\Gamma$ loses, it deviates a lot;
  - when $\Lambda$ loses, it stays quite close to the winner value of $\Gamma$
- When we ignore small suppressions, $\Lambda$ wins more times and with quite small errors, too

- Ultimately, it appears that $\Lambda(v)$ is the best metric to predict the usefulness of a node in the partial lattice
Research challenges & solutions for the case of the partial lattice

✓ Which nodes will be selected and how?
  ✓ Ultimately, it appears that \( \Lambda(v) = \Sigma (\log(y_i) \times \mu_i) \) is the best metric

• **Will this save time adequately (for the user time requirement)?**

• What is the effect to the quality of the solution?
Time to materialize the partial lattice with its histograms for the Adult data set

Full lattice

Partial lattice
Time to materialize the partial lattice with its histograms

- Adult data set (30162 tuples)

- IPUMS data set (600000 tuples)

Comparing same QIs for different data sets, we see that the effect of data size is important here.
Time to materialize the partial lattice with its histograms

• We observe a drastic decrease in the time to compute the partial lattice and its histograms: approx. 1 minute (for QI=6) compared to 20 minutes in the full lattice. The drop is linear to p (the % of nodes materialized).

• As in the case of the full lattice, the time to answer increases exponentially with the size of QI.

• As expected, the time breakdown indicates that the interaction with the underlying database consumes most of the time spendings.
Research challenges & solutions for the case of the partial lattice

✓ Which nodes will be selected and how?
  ✓ Ultimately, it appears that $\Lambda(v) = \sum (\log(\gamma_i) \cdot \mu_i)$ is the best metric

✓ Will this save time adequately (for the user time requirement)?
  ✓ Yes, we believe it did (~1 min preparation time for the worst case)

• What is the effect to the quality of the solution?
Algorithm for partial lattice: **Exact answer**

1. Search for $v_{\text{max}}$
2. If it belongs to the partial lattice and does not satisfy the request, then try approximations
3. If it does not belong to the partial lattice, OR, it belongs and gives an exact answer, then search the sublattice **exhaustively**
   - Attn: due to its partial nature, if a level fails to answer, it does not mean that a lower level does not include a node that can answer
   - In any case, the sublattice is too small; exhaustive search is anyway the fastest search
Algorithm for partial lattice: Approximate answers

- **Relaxations of k and maxSupp:** search exhaustively all the levels from 0 to $\text{height}(v_{\text{max}})$ for the best possible answer
- **Relaxation of h:** same as exact lattice
  - Try the upper part of lattice; if a node answers $(k, \text{maxSupp})$, search downwards
  - Else, if a level is exhausted and fails, search upwards
Effect to the quality of solution

• We conducted the full lattice’s experiments for the partial lattice with the same parameters (k-anonymity) for both the Adult and the IPUMS data set

• Efficiency:
  – All execution times for the algorithm range in the area of 0.33-2 msecs

• Effectiveness (summary of findings):
  – The exact solution performs very well
  – Height relaxation has very good results.
  – MaxSupp relaxation responds with quite larger suppression.
  – Relaxation of k does not always return an answer; and if it does, it is not always the best possible.

  – Astonishingly, p% does not change practically anything (tried 1%, 5%, 10%): only height relaxation is slightly better, all the rest are the same
Quality of Solution: good news

• **Exact answer** (search the sub-lattice of $v_{\text{max}}$)
  
  – The partial lattice achieves **exact answer** (like the full lattice) **with no deviations**!
    
    • **Exception**: 3 cases, where the partial lattice gives approximate answers and the full lattice gives exact ones ((i) $QI=3$, $k=50$, (ii) $QI=4$, $k=3$ (iii) $QI=5$, $\text{maxSupp}=3126$).

• **Height approximation** (respect $k$ & max supp, relax $h$):
  
  – **Very good results** compared to the full lattice, with
    
    • small deviations for $\text{maxSupp}$$\&\&$
    • zero deviations for $k$
Quality of Solution: bad news

- MaxSupp approximation (respect k & h, relax maxSupp):
  - A solution is returned
  - Still: descendants of vmax are typically found at low heights, resulting in a large volume of suppressed tuples compared to the full lattice

- K approximation (respect maxSupp & h, relax k):
  - It is hard to find an answer (actually, both in partial and the full lattice). Still, whenever this happens, the full lattice always has a better suggestion for k.

- Remember: $v_{\text{max}}$ might not be part of the lattice
Heuristic extension: $v_{\text{max}}$ histogram at runtime

- **Observation**: The two approximations that suffer are the ones executed locally at $v_{\text{max}}$
- **Heuristic**: For each user request, at runtime, compute the histogram of $v_{\text{max}}$ too
- Apart from the above addition, the algorithm for the partial lattice remains unaffected
Heuristic extension: $v_{\text{max}}$ histogram at runtime

- We pay a time penalty of 0.1 – 0.3 sec per query for the (one) extra histogram of $v_{\text{max}}$
- Small improvement for exact answers and no improvement for the relaxation of $h$
- Identical behavior to the full lattice for the two approximations taking place locally at $v_{\text{max}}$

<table>
<thead>
<tr>
<th></th>
<th>Exact P</th>
<th>PR</th>
<th>Relax M P</th>
<th>PR</th>
<th>Relax H P</th>
<th>PR</th>
<th>Relax k P</th>
<th>PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same Sol.</td>
<td>6</td>
<td>8</td>
<td>1</td>
<td>18</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Other Sol.</td>
<td>1+3</td>
<td>2+0</td>
<td>17</td>
<td>0</td>
<td>9</td>
<td>9</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Failed</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>No Sol both</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

Effectiveness comparison of
- Partial (P, shaded) vs
- PartialWRuntime (PR) wrt Full Lattice

Legend: “1+3” means “1 occasion where partial gave exact answer different that the full lattice + 3 occasions where it gave approximations, instead”
Roadmap

• Introduction
• Study of the relationship between suppression, generalization and privacy
• User-time anonymization with an exhaustive, off-line pre-processing
• User-time anonymization with user-time pre-processing
• Conclusions
Conclusions (1)

• What is the relationship between suppression, generalization and privacy?
  – In low heights and/or large QI’s suppression takes large values => the problem is important
  – The larger the height, the lower the suppression – minimum suppression is characterized by steep reductions (as opposed to average suppression) => it is important to detect the proper generalization schemes that allow BOTH small suppression and low height
  – The value of k has a clear effect to the suppression in low heights (where, still, one can find “good” solutions) => there a meaning, indeed, to negotiate k, if necessary
  – All the results are consistent in 2 data sets and 2 privacy criteria: k-anonymity & l-diversity
Conclusions (2)

• Can we respond in user time to anonymization requests? Can we suggest anonymization schemes that are approximately close to the original user request?
  – Yes to both! We have two ways to address the above, depending on the price we are willing to pay wrt the offline preprocessing of the lattice
  – Full lattice preprocessing:
    • 18 sec – 20 min preprocessing time
    • Exact answers and approximations in less than 10msec (depends upon lattice size)
  – Partial lattice preprocessing:
    • 4 sec – 1 min preprocessing time
    • Works well with exact answers and some approximations;
    • Misses some approximations; typically results in lower heights and higher suppressions
    • Answers faster than full lattice (0.3 – 2msec)
  – Partial lattice with $v_{\text{max}}$ histogram at runtime:
    • 0.1 – 0.3 sec overhead
    • Small improvement for exact answer; identical behavior to full lattice for approximations local to $v_{\text{max}}$
Future Work

• Histogram computation at query time
• Negotiate the QI size too with the user as part of the algorithm
• Instead of global, try local recoding (but with fixed hierarchies)
  – Significantly harder problem, lattice in its current form cannot help
Thank you!

Questions?
References (1)


References (2)

Auxiliary Slides
### Two dimension illustration

<table>
<thead>
<tr>
<th>a</th>
<th>2</th>
<th>10</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>8</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>c</td>
<td>30</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(α, β)</th>
<th>γ</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>12</td>
</tr>
<tr>
<td>b</td>
<td>11</td>
</tr>
<tr>
<td>c</td>
<td>55</td>
</tr>
<tr>
<td>d</td>
<td>20</td>
</tr>
</tbody>
</table>

| a  | 5  | 7 | 0 |
|----|----|---|
| b  | 6  | 5 | 5 |
| c  | 30 | 25 | 0 |
| d  | 0  | 20 | 0 |

### Table Summary

<table>
<thead>
<tr>
<th>Attl</th>
<th>Att2</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>α</td>
<td>2</td>
</tr>
<tr>
<td>a</td>
<td>β</td>
<td>10</td>
</tr>
<tr>
<td>b</td>
<td>α</td>
<td>8</td>
</tr>
<tr>
<td>c</td>
<td>γ</td>
<td>5</td>
</tr>
<tr>
<td>b</td>
<td>(α, β)</td>
<td>30</td>
</tr>
<tr>
<td>c</td>
<td>(α, β)</td>
<td>25</td>
</tr>
<tr>
<td>d</td>
<td>β</td>
<td>20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attl</th>
<th>Att2</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>(α, β)</td>
<td>12</td>
</tr>
<tr>
<td>b</td>
<td>(α, β)</td>
<td>11</td>
</tr>
<tr>
<td>c</td>
<td>γ</td>
<td>5</td>
</tr>
<tr>
<td>c</td>
<td>β</td>
<td>20</td>
</tr>
</tbody>
</table>

(a) original  
(b) global  
(c) m/d  
(d) local
IS THE PROBLEM VALID?
Effect of k and height to suppression (1)

Avg #suppressed tuples, QI = 3

Min #suppressed tuples, QI = 3
Effect of $k$ and height to suppression (2)

- As the height increases, suppression drops with high rate
  - The drop follows the same trend for different values of $k$.
  - Big heights have quite low suppression volumes to bother us; however, the **low heights** that are important in terms of information utility have quite high values – esp. for somehow larger $k$
- As $k$ increases, the number of suppressed tuples increases too, esp. in low heights
- The differences between average and min suppression is significant and typically the **average** value is 2-3 times lower than the **minimum** one.
  - Therefore, we conclude that **not all nodes are “equally” good solutions** (some nodes are more important than others).
  - Paying the price to find the best possible solution can result in significant improvement to the performed suppression.
Effect of QI to suppression (1)

\[ k = 3 \]
Effect of QI to suppression (2)

• Clearly, different QI sizes at the same level have on average an increase of the scale of 2 -3 times, for large volumes of suppressed tuples. This scale factor changes as the volume of suppressed tuples drops.

• Moreover, it is clear that statistically tolerable amounts of suppressed tuples are attained slower as the size of QI grows. For example, the suppression percentage falls under 1% of the total volume of data at height H1 for QI = 3, H3 for QI = 4, H6 for QI = 5 and after H8 for QI = 6.

• The most important observation is that a QI of size n drops to the levels of suppression of the QI of size n-1 around 3-4 levels of generalization later for smaller QI’s and 1-2 levels for larger QI’s.
The study for $l$-diversity: the effect of $l$ and height

- As $l$ increases, so does the amount of suppressed values (for the same height and QI size). The amount of suppression is not directly analogous to the value of $l$, however the scaling of the suppression is quite close to the scaling of the value of $l$.
- For different values of $l$, consistently, as the height increases, the number of suppressed tuples drops quite quickly.
- As in the case of k-anonymity, the ratio of minimum to average value is approximately 2 (in fact it rises to quite large values at big heights; if one removes the outliers the average ratio of average to minimum value is around 3).
The study for $l$-diversity: effect of QI

- Clearly, different QI sizes at the same level have on average an increase of the scale of 2-3 times, for large volumes of suppressed tuples. This scale factor changes as the volume of suppressed tuples drops.

- Moreover, it is clear that statistically tolerable amounts of suppressed tuples are attained slower as the size of QI grows. For example, the suppression percentage falls under 1% of the total volume of data at height $H_1$ for QI = 3, $H_3$ for QI = 4, $H_6$ for QI = 5 and after $H_8$ for QI = 6.

- The most important observation is that a QI of size $n$ drops to the levels of suppression of the QI of size $n-1$ around 3-4 levels of generalization later for smaller QI’s and 1-2 levels for larger QI’s.
The study for the IPUMS data set

- **k-anonymity**
  - Similarly to the Adult data set, the amount of suppressed tuples drops rapidly as we increase the height.
  - The behavior of the average with respect to the min suppression is quite different with respect to Adult data set. Min suppression drops much faster with respect to the average => it much more important to pick the right solution.

- **\(l\)-diversity**
  - Again, the number of suppressed tuples drops rapidly as we increase the height.
  - We can achieve tolerable amounts of suppressed tuples quite low in the lattice – at levels H1 and H2 that is.
Answers to the original questions

• Is the amount of suppressed tuples significant?
  – In low heights and/or large QI’s suppression takes large values => the problem is important

• What is the relationship between suppression, generalization and privacy?
  – The larger the height, the lower the suppression – minimum suppression is characterized by steep reductions (as opposed to average suppression) => it is important to detect the proper generalization schemes that allow BOTH small suppression and low height
  – The value of k has a clear effect to the suppression in low heights (where, still, one can find “good” solutions) => there a meaning, indeed, to negotiate k, if necessary
  – All the results are consistent in 2 data sets and 2 privacy criteria: k-anonymity & l-diversity; as a side remark, k-anonymity is a good estimator for naïve l-diversity
Cumulative histogram για QI-3 (k-anonymity)
Cumulative histogram for QI-3 (I-diversity)

A0W0R0O0E0

A1W1R1O1E1
Partial vs. Full Lattice for QI=5 wrt Avg. # suppressed tuples

![Graph showing the comparison between partial and full lattice for QI=5. The x-axis represents height, and the y-axis represents the average number of suppressed tuples. The graph depicts a downward trend as height increases, with the partial lattice showing a slightly steeper decrease compared to the full lattice.](image)
Min - Avg - Max # suppressed tuples for QI=3 & QI=5

<table>
<thead>
<tr>
<th></th>
<th>QI</th>
<th>= 3</th>
<th></th>
<th>QI</th>
<th>= 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Avg</td>
<td>Max</td>
<td>Avg % over full</td>
<td>% over previous</td>
</tr>
<tr>
<td>H0</td>
<td>554</td>
<td>554</td>
<td>554</td>
<td>1.83</td>
<td>-</td>
</tr>
<tr>
<td>H1</td>
<td>125</td>
<td>207</td>
<td>295</td>
<td>0.69</td>
<td>62.33</td>
</tr>
<tr>
<td>H2</td>
<td>28</td>
<td>56</td>
<td>69</td>
<td>0.19</td>
<td>72.92</td>
</tr>
<tr>
<td>H3</td>
<td>12</td>
<td>24</td>
<td>54</td>
<td>0.08</td>
<td>57.52</td>
</tr>
<tr>
<td>H4</td>
<td>4</td>
<td>8</td>
<td>15</td>
<td>0.03</td>
<td>64.79</td>
</tr>
<tr>
<td>H5</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>0.01</td>
<td>52.66</td>
</tr>
<tr>
<td>H6</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>0.01</td>
<td>58.50</td>
</tr>
</tbody>
</table>
Ratio of minimum values for different values of k, QI size and height

|       | QI = 3 | |QI| =4 | QI = 5 | |QI| =6 |
|-------|--------|--------|--------|--------|--------|--------|
|       | min(k=10) / min (k=3) | min(k=25) / min(k=10) | min(k=10) / min (k=3) | min(k=25) / min(k=10) | min(k=10) / min (k=3) | min(k=25) / min(k=10) |
| H0    | 3,47   | 2,38   | 2,86   | 1,56   | 1,81   | 1,37   |
| H1    | 4,18   | 2,27   | 3,14   | 2,02   | 2,42   | 1,51   |
| H2    | 6,07   | 3,59   | 3,97   | 2,28   | 2,84   | 1,73   |
| H3    | 4,25   | 3,82   | 4,75   | 2,27   | 2,98   | 1,77   |
| H4    | 7      | 2      | 6,07   | 3,59   | 3,8    | 1,88   |
| H5    | 2      | 7      | 4,25   | 3,12   | 4,22   | 2,06   |
|       |        |        |        |        |        |        |
## Avg # suppressed tuples for various QI sizes

|       | | QI | =3 |       | | QI | =4 |       | | QI | =5 |       | | QI | =6 |
|-------|-------|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|-------|-----|
|       | Avg   | Avg % over full | Avg | Avg % over full | Avg | Avg % over full | Avg | Avg % over full |
| H0    | 554,0 | 1,8 | 3297,0 | 10,9 | 10458,0 | 34,7 | 15318,0 | 50,8 |
| H1    | 208,7 | 0,7 | 1847,8 | 6,1 | 7795,2 | 25,8 | 12808,7 | 42,5 |
| H2    | 56,5  | 0,2 | 868,6  | 2,9 | 5537,1 | 18,4 | 10369,3 | 34,4 |
| H3    | 24,0  | 0,1 | 354,3  | 1,2 | 3711,9 | 12,3 | 8105,1  | 26,9 |
| H4    | 8,5   | 0,0 | 121,0  | 0,4 | 2296,3 | 7,6  | 6036,7  | 20,0 |
| H5    | 4,0   | 0,0 | 42,9   | 0,1 | 1295,1 | 4,3  | 4255,2  | 14,1 |
| H6    | 1,7   | 0,0 | 15,1   | 0,0 | 644,3  | 2,1  | 2803,0  | 9,3  |
| H7    | 0,7   | 0,0 | 6,1    | 0,0 | 283,0  | 0,9  | 1703,8  | 5,6  |
| H8    | 0,0   | 0,0 | 2,1    | 0,0 | 110,4  | 0,4  | 941,1   | 3,1  |
| H9    | 0,0   | 0,0 | 0,4    | 0,0 | 40,5   | 0,1  | 465,5   | 1,5  |
### Min # of suppressed tuples for various QI sizes

|       | $|QI| = 3$ | $|QI| = 4$ | $|QI| = 5$ | $|QI| = 6$ |
|-------|---------|---------|---------|---------|
| H0    | 554     | 3297    | 10458   | 15318   |
| H1    | 125     | 1042    | 4514    | 8304    |
| H2    | 28      | 318     | 2169    | 4901    |
| H3    | 12      | 110     | 1123    | 2867    |
| H4    | 4       | 28      | 716     | 1941    |
| H5    | 1       | 12      | 322     | 1177    |
| H6    | 0       | 4       | 108     | 629     |
| H7    | 0       | 0       | 41      | 354     |
| H8    | 0       | 0       | 8       | 155     |
| H9    | 0       | 0       | 2       | 33      |
Effect of $k$ to suppression ($\text{QI}=3$)

<table>
<thead>
<tr>
<th>$</th>
<th>\text{QI}</th>
<th>=3$ (lattice up to height H6)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k=3$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>min</td>
<td>avg</td>
</tr>
<tr>
<td>H0</td>
<td>554</td>
<td>554</td>
</tr>
<tr>
<td>H1</td>
<td>125</td>
<td>209</td>
</tr>
<tr>
<td>H2</td>
<td>28</td>
<td>57</td>
</tr>
<tr>
<td>H3</td>
<td>12</td>
<td>24</td>
</tr>
<tr>
<td>H4</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>H5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>H6</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>
Auxiliary slides

FULL LATTICE RESULTS
Experiments for variant k

• Small QI (QI=3) results in low-level solutions; an increase of k results in faster detection of the solution with less searches.

• When we fail to find an exact answer, the increase of k results in an increase of the nodes that we visit.

• Every increment of QI by one practically results in an increase of the visited nodes by a factor of 5

• In all cases, the time needed to detect the solutions ranges in 1-8 milliseconds!
Experiments for variant height

• Small QI pushes the solution low and returns exact answers

• Concerning approximate solutions, the higher $v_{\text{max}}$ is placed by the query, the less nodes we visit to find the approximate answer.

• The size of QI is again the main factor for the number of nodes visited.

• All times are again in the previous range!
Experiments for variant maxSupp

• Small QI size (QI=3) allow the detection of exact answers: in this case, as maxSupp increases the number of visited nodes increases too.

• When approximate solutions are involved, the increase of maxSupp results in a decrease of the nodes visited (remember: increasing maxSupp allows more tuples to be deleted, so it is easier to obtain low level solutions).

• The size of QI is the main factor for the number of nodes visited.

• All times are again in the previous range!
Experiments for $l$-diversity

• Similarly to k-anonymity, experiments for $l$-diversity ($l$ in 3, 6, 9) have been conducted.

• Similar results with k-anonymity:
  – Time is always within 1-8 msec
  – The QI size is always the most dominant factor for the number of visited nodes.
  – Small QI sizes allow the achievement of exact answers.
  – Approximations behave in accordance to the case of k-anonymity (for variant $l$, height, and maxsupp).
Experiments with the IPUMS data set

• Parameters

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>3, 30, 50, 100, 150</td>
</tr>
<tr>
<td>l</td>
<td>3, 6, 10</td>
</tr>
<tr>
<td>Topmost node</td>
<td>low (1010), middle-low (2110), middle (2220)</td>
</tr>
<tr>
<td>MaxSupp</td>
<td>600, 6000, 60000</td>
</tr>
</tbody>
</table>

• Results are similar to the ones of the Adult data set (attn: here, we have small QI size, QI = 4)
  • When is k or l are small, then we can have exact answers for both k-anonymity and l-diversity (again: the small QI is important here and allows many exact answers).
  • As the height of vmax increases, exact answers slow down when they are present (we must descent the sublattice), but the approximations are detected faster, because we are already high in the lattice.
  • As the value of maxSupp increases, the approximations are detected faster because the solution is found low and we do not have to climb more.
  • Time ranges in 2-4 milliseconds for k-anonymity and 2-8 milliseconds for l-diversity
Variant $k$

| $|Q| = 3$ | $|Q| = 4$ | $|Q| = 5$ | $|Q| = 6$ |
|---|---|---|---|
| ![Graph](image1) | ![Graph](image2) | ![Graph](image3) | ![Graph](image4) |

**Parameters:** 3, 321, 211

**Solution:** Move down find id: 4 supp tuples: 125 level: 100

**Parameters:** 10, 321, 211

**Solution:** Move down find id: 5 supp tuples: 170 level: 110

**Parameters:** 50, 321, 211

**Solution:** Move down find id: 23 supp tuples: 251 level: 211

**Parameters:** 3, 321, 211

**Solution:** Move down find id: 56 supp tuples: 283 level: 1011

**Parameters:** 10, 321, 211

**Solution:** Rlx1: id: 551 supp tuples: 656 level: 21012
Rlx2: id: 503 supp tuples: 108 level: 10312
Rlx3: No solution.

**Parameters:** 50, 321, 211

**Solution:** Rlx1: id: 65 supp tuples: 655 level: 2011
Rlx2: id: 17 supp tuples: 170 level: 1210
Rlx3: id: 65 k: 5 Supp tups: 301

**Parameters:** 3, 321, 211

**Solution:** Rlx1: id: 65 supp tuples: 4077 level: 2112
Rlx2: id: 107 supp tuples: 137 level: 1212
Rlx3: id: 65 k: 5 Supp tups: 301

**Parameters:** 3, 321, 211

**Solution:** Rlx1: id: 551 supp tuples: 9214 level: 21012
Rlx2: id: 636 supp tuples: 169 level: 40122
Relaxation3: No solution.

**Parameters:** 3, 321, 211

**Solution:**
Rlx1: id: 2591 supp tuples: 1611 level: 211012
Rlx2: id: 763 supp tuples: 155 level: 400202
Rlx3: No solution.

**Parameters:** 50, 321, 211

**Solution:**
Rlx1: id: 2591 supp tuples: 5362 level: 211012
Rlx2: id: 2171 supp tuples: 285 level: 401211
Rlx3: No solution.

**Parameters:** 50, 321, 211

**Solution:**
Rlx1: id: 2591 supp tuples: 15106 level: 211012
Rlx2: id: 2812 supp tuples: 137 level: 40122
Rlx3: No solution.
Variant constraint on upper levels

<table>
<thead>
<tr>
<th>QI</th>
<th>=3</th>
<th>QI</th>
<th>=4</th>
<th>QI</th>
<th>=5</th>
<th>QI</th>
<th>=6</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Graphs" /></td>
<td><img src="image2.png" alt="Graphs" /></td>
<td><img src="image3.png" alt="Graphs" /></td>
<td><img src="image4.png" alt="Graphs" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Parameters:** 10, 321, 101  
**Solution:** Move down find, id: 19 supp tuples:257 level:101

**Parameters:** 10, 321, 211  
**Solution:** Move down find, id: 5 supp tuples:170 level:110

**Parameters:** 10, 321, 212  
**Solution:** Move down find, id: 5 supp tuples:170 level:110

**Parameters:** 10, 321, 1001  
**Solution:**  
Rlx1: id:55 supp tuples:2349 lvl:1001  
Rlx2: id:17 supp tuples:170 lvl:1210  
Rlx3: No solution

**Parameters:** 10, 321, 2101  
**Solution:**  
Rlx2: id:17 supp tuples:170 lvl:1210  
Rlx3: id:55 k:5 Supp tuples:301

**Parameters:** 10, 321, 2112  
**Solution:** Move down find, id: 59 supp tuples:285 level:1111

**Parameters:** 10, 321, 11001  
**Solution:**  
Rlx1: id:280 supp tuples:8169 lvl:11001  
Rlx2: id:504 supp tuples:230 lvl:10222  
Rlx3: No solution

**Parameters:** 10, 321, 21012  
**Solution:**  
Rlx1: id:551 supp tuples:2533 lvl:21012  
Rlx2: id:504 supp tuples:230 lvl:10222  
Rlx3: No solution

**Parameters:** 10, 321, 22112  
**Solution:**  
Rlx1: id:563 supp tuples:369 lvl:22112  
Rlx2: id:635 supp tuples:60 lvl:40112  
Rlx3: id:563 k:9 Supp tuples:315

**Parameters:** 10, 321, 22112  
**Solution:**  
Rlx1: id:2623 supp tuples:712 lvl:22112  
Rlx2: id:2611 supp tuples:54 lvl:40112  
Rlx3: id:2623 k:5 Supp tuples:298
Variant max_supp

| |Q| = 3 | |Q| = 4 | |Q| = 5 | |Q| = 6 |
|---|---|---|---|---|---|
| **time (ms)** | | | | |
| 32 | 321 | 3216 | 32 | 321 | 3216 | 32 | 321 | 3216 | 32 | 321 | 3216 |

| **max_supp** | **max_supp** | **max_supp** | **max_supp** |

| **# visited nodes** | | | |
| 32 | 321 | 3216 | 32 | 321 | 3216 | 32 | 321 | 3216 | 32 | 321 | 3216 |

**Parameters**: 10, 32, 211
Solution:
Rlx1: id:23 supp tuples:55 level:211
Rlx2: id:11 supp tuples:28 level:310
Rlx3: id:23 k:7 Supp_sup:31

**Parameters**: 10, 321, 211
Solution:
Move down find, id: 5 supp tuples:170 level:1

**Parameters**: 10, 321, 211
Solution:
Move down find, id: 1 supp tuples:1921 level:000

**Parameters**: 10, 321, 21012
Solution:
Rlx1: id:551 supp tuples:2533 lvl:21012
Rlx2: id:504 supp tuples:230 lvl:1922
Rlx3: No solution

**Parameters**: 10, 321, 21012
Solution:
Rlx1: id:551 supp tuples:2533 lvl:21012
Rlx2: id:2812 supp tuples:21 lvl:40122
Rlx3: No solution

**Parameters**: 10, 321, 211012
Solution:
Rlx1: id:2591 supp tuples:5362 lvl:211012
Rlx2: id:2171 supp tuples:285 lvl:401211
Rlx3: No solution

**Parameters**: 10, 321, 21012
Solution:
Rlx1: id:2591 supp tuples:5362 lvl:211012
Rlx2: id:1525 supp tuples:122 lvl:400210
Rlx3: id:2591 k:8 Supp tuples:2915

119
### I-diversity over Adult. Variant value for the privacy criterion, I

<table>
<thead>
<tr>
<th>QI</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>time (ms)</td>
<td>![Graph for QI=3]</td>
<td>![Graph for QI=4]</td>
<td>![Graph for QI=5]</td>
<td>![Graph for QI=6]</td>
</tr>
<tr>
<td># of visited nodes</td>
<td>![Bar chart for QI=3]</td>
<td>![Bar chart for QI=4]</td>
<td>![Bar chart for QI=5]</td>
<td>![Bar chart for QI=6]</td>
</tr>
</tbody>
</table>

**Parameters:**

<table>
<thead>
<tr>
<th>QI</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters:</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
</tr>
<tr>
<td>Parameters:</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
</tr>
<tr>
<td>Parameters:</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
</tr>
<tr>
<td>Parameters:</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
</tr>
<tr>
<td>Solution:</td>
<td>Move down find. id: 503 supp tpls:244 lvl:10212</td>
<td>No solution</td>
<td>Move down find. id: 503 supp tpls:244 lvl:10212</td>
<td>No solution</td>
</tr>
<tr>
<td>Parameters:</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
</tr>
<tr>
<td>Solution:</td>
<td>Move down find. id: 504 supp tpls:302 lvl:10222</td>
<td>No solution</td>
<td>Move down find. id: 504 supp tpls:302 lvl:10222</td>
<td>No solution</td>
</tr>
<tr>
<td>Parameters:</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
</tr>
<tr>
<td>Solution:</td>
<td>Move down find. id: 504 supp tpls:302 lvl:10222</td>
<td>No solution</td>
<td>Move down find. id: 504 supp tpls:302 lvl:10222</td>
<td>No solution</td>
</tr>
<tr>
<td>Parameters:</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
</tr>
<tr>
<td>Solution:</td>
<td>Move down find. id: 504 supp tpls:302 lvl:10222</td>
<td>No solution</td>
<td>Move down find. id: 504 supp tpls:302 lvl:10222</td>
<td>No solution</td>
</tr>
<tr>
<td>Parameters:</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
<td>3, 321, 211</td>
</tr>
<tr>
<td>Solution:</td>
<td>Move down find. id: 504 supp tpls:302 lvl:10222</td>
<td>No solution</td>
<td>Move down find. id: 504 supp tpls:302 lvl:10222</td>
<td>No solution</td>
</tr>
</tbody>
</table>

---

120
**I-diversity - Adult** Variant constraint on upper levels

<table>
<thead>
<tr>
<th>QI</th>
<th>3</th>
<th>QI</th>
<th>4</th>
<th>QI</th>
<th>5</th>
<th>QI</th>
<th>6</th>
</tr>
</thead>
</table>

![Graphs and tables showing the distribution of time and number of visited nodes across different QI values.](image)

**Parameters: 6, 321, 1 0 1**

**Solution:**
Rlx1: id=19 supp tpls:368 lvl:1 0 1
Rlx2: id=6 supp tpls:54 lvl:1 2 0
Rlx3: id=19 l:5 Supp, tupp:266
Parameters: 6, 321, 2 1 1

**Parameters: 5, 321, 1 0 0 1**

**Solution:**
Rlx1: id=55 supp tpls:3081 lvl:1 0 0 1
Rlx2: id=18 supp tpls:54 lvl:1 2 2 0
Rlx3: No Solution
Parameters: 6, 321, 2 0 1 1

**Parameters: 6, 321, 1 1 0 0 1**

**Solution:**
Rlx1: id=280 supp tpls:9444 lvl:11001
Rlx2: id=504 supp tpls:302 lvl:10222
Rlx3: No Solution
Parameters: 6, 321, 2 1 0 1 2

**Parameters: 6, 321, 2 2 1 1 2**

**Solution:**
Rlx1: id=563 supp tpls:434 lvl:22112
Rlx2: id=635 supp tpls:63 lvl:40112
Rlx3: id=563 l:5 Supp, tupp:282

**Parameters: 6, 321, 1 1 0 0 1**

**Solution:**
Rlx1: id=336 sup tpl:14076 lvl:111001
Rlx2: id=2811 supp tpls:90 lvl:401212
Rlx3: No Solution
Parameters: 6, 321, 2 1 0 1 2

**Parameters: 6, 321, 2 2 1 1 2**

**Solution:**
Rlx1: id=2623 sup tpls:857 lvl:222112
Rlx2: id=2811 supp tpls:90 lvl:401212
Rlx3: id=2623 l:3 Supp, tupp:253
### 1-diversity - Adult Variant max supp

<table>
<thead>
<tr>
<th>QI</th>
<th>Time (ms)</th>
<th># of visited nodes</th>
<th># of delayed nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td><img src="image1" alt="Graph" /></td>
<td><img src="image2" alt="Graph" /></td>
<td><img src="image3" alt="Graph" /></td>
</tr>
<tr>
<td>4</td>
<td><img src="image4" alt="Graph" /></td>
<td><img src="image5" alt="Graph" /></td>
<td><img src="image6" alt="Graph" /></td>
</tr>
<tr>
<td>5</td>
<td><img src="image7" alt="Graph" /></td>
<td><img src="image8" alt="Graph" /></td>
<td><img src="image9" alt="Graph" /></td>
</tr>
<tr>
<td>6</td>
<td><img src="image10" alt="Graph" /></td>
<td><img src="image11" alt="Graph" /></td>
<td><img src="image12" alt="Graph" /></td>
</tr>
</tbody>
</table>

#### Parameters:

- **QI = 3**
  - Parameters: 6, 32, 211
  - Solution:
    - Rx1: id:65 supp tpls:1013 lvl:2 0 1 1
    - Rx2: id:41 supp tpls:12 lvl:4 0 1 1
    - Rx3: No Solution
    - Parameters: 321, 211
    - Solution:
      - Move down find, id: 20 supp tpls:70 lvl:111

- **QI = 4**
  - Parameters: 6, 32, 2011
  - Solution:
    - Rx1: id:65 supp tpls:1013 lvl:2 0 1 1
    - Rx2: id:41 supp tpls:12 lvl:4 0 1 1
    - Rx3: No Solution
    - Parameters: 321, 211
    - Solution:
      - Move down find, id: 20 supp tpls:70 lvl:111

- **QI = 5**
  - Parameters: 6, 32, 21012
  - Solution:
    - Rx1: id:551 supp tpls:3205 lvl:21012
    - Rx2: id:638 supp tpls:7 lvl:40321
    - Rx3: No Solution
    - Parameters: 321, 21012
    - Solution:
      - Move down find, id: 551 supp tpls:3205 lvl:21012

- **QI = 6**
  - Parameters: 6, 32, 211012
  - Solution:
    - Rx1: id:2591 supp tpls:6002 lvl:211012
    - Rx2: id:2823 supp tpls:7 lvl:40321
    - Rx3: No Solution
    - Parameters: 321, 211012
    - Solution:
      - Move down find, id: 2591 supp tpls:6002 lvl:211012
### K-anonymity IPUMS (Q=4)

<table>
<thead>
<tr>
<th>Variant k</th>
<th>Variant level</th>
<th>Variant max_supp</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="3-k.png" alt="Graph" /></td>
<td><img src="3-level.png" alt="Graph" /></td>
<td><img src="3-max.png" alt="Graph" /></td>
</tr>
</tbody>
</table>

**Parameters:** 3, 6000, 2 1 1 0  
**Solution:**  
Move down find, id: 6 supp tuples:4462 lvl:0100  
**Parameters:** 10, 6000, 2 1 1 0  
**Solution:**  
Move down find, id: 26 supp tpls:3778 lvl:1100  
**Parameters:** 50, 6000, 2 1 1 0  
**Solution:**  
Rlx1: id:47 supp tuples:9476 level:2 1 1 0  
Rlx2: id:36 supp tuples:2037 level:1 3 0 0  
Rlx3: id:47 k:34 supp tpls:5948  
**Parameters:** 100, 6000, 2 1 1 0  
**Solution:**  
Rlx1: id:47 supp tuples:19968 level:2 1 1 0  
Rlx2: id:36 supp tuples:4775 level:1 3 0 0  
Rlx3: id:47 k:34 Supp_tupp:5948  
**Parameters:** 150, 6000, 2 1 1 0  
**Solution:**  
Rlx1: id:47 supp tuples:32572 level:2 1 1 0  
Rlx2: id:86 supp tuples:2482 level:4 1 0 0  
Rlx3: id:47 k:34 Supp_tupp:5948  
**Parameters:** 50, 6000, 1 0 1 0  
**Solution:**  
Rlx1: id:22 supp tuples:110173 level:1 0 1 0  
Rlx2: id:36 supp tuples:2037 level:1 3 0 0  
Rlx3: id:22 k:3 Supp_tupp:4146  
**Parameters:** 50, 6000, 2 1 1 0  
**Solution:**  
Rlx1: id:47 supp tuples:9476 level:2 1 1 0  
Rlx2: id:36 supp tuples:2037 level:1 3 0 0  
Rlx3: id:47 k:34 Supp_tupp:5948  
**Parameters:** 50, 6000, 2 2 2 0  
**Solution:**  
Move down find, id: 51 supp tuples:4742 lvl:2 2 0 0  
**Parameters:** 50, 6000, 2 1 1 0  
**Solution:**  
Rlx1: id:47 supp tuples:9476 level:2 1 1 0  
Rlx2: id:36 supp tuples:2037 level:1 3 0 0  
Rlx3: id:47 k:34 Supp_tupp:5948  
**Parameters:** 50, 60000, 2 1 1 0  
**Solution:**  
Move down find, id: 26 supp tuples:30578 lvl:1 1 0 0
PARTIAL LATTICE RESULTS
## Grouping power for QI=6

<table>
<thead>
<tr>
<th>Variable</th>
<th>Level</th>
<th>Num Groups</th>
<th>Avg Group Size</th>
<th>relImp()</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>400000</td>
<td>3455</td>
<td>8.73</td>
<td>1.56</td>
</tr>
<tr>
<td></td>
<td>300000</td>
<td>5380</td>
<td>5.61</td>
<td>1.30</td>
</tr>
<tr>
<td></td>
<td>200000</td>
<td>7015</td>
<td>4.30</td>
<td>1.30</td>
</tr>
<tr>
<td></td>
<td>100000</td>
<td>9117</td>
<td>3.31</td>
<td>1.70</td>
</tr>
<tr>
<td></td>
<td>000000</td>
<td>15537</td>
<td>1.94</td>
<td>0.59</td>
</tr>
<tr>
<td>education</td>
<td>040000</td>
<td>8247</td>
<td>3.66</td>
<td>1.26</td>
</tr>
<tr>
<td></td>
<td>030000</td>
<td>10407</td>
<td>2.90</td>
<td>1.30</td>
</tr>
<tr>
<td></td>
<td>020000</td>
<td>13526</td>
<td>2.23</td>
<td>1.09</td>
</tr>
<tr>
<td></td>
<td>010000</td>
<td>14796</td>
<td>2.04</td>
<td>1.05</td>
</tr>
<tr>
<td></td>
<td>000000</td>
<td>15537</td>
<td>1.94</td>
<td>0.95</td>
</tr>
<tr>
<td>marital_status</td>
<td>003000</td>
<td>11190</td>
<td>2.70</td>
<td>1.16</td>
</tr>
<tr>
<td></td>
<td>002000</td>
<td>13018</td>
<td>2.32</td>
<td>1.14</td>
</tr>
<tr>
<td></td>
<td>001000</td>
<td>14855</td>
<td>2.03</td>
<td>1.05</td>
</tr>
<tr>
<td></td>
<td>000000</td>
<td>15537</td>
<td>1.94</td>
<td>0.96</td>
</tr>
<tr>
<td>occupation</td>
<td>000200</td>
<td>7932</td>
<td>3.80</td>
<td>1.38</td>
</tr>
<tr>
<td></td>
<td>000100</td>
<td>10975</td>
<td>2.75</td>
<td>1.42</td>
</tr>
<tr>
<td></td>
<td>000000</td>
<td>15537</td>
<td>1.94</td>
<td>0.71</td>
</tr>
<tr>
<td>race</td>
<td>000020</td>
<td>13478</td>
<td>2.24</td>
<td>1.13</td>
</tr>
<tr>
<td></td>
<td>000010</td>
<td>15210</td>
<td>1.98</td>
<td>1.02</td>
</tr>
<tr>
<td></td>
<td>000000</td>
<td>15537</td>
<td>1.94</td>
<td>0.98</td>
</tr>
<tr>
<td>work_class</td>
<td>000003</td>
<td>11790</td>
<td>2.56</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>000002</td>
<td>11798</td>
<td>2.56</td>
<td>1.24</td>
</tr>
<tr>
<td></td>
<td>000001</td>
<td>14668</td>
<td>2.06</td>
<td>1.06</td>
</tr>
<tr>
<td></td>
<td>000000</td>
<td>15537</td>
<td>1.94</td>
<td>0.94</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Level</th>
<th>relImp()</th>
</tr>
</thead>
<tbody>
<tr>
<td>age1</td>
<td>1.70</td>
<td></td>
</tr>
<tr>
<td>age4</td>
<td>1.56</td>
<td></td>
</tr>
<tr>
<td>occupation1</td>
<td>1.42</td>
<td></td>
</tr>
<tr>
<td>occupation2</td>
<td>1.38</td>
<td></td>
</tr>
<tr>
<td>age3</td>
<td>1.30</td>
<td></td>
</tr>
<tr>
<td>education3</td>
<td>1.30</td>
<td></td>
</tr>
<tr>
<td>age2</td>
<td>1.30</td>
<td></td>
</tr>
<tr>
<td>education4</td>
<td>1.26</td>
<td></td>
</tr>
<tr>
<td>work_class2</td>
<td>1.24</td>
<td></td>
</tr>
<tr>
<td>marital_status3</td>
<td>1.16</td>
<td></td>
</tr>
<tr>
<td>marital_status2</td>
<td>1.14</td>
<td></td>
</tr>
<tr>
<td>race2</td>
<td>1.13</td>
<td></td>
</tr>
<tr>
<td>education2</td>
<td>1.09</td>
<td></td>
</tr>
<tr>
<td>work_class1</td>
<td>1.06</td>
<td></td>
</tr>
<tr>
<td>education1</td>
<td>1.05</td>
<td></td>
</tr>
<tr>
<td>marital_status1</td>
<td>1.05</td>
<td></td>
</tr>
<tr>
<td>race1</td>
<td>1.02</td>
<td></td>
</tr>
<tr>
<td>work_class3</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>race0</td>
<td>0.98</td>
<td></td>
</tr>
<tr>
<td>marital_status0</td>
<td>0.96</td>
<td></td>
</tr>
<tr>
<td>education0</td>
<td>0.95</td>
<td></td>
</tr>
<tr>
<td>work_class0</td>
<td>0.94</td>
<td></td>
</tr>
<tr>
<td>occupation0</td>
<td>0.71</td>
<td></td>
</tr>
<tr>
<td>age0</td>
<td>0.59</td>
<td></td>
</tr>
</tbody>
</table>
Negotiation Experiments – other observations

• Small QI sizes achieve exact answers
• Approximate answers: when k increases, the number of nodes searched increases.
• As maxSupp increases the number of visited nodes drops.
• Again, QI size is the dominant factor of the execution time for detecting an exact answer.
  – Compared to the full lattice, the number of visited nodes is much smaller, of course. For example, in the partial lattice, the maximum number of visited nodes in any of our experiments has been 94 (compared to the 1792 visited nodes for the full lattice).