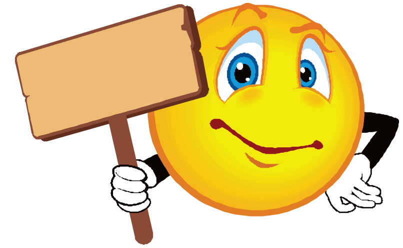
Online negotiation for privacy preserving data publishing

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

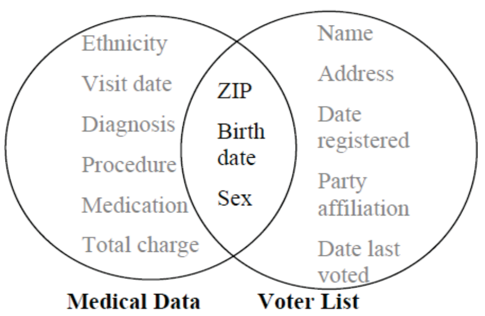
- Organizations (hospitals, ministries, internet providers, …) publicly release data concerning individual records (internet searches, medical records, …)
- Although data re stripped from identity-revealing attributes, it is still possible to identify individuals via various forms of attacks

Voter Registration Data

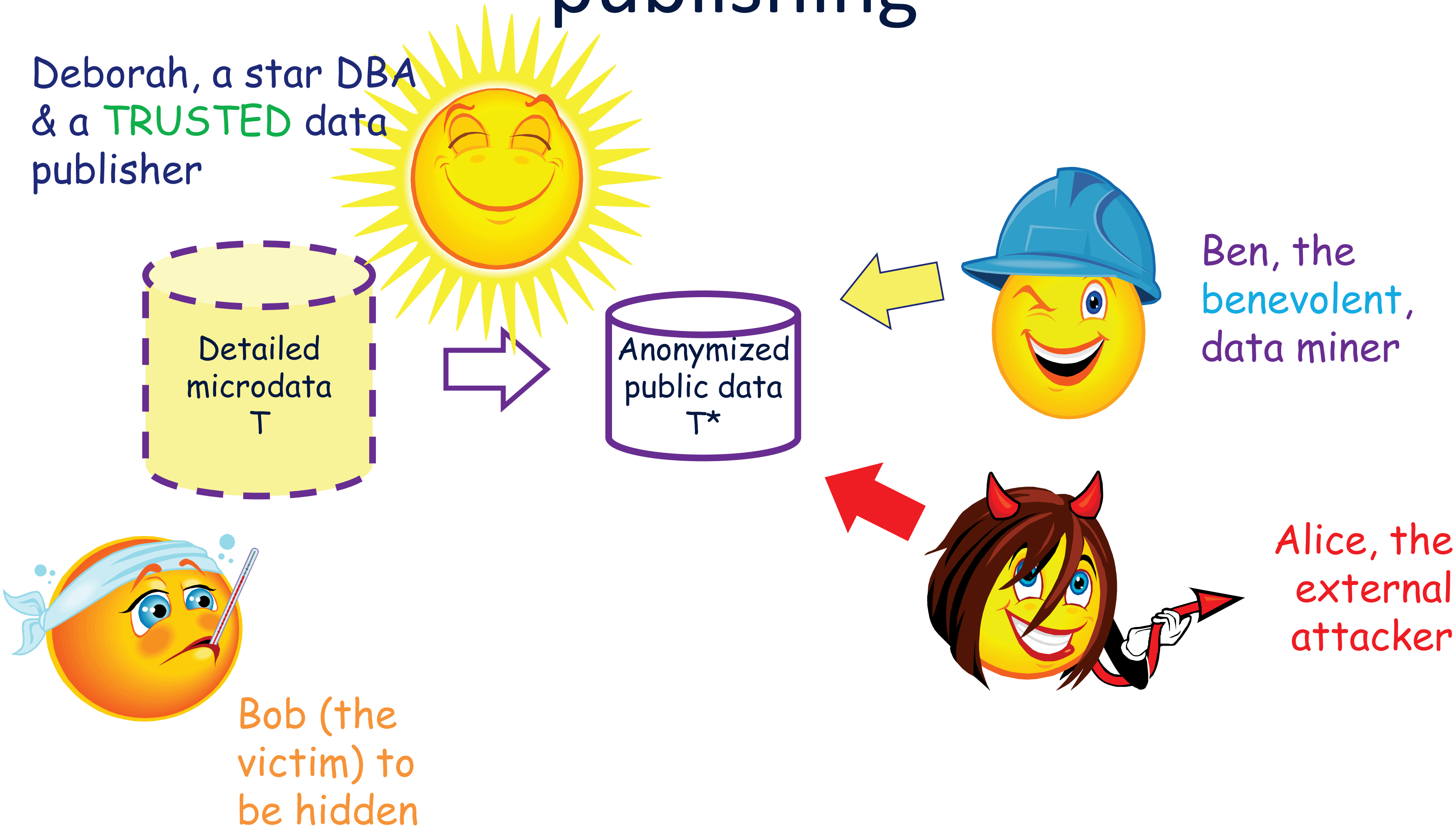
Name	Birthdate	Sex	Zipcode
Andre	1/21/76	Male	53715
Beth	1/10/81	Female	55410
Carol	10/1/44	Female	90210
Dan	2/21/84	Male	02174
Ellen	4/19/72	Female	02237

Hospital Patient Data

Birthdate	Sex	Zipcode	Disease
1/21/76	Male	53715	Flu
4/13/86	Female	53715	Hepatitis
2/28/76	Male	53703	Brochitis
1/21/76	Male	53703	Broken Arm
4/13/86	Female	53706	Sprained Ankle
2/28/76	Female	53706	Hang Nail



The context of privacy-preserving data publishing



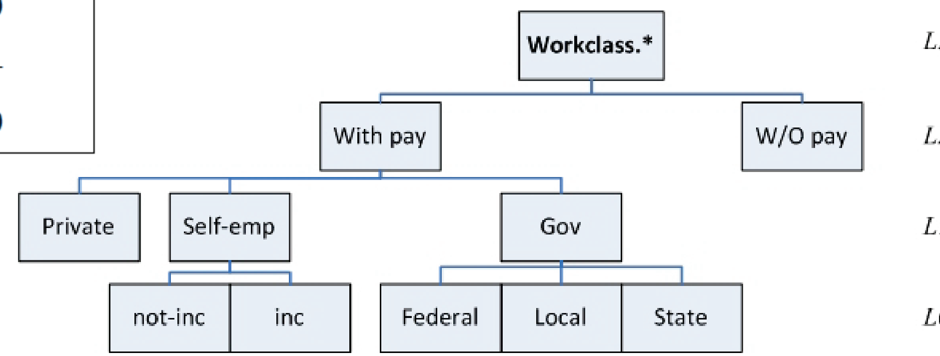
Anonymization

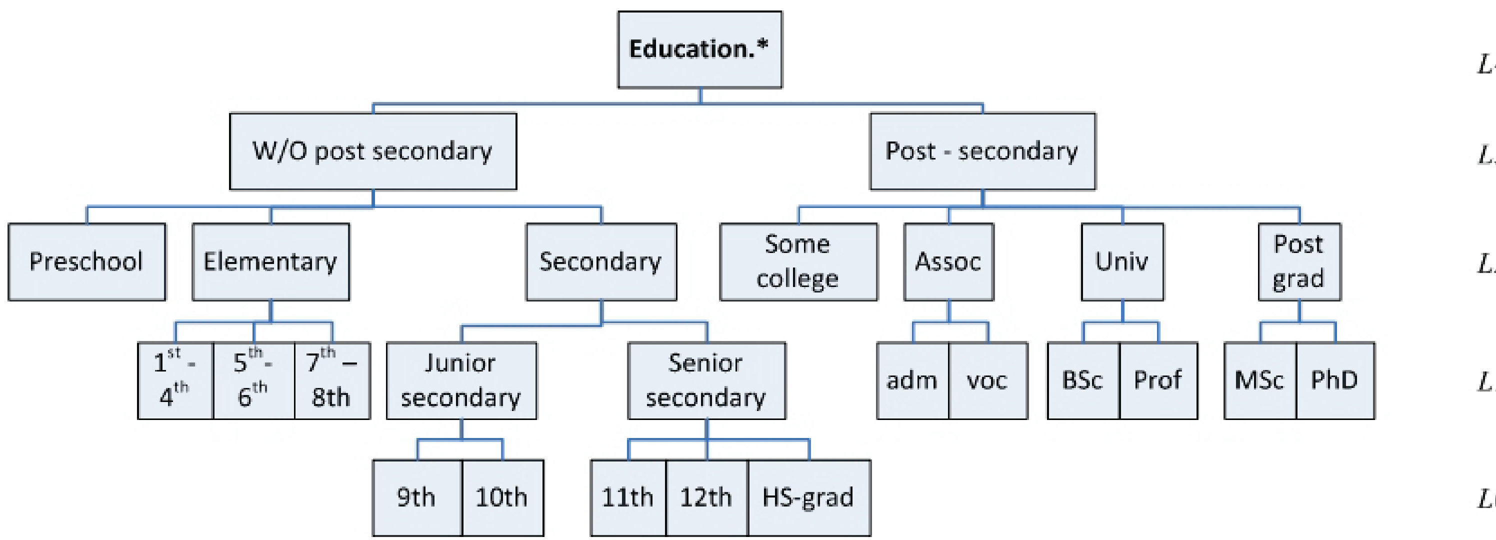
- To retain privacy one must:
  - Remove the attributes that directly identify individuals (name, SSN, …)
  - Organize the tuples and the cell values of the data set in such a way that:
    - The statistical properties of the data set are retained
    - The attacker cannot guess to which individual a 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,…)

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





General methods for Anonymization

- "Hide tuples in the crowd"
  - Generalization
  - Anatomization
- "Lies to the attacker, truth to the statistician"
  - Noise injection
  - Value perturbation

k-anonymity

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

Name	Age	Work_class	Education	Hours/week
Thales	**37-41**	**Private**	**Without-post-secondary**	**40**
Anaximander	**37-41**	**Private**	**Without-post-secondary**	**50**
Anaximenes	**37-41**	**Private**	**Without-post-secondary**	**40**
Pythagoras	**37-41**	**Private**	**Without-post-secondary**	**45**
Gorgias	27-31	Gov	Post-secondary	30
Heraclitus	27-31	Gov	Post-secondary	50
Empedocles	27-31	Gov	Post-secondary	60
Leucippus	*32-36*	*Self-emp*	*Post-secondary*	*50*
Democritus	*32-36*	*Self-emp*	*Post-secondary*	*54*
Protagoras	*32-36*	*Self-emp*	*Post-secondary*	*40*

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.

Naïve *l*-diversity

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

Name	Age	Work_class	Education	Hours/week
Thales	**37-41**	**Private**	**Without-post-secondary**	**40**
Anaximander	**37-41**	**Private**	**Without-post-secondary**	**50**
Anaximenes	**37-41**	**Private**	**Without-post-secondary**	**40**
Pythagoras	**37-41**	**Private**	**Without-post-secondary**	**45**
Gorgias	27-31	Gov	Post-secondary	30
Heraclitus	27-31	Gov	Post-secondary	50
Empedocles	27-31	Gov	Post-secondary	60
Leucippus	*32-36*	*Self-emp*	*Post-secondary*	*50*
Democritus	*32-36*	*Self-emp*	*Post-secondary*	*54*
Protagoras	*32-36*	*Self-emp*	*Post-secondary*	*40*

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.

Generalization vs suppression

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

Name	Age	Work_class	Education	Hours/week
Thales	37-41	Private	Without-post-secondary	40
Anaximander	37-41	Private	Without-post-secondary	50
Anaximenes	37-41	Private	Without-post-secondary	40
Pythagoras	37-41	Private	Without-post-secondary	45
Gorgias	27-31	Gov	Post-secondary	30
Heraclitus	27-31	Gov	Post-secondary	50
Empedocles	27-31	Gov	Post-secondary	60
Leucippus	*32-36*	*Self-emp*	*Post-secondary*	*50*
Democritus	*32-36*	*Self-emp*	*Post-secondary*	*54*
Protagoras	*32-36*	*Self-emp*	*Post-secondary*	*40*

This anonymization suppressed no tuples, and guarantees 3-anonymity.

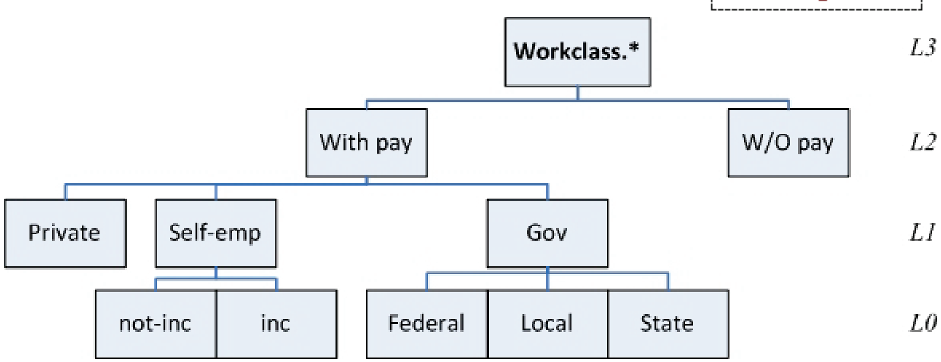
What if we want 4-anonymity?

Generalization vs suppression

Name	Age	Work_class	Education	Hours/week
Thales	37-41	Private	Without-post-secondary	40
Anaximander	37-41	Private	Without-post-secondary	50
Anaximenes	37-41	Private	Without-post-secondary	40
Pythagoras	37-41	Private	Without-post-secondary	45

Low height,
6 tuples suppressed

Name	Age	Work_class	Education	Hours/week
Thales	37-46	With pay	Without-post-secondary	40
Anaximander	37-46	With pay	Without-post-secondary	50
Anaximenes	37-46	With pay	Without-post-secondary	40
Pythagoras	37-46	With pay	Without-post-secondary	45
Gorgias	27-36	With pay	Post-secondary	30
Heraclitus	27-36	With pay	Post-secondary	50
Empedocles	27-36	With pay	Post-secondary	60
Leucippus	27-36	With pay	Post-secondary	50
Democritus	27-36	With pay	Post-secondary	54
Protagoras	27-36	With pay	Post-secondary	40



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

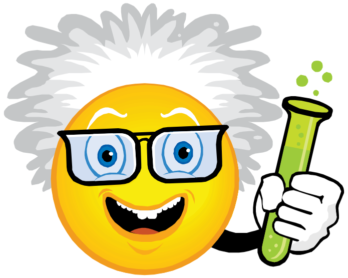
State-of-the-art

- 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 / …)

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?

Problem setting



- 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

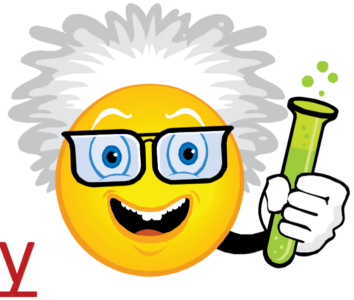
- The **combination of user time and guidance** via approximate answers allows the user to **negotiate** in user time the anonymization scheme for the published data set

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
- <u>Study of the relationship between suppression, generalization and privacy</u>
- 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

Quasi-identifiers: Age, Gender, Race, Marital Status, Education, Native country, Work Class, Occupation

Sensitive attribute: Hours per week

Attribute	Distinct Value
Age	72
Gender	2
Race	2
Marital Status	7
Education	16
Native country	41
Work Class	7
Occupation	14
Hours per week	94
Salary	2

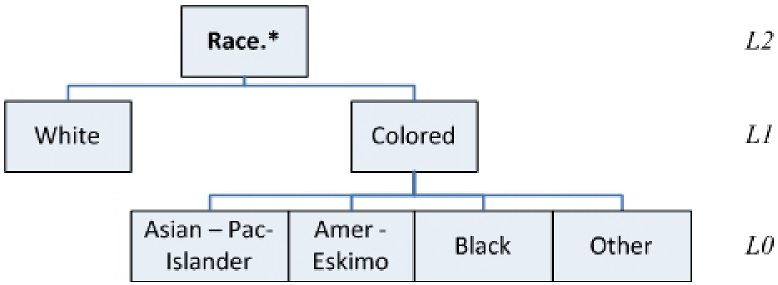
- IPUMS with 600.000 tuples

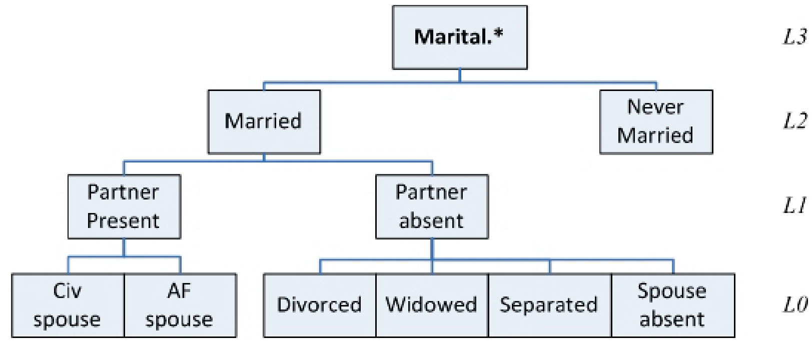
Quasi-identifiers: Age, Birthplace, Education, Gender

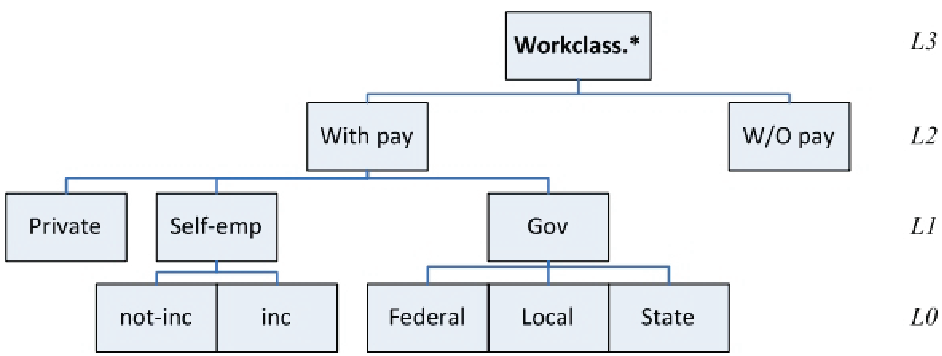
Sensitive attribute: Occupation

Attribute	Distinct Values
Age	
Birthplace	
Education	
Gender	
Occupation	

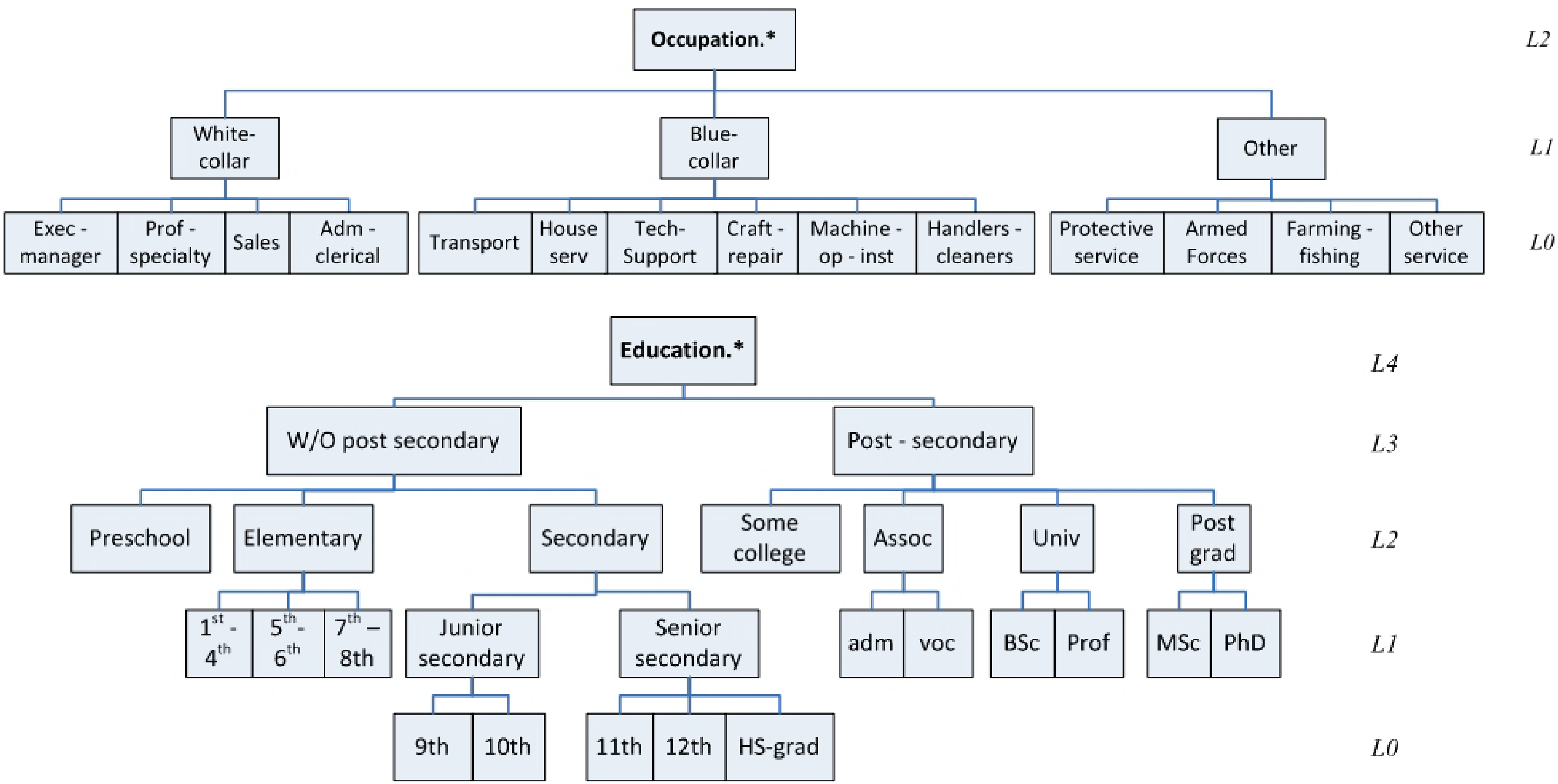
Generalization Hierarchies







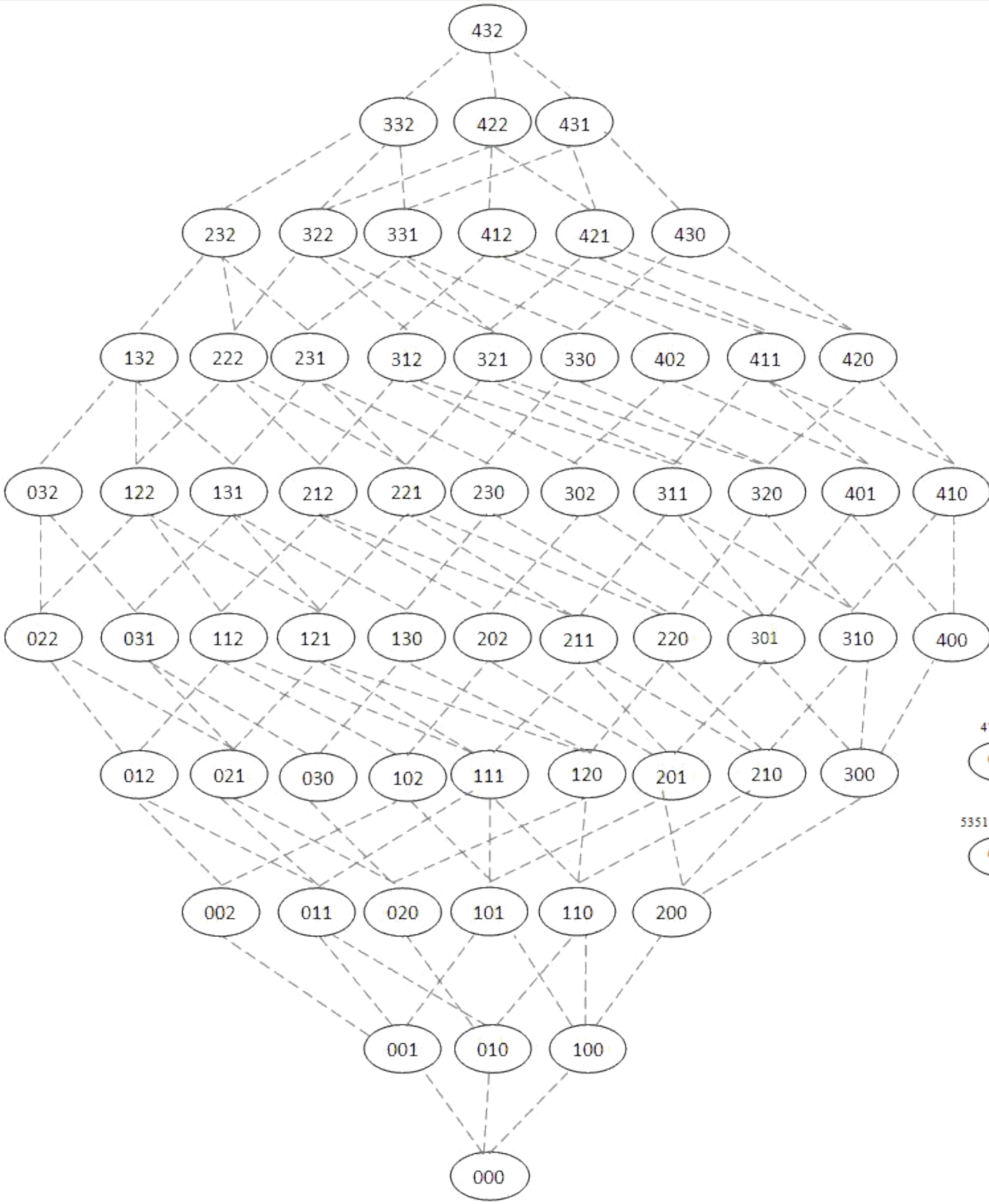
Generalization Hierarchies (2)



First Steps

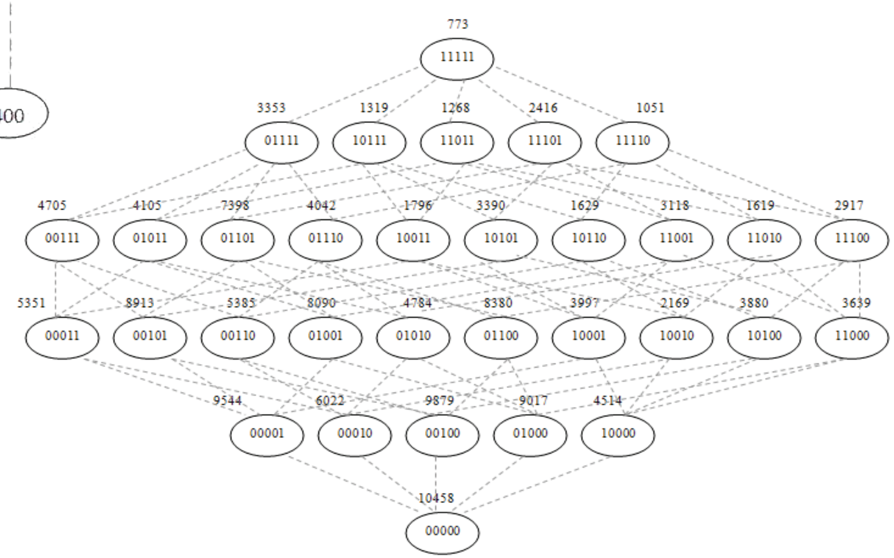
- We construct the **lattice of all the possible combinations**, for all the hierarchy levels of all the quasi-identifier attributes.
  - Every such combination (aka anonymization scheme) is a node in the lattice
  - A node v_{child} has an edge to a node v_{father} if
    - (a) for all but one of the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

Lattice & induced lattice



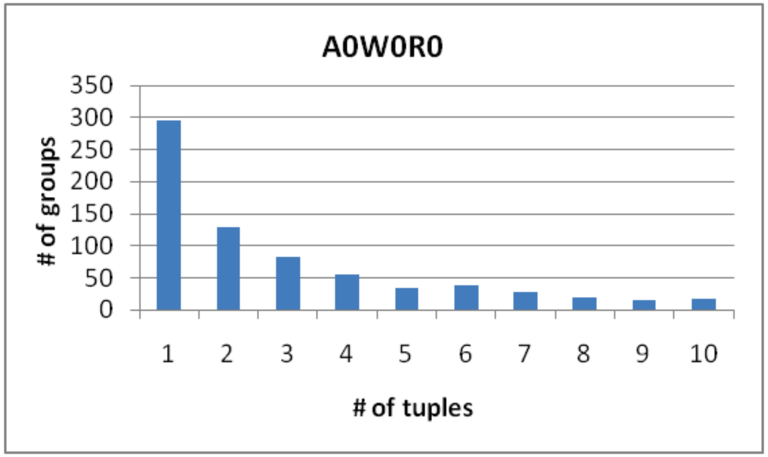
QI=3 – full lattice

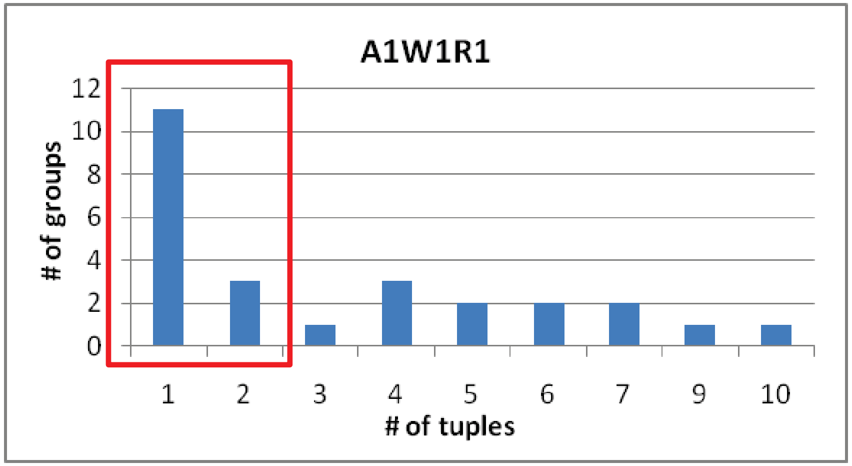
(Age, Race, Work_class)



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

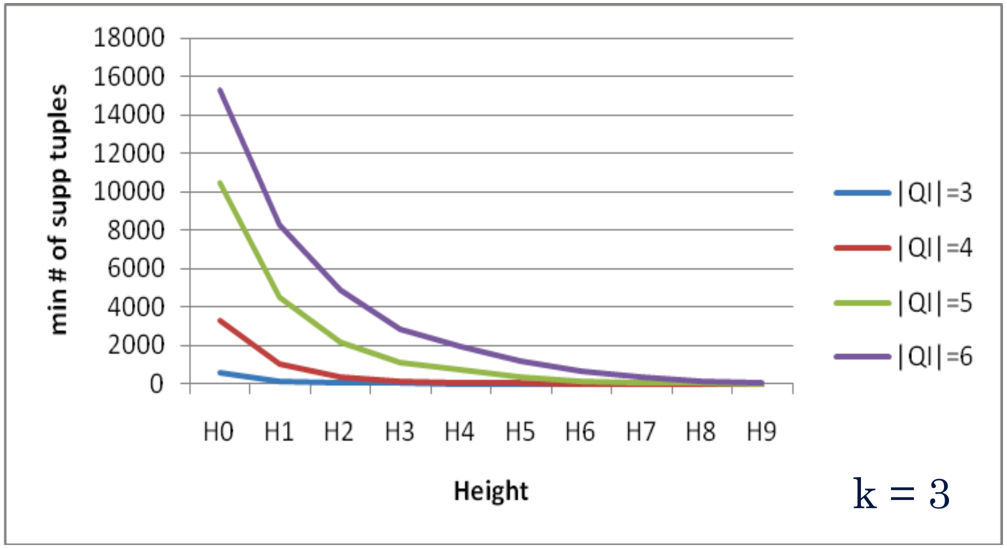
- For each node we need an auxiliary view and a query defined over it
  - k-anonymity

```
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.lavel0 and  Adult.Work_class=Work_class.level0
GROUP BY Age.level0,Work_class.level2,Race.level0)

SELECT plithos, count(*) FROM test
```

  - *l*-diversity

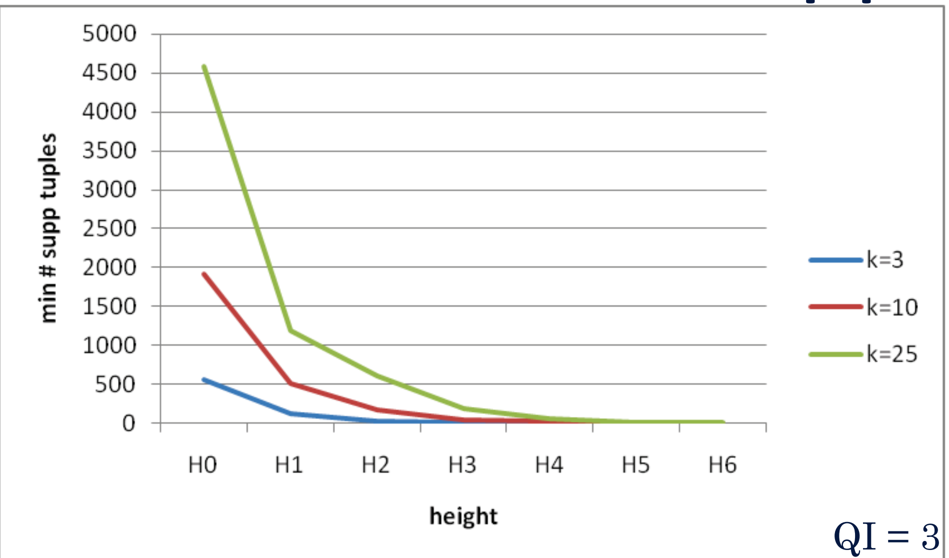
```
CREATE VIEW test(a,w,r,l,plithos) AS (
SELECT Age.level0,Work_class.level2,Race.level0,  distinct  hourse_per_week,count(*)
FROM Adult, Age,Race,Work_class
WHERE Adult.age=age.level0 and Adult.race=race.lavel0 and Adult.Work_class=Work_class.level0
GROUP BY Age.level0,Work_class.level2,Race.level0)

SELECT plithos, count(*) FROM test
```

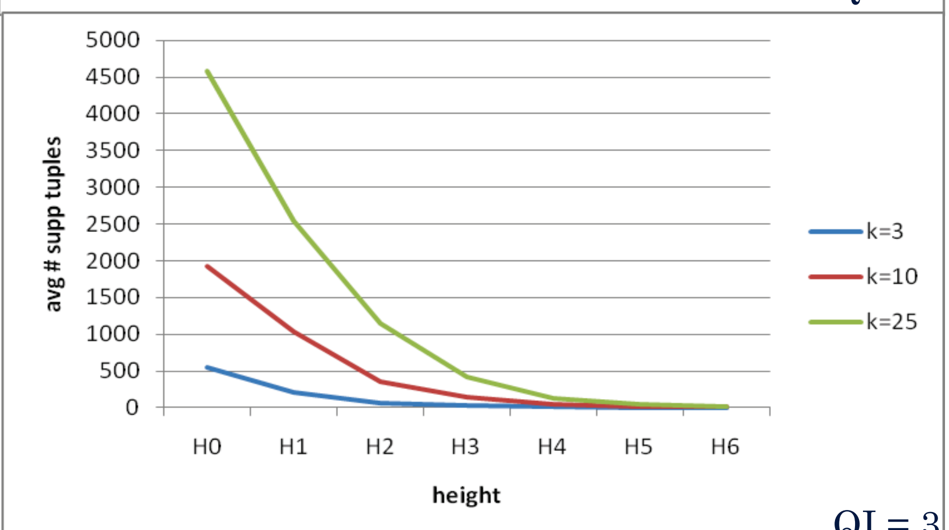
QI size is the most important factor for suppression



- 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

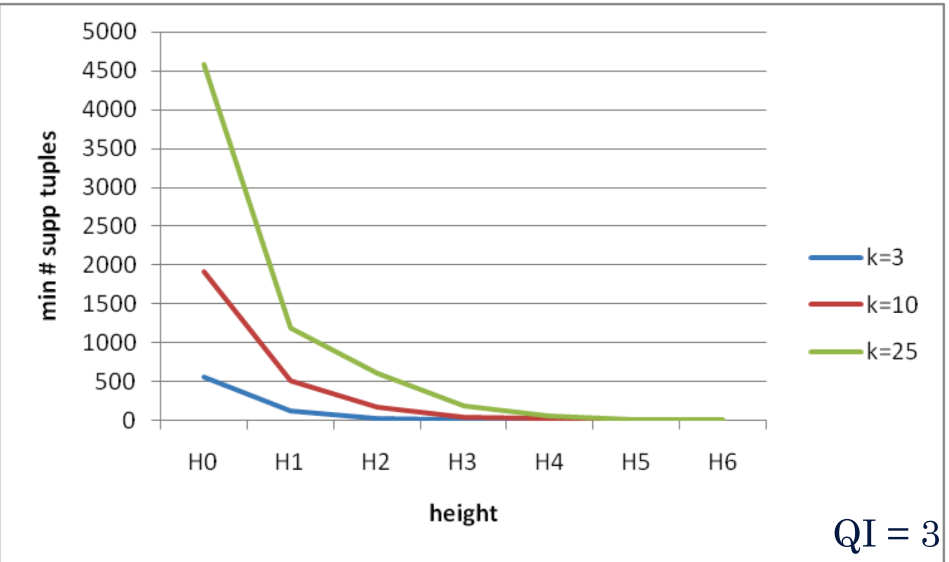


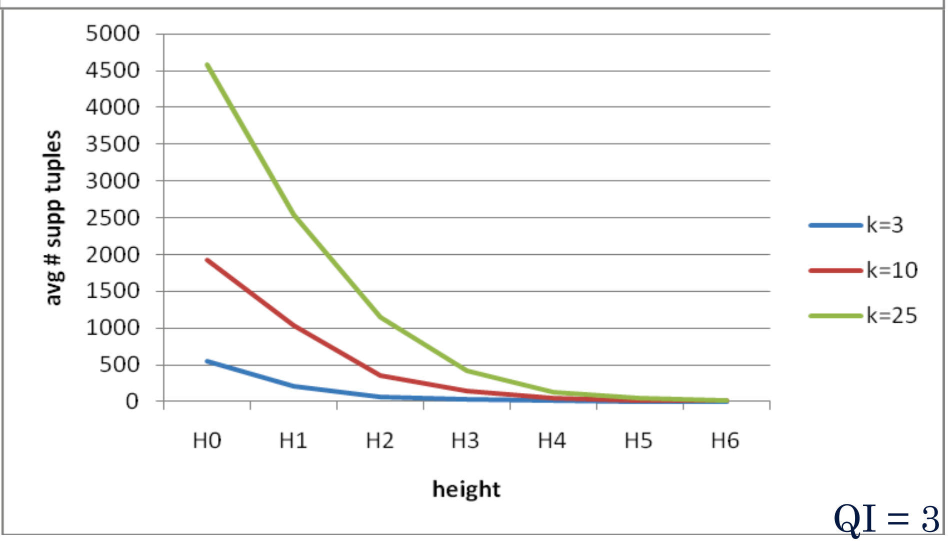


- As the height increases:
  - suppression drops quickly at small heights
  - the drop in suppression is less important in higher heights, where the number of suppressed tuples becomes statistically small and drops slowly.
- As the value for the privacy criterion (e.g., *k* in k-anonymity) increases, 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

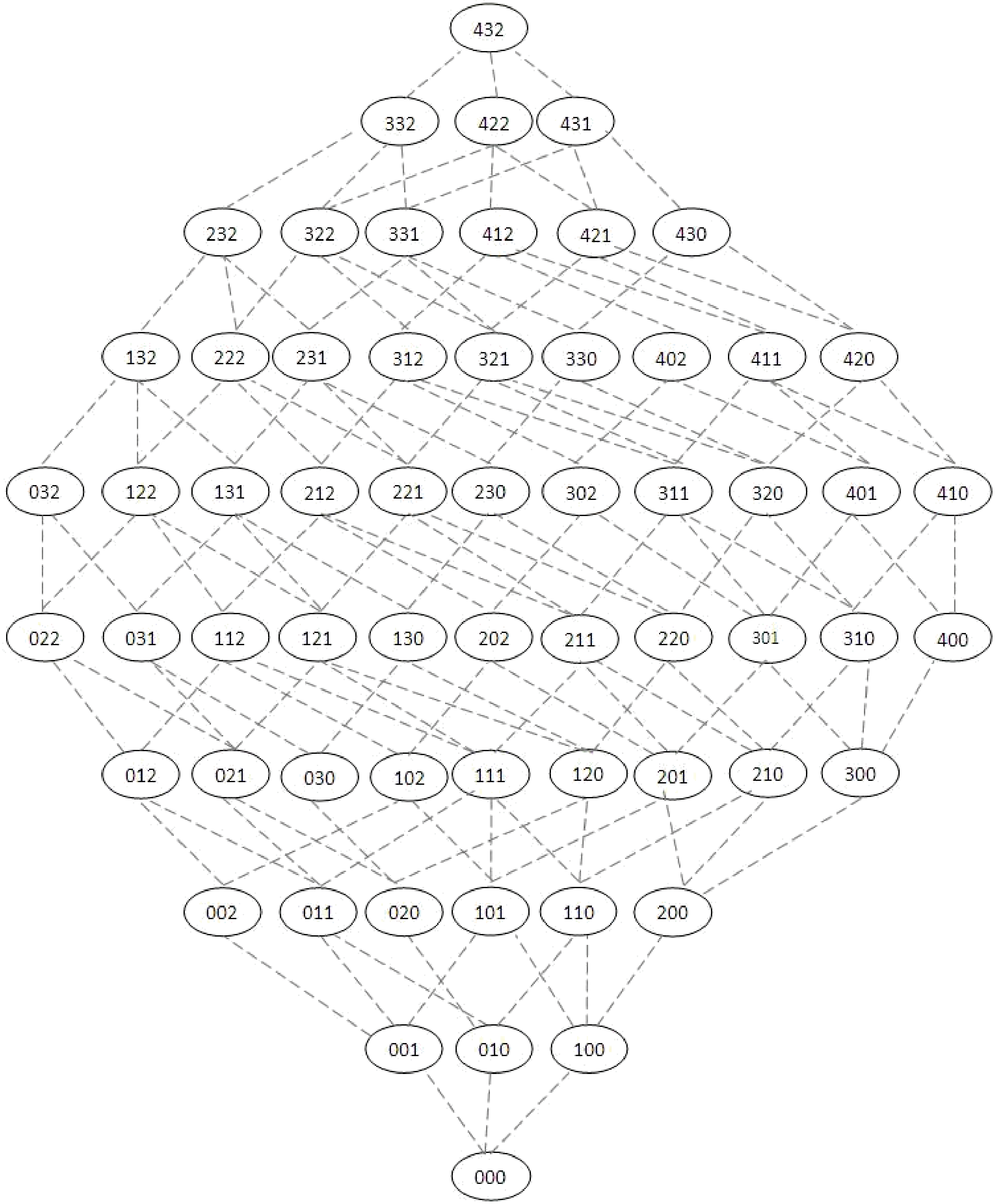
- Offline phase
  - Lattice construction
  - Histogram construction for all nodes of the lattice
- Online negotiation
  - User submits request with parameters (κόμβος vmax)
    - MaxSupp (maximum tolerable no. of suppressed tuples)
    - **h**=[h_1, …, h_n] (maximum tolerable height in the hierarchy 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

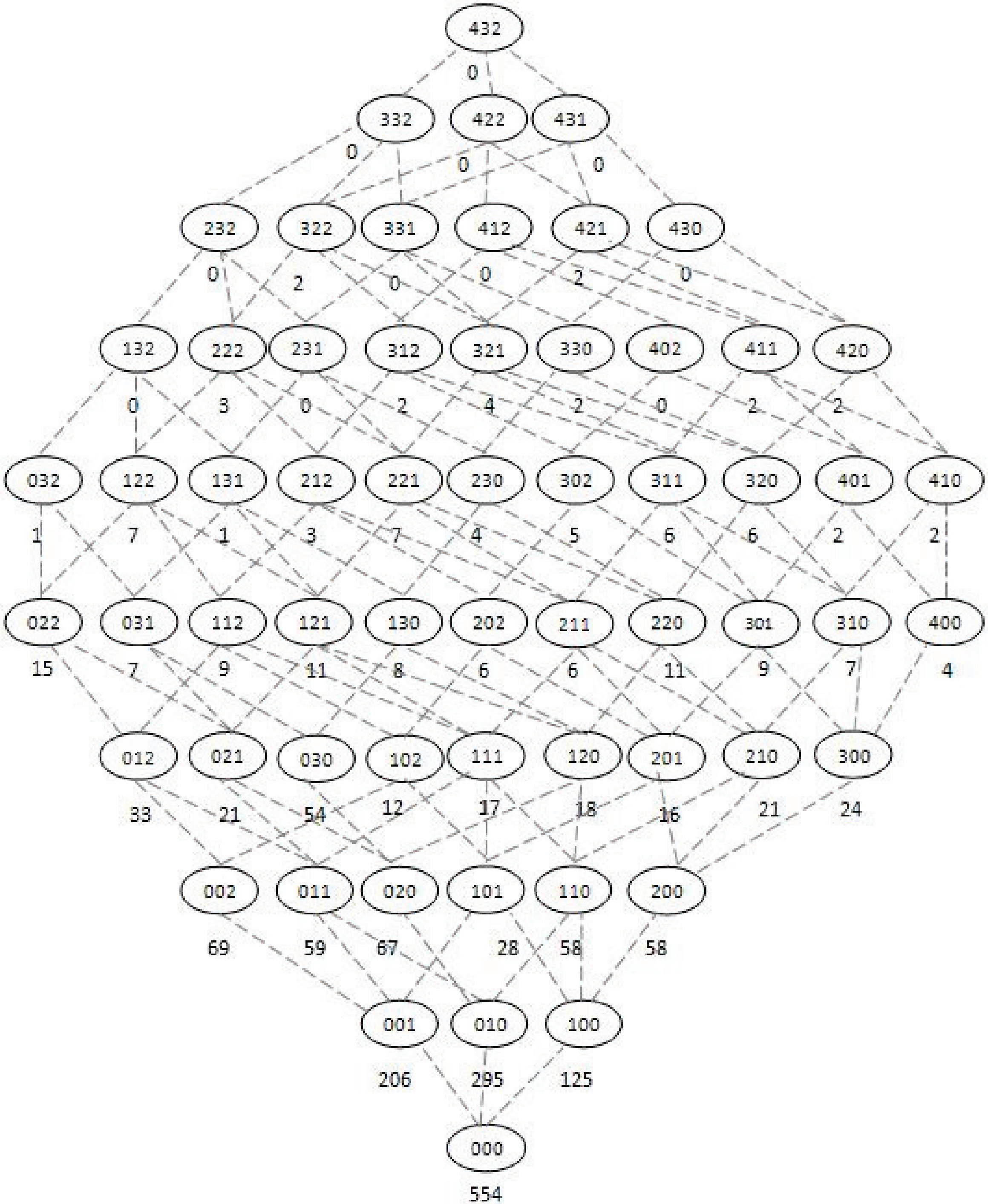
- Solution #1: Keep k, **h** fixed and relax maxSupp – i.e., search for the minimum amount of suppressed tuples while respecting k,h

- Solution #2: Keep k, maxSupp fixed and relax **h** – i.e., search for the minimum height while respecting k, maxSupp

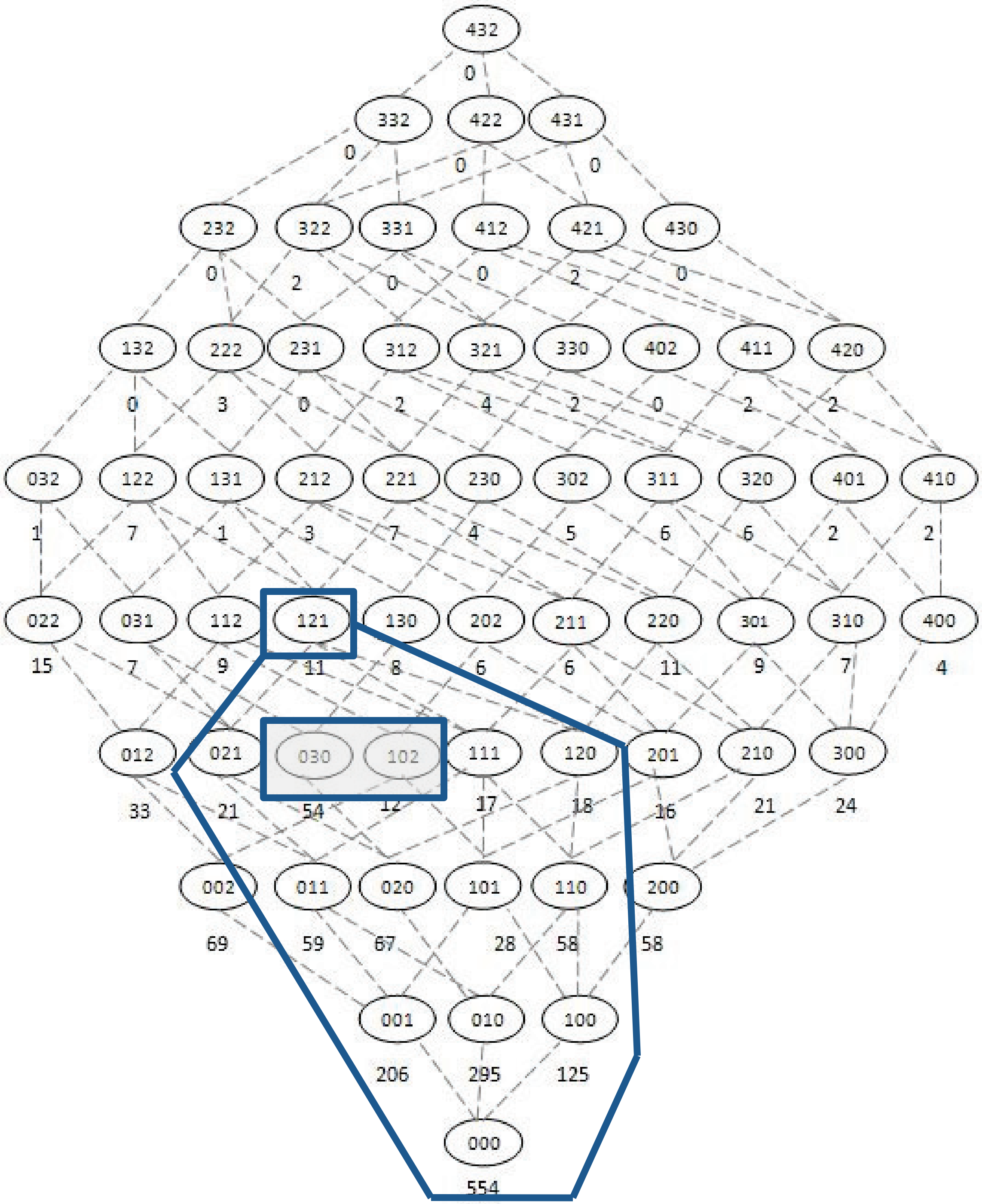
- Solution #3: Keep **h**, maxSupp fixed and relax k – i.e., search for the maximum possible k which respects **h**,maxSupp



This is the lattice for QI=3

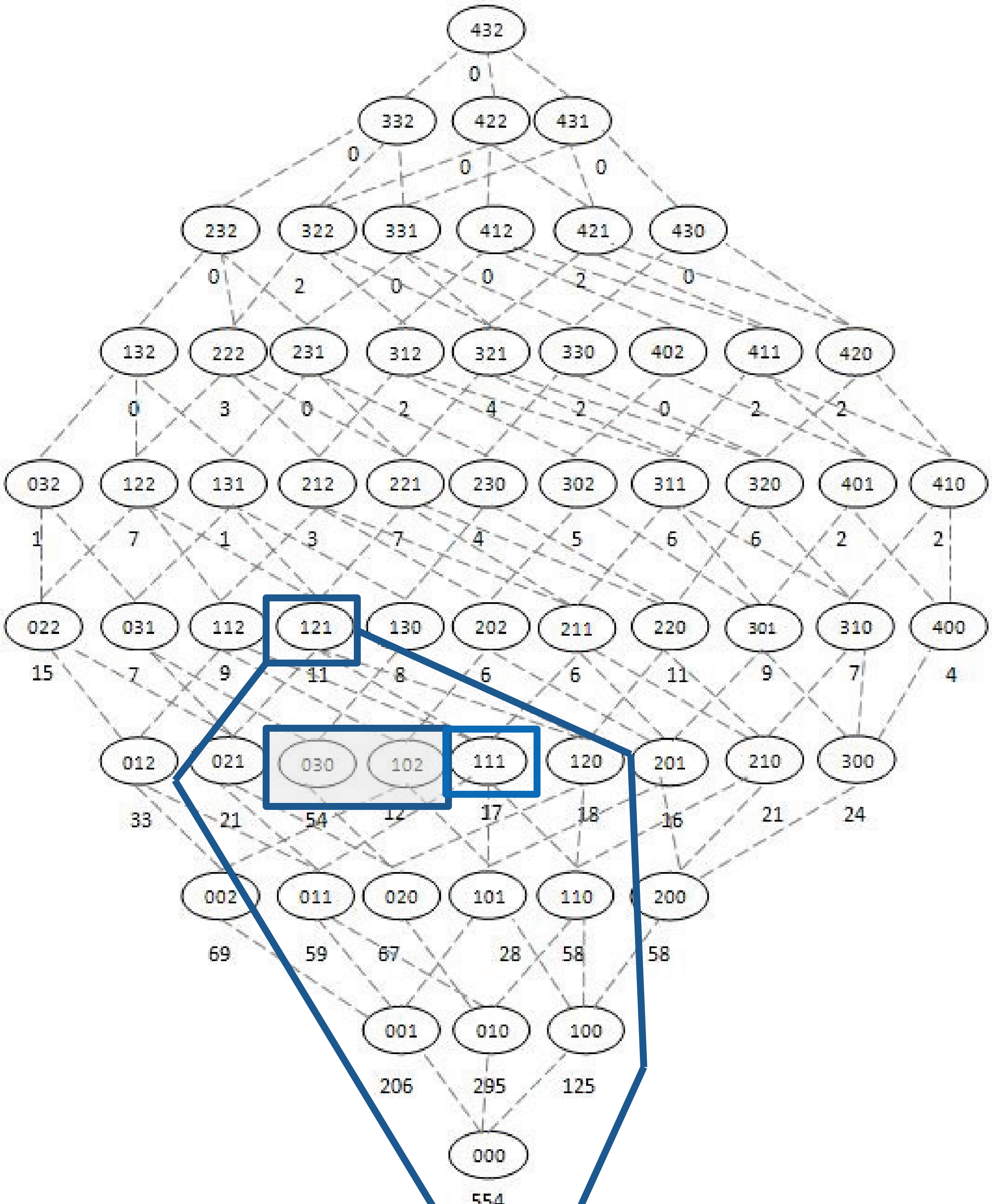


This is the lattice for QI=3 annotated with the number of suppressed tuples for k=3



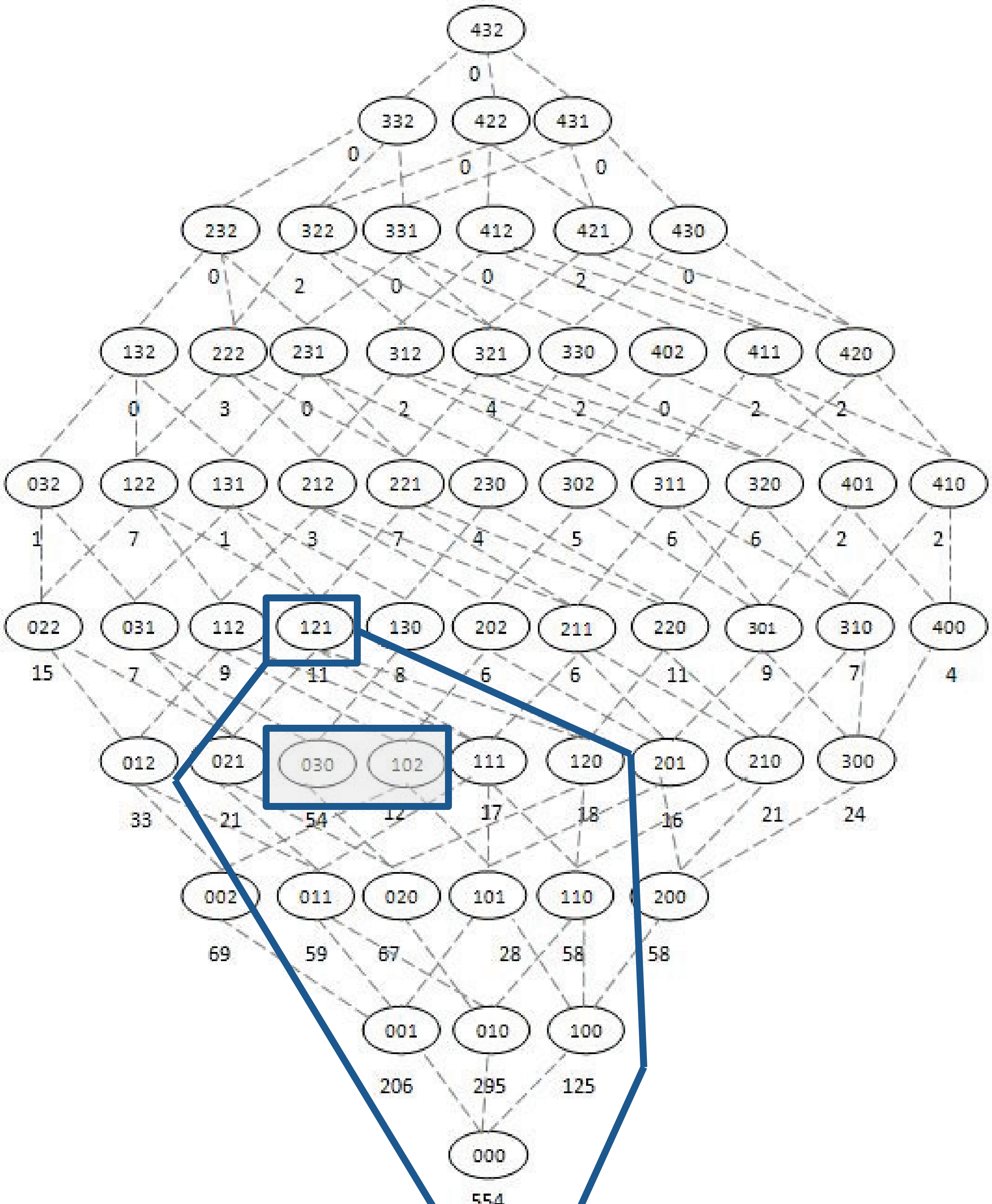
Assume the user requests:
h = 121
K = 3
MaxSupp = 20

Observe vmax 121



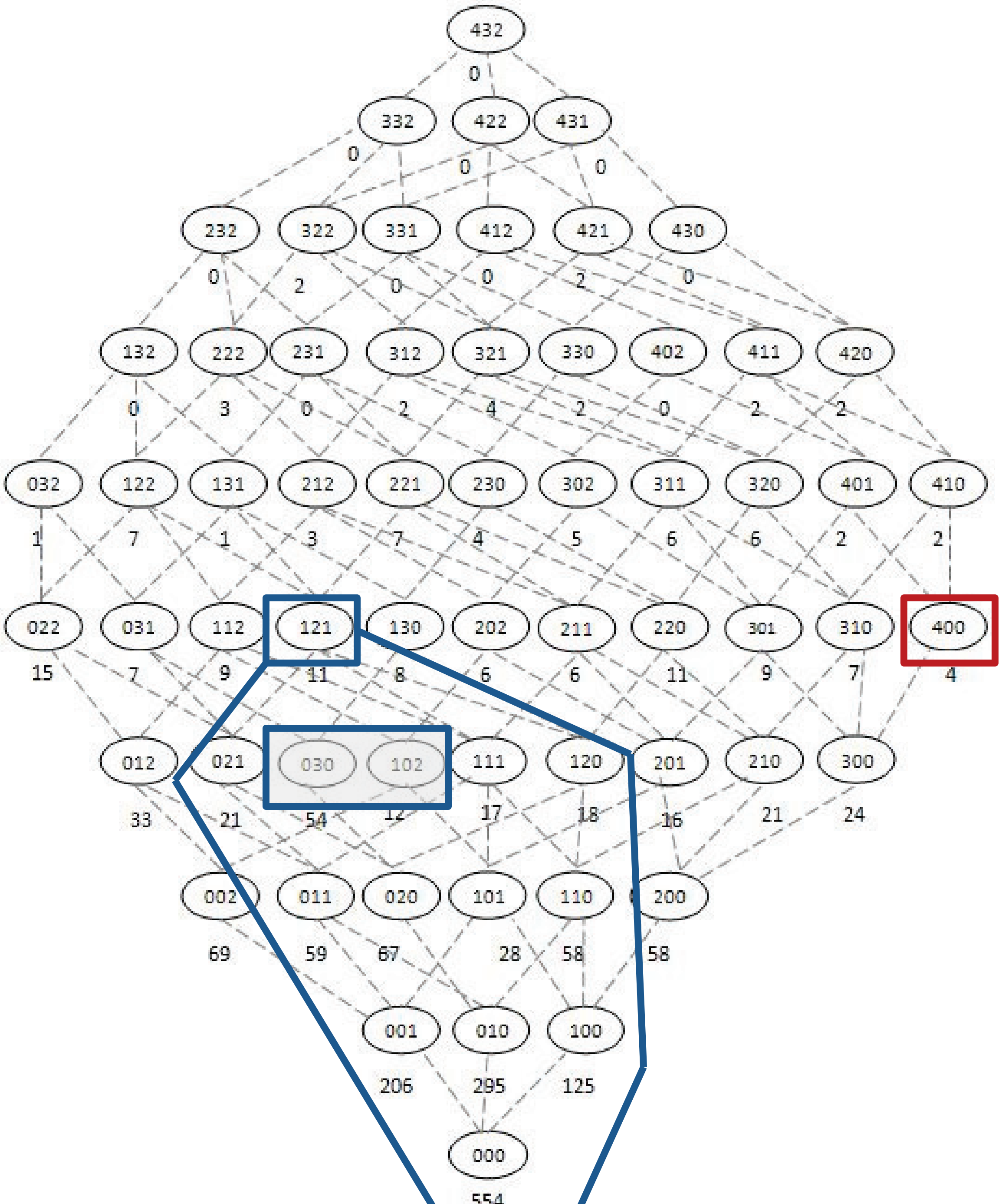
Assume the user requests:
h = 121
K = 3
MaxSupp = 20

The exact solution is 111 with #supp.=17



Assume the user requests:
h = 121
K = 3
MaxSupp = 8

Observe vmax 121: it fails to meet all three constraints



Assume the user requests:
h = 121
K = 3
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**, maxSupp) with the user constraints

1. Identify top-acceptable node v_{max}
2. If v_{max} answers the (k, **h**, maxSupp)
   – Search within the sublattice of vmax for the lowest possible node that also answers (k, **h**, maxSupp)
3. Else
   – Relax MaxSupp: stay at v_{max} (respect **h**) and find the suppression value for k (respect k)
   – Relax k: stay at v_{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_{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

```
Algorithm SimpleAnonymityNegotiation(L,k,h,MaxSupp)
In: Lattice L with the histograms for R,H, constraints for k, h, MaxSupp
Out: an exact solution s[v,k,h,supp] or s1,s2,s3, si=[v_i,k_i,h_i,supp_i]
Var: a 2D vector of candidate solutions Candidates[hmax][]
Begin
Let v_max be the node that corresponds to the constraint h ;
if v_max is visited then exit;
mark v_max as visited;
if (checkExactSolution(v_max,L,k,h,MaxSupp) == true){
        Candidates[height(v_max)] = Candidates[height(v_max)] ∪ {v_max};
        for all v_c in lower(v_max)
                ExactSublatticeSearch(v_c,L,k,h,MaxSupp,Candidates);
//when the recursion is over, the Candidates has the full list of nodes
//that can serve as candidate solutions
        minHeight = minimum height having Candidates[minHeight] != {};
        v_win = v in Candidates[minHeight] with the lowest possible suppression for k;
        return(v_win,k,minHeight,suppressed(v_win,k));
}
else{
        approxSol_1 = ApproximateMaxSupp(L,v_max,k,h,MaxSupp);
        approxSol_2 = ApproximateH(L,v_max,height(v_max),height(top),k,h,MaxSupp);
        approxSol_3 = ApproximateK(L,v_max,k,h,MaxSupp);

        return approxSol_1, approxSol_2, approxSol_3;
}
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)

```
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 γ 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 γ
  - 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 that both the exact and the approximate search are correct

Theoretical results

- If the user request is satisfied by v_{max} (exact search)
  - For any value α, the cumulative histogram for v_{max} has a smaller or equal value than the cumulative histogram for any node v in the descendants of v_{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_{max}
  - None of the nodes of the sublattice induced by vmax has a value k* which is larger than v_{max}.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_{max}
    - Also remember that the relaxation of **h** explores all the lattice's levels

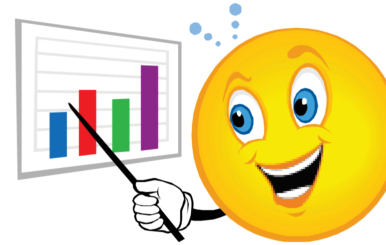
Theorem 2 *Given a node v_{max}, the sublattice it induces $L(v_{max})$, and an integer α, the following hold:*

$$\mathrm{cumKA}(v_{max}|\alpha) \leq \mathrm{cumKA}(v|\alpha),\ v \in L(v_{max})$$
$$\mathrm{cumSLD}(v_{max}|\alpha) \leq \mathrm{cumSLD}(v|\alpha),\ v \in L(v_{max})$$

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

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

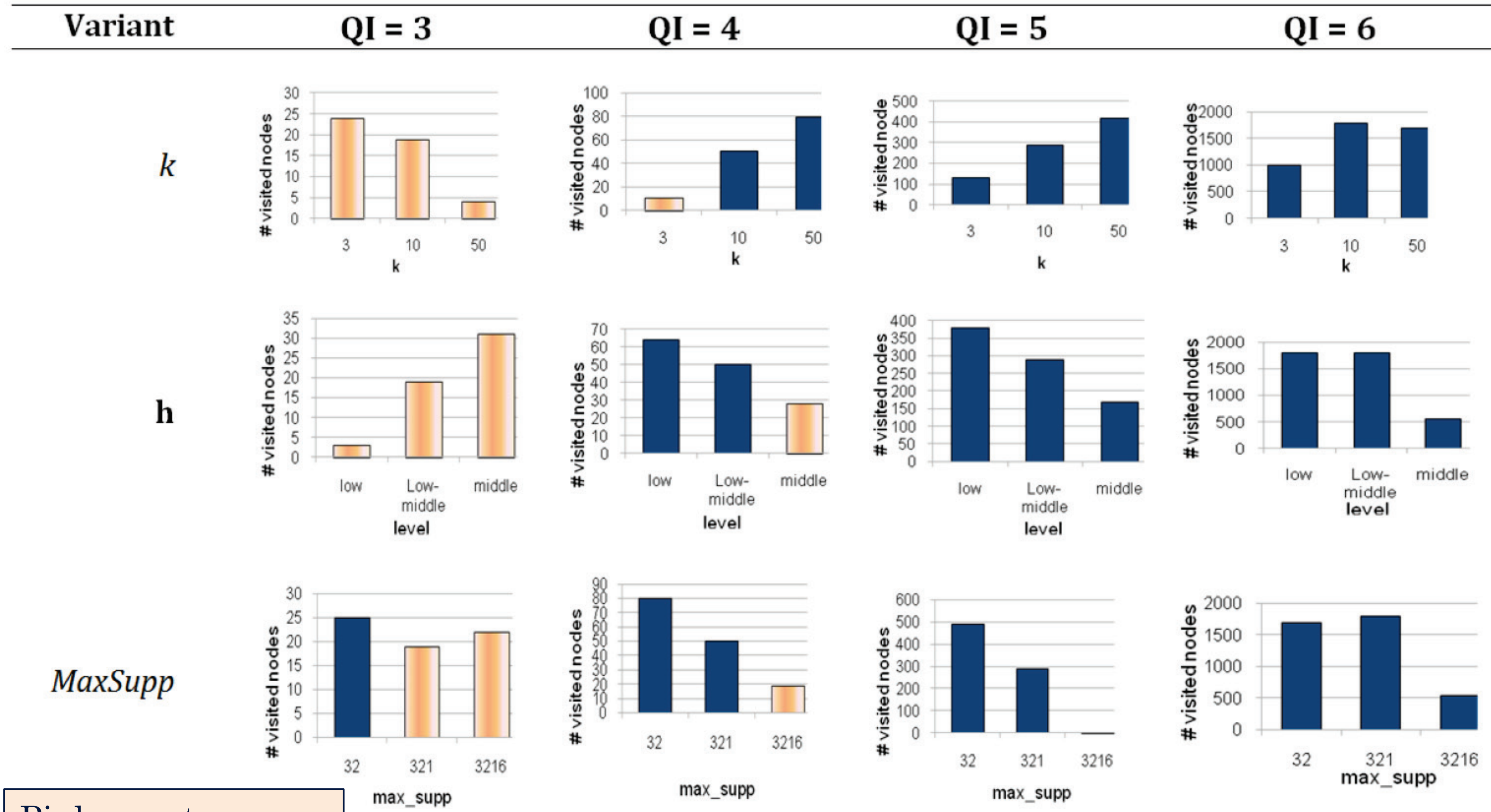
Adult data set Experiments



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

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



Pink: exact answer
Blue: approximations

Results in a nutshell (1)

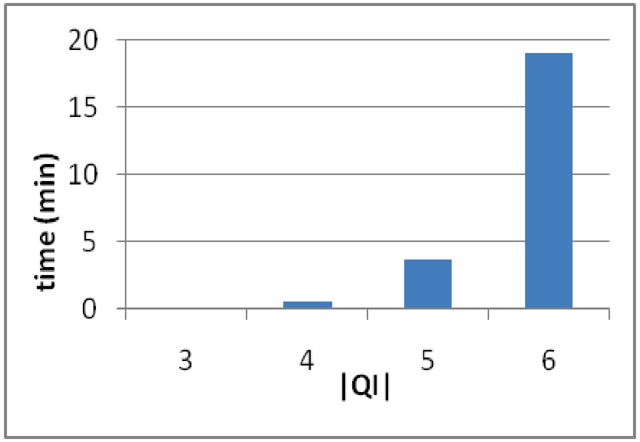
- When increasing the privacy criterion
  - When QI is small: exact answer + search is directed towards lower heights. Consequently, as k increases the solution is found earlier.
  - For larger QIs: need to resort to relaxations + the increase of k sublinearly increases the search space.
- When increasing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

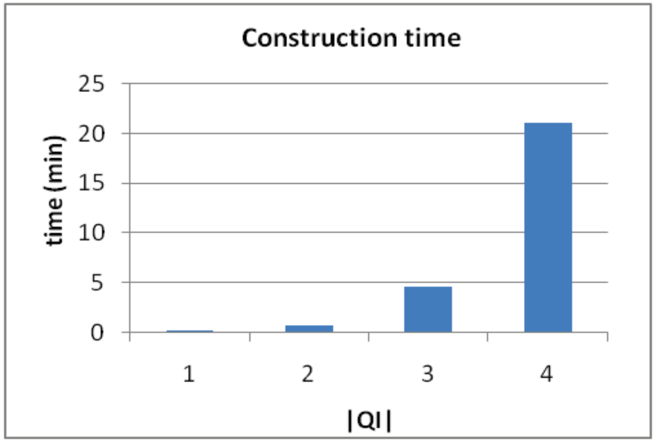
- In all cases, the time needed to detect the solutions ranges in 1-8 milliseconds!
- In all experiments, it is clear that the costs are dominated by the QI size.
- Results present the same behavior
  - for l-diversity and k-anonymity
  - For the Adult and the IPUMS data set

Lattice (in fact: Histogram) Construction time

- Adult data set (30162 tuples)



k-anonymity



l-diversity

- IPUMS data set (600000 tuples)

	k-anonymity	l-diversity
Average time (minutes)	10,5	38,087
Histo size (Kbytes)	530,688	58,424

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 Λ(v)= $\Sigma(\log(\gamma_i)*\mu_i)$ 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_{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_{max} histogram at runtime:
  - Small improvement for exact answer; identical behavior to full lattice for approximations local to v_{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 $A_{1@0}$, …, $A_{k@h}$, …, $A_{n@0}$ 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 (|levels| << |nodes|)

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

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

Level	relImp()
age1	1.70
age4	1.56
occupation1	1.42
occupation2	1.38
age3	1.30
education3	1.30
age2	1.30
education4	1.26
work_class2	1.24
marital_status3	1.16
marital_status2	1.14
race2	1.13
education2	1.09
work_class1	1.06
education1	1.05
marital_status1	1.05
race1	1.02
work_class3	1.00
race0	0.98
marital_status0	0.96
education0	0.95
work_class0	0.94
occupation0	0.71
age0	0.59

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 (γ)
  - Relative importance of a level (μ)

$$relImp\,(A^{h}) = \begin{cases} \dfrac{avgGroupSize(A^{h})}{avgGroupSize(A^{h-1})}, & \text{for all heights } h \text{ in } A^{H}, \ldots, 1, \text{ or} \\ \dfrac{1}{power\,(A^{1})}, & \text{for } h=0 \end{cases}$$

  - The product γ*μ
  - The product $\log_2(\gamma)*\mu$

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 $\log_2(\gamma)*\mu$

k=3	Till 30 tuples				Overall			
	#dev(Γ)	Err(Γ)	#dev(Λ)	Err(Λ)	#dev(Γ)	Err(Γ)	#dev(Λ)	Err(Λ)
QI=3	0	0,00%	0	0,00%	3	28,21%	4	26,67%
QI=4	2	21,96%	1	3,41%	5	29,64%	5	11,85%
QI=5	2	2,86%	2	2,35%	4	1,43%	6	7,42%
QI=6	1	0,60%	1	1,31%	4	8,21%	6	8,59%

k=10	Till 30 tuples				Overall			
	#dev(Γ)	Err(Γ)	#dev(Λ)	Err(Λ)	#dev(Γ)	Err(Γ)	#dev(Λ)	Err(Λ)
QI=3	2	52,99%	0	0%	3	22,27%	2	0,36%
QI=4	3	57,87%	1	8,53%	4	29,83%	5	54,56%
QI=5	3	6,84%	2	4,51%	5	3,85%	6	2,54%
QI=6	0	0,00%	2	1,85%	2	5,27%	7	6,34%

k=50	Till 30 tuples				Overall			
	#dev(Γ)	Err(Γ)	#dev(Λ)	Err(Λ)	#dev(Γ)	Err(Γ)	#dev(Λ)	Err(Λ)
QI=3	2	33,20%	2	12,30%	2	16,60%	4	44,01%
QI=4	4	38,38%	1	0,61%	5	22,39%	3	0,36%
QI=5	1	0,05%	1	1,30%	2	0,04%	3	0,90%
QI=6	0	0,00%	2	1,00%	3	24,44%	7	25,08%

Eventually …

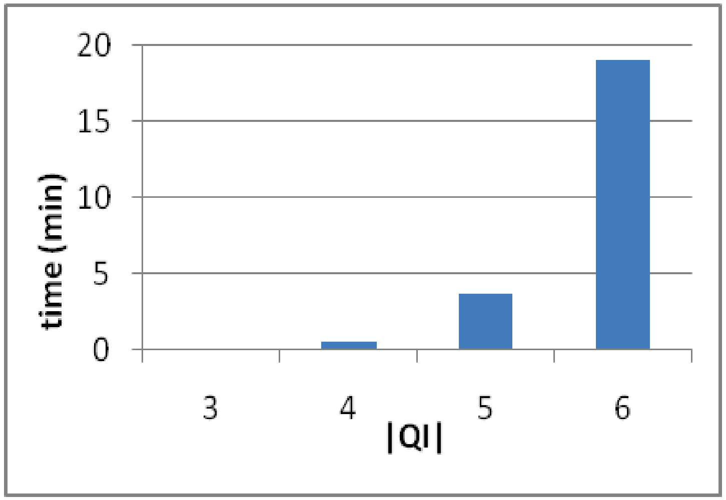
- Given $\Gamma(v) = \Sigma(\gamma_i)$, and, $\Lambda(v) = \Sigma(\log(\gamma_i) * \mu_i)$
- In the "full" case, it appears that
  - wins are ~ equally split
  - when Γ loses, it deviates a lot;
  - when Λ loses, it stays quite close to the winner value of Γ
- When we ignore small suppressions, Λ wins 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

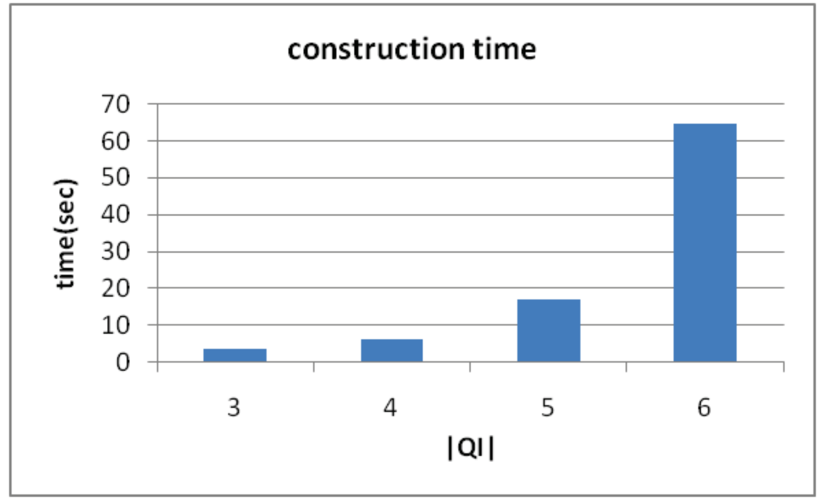
Research challenges & solutions for the case of the partial lattice

- ✓ Which nodes will be selected and how?
  - ✓ Ultimately, it appears that $\Lambda(v) = \Sigma(\log(\gamma_i) * \mu_i)$ 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

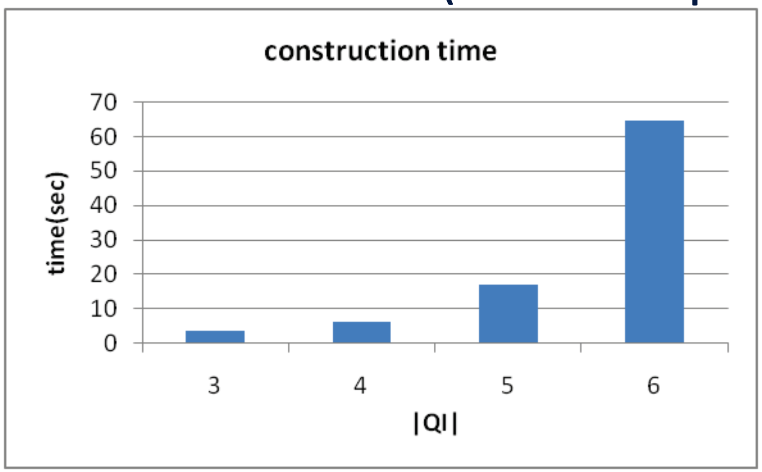


Partial lattice

Time to materialize the partial lattice with its histograms

All numbers are secs

- Adult data set (30162 tuples)



	QI=3	QI=4	QI=5	QI=6
Level importance comp.	1.1522	1.8028	3.27	5.2404
Node importance comp.	0.0522	0.1844	0.4092	0.995
Histogram computation	2.5162	4.2512	13.3228	58.469
Overall	3.7206	6.238	17.002	64.704

- IPUMS data set (600000 tuples)

	QI=4
Level importance comp.	31,6438
Node importance comp.	0,183
Histogram computation	77,6478
Overall	109,474

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 $\Lambda(v) = \Sigma(\log(\gamma_i)*\mu_i)$ 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)
- <u>What is the effect to the quality of the solution?</u>

Algorithm for partial lattice: Exact answer

1. Search for v_{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 height(v_{max}) for the best possible answer
- Relaxation of h: same as exact lattice
  - Try the upper part of lattice; if a node answers (k, 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_{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, maxSupp=3126).
- Height approximation (respect k & max supp, relax h):
  - Very good results compared to the full lattice, with
    - small deviations for 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_{max} might not be part of the lattice

Heuristic extension: v_{max} histogram at runtime

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

Heuristic extension: v_{max} histogram at runtime

	Exact		Relax M		Relax H		Relax k	
	P	PR	P	PR	P	PR	P	PR
Same Sol.	6	8	1	18	9	9	0	6
Other Sol.	1+3	2+0	17	0	9	9	1	0
Failed	-	-	-	-	-	-	5	0
No Sol both	-	-	-	-	-	-	12	12

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"

- We pay a time penalty of 0.1 – 0.3 sec per query for the (one) extra histogram of v_{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_{max}

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
- <u>Conclusions</u>

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_{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_{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?

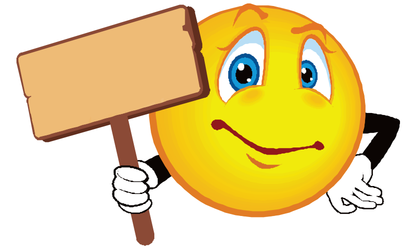
References (1)

- **[Sama01]** P. Samarati. Protecting respondents' identities in microdata release. IEEE Trans. Knowl. Data Eng. (TKDE), 13(6):1010–1027, 2001.
- **[Swee02a]** Latanya Sweeney. k-Anonymity: A Model for Protecting Privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5): 557-570 (2002)
- **[Swee02b]** Latanya Sweeney. Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5): 571-588 (2002)
- **[LeDR05]** K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In SIGMOD, pages 49–60, 2005.
- **[LeDR06]** Kristen LeFevre, David J. DeWitt, Raghu Ramakrishnan. Mondrian Multidimensional K-Anonymity. ICDE 2006: 25
- **[MaGK06]** A. Machanavajjhala, J. Gehrke, and D. Kifer. l-diversity: Privacy beyond k-anonymity. ICDE, 2006.

References (2)

- **[Xu+06]** Jian Xu, Wei Wang, Jian Pei, Xiaoyuan Wang, Baile Shi, Ada Wai-Chee Fu. Utility-based anonymization using local recoding. KDD 2006: 785-790
- **[Agg05]** Charu C.Aggarwal. On k-anonymity an the curse of Dimensionality. VLDB 2005.
- **[PaSh07]** Hyoungmin Park, Kyuseok Shim. Approximate Algorithms for k-anonymity. SIGMOD 2007.
- **[UCI]** U.C. Irvine Repository of Machine Learning Databases. 1998. http://www.ics.uci.edu/~mlearn
- **[IPUMS]** Data set obtained from the web site of Y. Tao for the [XiTa07] paper http://www.cse.cuhk.edu.hk/~taoyf/paper/sigmod07.html

Auxiliary Slides



From Li et al., TKDE, VOL. 20, NO. 9, SEPT. 2008

Two dimention illustration

<table>
<tr><th></th><th>α</th><th>β</th><th>γ</th></tr>
<tr><td>a</td><td>2</td><td>10</td><td>0</td></tr>
<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>
</table>

<table>
<tr><th></th><th>(α, β)</th><th>γ</th></tr>
<tr><td>a</td><td>12</td><td>0</td></tr>
<tr><td>b</td><td>11</td><td>5</td></tr>
<tr><td>c</td><td>55</td><td>0</td></tr>
<tr><td>d</td><td>20</td><td>0</td></tr>
</table>

<table>
<tr><th></th><th>α</th><th>β</th><th>γ</th></tr>
<tr><td>a</td><td colspan="2">12</td><td>0</td></tr>
<tr><td>b</td><td colspan="2">11</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>
</table>

<table>
<tr><th></th><th>α</th><th>β</th><th>γ</th></tr>
<tr><td>a</td><td>5</td><td>7</td><td>0</td></tr>
<tr><td>b</td><td>6</td><td>5</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>
</table>

Table Summary

Att1	Att2	Freq
a	α	2
a	β	10
b	α	8
b	β	3
b	γ	5
c	α	30
c	β	25
d	β	20

(a)

original

Att1	Att2	Freq
a	(α, β)	12
b	(α, β)	11
b	γ	5
c	(α, β)	55
d	(α, β)	20

(b)

global

Att1	Att2	Freq
a	(α, β)	12
b	(α, β)	11
b	γ	5
c	α	30
c	β	25
d	β	20

(c)

m/d

Att1	Att2	Freq
a	(α, β)	5
a	β	7
b	α	6
b	(α, β)	5
b	γ	5
c	α	30
c	β	25
d	β	20

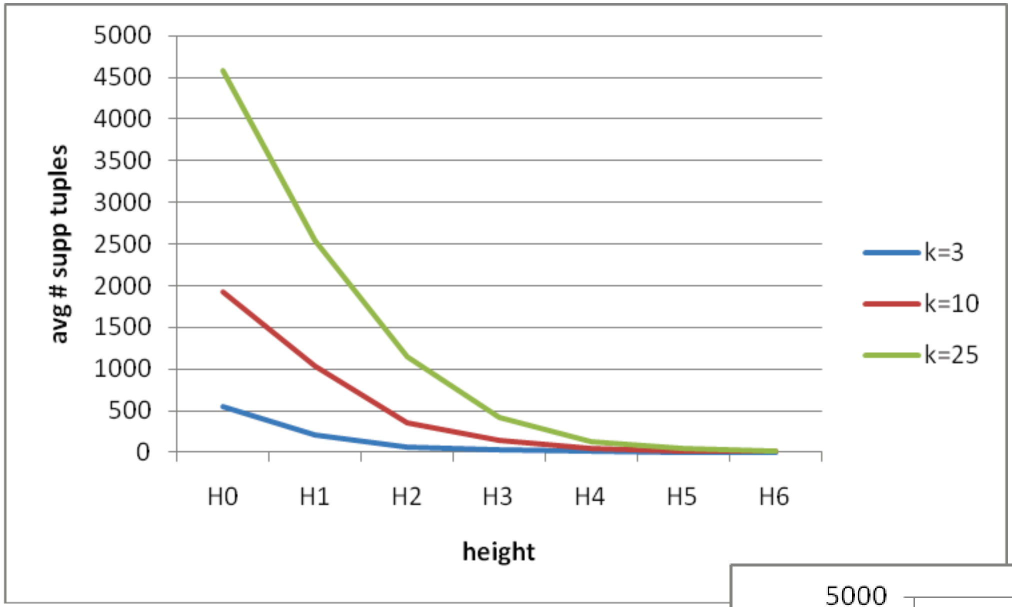
(d)

local

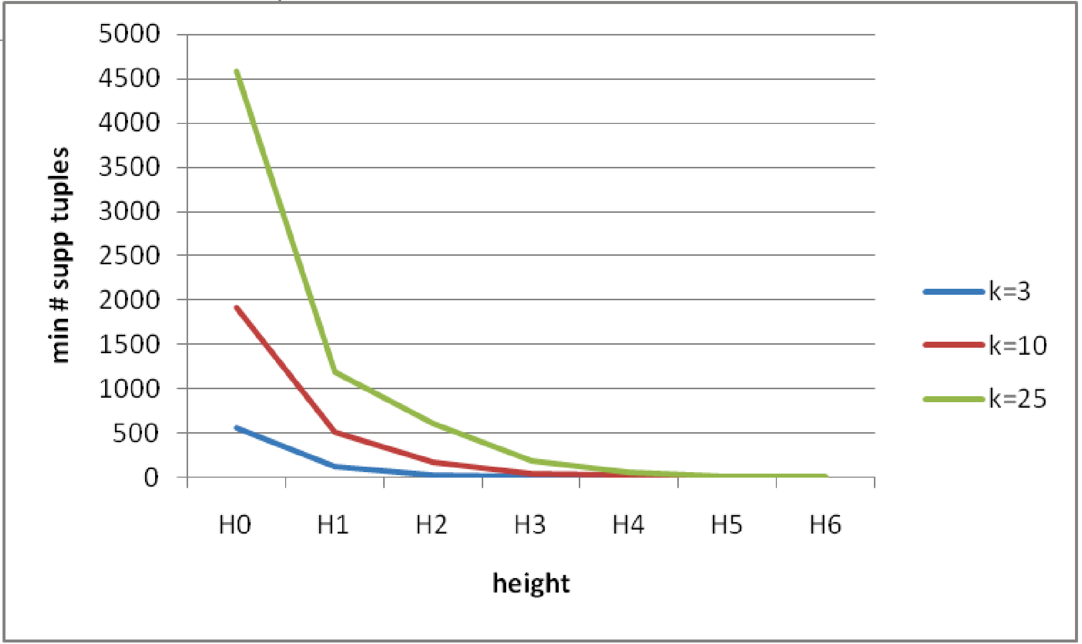
Auxiliary slides

IS THE PROBLEM VALID?

Effect of k and height to suppression (1)



Avg #suppressed tuples, QI = 3

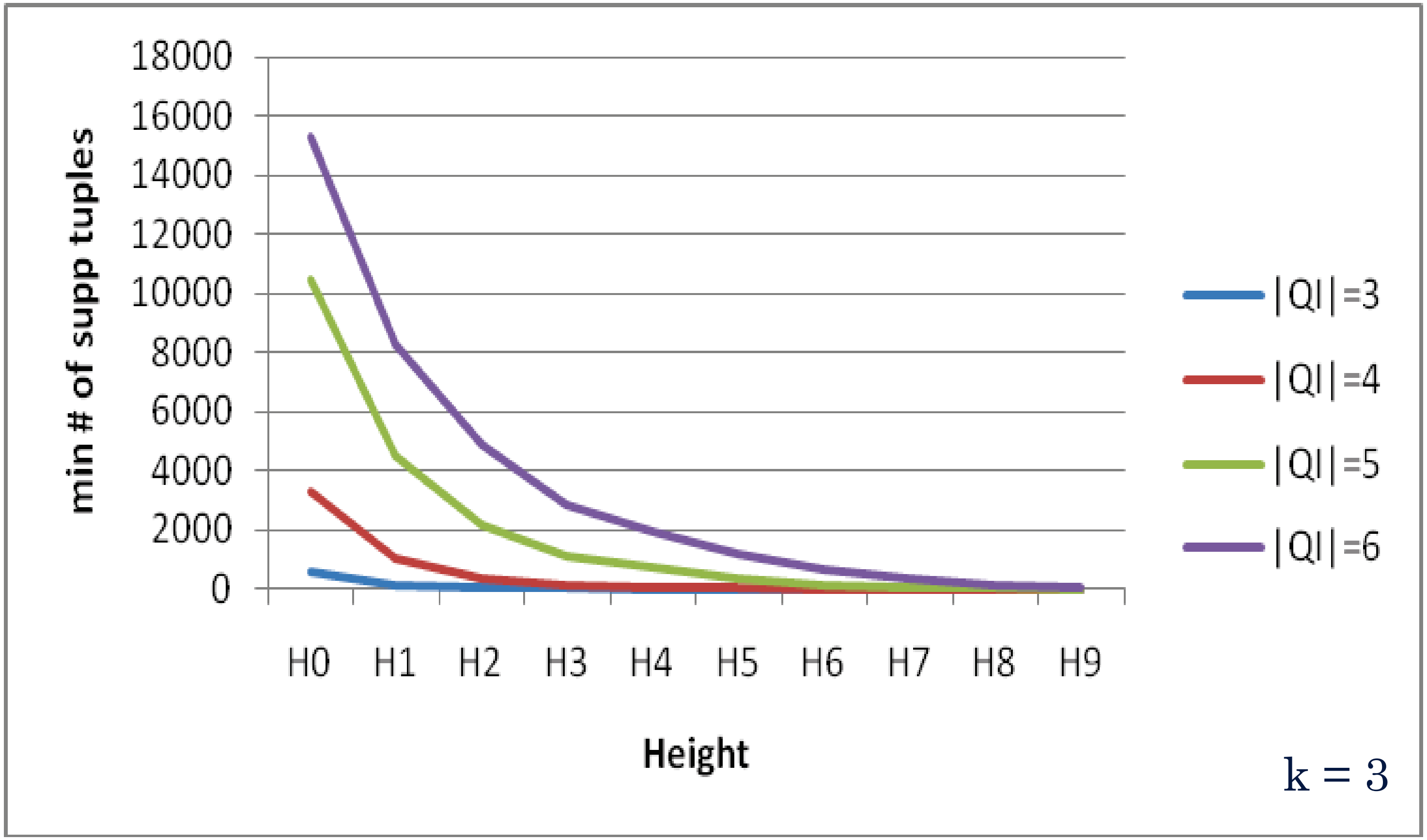


Min #suppressed tuples QI = 3

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)



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 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*.

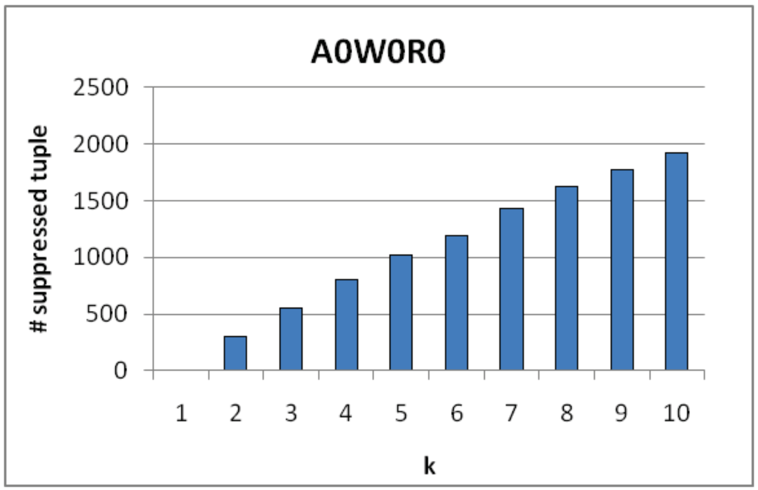
The study for the IPUMS data set

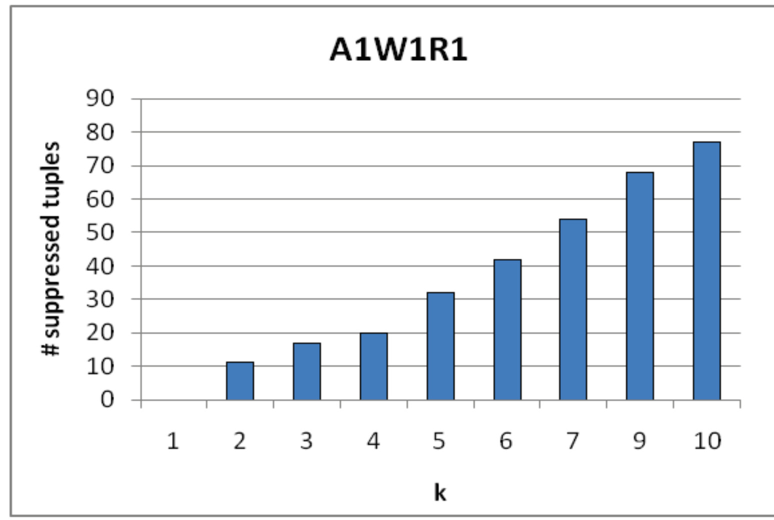
- k-anonymity
  – Similarly to the Adult data set , the amount of suppressed tuples drops rapidly as we increase the height.
  – The behavior of the average with respect to the min suppression is quite different with respect to Adult data set. Min suppression drops much faster with respect to the average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

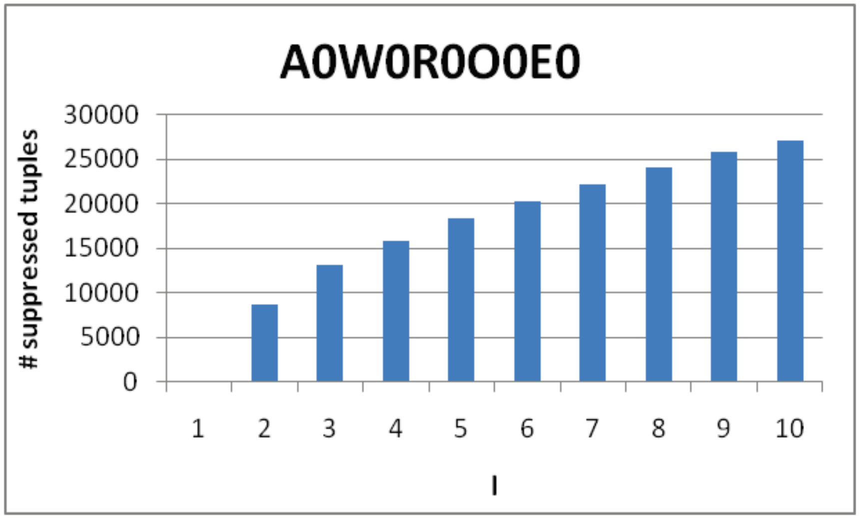
- Is the amount of suppressed tuples significant?
  - In low heights and/or large QI's suppression takes large values => the problem is important
- What is the relationship between suppression, generalization and privacy?
  - The larger the height, the lower the suppression – minimum suppression is characterized 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

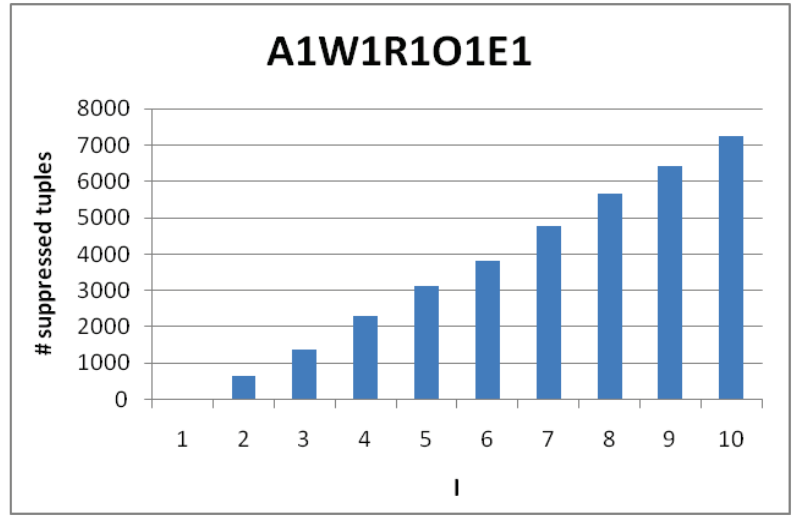
Cumulative histogram για QI-3 (k-anonymity



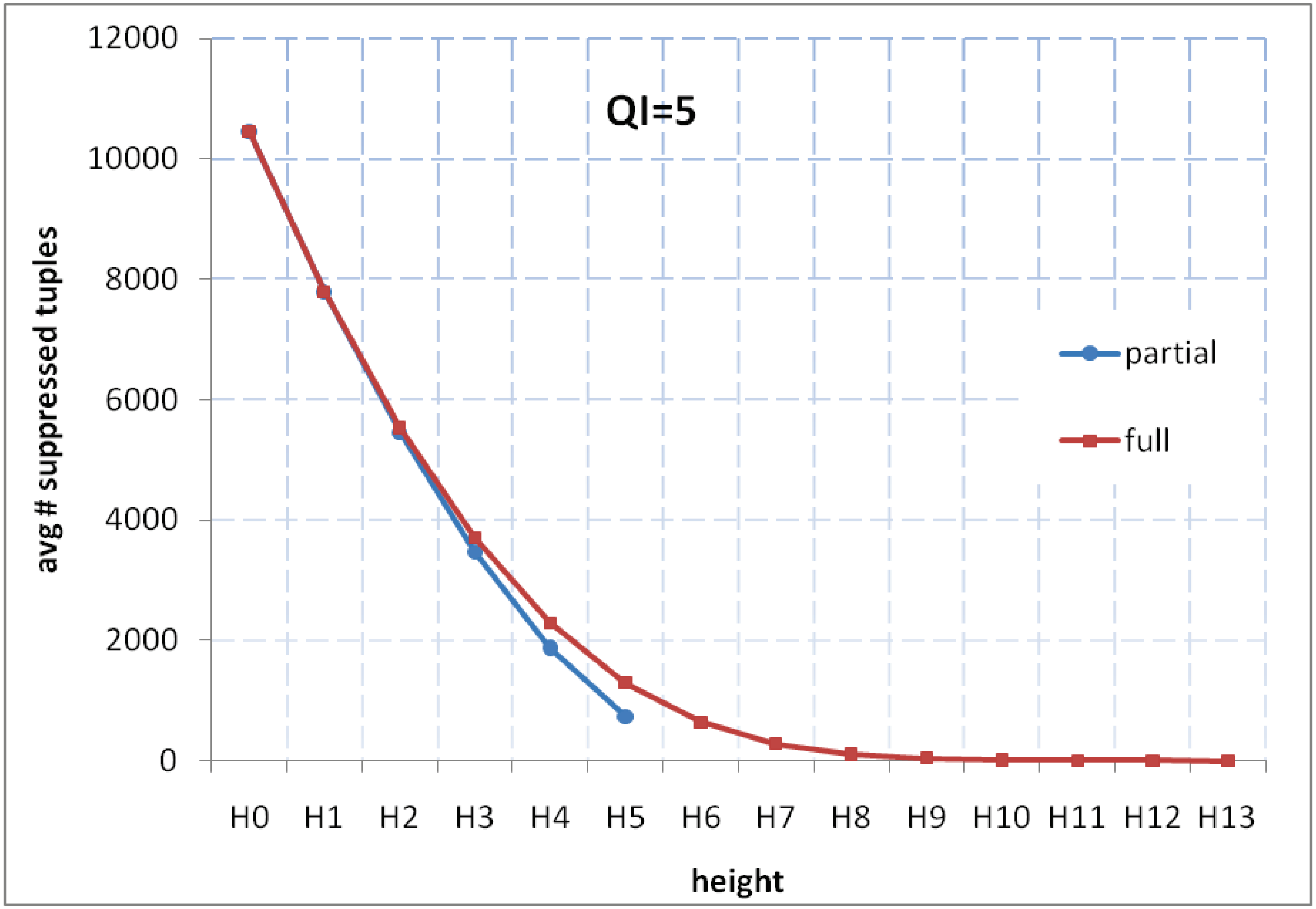


Cumulative histogram for QI-3 (*l*-diversity)





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



Min -Avg -Max # suppressed tuples for QI=3 & QI=5

	|QI| = 3					|QI| = 5				
	Min	Avg	Max	Avg % over full	% over previous	Min	Avg	Max	Avg % over full	% over previous
H0	554	554	554	1,83	-	10458	10458	10458	34,67	-
H1	125	207	295	0,69	62,33	4514	7795	9879	25,84	34,15
H2	28	56	69	0,19	72,92	2169	5459	8913	18,10	42,80
H3	12	24	54	0,08	57,52	1619	3472	7398	11,51	57,22
H4	4	8	15	0,03	64,79	1051	1881	3353	6,24	84,53
H5	1	4	7	0,01	52,66	773	733	733	2,43	156,67
H6	0	2	4	0,01	58,50	-	-	-	-	-

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)	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	1,57	1,18
H1	4,18	2,27	3,14	2,02	2,42	1,51	1,91	1,34
H2	6,07	3,59	3,97	2,28	2,84	1,73	2,17	1,47
H3	4,25	3,82	4,75	2,27	2,98	1,77	2,49	1,64
H4	7	2	6,07	3,59	3,8	1,88	2,61	1,68
H5	2	7	4,25	3,12	4,22	2,06	3,03	1,71

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 # suppressed tuples for various QI sizes

	Min # of suppressed tuples			
	|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 (QI=3)

|QI|=3 (lattice up to height H6)									
	k=3			k=10			k=25		
	min	**avg**	**max**	**min**	**avg**	**max**	**min**	**avg**	**max**
H0	554	554	554	1921	1921	1921	4578	4578	4578
H1	125	209	295	522	1030	1357	1184	2546	3573
H2	28	57	69	170	352	508	610	1153	1926
H3	12	24	54	51	148	484	195	419	1236
H4	4	8	15	28	45	94	56	127	222
H5	1	4	7	2	19	37	14	48	105
H6	0	2	4	0	9	23	14	21	40

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_{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 fror 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).

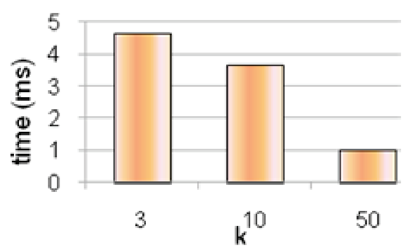
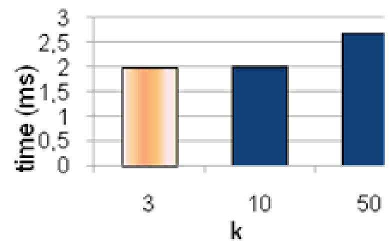
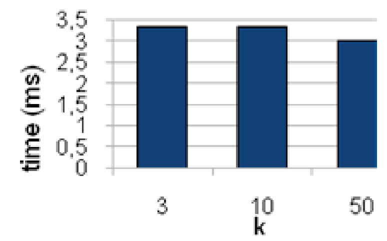
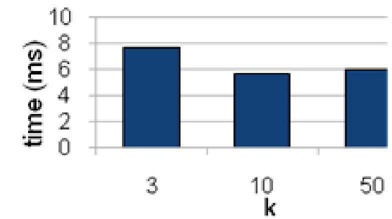
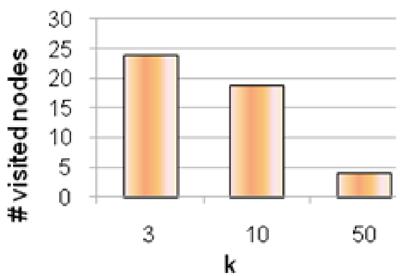
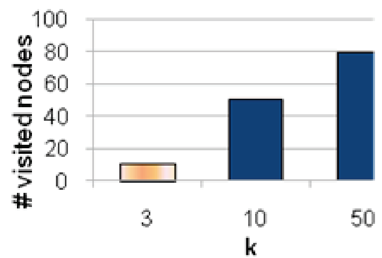
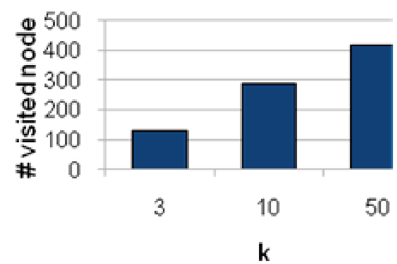
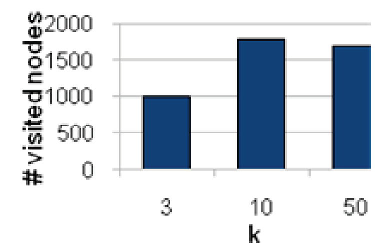
Experiments with the IPUMS data set

- Parameters

k	3,30,50,100,150
l	3,6,10
Topmost node	low (1010), middle-low (2110), middle (2220)
MaxSupp	600, 6000, 60000

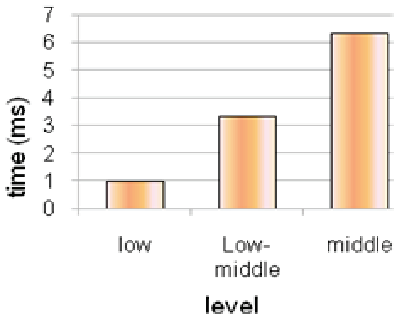
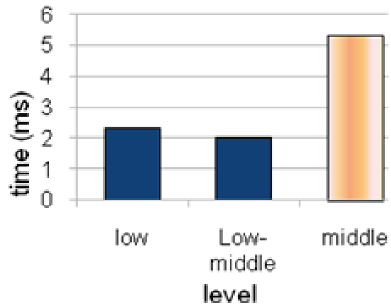
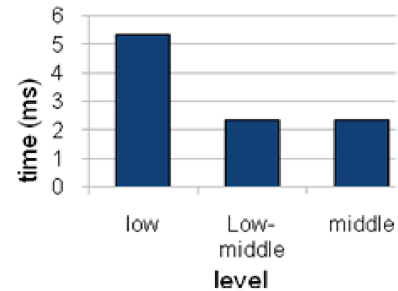
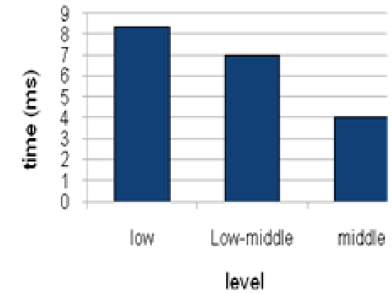
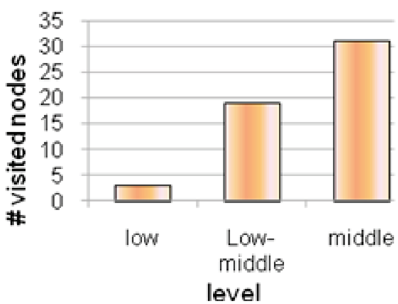
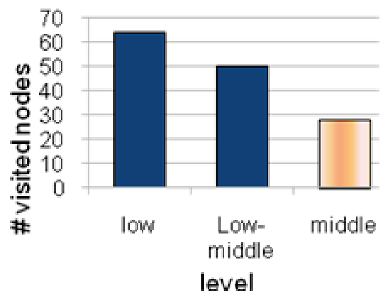
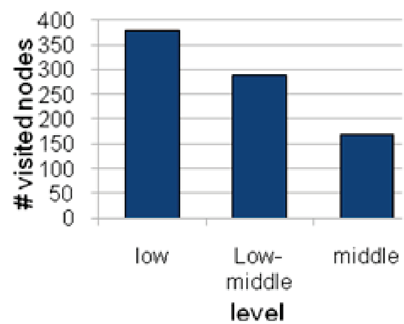
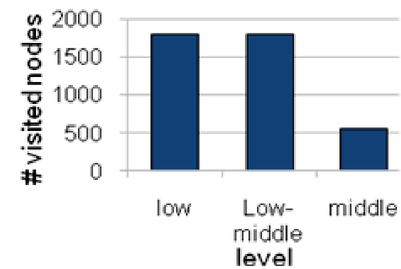
- Results are similar to the ones of the Adult data set (attn: here, we have small QI size, QI =4)
  - When is k or l are small, then we can have exact answers for both k-anonymity and l-diversity (again: the small QI is important here and allows many exact answers) .
  - As the height of v_{max} increases, 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

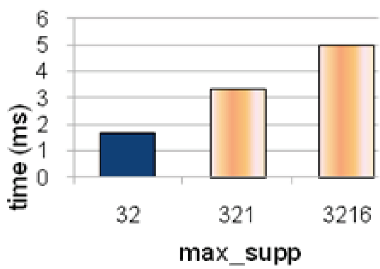
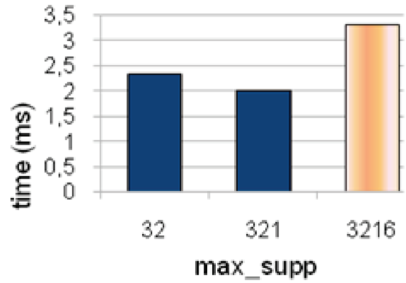
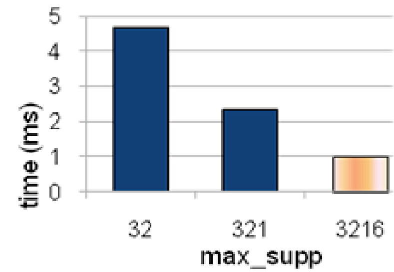
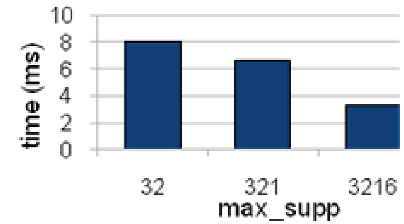
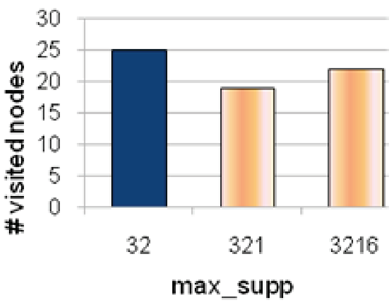
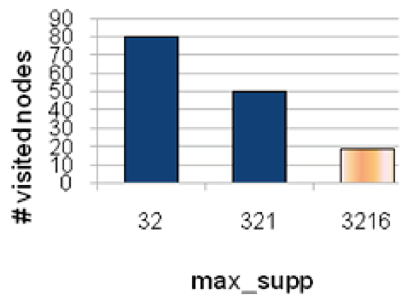
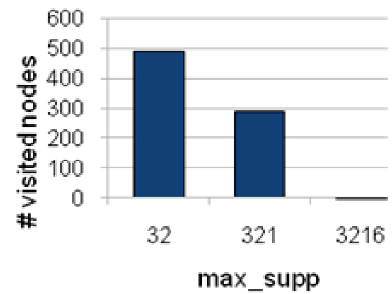
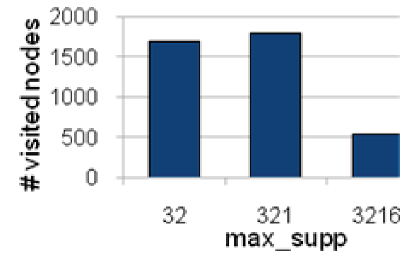
|QI|=3	|QI|=4	|QI|=5	|QI|=6
			
			
Parameters: 3, 321, 211 ***Solution:*** Move down find, id: 4 supp tuples: 125 level:100 **Parameters:** 10,321,211 ***Solution:*** Move down find, id: 5 supp tuples:170 level:110 **Parameters:** 50, 321, 211 ***Solution:*** Move down find, id: 23 supp tuples:251 level:211 **Parameters:** 50, 321, 211 ***Solution:*** Move down find, id: 23 supp tuples:251 level:211	**Parameters:** 3, 321, 2011 ***Solution:*** Move down find, id: 56 supp tuples:283 level:1011 **Parameters:** 10, 321, 2011 ***Solution:*** Rlx1: id:65 supp tuples:655 level:2011 Rlx2 id:17 supp tuples:170 level:1210 Rlx3 id:65 k:5 Supp_tupp:301 **Parameters:** 50, 321, 2011 Rlx2 id:17 supp tuples:170 level:1210 Rlx3 id:65 k:5 Supp_tupp:301 **Parameters:** 50, 321, 2011 ***Solution:*** Rlx1: id:65 supp tuples:4077 level:2011 Rlx2 id:107 supp tuples:137 level:1212 Rlx3 id:65 k:5 Supp tpls:301	**Parameters:** 3, 321, 21012 ***Solution:*** Rlx1: id:551 supp tuples:656 lv:21012 Rlx2 id:503 supp tuples:108 lv:10212 Rlx3:No solution. **Parameters:**: 10, 321, 21012 ***Solution:*** Rlx1: id:551 supp tuples:2533 lvl:21012 Rlx2: id:504 supp tuples:230 lvl:10222 ***Solution:*** Rlx1: id:551 supp tuples:2533 lvl:21012 Rlx2: id:504 supp tuples:230 lvl:10222 Rlx3: No solution. **Parameters:** 50, 321, 21012 ***Solution:*** Rlx1: id:551 supp tuples:9214 lvl:21012 Rlx2 id:636 supp tuples:169 lvl:40122 Relaxation3 : No solution.	**Parameters:** 3, 321, 211012 ***Solution:*** Rlx1: id:2591 supp tpls:1611 lvl:211012 Rlx2 id:763 supp tpls:155 lvl:400202 Rlx3: No solution. **Parameters:** 10, 321, 211012 ***Solution:*** Rlx1: id:2591 supp tpls:5362 lvl:211012 Rlx2 id:2171 supp tpls:285 lvl:401211 ***Solution:*** Rlx1: id:2591 supp tpls:5362 lvl:211012 Rlx2 id:2171 supp tpls:285 lvl:401211 Rlx3: No solution. **Parameters:** 50, 321, 2 1 1 0 1 2 ***Solution:*** Rlx1 id:2591 supp tpls:15106 lv:211012 Rlx2 id:2812 supp tpls:137 lvl:401222 Rlx3: No solution.

Pink: exact answer
Blue: approximations

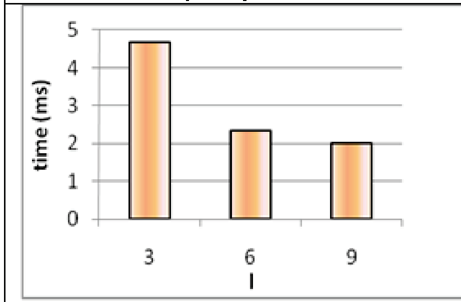
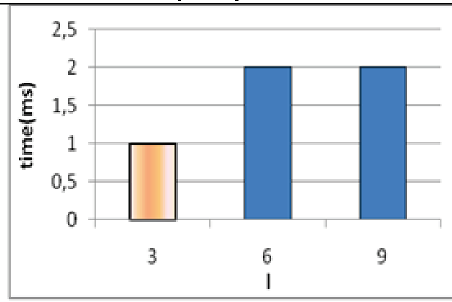
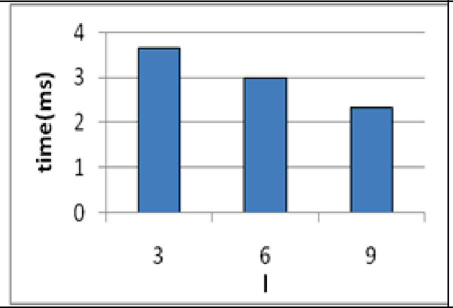
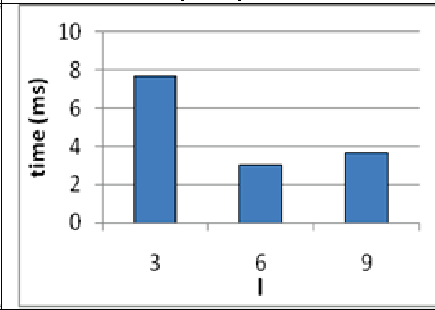
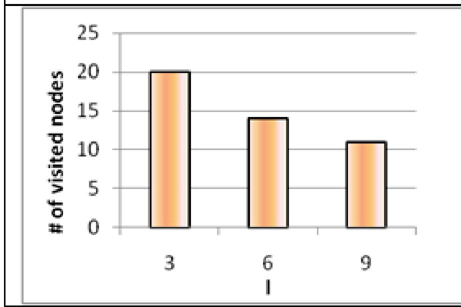
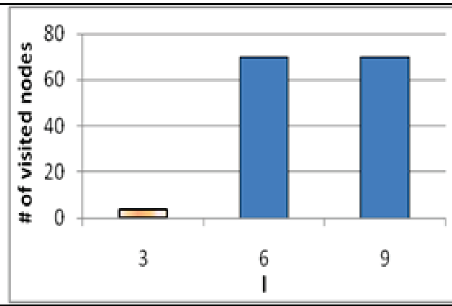
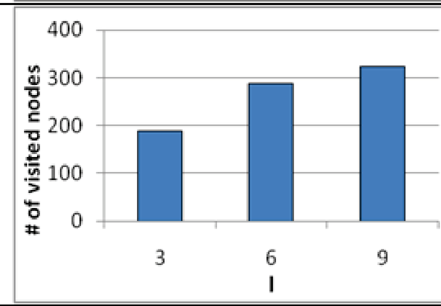
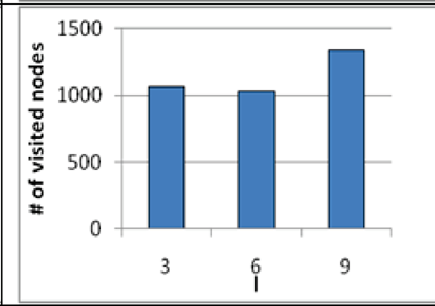
Variant constraint on upper levels

<table>
<tr><th>|QI|=3</th><th>|QI|=4</th><th>|QI|=5</th><th>|QI|=6</th></tr>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>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</td>
<td>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, 2011 Solution:
Rlx1: id:65 supp tuples:655 lvl:2011
Rlx2 id:17 supp tuples:170 lvl:1210
Rlx3 id:65 k:5 Supp_tupp:301
Parameters: 10, 321, 2112 Solution:
Move down find, id: 59 supp tuples:285 level:1111</td>
<td>Parameters: 10, 321, 11001, Solution
Rlx1: id:280 supp tpls: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 tpls:369 lvl:22112
Rlx2 id:635 supp tpls:60 lvl:40112
Rlx3 id:563 k:9 Supp tpls:315</td>
<td>Parameters: 10, 321, 111001 Solution:
Rlx1: id:336 supp tpls:12823 lvl:111001
Rlx2 id:2171 supp tpls:285 lvl:401211
Rlx3: No solution.
Parameters: 10, 321, 211012 Solution:
Rlx1: id:2591 supp tpls:5362 lvl:211012
Rlx2 id:2171 supp tpls:285 lvl:401211
Rlx3: No solution.
Parameters: 10,321, 222112 Solution:
Rlx1: id:2623 supp tpls:712 lvl:222112
Rlx2 id:2811 supp tpls:54 lvl:401212
Rlx3 id:2623 k:5 Supp tpls:298</td>
</tr>
</table>

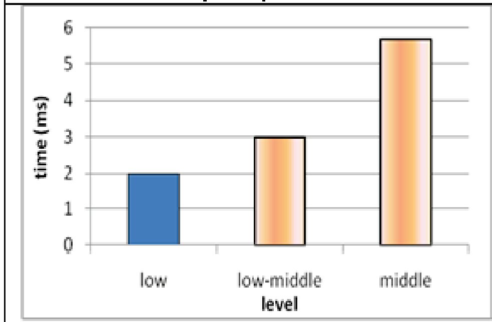
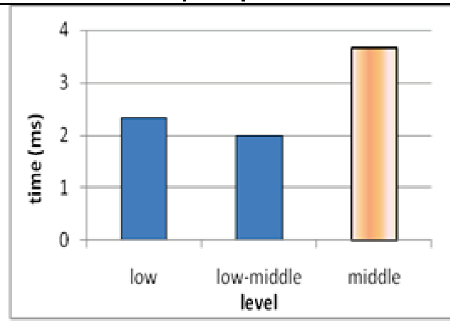
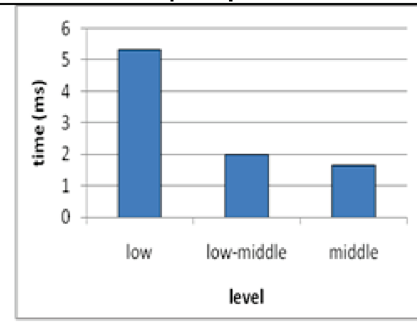
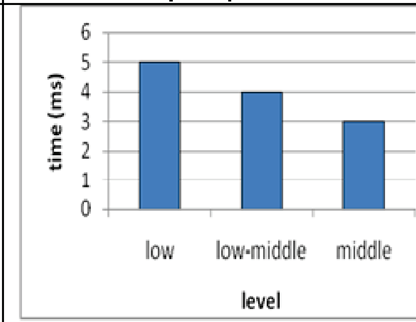
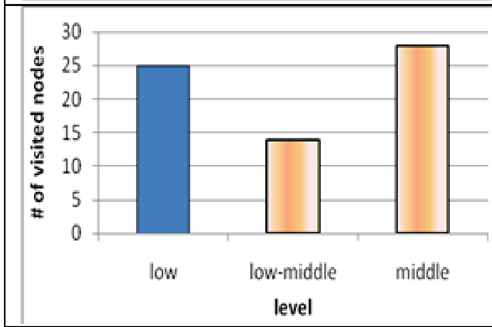
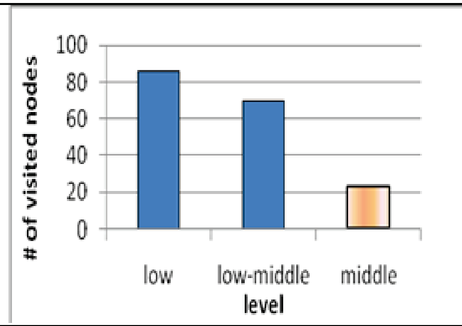
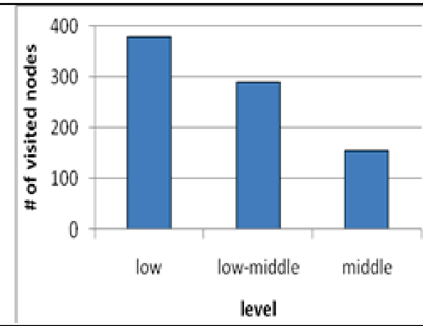
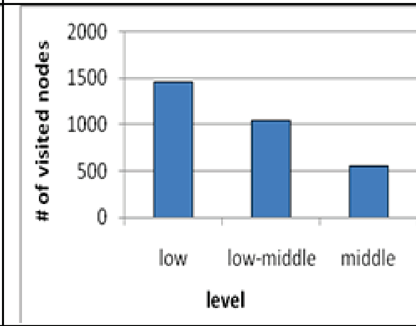
Variant max_supp

|QI|=3	|QI|=4	|QI|=5	|QI|=6
			
			
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_tupp:31 **Parameters:** 10, 321, 211**Solution:** Move down find, id: 5 supp tuples:170 level:1 1 0 **Parameters:**: 10, 3216, 211**Solution:** Move down find, id: 1 supp tuples:1921 level:000	**Parameters:** 10, 32, 2011 **Solution:** Rlx1: id:65 supp tuples:655 lvl:2011 Rlx2 id:35 supp tuples:28 lvl:3210 Rlx3 No solution **Parameters:** 10, 321, 2011 **Solution:** Rlx1: id:65 supp tuples:655 lvl:2011 Rlx2: id:17 supp tuples:170 lvl:1210 Rlx3: id:65 k:5 Supp_tupp:301 **Parameters:** 10, 3216, 2011 **Solution:** Move down find, id: 19 supp tuples:2110 level:2000	**Parameters:** 10, 32, 21012 **Solution:** Rlx1: id:551 supp tpls:2533 lvl:21012 Rlx2: id:638 supp tpls:14 lvl:40212 Rlx3: No solution **Parameters:** 10, 321, 21012 **Solution:** Rlx1: id:551 supp tpls:2533 lvl:21012 Rlx2: id:504 supp tpls:230 lvl:10222 Rlx3: No solution **Parameters:** 10, 3216, 21012 **Solution:** Move down find, id: 551 supp tuples:2533 level:21012	**Parameters:** 10, 32, 211012 **Solution:** Rlx1: id:2591 supp tpls:5362 lvl:211012 Rlx2: id:2812 supp tpls:21 lvl:401222 Rlx3: No solution **Parameters:** 10, 321, 211012 **Solution:** Rlx1: id:2591 supp tpls:5362 lvl:211012 Rlx2: id:2171 supp tpls:285 lvl:401211 Rlx3: No solution **Parameters:** 10, 3216, 211012 **Solution:** Rlx1: id:2591 supp tpls:5362 lvl:211012 Rlx2: id:1525 supp tpls:1222 lvl:400210 Rlx3: id:2591 k:5 Supp tpls:2915

l-diversity over Adult. Variant value for the privacy criterion, l

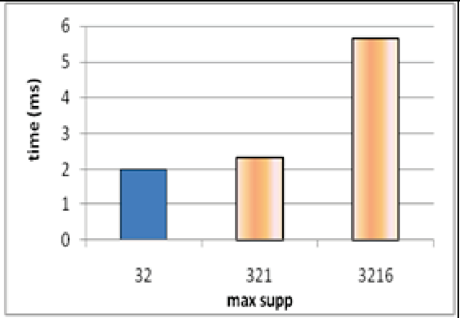
<table>
<tr><th>|QI|=3</th><th>|QI|=4</th><th>|QI|=5</th><th>|QI|=6</th></tr>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>Parameters:3, 321, 2 1 1
Solution: Move down find, id: 4 supp tpls:240 lvl:100
Parameters:6, 321, 2 1 1
Solution: Move down find, id: 20 supp tpls:70 lvl:111
Parameters:9, 321, 2 1 1
Solution: Move down find, id: 20 supp tpls:186 lvl:111</td>
<td>Parameters:3, 321, 2 0 1 1
Solution: Move down find, id: 65 supp tpls:319 lvl:2011
Parameters:6, 321, 2 0 1 1
Solution:
Rlx1: id:65 supp tpls:1013 lvl:2011
Rlx2 id:18 supp tpls:54 lvl:1220
Rlx3 id:65 l:3 Supp tpls:319
Parameters:9, 321, 2 0 1 1
Solution:
Rlx1: id:65 supp tpls:2005 lvl:2 0 1 1
Rlx2 id:18 supp tpls:104 lvl:1 2 2 0
Rlx3 id:65 l:3 Supp_tupp:319</td>
<td>Parameters:3, 321, 2 1 0 1 2
Rlx1: id:551 supp tpls:1139 lvl:2 1 0 1 2
Rlx2 id:503 supp tpls:244 lvl:1 0 2 1 2
Rlx3 No solution
Parameters:6, 321, 2 1 0 1 2
Solution:
Rlx1: id:551 supp tpls:3205 lvl:21012
Rlx2 id:504 supp tpls:302 lvl:10222
Rlx3 No solution
Parameters:9, 321, 2 1 0 1 2
Solution:
Rlx1: id:551 supp tpls:5418 lvl:21012
Rlx2 id:635 supp tpls:250 lvl:40112
Rlx: No solution</td>
<td>Parameters:3, 321, 2 1 1 0 1 2
Solution:
Rlx1: id:2591 sup tpls:2715 lvl:211012
Rlx2 id:2452 sup tpls:288 lvl:101222
Rlx3 No solution
Parameters:6, 321, 2 1 1 0 1 2
Solution:
Rlx1: id:2591 sup tpls:6602 lvl:211012
Rlx2 id:2811 sup tpls:90 lvl:401212
Rlx3 No solution
Parameters:9, 321, 2 1 1 0 1 2
Solution:
Rlx1: id:2591sup tpls:10139 lvl:21101 2
Rlx2 id:2811 supp tpls:206 lvl:401212
Rlx3 No solution</td>
</tr>
</table>

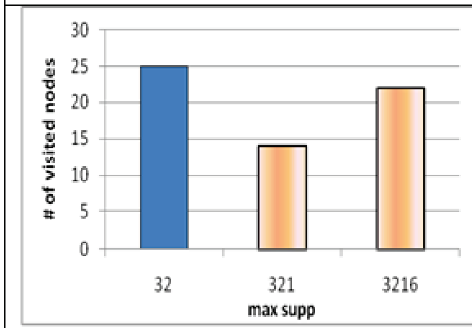
l-diversity- Adult Variant constraint on upper levels

|QI|=3	|QI|=4	|QI|=5	|QI|=6
			
			
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 **Solution:** Move down find, id: 20 supp tpls:70 lvl:111 **Parameters**:6, 321, 2 1 2 **Solution:** Move down find, id: 34 supp tpls:64 lvl:102	**Parameters**:6, 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 **Solution:** Rlx1: id:65 supp tpls:1013 lvl:2 0 1 1 Rlx2 id:18 supp tpls:54 lvl:1 2 2 0 Rlx3 id:65 l:3 Supp_tupp:319 **Parameters**:6, 321, 2 1 1 2 **Solution:** Move down find, id: 104 supp tpls:61 lvl:1112	**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 ***Solution:*** Rlx1: id:551 supp tpls:3205 lvl:21012 Rlx2 id:504 supp tpls:302 lvl:10222 Rlx3 No solution **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 1 0 0 1 **Solution:** Rlx1: id:336sup tpl:14076 vl:111001 Rlx2 id:2811 supp tpls:90 lvl:401212 Rlx3 No Solution **Parameters**:6, 321, 2 1 1 0 1 2 **Solution:** Rlx1: id:2591suptpl:6602 lvl:211012 Rlx2 id:2811 supp tpls:90 lvl:401212 Rlx3 No Solution **Parameters**:6, 321, 2 2 2 1 1 2 **Solution:** Rlx1: id:2623 sup tpl:857 lvl:222112 Rlx2 id:2811 supp tpl:90 lvl:401212 Rlx3 id:2623 l:3 Supp_tupp:253

l-diversity- Adult Variant max supp

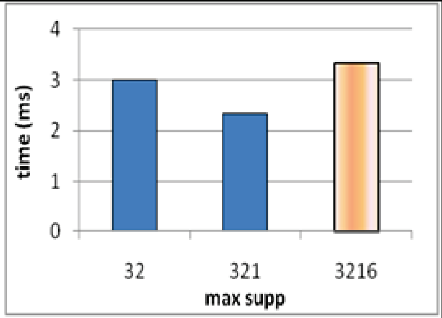
|QI|=3

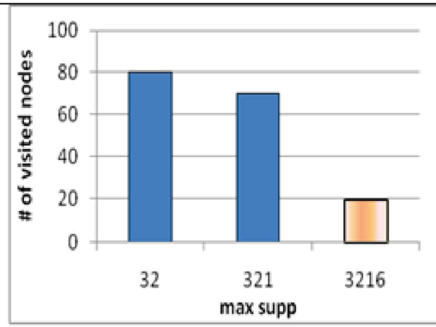




Parameters:6, 32, 2 1 1
Solution:
Rlx1:id:23 supp tpls:57 lvl:2 1 1
Rlx2 id:35 supp tpls:22 lvl:1 1 2
Rlx3 id:23 k:4 Supp_tupp:25
Parameters:6, 321, 2 1 1
Solution:
Move down find, id: 20 sup tpls:70 lvl:111
Parameters:6, 3216, 2 1 1
Solution:
Move down find, id: 1 sup tls:2476 lvl:000

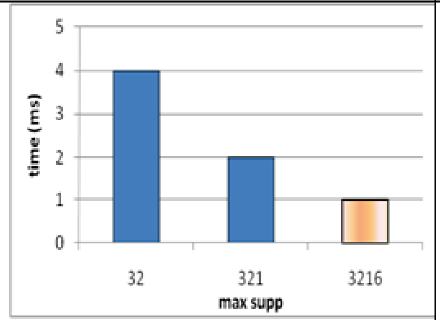
|QI|=4

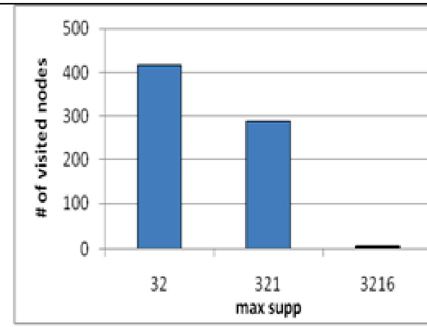




Parameters:6, 32, 2 0 1 1
Solution:
Rlx1: id:65 supp tpls:1013 lvl:2 0 1 1
Rlx2 id:41 supp tpls:12 lvl:4 1 1 0
Rlx3 No Solution
Parameters:6, 321, 2 0 1 1
Solution:
Rlx1: id:65 supp tpls:1013 lvl:2 0 1 1
Rlx2 id:18 supp tpls:54 lvl:1 2 2 0
Rlx3 id:65 l:3 Supp_tupp:319
Parameters:6, 3216, 2 0 1 1
Solution:
Move down find, id: 19 supp tpls:2826 lvl:2 0 0 0

|QI|=5

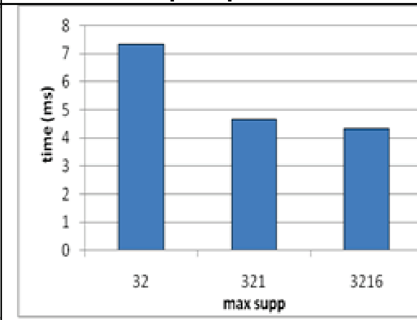


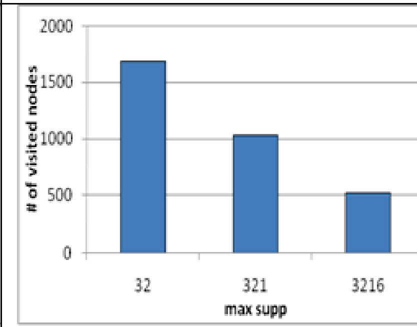


Parameters:6, 32, 2 1 0 1 2
Solution:
Rlx1: id:551 supp tpls:3205 lvl:21012
Rlx2 id:638 supp tpls:7 lvl:4 0 2 1 2
Rlx3 No Solution
Parameters:6, 321, 2 1 0 1 2
Solution:
Rlx1: id:551 supp tpls:3205 lvl:21012
Rlx2 id:504 supp tpls:302 lvl:10222
Rlx3 No Solution

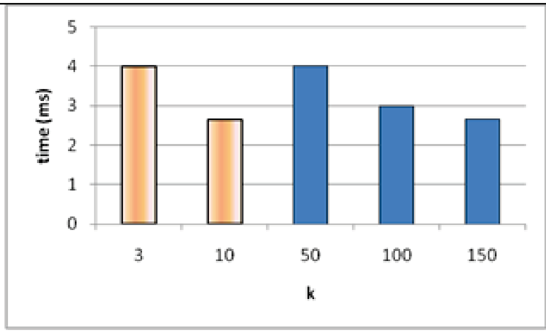
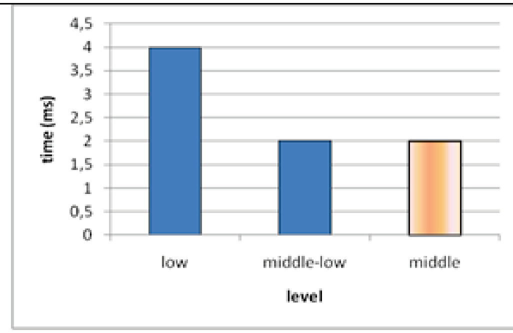
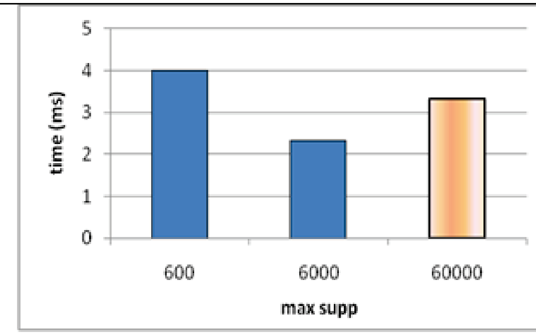
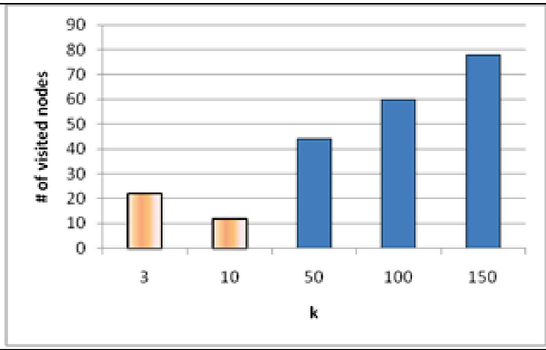
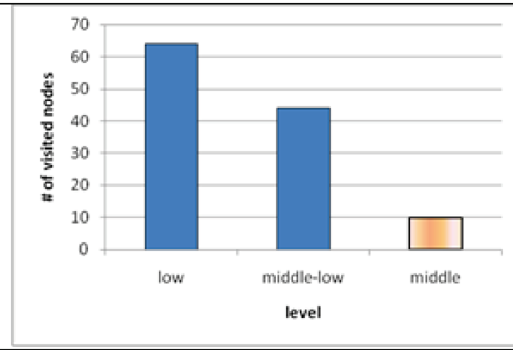
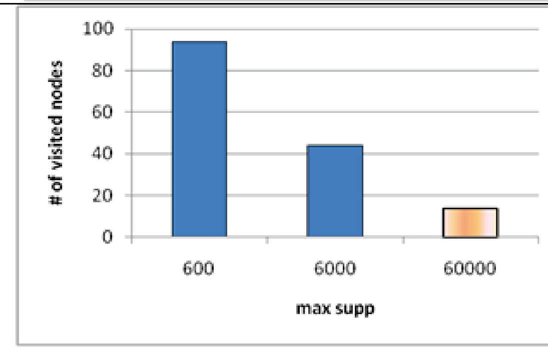
Parameters:6, 3216, 2 1 0 1 2
Solution:
Move down find, id: 551 supptpls:3205 lvl:21012

|QI|=6





Parameters:6, 32, 2 1 1 0 1 2
Solution:
Rlx1: id:2591 sup tpl:6602 lvl:211012
Rlx2 id:2823 supp tpls:7 lvl:403212
Rlx3 No Solution
Parameters:6, 321, 2 1 1 0 1 2
Solution:
Rlx1: id:2591 sup tpl:6602 lvl:211012
Rlx2 id:2811 supp tpls:90 lvl:401212
Rlx3 No Solution
Parameters:6, 3216, 2 1 1 0 1 2
Solution:
Rlx1: id:2591 sup tpl:6602 lvl:211012
Rlx2 id:503 sup tpl:1583 lvl:400201
Rlx3 id:2591 l:3 Supp_tupp:2715

K-anonymity IPUMS (QI=4)		
Variant k	Variant level	Variant max_supp
		
		
Parametres:3, 6000, 2110 **Solution:** Move down find, id: 6 supp tuples:4462 lvl:0100 **Parametres:**10, 6000, 2 1 1 0 Move down find, id: 26 supp tpls:3778 lvl:1100 **Parametres:**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 **Parametres:**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 **Parametres:**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 level:2 2 0 0	**Parameters:**50, 600, 2 1 1 0 **Solution:** Rlx1: id:47 supp tuples:9476 level:2 1 1 0 Rlx2: id:86 supp tuples:527 level:4 1 0 0 Rlx3: id:47 k:5 Supp_tupp:470 **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 level:1 1 0 0

Auxiliary slides

PARTIAL LATTICE RESULTS

Grouping power for QI=6

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

Level	relImp()
age1	1.70
age4	1.56
occupation1	1.42
occupation2	1.38
age3	1.30
education3	1.30
age2	1.30
education4	1.26
work_class2	1.24
marital_status3	1.16
marital_status2	1.14
race2	1.13
education2	1.09
work_class1	1.06
education1	1.05
marital_status1	1.05
race1	1.02
work_class3	1.00
race0	0.98
marital_status0	0.96
education0	0.95
work_class0	0.94
occupation0	0.71
age0	0.59

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).