[Title removed for anonymity]

Graham Cormode
graham@research.att.com

Magda Procopiuc (AT&T)
Divesh Srivastava (AT&T)
Thanh Tran (UMass Amherst)

Introduction

- Privacy is a common theme in public discourse
  - Privacy on social networks
  - Privacy of browsing activity on the web
  - Privacy in public places (e.g. in airports)
- Although there has been much effort in computer security, privacy is a growing field
  - Initial efforts: data swapping (1980s) k-anonymity (~2000), etc.
  - Current focus: differential privacy (2005 onwards)

Differential Privacy

- Principle: information released reveals little about any individual
  - Even if adversary knows (almost) everything about everyone else!
- Thus, individuals should be secure about contributing their data
  - What is learnt about them is about the same either way
- Much work on providing differential privacy
  - Simple recipe for some data types e.g. numeric answers
  - More complex for arbitrary data (exponential mechanism)

Differential Privacy

An adversary knowing information of everyone except one individual cannot deduce information about that individual

A randomized algorithm K satisfies ε-differential privacy if:

Given any pair of neighboring data sets, D_1 and D_2, and S in Range(K):

$$\Pr[K(D_1) = S] \leq e^{\varepsilon} \Pr[K(D_2) = S]$$

Achieving ε-Differential Privacy

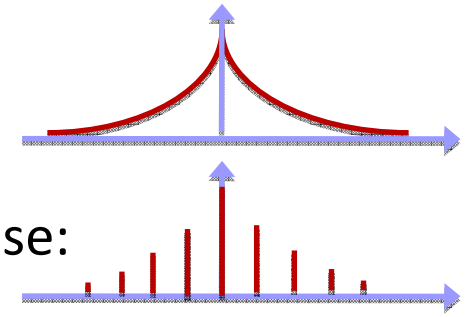
(Global) Sensitivity of publishing:

$s = \max_{x,x'} |F(x) - F(x')|$, x, x' differ by 1 individual

E.g., one individual changing his/her info affects at most 2 entries in the frequency matrix; hence $s = 2$

For every value that is output:

- Add Laplacian noise, Lap(ε/s):
- Or Geometric noise for discrete case:



Applying Differential Privacy

- This talk: applying differential privacy to real data
- Part 1: Applying differential privacy to sparse data
  - How to take arbitrary data and publish under DP?
  - Make the data efficient to generate and use
- Part 2: Understanding differential privacy guarantees
  - Aim to better understand the privacy guarantees
  - Show the limitations due to population statistics

Publishing Anonymized Data

- Census data:
  - On-the-Map dataset contains micro data on home and work location distribution for the states in the US
- Tale of two cities:
  - Location information from wireless cellular networks is used to understand human mobility patterns
- In general, given arbitrary data, we want to publish a differentially private version of it
  - General approach: represent as contingency table, add noise

Data Sparsity: Examples

- To publish the work-home commute data in the US:
  - Number of census tracts in the US: ~ 10^6
  - Size of the frequency matrix: $(10^5)^2 \approx 10^{10}$
  - Consider data of 10 million residents
  - At most 10^{-3} density, the data is 99.9% sparse!
  - Only gets worse as dimensionality increases

Achieving ε-Differential Privacy

Input Data Table → transform → Frequency Matrix (sparse) → add noise → Anonymized Data (dense)

TID	Home	Work
1	A	X
2	B	X
3	B	X
4	A	Y
5	Z	C
6	A	X
7	B	X
…	…	…

	A	B	C	X	Y	Z
A	0	0	0	40	25	0
B	0	0	0	30	0	0
C	0	0	0	0	0	0
X	0	5	0	0	0	0
Y	0	0	10	0	0	0
Z	0	0	5	0	0	0

	A	B	C	X	Y	Z
A	0	2	0	43	22	0
B	4	1	2	35	0	0
C	1	0	0	3	0	3
X	0	3	5	1	1	2
Y	2	0	6	2	0	2
Z	4	3	6	0	3	0

Objectives

Anonymization can make sparse input dense and noisy

$\Rightarrow$ High cost for storing and processing

$\Rightarrow$ Reduced data utility

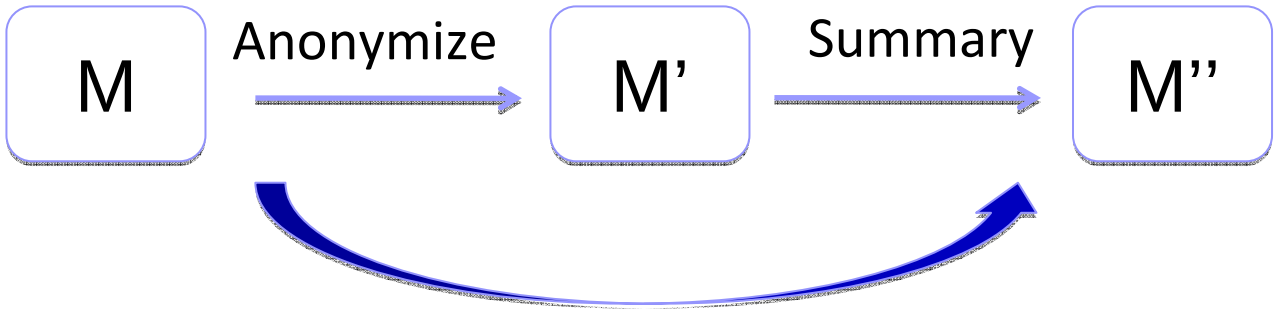
Objectives: To publish data sets so that:

- they have differential privacy
- the publishing process is efficient
- queries are answered with high utility

Approaches: adopt techniques from data summarization, with new analysis & algorithms to meet these different objectives

Shortcut Approach

- Post-processing anonymized data doesn't breach privacy
- So publish a summary of the anonymized data



- Goal: generate M'' directly from M, shortcutting M'

Summarization Techniques

- Goals of summarization:
  - Size of the published data ≈ size of original data, n
  - Give good answers e.g. for count queries over ranges
  - Speed up the anonymization process
- Uniform sampling does not suffice for sparse data since the sample contains most of the noise from zero entries
- Consider various data-dependent summarization techniques:
  - Simple filtering
  - Threshold/priority sampling
  - (Sketches)
  - Combinations of the above

Filtering

- Goal: retain all (noisy) values greater than θ
- Handle non-zero entries directly
  - Add noise from geometric dbn $\Pr[|X|=x] = (1-\alpha)/(1+\alpha)\ \alpha^{|x|}$, ($\alpha = \exp(-\varepsilon/2)$), test against filter
- For zero values, determine probability of passing filter:
  - $\Pr[\ M'(i) > \theta \mid M(i)=0\ ] = p_\theta = \sum_{x>\theta} (1-\alpha)/(1+\alpha)\ \alpha^{|x|} = \alpha^\theta/(1+\alpha)$
- Binomial dbn $B(m-n,p_\theta)$ sets how many locations to pick
- Draw noise values conditioned on being > θ:
  - $\Pr[M'(i) = b \mid M'(i) > \theta] = \Pr[M'(i)=b] / \Pr[M'(i)>\theta\ ] = (1-\alpha)\alpha^{b-\theta}$
  - Use CDF: $\Pr[\ M'(i) < x \mid M'(i) > \theta] = \sum_{\theta \le b \le x} (1-\alpha)\alpha^{b-\theta} = 1 - \alpha^{x-\theta+1}$

Threshold Sampling

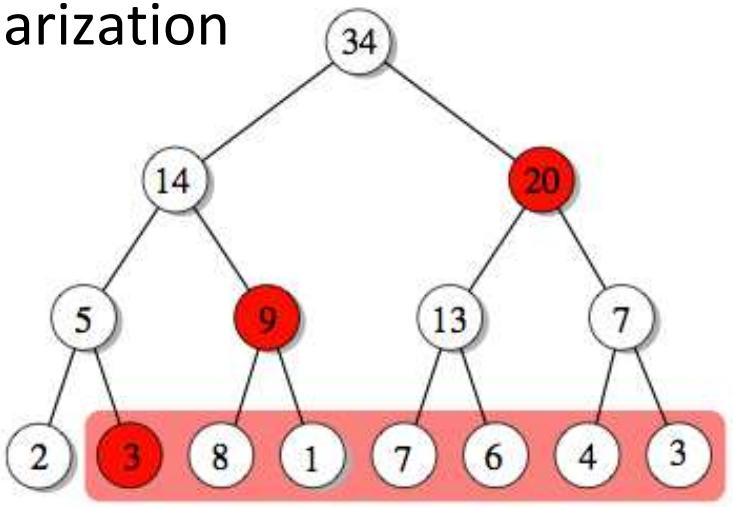
- Goal: sample weight w with probability $\min\{1, w/\tau\}$
- Again, handle non-zeros directly
- For zero-values, find probability of being sampled:
  - $\Pr[\, i \in S\,] = \sum_v \Pr[\, i \in S \mid M'(i) = v]\, \Pr[\, M'(i)=v] = p_\tau$
  - Calculate $p_\tau = 2\alpha(1-\alpha^\tau)/(\tau(1-\alpha^2))$
- Can find CDF of values, conditioned on being in the sample:
  - $\Pr[X = v \le -\tau] = \tau\, \alpha^{-v} C_\tau (1-\alpha)$
  - $\Pr[-\tau < X = v \le 0] = C_\tau\, (-\, v\, \alpha^{-v} + (v+1)\, \alpha^{-v+1} - \alpha^{\tau+1})$
  - $\Pr[\, 0 < X = v \le \tau] = \tfrac{1}{2} + \alpha\, C_\tau (1 - (v+1)\alpha^v + v\alpha^{v+1})$
  - $\Pr[\tau < X = v] = \tfrac{1}{2} + \alpha\, C_\tau (1 - \alpha^\tau - \tau\alpha^v\, (1-\alpha))$
  - where $C_\tau = 1/(2\alpha\, (1-\alpha^\tau))$

Filter + Threshold

- Combine both methods: sample with τ values above θ
- For zero-values, find probability of being sampled:
  - $\Pr[\, i \in S \,] = \sum_{\nu} \Pr[\, i \in S \mid M'(i) = \nu > \theta] \Pr[\, M'(i) = \nu > \theta \,] = p_{\theta,\tau}$
  - Calculate $p_{\theta,\tau} = 2/(\tau(1-\alpha^2)) \cdot (\theta\, \alpha^{\theta} - (\theta-1)\, \alpha^{\theta+1} - \alpha^{\tau+1})$
- CDF of values, conditioned on being in the sample:
  - $\tau\, C_{\theta,\tau}\ (1-\alpha)\ \alpha^{-\nu}$, if $\nu \le -\tau$
  - $C_{\theta,\tau}\ (-\nu\, \alpha^{-\nu} + (\nu+1)\alpha^{-\nu+1} - \alpha^{\tau+1})$, if $-\tau < \nu \le -\theta$
  - $\frac{1}{2} + C_{\theta,\tau}\, (\theta\, \alpha^{\theta} - (\theta-1)\, \alpha^{\theta+1} - (\nu+1)\alpha^{\nu+1} + \nu\, \alpha^{\nu+2})$, if $\theta \le \nu \le \tau$
  - $1 - \tau\, C_{\theta,\tau}\ (1-\alpha)\, \alpha^{\nu+1}$, if $\nu > \tau$
  - where $C_{\theta,\tau} = 1/(2(\theta\, \alpha^{\theta} - (\theta-1)\, \alpha^{\theta+1} - \alpha^{\tau+1}))$
- Can carefully manipulate sample to get to desired size

Optimizing for Range Queries

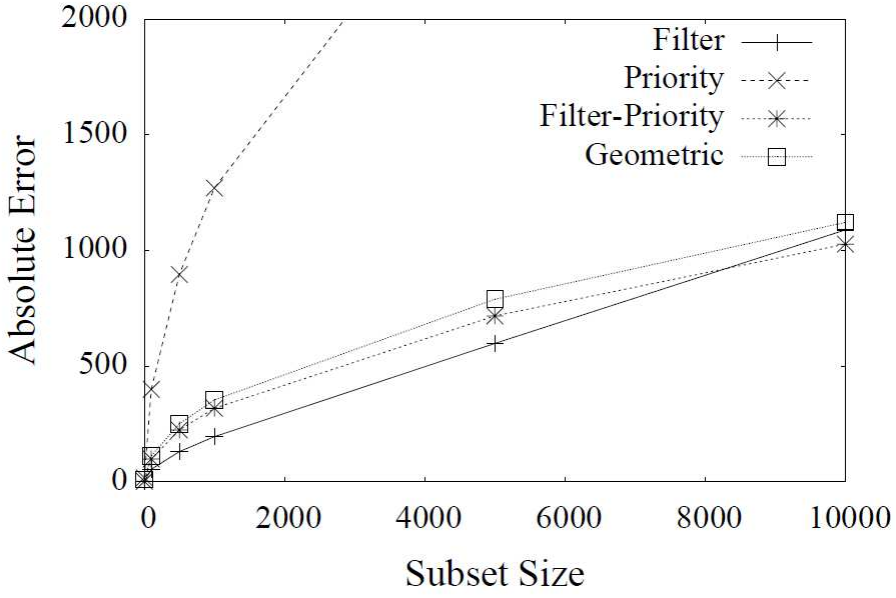
- Standard technique: decompose range into subranges
  - Dyadic ranges: length is power of 2, at a multiple of length
  - Any range < u can be written as log u subranges
- Useful for accuracy:
  - Only log u noise, not u (volume of noise is a log factor more)
- Can combine with summarization



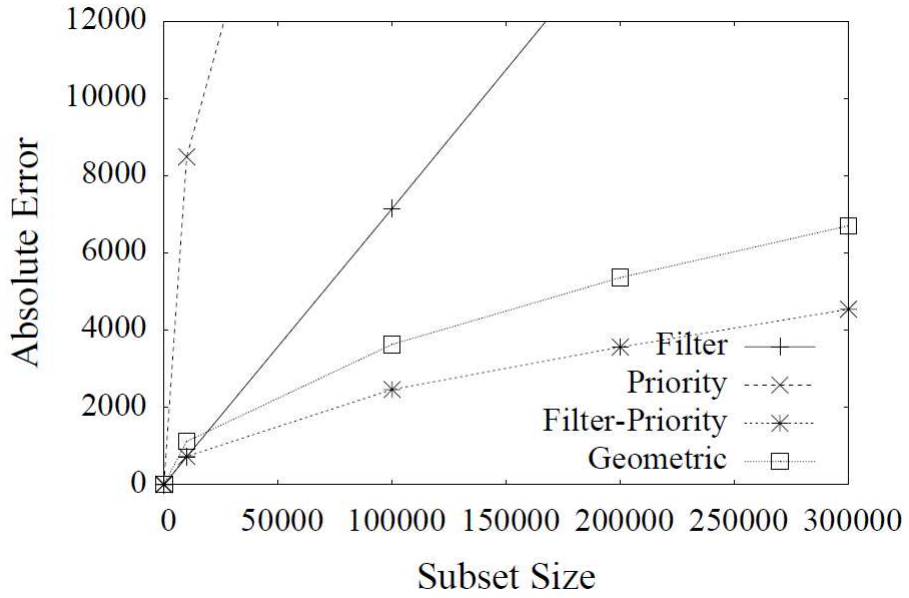
Experiments

- Compare different techniques to (exhaustive) addition of noise to every cell
- Use “real” and synthetic data
- Control parameters:
  - Data density (n/m)
  - Mean and standard deviation of (non-zero) data
  - Size of range queries
- Measure absolute error in range query answer

Synthetic Data



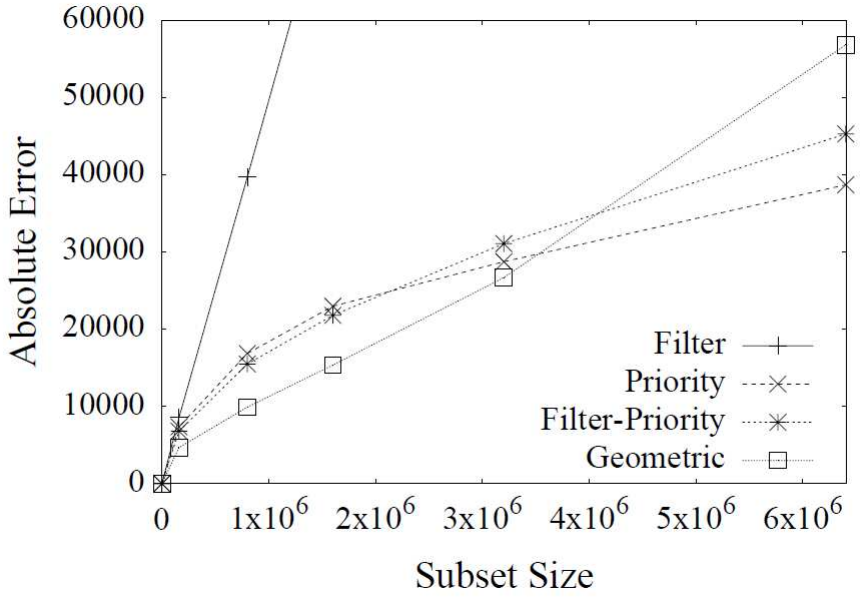
Small subsets, μ=100, σ=20, n/m=0.1

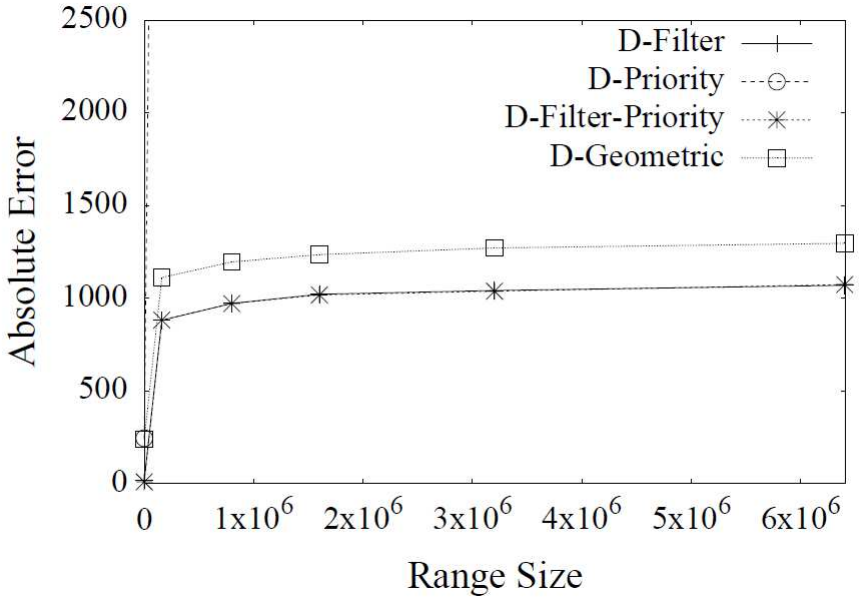


Large subsets, μ=100, σ=20, n/m=0.01

- Build summary of size n
- Sampling is capturing M', including noise
- Combination of filtering and sampling most effective
  - Improves over "baseline" method

OnTheMap data





- Here, sampling is more accurate
- Dyadic ranges effective at bounding error
  - For even short ranges
  - Combination of dyadic and summarization works well

Summary

- Effective way to publish data with differential privacy
- Combination of filtering and sampling seems most effective over a variety of settings
- Dyadic ranges are useful, when ranges are sufficiently large (a constant fraction of the data space)
- Future work: differentially private space decompositions
  - E.g. kd-trees, histograms

Outline

- Introduction
- Part 1 – differential privacy on sparse data
- Part 2 – when differential privacy isn't enough

Revisiting the definition

- Differential privacy guarantees that what I learn about an individual from the released data is about the same whether or not they are in the data
- So I can't learn much about an individual from the released data, right?
- WRONG!
  - Will show how applying differential privacy can still allow us to learn about individuals

Build a Classifier

- Key idea: build an accurate classifier under DP
  - Similar ideas have been used to attack other privacy definitions
- Data model: target ("sensitive") attribute $s \in SA$
  - Think disease status, salary band, etc.
- "Observable" attributes $t_1, t_2 \ldots t_m$
  - Think age, gender, zip code, height etc.
- Goal: build a classifier that given $(t_1, t_2, \ldots t_m)_i$ predicts s_i
  - An accurate classifier reveals the private information

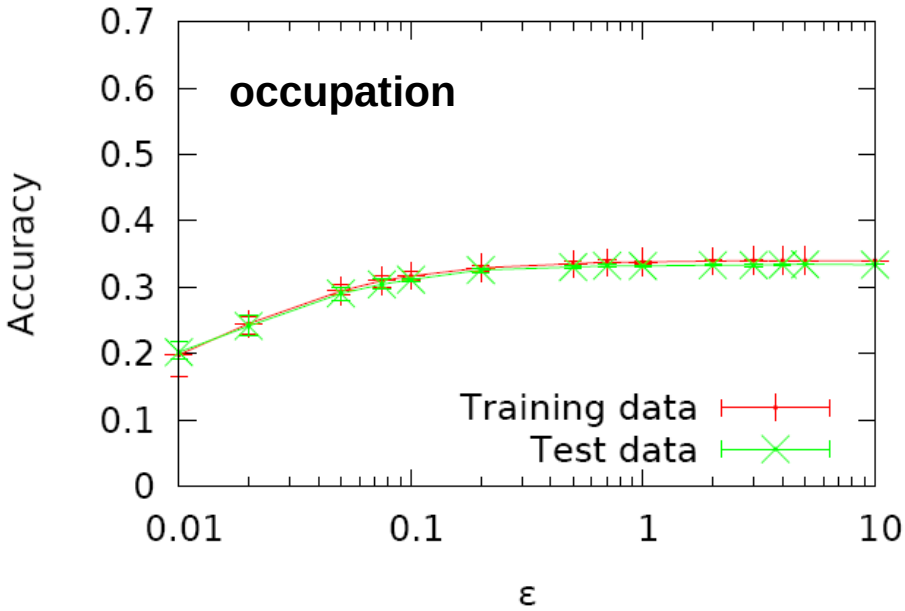
Building the Classifier

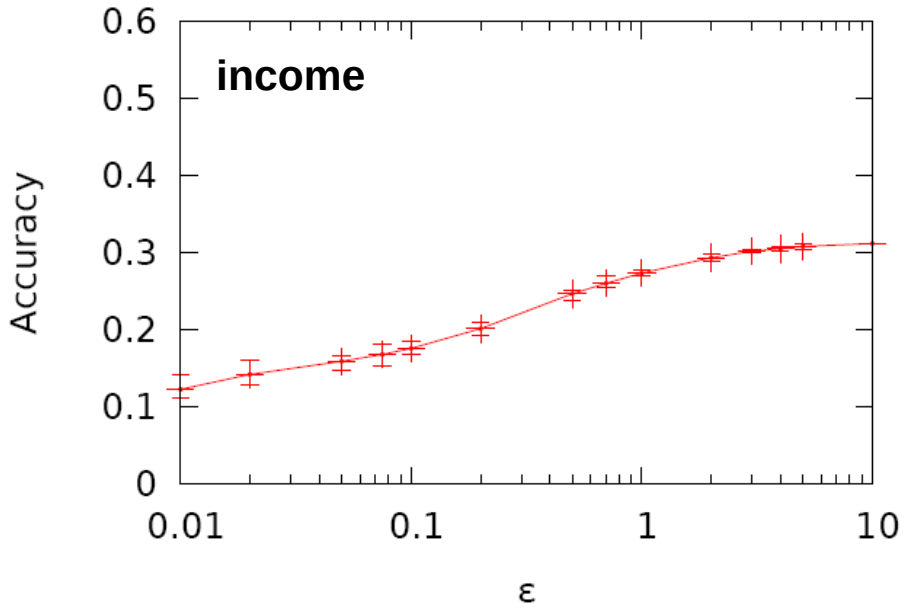
- Build a naïve Bayes classifier for s:
  - Prediction is $s\tilde{} = \arg\max_{s \in SA} \Pr[s] \prod_{j=1}^{m} \Pr[t_i \mid s]$
- Parameters are the marginal distributions
  $\Pr[t_i|s] = \Pr[t_i \cap s]/\Pr[s] \approx |\{r \in T : r_i = t_i \cap r_s = s\}|/|\{r \in T : r_s = s\}|$
- Just need the counts $\forall s \in SA, i, v \in T_i \; |\{r \in T : t_i = v \cap r_s = s\}|$
  - Can obtain "noisy" versions of these under differential privacy
- Minor corrections: add 1 to counts (Laplacian correction), round up to 1 if negative due to noise

Impact of Noise

- DP adds noise to mask out the effect of one individual
- But when data contains many individuals, noise will be minimal – essentially at the level of sampling error
- May need to do some grouping / bucketing of values to ensure sufficient density (e.g. replace salary with ranges)

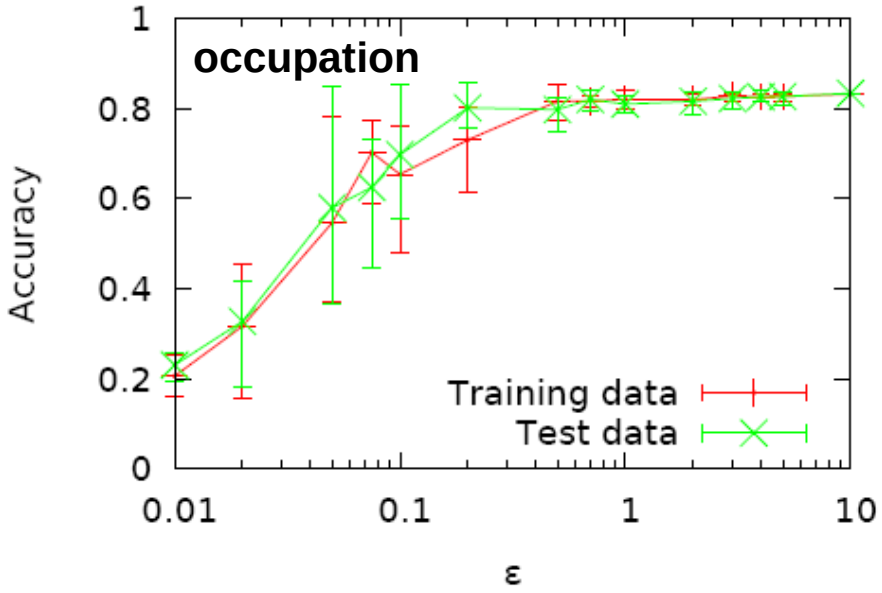
Experimental Study

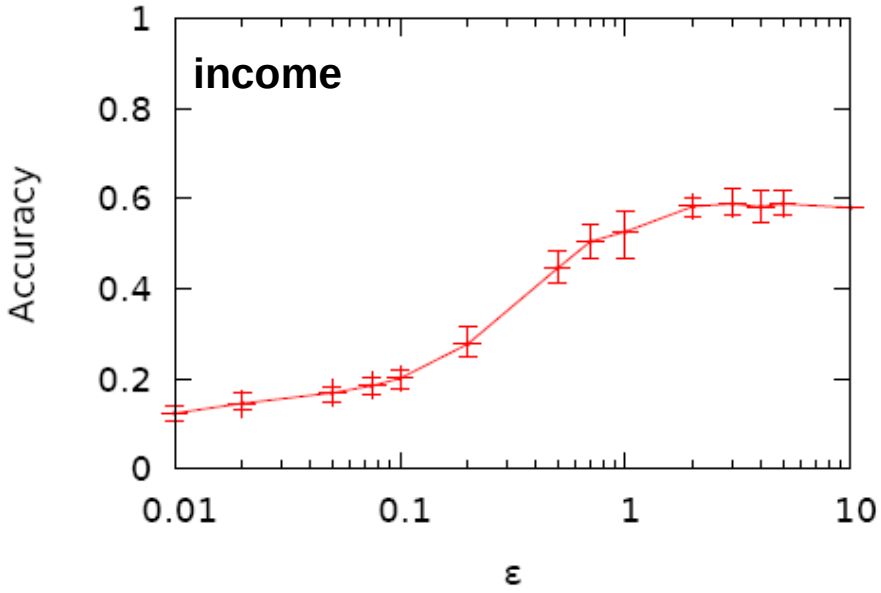




- Tested accuracy of predicting
  - 'occupation' in UCI Adult data set (focus of previous work)
  - 'income' in UCI Internet-usage data set
- Clear improvement in accuracy over baseline methods
  - E.g. just predict most common attribute value

High Confidence Results





- When restricting to high-confidence predictions (~ 10% of the data), accuracy is yet higher

Discussion

- Why does this work?
  - The classifier is based on correlations between the observable attributes and the target attribute
  - These are *population statistics*: they arise from the coarse behavior of the whole population
  - One individual has almost no influence on them
  - More directly, the noise added to mask an individual does not substantially change them until the noise is very large
- Differential privacy is behaving as advertised
  - What we learn about the individual really is the same whether they are there or not

Enabling Disclosure

- Should we be worried? Correlations are inherent in the data?
  - Suppose the data set covers thousands of individuals, took great expense and effort to gather
  - An 'attacker' would never go to this effort to collect data
  - But almost 'for free' they can use the data (with privacy) and potentially compromise an individual's privacy
- "If the release of the statistic S makes it possible to determine the (microdata) value more accurately than without access to S, a disclosure has taken place" – T. Dalenius, 1977
  - DP does not prevent disclosure, even when the attacker has no other information

Limitations

- Ultimately, the information revealed is no more than the best classifier that can be build on the data
  - We have to consider the 'sensitivity' of the classifier: Naïve Bayes has low sensitivity, and so is effective; others have higher sensitivity
- The classifier is only probabilistically accurate: we don't know when it is correct (though its confidence is correlated)
- This approach: require target to have small cardinality
  - But regression approaches should also work

Comparison to other attacks

- Learning attacks have been used on other anonymization schemes
  - "deFinetti attack" [Kifer 09] follows a similar tack: build classifier from released l-diverse data, use this to infer sensitive data
  - Extra strength: can use additional 'grouping' information to eliminate possibilities and further improve accuracy
  - However, increasing the group size weakens this
  - Bottom line: ultimately, similar accuracy for attacker with DP or l-diversity under this kind of learning attack

Concluding Remarks

- Have demonstrated that differentially private data can still be used to learn private data about individuals
- Relies on basic correlations in data – we would not in general want to remove these (these are the utility of the data!)
- Means that merely ensuring DP is not sufficient for privacy
  - Can't just apply DP and forget it: must think more deeply about whether data release provides sufficient privacy for data subjects