Differential Privacy in the Wild (Part 2)

A Tutorial on Current Practices and Open Challenges

Outline of the Tutorial

1. What is Privacy?
2. Differential Privacy
3. Answering Queries on Tabular Data

   Break

4. Applications I: Machine Learning
5. Privacy in the Real World
6. Applications II: Networks and Trajectories

MODULE 4: APPLICATIONS I: MACHINE LEARNING

Module 4: Applications I

- Private Empirical Risk Minimization
  - E.g. SVM, logistic regression
  - Make a specific learning approach private
- Private Stochastic Gradient Descent
  - E.g. Deep learning
  - Make a general purposed fitting technique private
- Other Important Problems in Private Learning

Differentially Private Machine Learning

Predicts flu or not, based on patient symptoms

Trained on sensitive patient data

Credit: Chaudhuri

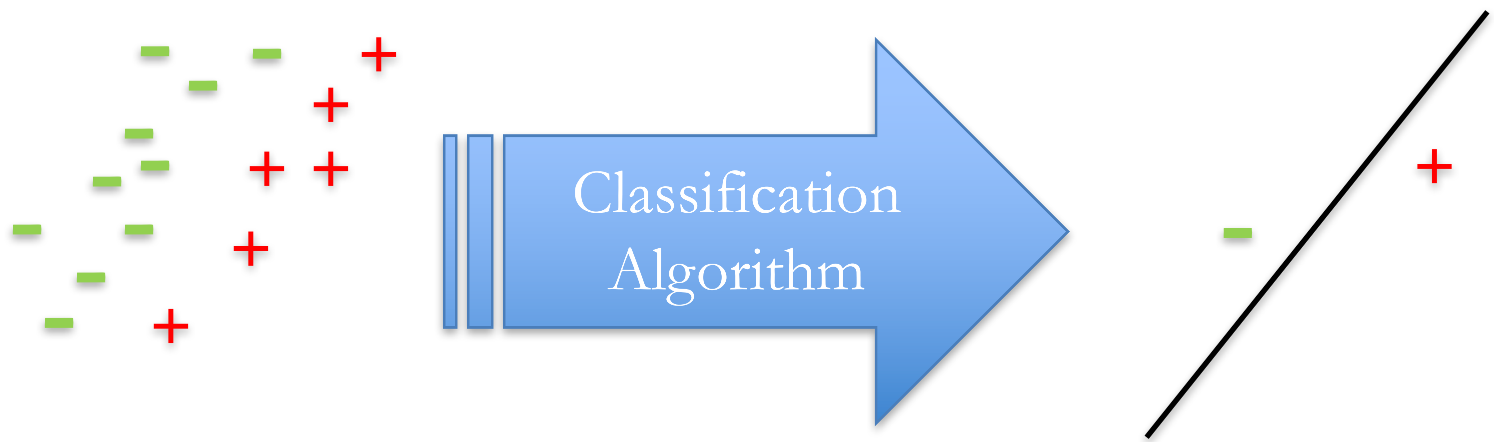
From Attributes to Labeled Data

Yes	No	99F	No
Sore Throat	Fever	Temperature	Flu?

↓

1	0	99	-
Data			Label

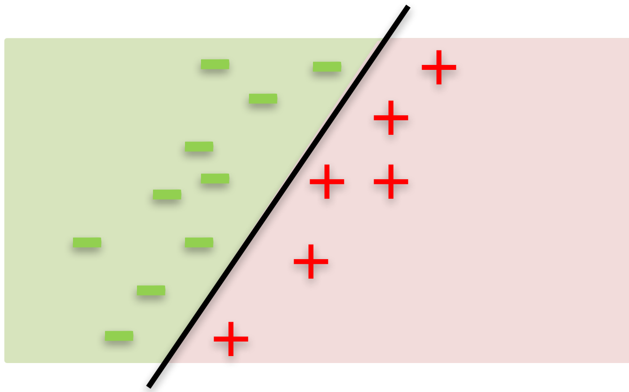
Classifying Sensitive Data



Private
Data

Public
Classifier

Classifying Sensitive Data



Distribution P over labelled examples

Goal: Find a vector w that separates + from - for most points from P

Key: Find a simple model to fit the samples

Empirical Risk Minimization

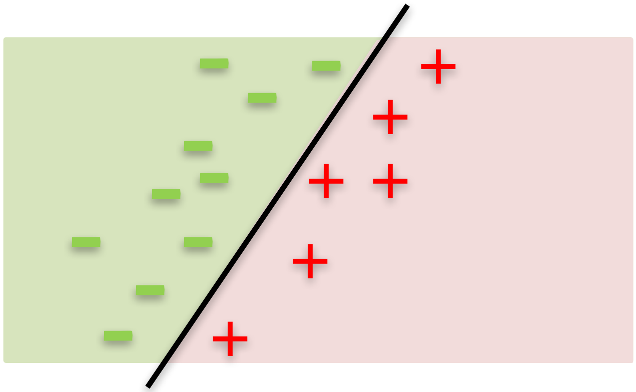
- Training dataset:
  - Labeled data $D = \{(x_i, y_i) \in X \times Y : i = 1,2, \ldots, n\}$
  - e.g binary classification $X = R^d$, $Y = \{-1, +1\}$
  - Train predictor over D: $\omega : X \to Y$

- **Empirical risk** (or error) of ω over D is

$$\frac{1}{n}\sum_{i=1}^{n} l(\omega, (x_i, y_i))$$

  - l is a loss function: how well ω classifies (x_i, y_i)

Examples of Loss Function



Risk: Hinge loss $l(z) = \max(0, 1 - z)$
Optimizer: Support vector machines (SVM)

Risk: Logistic loss $l(z) = \log(1 + \exp(-z))$
Optimizer: Logistic regression

Regularized ERM

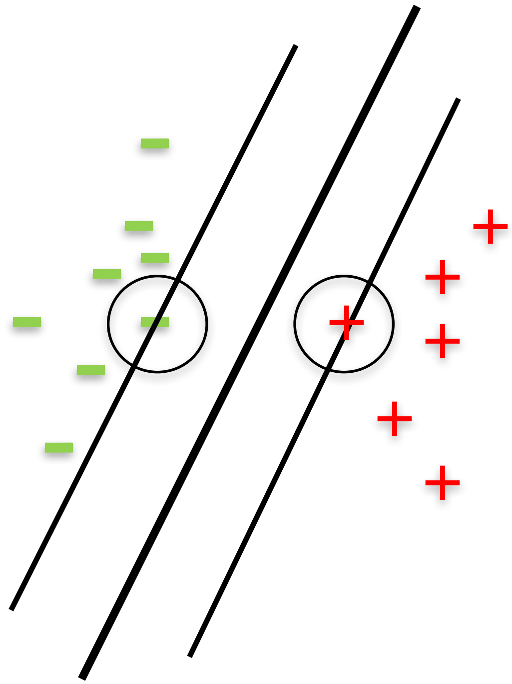
- **Goal:** Labeled data $D = \{(x_i, y_i)\}$, find

$$f(D) = argmin_{\omega} \frac{1}{2}\lambda \parallel \omega \parallel^2 + \frac{1}{n}\sum_{i=1}^{n} l(\omega, (x_i, y_i))$$

Regularizer
(Model Complexity)

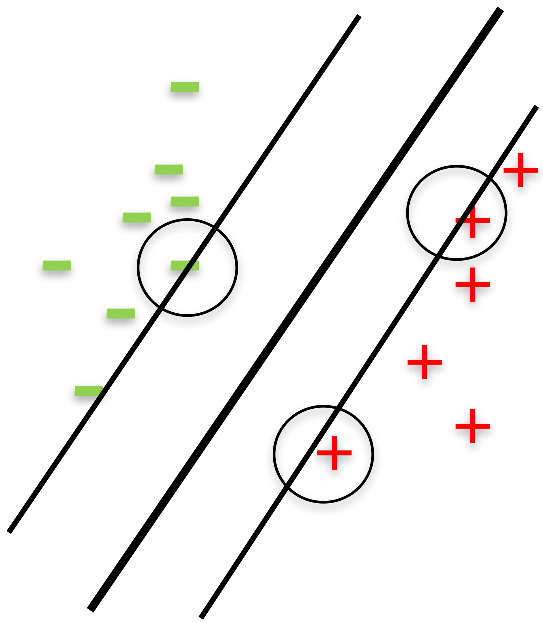
Risk
(Training Error)

Why ERM is not private for Support Vector Machine (SVM)?



SVM solution is a combination of support vectors
If one support vector moves, solution changes

Why ERM is not private for Support Vector Machine (SVM)?

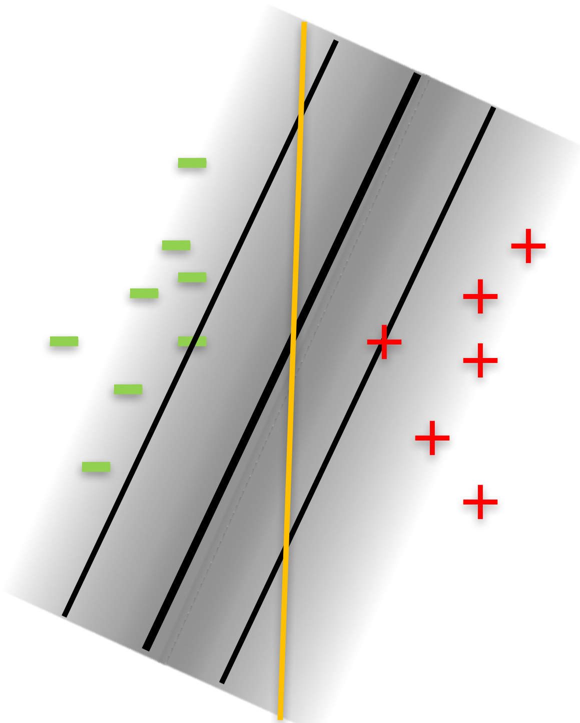


SVM solution is a combination of support vectors
If one support vector moves, solution changes

Module 4: Applications I

- Private Empirical Risk Minimization
  - E.g. SVM, logistic regression
  - Make a specific learning approach private
- Private Stochastic Gradient Descent
  - E.g Deep learning
  - Make a general purposed fitting technique private
- Other Important Problems in Private Learning

How to make ERM private?



Pick ω from distribution near the optimal solution

Output Perturbation

- **Goal:**

$$\tilde{f}(D) = f(D) + noise = \left[argmin_{\omega} \frac{1}{2}\lambda \parallel \omega \parallel^2 + \frac{1}{n}\sum_{i=1}^{n} l(\omega, (x_i, y_i))\right] + noise$$

Theorem: [CMS11] If $\parallel x_i \parallel \leq 1$ and l is 1-Lipschitz, then for any D, D' with $\text{dist}(D, D') = 1$,

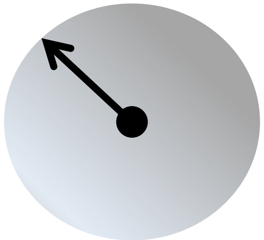
$$||f(D) - f(D')||_2 \leq \frac{2}{\lambda n} \quad (L_2\text{-sensitivity})$$

Output Perturbation

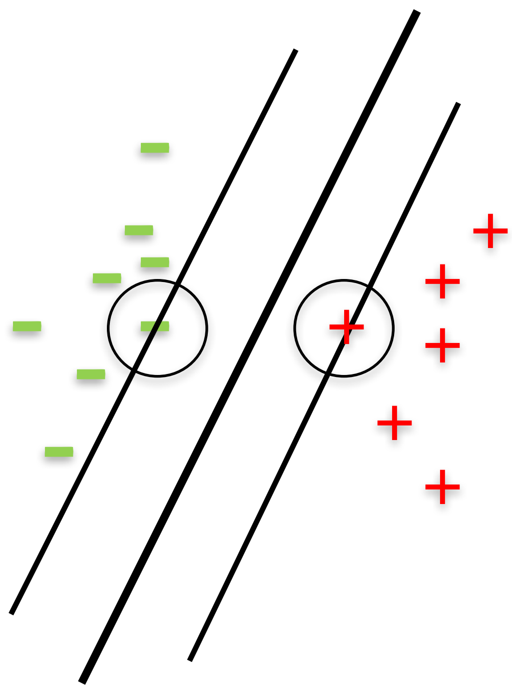
- **Goal:**

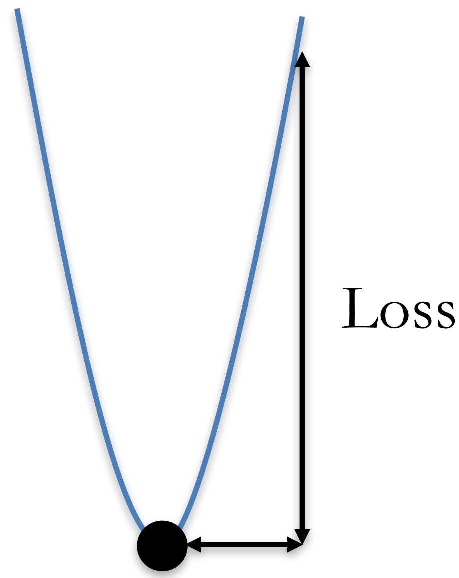
$$\tilde{f}(D) = f(D) + noise = \left[argmin_{\omega} \frac{1}{2}\lambda \parallel \omega \parallel^2 + \frac{1}{n}\sum_{i=1}^{n} l(\omega,(x_i,y_i))\right] + noise$$

- Laplace $noise$ drawn from
  - Magnitude: drawn from $\Gamma(\mathrm{d}, \frac{2}{\lambda n \epsilon})$
  - Direction: uniform at random



Property of Real Data





Optimization surface is very steep in some direction
→ High loss if perturbed in those directions

Objective Perturbation

- **Insight:** Perturb optimization surface and then **optimize**

$$\tilde{f}(D) = argmin_{\omega} \left[\frac{1}{2}\lambda \parallel \omega \parallel^2 + \frac{1}{n}\sum_{i=1}^{n} l(\omega,(x_i,y_i)) + noise\right]$$

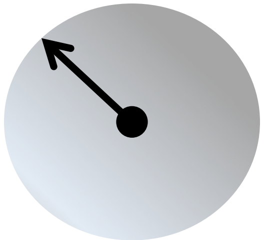
- **Main idea:** *add noise as part of the computation*:
  – Regularization already changes the objective to protects against overfitting.
  – Change the objective a little bit more to protect privacy.

Objective Perturbation

- **Insight:** Perturb optimization surface and then **optimize**

$$\tilde{f}(D) = argmin_{\omega} \left[\frac{1}{2} \lambda \parallel \omega \parallel^2 + \frac{1}{n} \sum_{i=1}^{n} l(\omega, (x_i, y_i)) + noise \right]$$

- $noise$ drawn from
  - Magnitude: drawn from $\Gamma(\mathrm{d}, \frac{1}{\epsilon})$
  - Direction: uniform at random



Objective Perturbation

- **Insight:** Perturb optimization surface and then **optimize**

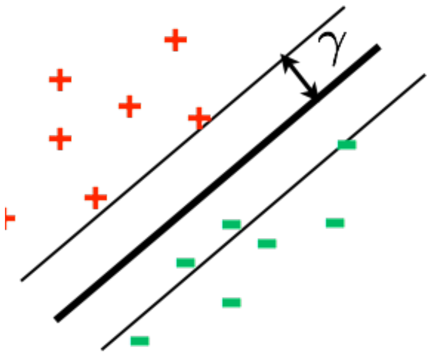
$$\tilde{f}(D) = argmin_{\omega} \left[\frac{1}{2} \lambda \parallel \omega \parallel^2 + \frac{1}{n} \sum_{i=1}^{n} l(\omega, (x_i, y_i)) + noise \right]$$

- **Theorem:** If l is convex and double-differentiable with $|l'(\mathrm{z})| \leq 1, |l''(\mathrm{z})| \leq c$ then Algorithm satisfy $\epsilon + 2 \log \left(1 + \frac{c}{n\lambda}\right)$-DP. [CMS11]

Accuracy

- Number of samples for error α w.r.t the best predictor
  - Fewer samples implies higher accuracy

d: #dimensions
γ: margin
ϵ: privacy
α: error
$\gamma, \alpha, \epsilon < 1$



- **Normal SVM:** $\frac{1}{(\alpha\gamma)^2}$
- **Objective perturbation:** $\frac{1}{(\alpha\gamma)^2} + \frac{d}{\alpha\epsilon\gamma}$
- **Output perturbation:** $\frac{1}{(\alpha\gamma)^2} + \frac{d}{\alpha^{1.5}\epsilon\gamma}$

Module 4: Applications I

- Private Empirical Risk Minimization
  - E.g. SVM, logistic regression
  - Make a specific learning approach private
- Private Stochastic Gradient Descent
  - E.g. Deep learning
  - Make a general purposed fitting technique private
- Other Important Problems in Private Learning

Stochastic Gradient Descent (SGD)

- Initial ω_0
- Incremental gradient update for $t = 0 \ldots T - 1$
  - Take a random example $(x_t, y_t) \in D$
  - Update $\omega_{t+1} = \omega_t - \eta_t(\nabla l(\omega_t, (x_t, y_t)))$
    - η_t is the step size

- Permutation-based SGD (PSGD)
  - Randomly permute training examples $D = \{(x_i, y_i)\}$ to feed each pass of SGD
  - Cycle D for k times: k-pass PSGD

White Box Approaches

- Initial ω_0
- Incremental gradient update for $t = 0 \ldots T-1$
  - Take a random example $(x_t, y_t) \in D$
  - Update $\omega_{t+1} = \omega_t - \eta_t(\nabla l(\omega_t, (x_t, y_t)) + noise)$
    - η_t is the step size

- Permutation-based SGD (PSGD)
  - Randomly permute training examples $D = \{(x_i, y_i)\}$ to feed each pass of SGD
  - Cycle D for k times: k-pass PSGD

White Box Approaches

- Cycle D for k times
- Basic composition:
  - Each pass is ϵ-DP, then k-pass is ϵk-DP.
  - Privacy loss grows linearly with the number of passes. [CSC13, SS15]
- Tighter privacy loss with advanced composition
  - Convex objectives [JKT12, BST14]
  - Deep learning with non-convex objectives [ACG16]

Advanced Composition

[DRV10]

- Composing k algorithms, each satisfying ϵ-DP ensures ϵ_g-DP with probability $1-\delta$

$$\epsilon_g = O\left(\epsilon\sqrt{k\ln\frac{1}{\delta}} + k\epsilon^2\right)$$

- Analyze privacy loss as a random variable: given output o and neighbors (D, D')

$$PL(o) = \ln\frac{\Pr[M(D)=o]}{\Pr[M(D')=o]}$$

Advanced Composition

[DRV10]

- Composing k algorithms, each satisfying ϵ-DP ensures ϵ_g-DP with probability $1-\delta$

$$\epsilon_g = O\left(\epsilon\sqrt{k\ln\frac{1}{\delta}} + k\epsilon^2\right)$$

- Each algorithm has privacy loss $PL(o)$
  - Worst case (DP): $\Pr[|PL(o)| \leq \epsilon] = 1$
  - Expected loss: $\mathrm{E}[PL(o)] \leq \epsilon(e^{\epsilon}-1)$
  - Total privacy loss ϵ_g is bounded by Azuma's inequality

Black Box Approaches

- Add noise to the final output of SGD [WLK17]
  - No need code change to the SGD program
  - Only sample noise once
  - Allow ϵ-DP and (ϵ, δ)-DP
  - Better convergence for constant number of passes based on the new bound over L_2 sensitivity of k-pass PSGD

"Bolt-on DP"
@ Thursday 2 PM DP Session)

Initialize ω_0
For $t = 0 \ldots T-1$
    $\omega_{t+1} \leftarrow$ update ω_t
Output ω_T

$\omega_T \leftarrow \omega_T$ + noise

L_2 sensitivity of k-pass PSGD

- l is β-smooth and L-Lipschitz, the L_2 sensitivity is
  - $2kL\eta$ if l is convex, $\eta_t = \eta \leq \frac{2}{\beta}$
  - $\frac{2L}{\lambda n}$ if l is λ-strongly convex, $\eta_t = \min(\frac{1}{\beta}, \frac{1}{\lambda t})$, $|D| = n$
  - Convergence when $k = O(1)$

	[WLK17] (black box)	[BST14] (white box)
Convex	$O(\frac{\sqrt{d}}{\sqrt{n}})$	$O(\frac{\sqrt{d}\log^{\frac{3}{2}} n}{\sqrt{n}})$
Strongly convex	$O(\frac{\sqrt{d}\log n}{n})$	$O(\frac{d\log^2 n}{n})$

Other Fitting Techniques

- Mini-batching SGD
  - At step t, the gradient is updated with a batch of examples B_t from D
  - Add noise per iteration
    - $\omega_{t+1} = \omega_t - \eta_t(E_{(x_i,y_i)\in B_t} \nabla l(\omega_t, (x_i, y_i)) + noise)$
  - Or add noise to the final output

- Proximal algorithm for strongly convex optimization [JKT12]
  - Add noise per iteration
  - Hard to implement than SGD

Module 4: Applications I

- Private Empirical Risk Minimization
  - E.g. SVM, logistic regression
  - Make a specific learning approach private
- Private Stochastic Gradient Descent
  - E.g. Deep learning
  - Make a general purposed fitting technique private
- Other Important Problems in Private Learning

Other Important Problems

- Practical issues
  - Parameter tuning: exponential mechanism [CMS11]
  - High dimensional data: random projection [WLK17]

- Solve non-convex optimization
  - Deep learning [SS15, ACG16]

- Understand what can be learned privately [KLNR11]
  - Private learning w/o efficiency: PAC, SQ
  - What cannot be learned privately? e.g. threshold functions where hypothesis space is infinite

DP Algorithms for ML

- Private ERM– a specific learning approach

Output perturbation

argmin (objective) + noise

Objective perturbation

argmin (objective + noise)

- Private SGD – a fitting technique

White box approaches

Initialize ω_0
For $t = 0 \dots T-1$
$\quad \omega_{t+1} \leftarrow$ update ω_t
$\quad \omega_{t+1} \leftarrow \omega_{t+1}$ + noise
Output ω_T

Black box approaches

Initialize ω_0
For $t = 0 \dots T-1$
$\quad \omega_{t+1} \leftarrow$ update ω_t
Output ω_T

$\omega_T \leftarrow \omega_T$ + noise

MODULE 5:
PRIVACY IN THE REAL WORLD

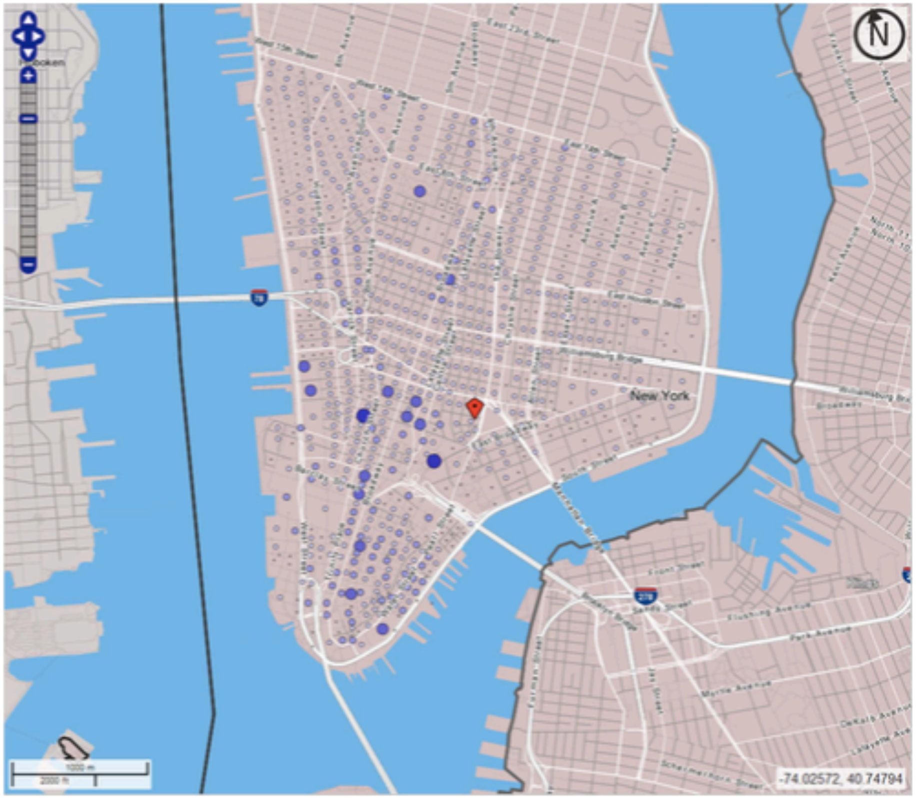
Module 5: Privacy in the real world

- Real world deployments of differential privacy
  - OnTheMap United States Census Bureau   RAPPOR chrome
- Privacy beyond Tabular Data
  - No Free Lunch Theorem
  - Customizing differential privacy using Pufferfish

OnTheMap

http://onthemap.ces.census.gov/

Employment in Lower Manhattan



Residential pattern of workers employed in Lower Manhattan

The maps above show LODES data in New York City in the OnTheMap application. The map on the left shows employment by census block in Lower Manhattan (in dense urban areas one census block is often equivalent to one city block). Large, dark dots have more employment than small, light dots. The map on the right shows the residential patterns of the same workers (those employed in Lower Manhattan). Workers employed in Lower Manhattan live throughout New York City as well as in New Jersey and other areas of New York state.

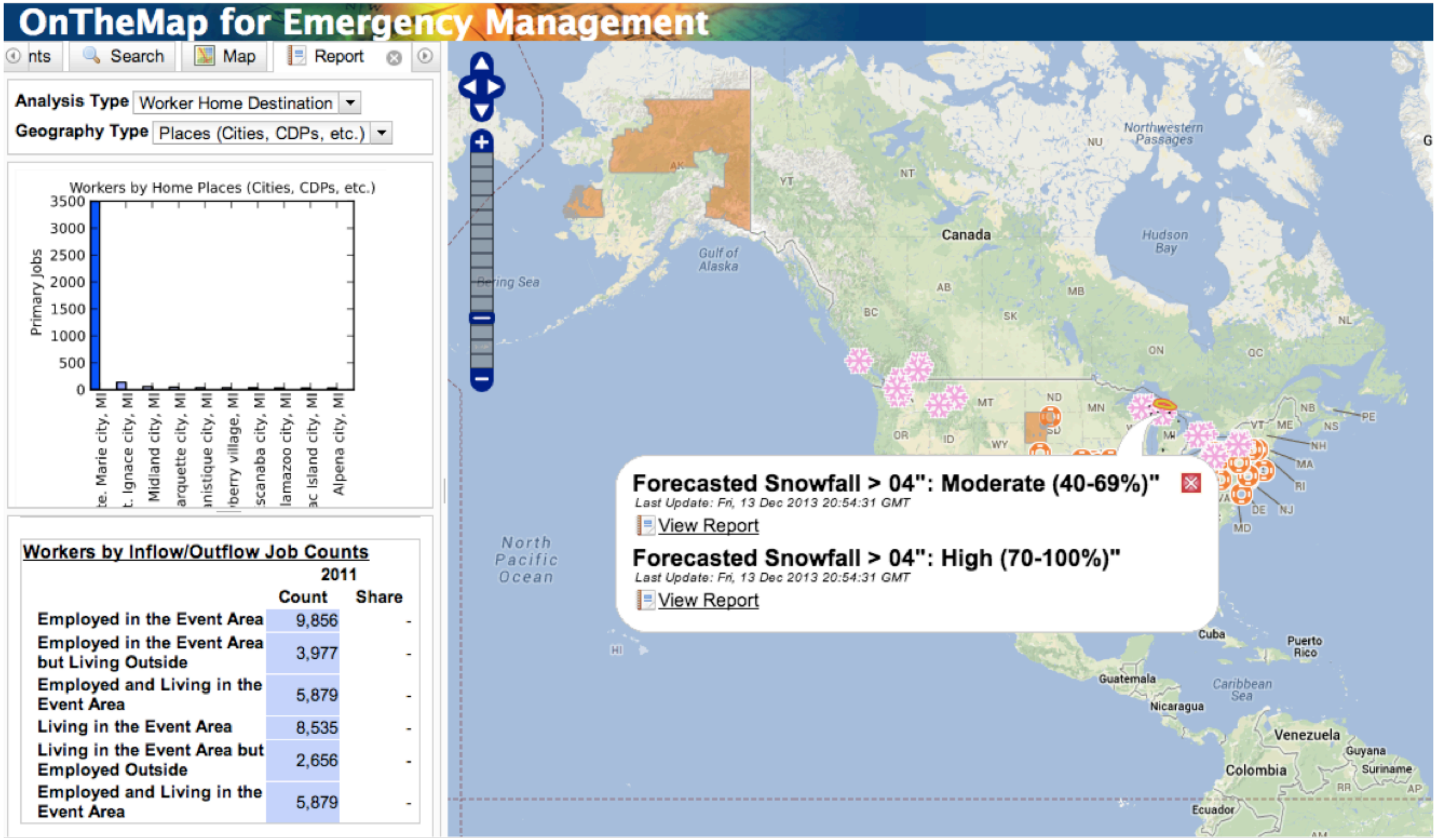
Data underlying OnTheMap

- Employee
  - Age
  - Sex
  - Race & Ethnicity
  - Education
  - Home location (Census block)

- Job
  - Start date
  - End date
  - Worker & Workplace IDs
  - Earnings

- Employer
  - Geography (Census blocks)
  - Industry
  - Ownership (Public vs Private)

Why release such data?



Why privacy is needed?

US Code: Title 13 CENSUS

It is against the law to make any publication whereby the data furnished by any particular establishment or individual under this title can be identified.

Violating the statutory confidentiality pledge can result in fines of up to $250,000 and potential imprisonment for up to five years.

OnTheMap

Residence (Sensitive)

Workplace (Quasi-identifier)

Worker ID	Origin	Destination
1223	MD11511	DC22122
1332	MD2123	DC22122
1432	VA11211	DC22122
2345	PA12121	DC24132
1432	PA11122	DC24132
1665	MD1121	DC24132
1244	DC22122	DC22122

Census Blocks

Current approach: Synthetic Database

- Sanitize the dataset one time
- Analyst can perform arbitrary computations on the synthetic datasets

- Unlike in query answering systems
  - No need to maintain state (of queries asked)
  - No need to track privacy loss across queries or across analysts

Synthetic Residence Generator (circa 2007)

Origin	Destination	# Workers
MD11511	DC22122	1
MD2123	DC22122	3
VA11211	DC22122	12
PA12121	DC24132	43
PA11122	DC24132	5
MD1121	DC24132	2
DC22122	DC22122	1

+

Noise
2
0
1
2
1
9
0

+

Dirichlet Resampling

No noise is added to origin-destination pairs with true count 0
Can lead to re-identification attacks.

Differentially Private Synthetic Data Generator

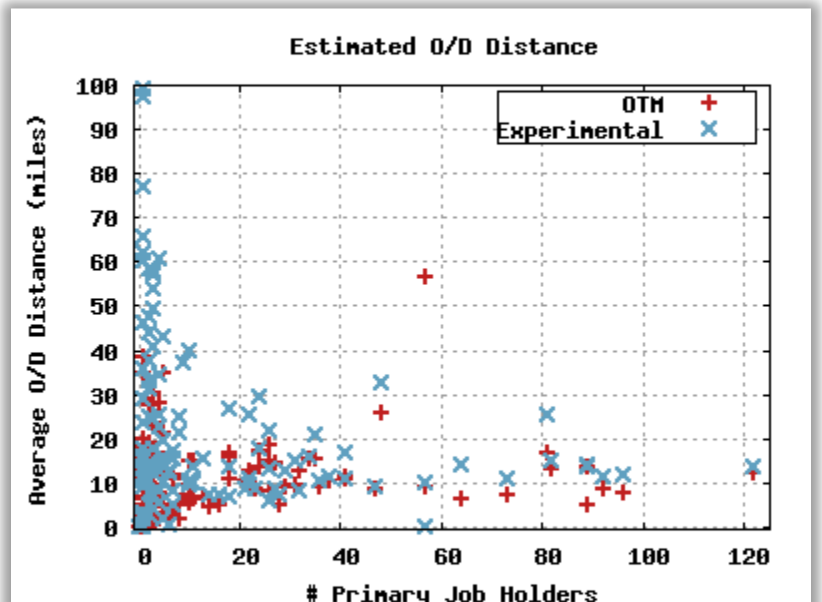
- Noise added to all origin-destination (o-d) pairs
  - Even if 0 count in the original dataset

- Noise calibrated to ensure a variant called *probabilistic* differential privacy

- Utility ensured by coarsening the domain and probabilistically dropping o-d pairs with no support.

Evaluation

- Utility measured by average commute distance for each destination block.



Experimental Setup:

- **OTM**: Currently published OnTheMap data used as original data.
- All destinations in Minnesota.
- 120,690 origins per destination.
  - chosen by pruning out blocks that are > 100 miles from the destination.

- Total $\varepsilon = 8.3$, $\delta = 10^{-5}$

Module 5: Privacy in the real world

- Real world deployments of differential privacy
  - OnTheMap United States Census Bureau RAPPOR chrome
- Privacy beyond Tabular Data
  - No Free Lunch Theorem
  - Customizing differential privacy using Pufferfish

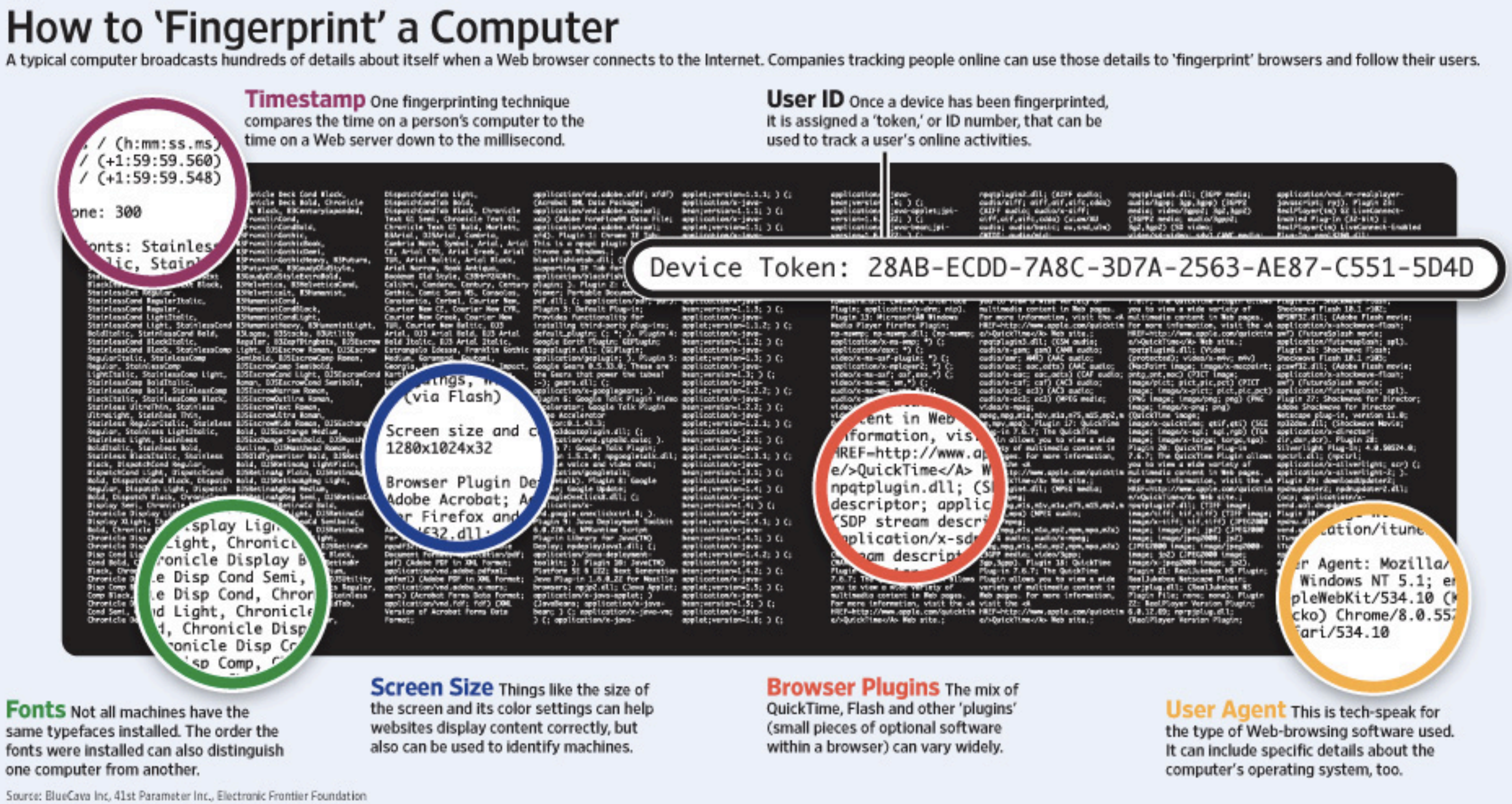
A dilemma chrome

- Cloud services want to protect their users, clients and the service itself from abuse.

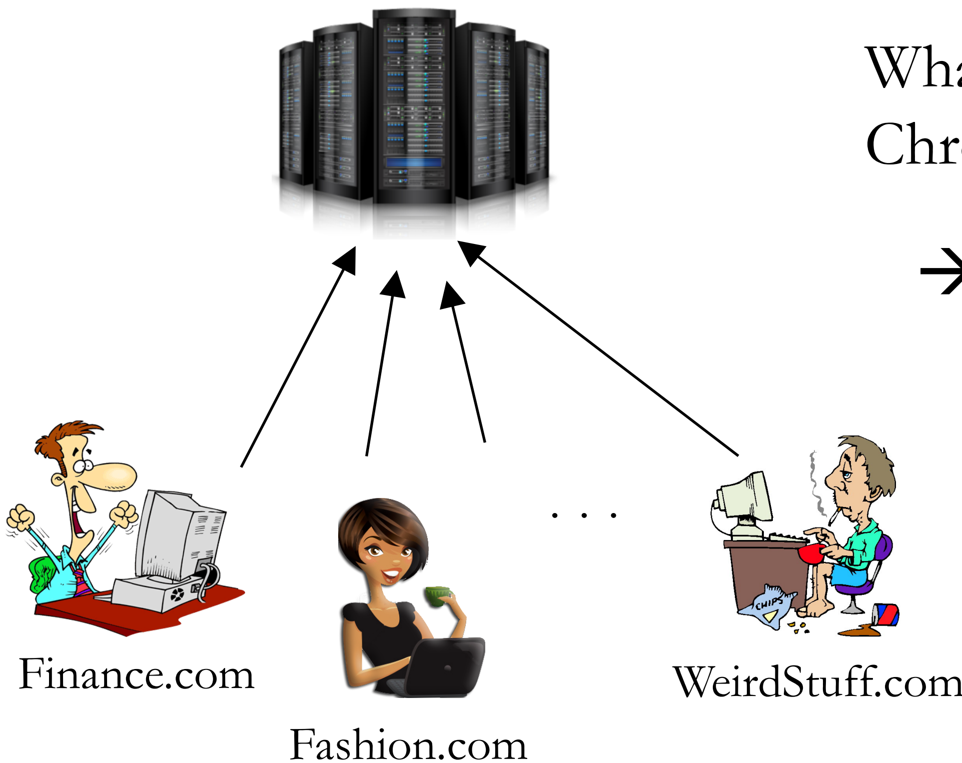
- Need to monitor statistics of, for instance, browser configurations.
  – Did a large number of users have their home page redirected to a malicious page in the last few hours?

- But users do not want to give up their data

Browser configurations can identify users



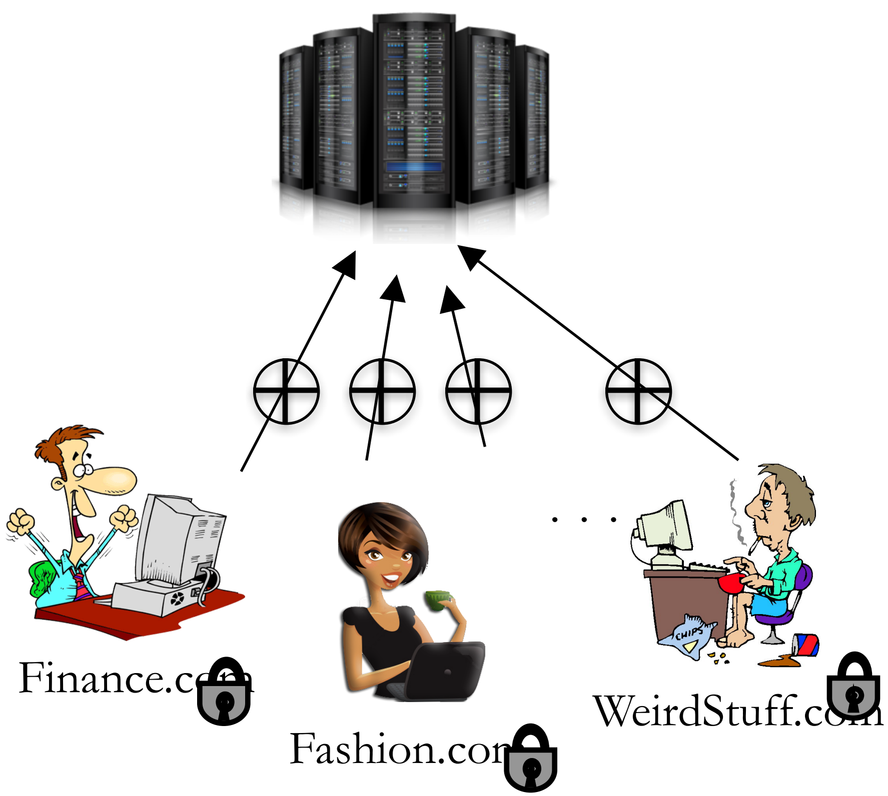
Problem
[Erlingsson et al CCS'14]



What are the *frequent* unexpected Chrome homepage domains?

→ To learn malicious software that change Chrome setting without users' consent

Why privacy is needed?



Liability (for server)

Storing unperturbed sensitive data makes server accountable (breaches, subpoenas, privacy policy violations)

Solution

Can use Randomized Response …

On a binary domain:

With probability p report true value

With probability 1-p report false value

… but the domain of all urls is very large …

… original value is reported with very low prob.

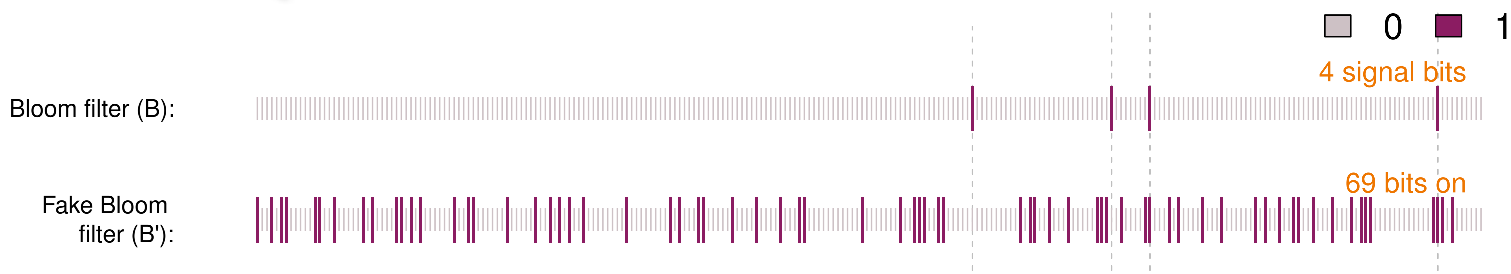
RAPPOR Solution

- Idea 1: Use bloom filters to reduce the domain size

Finance.com

RAPPOR Solution

- Idea 2: Use RR on bloom filter bits



RAPPOR Solution

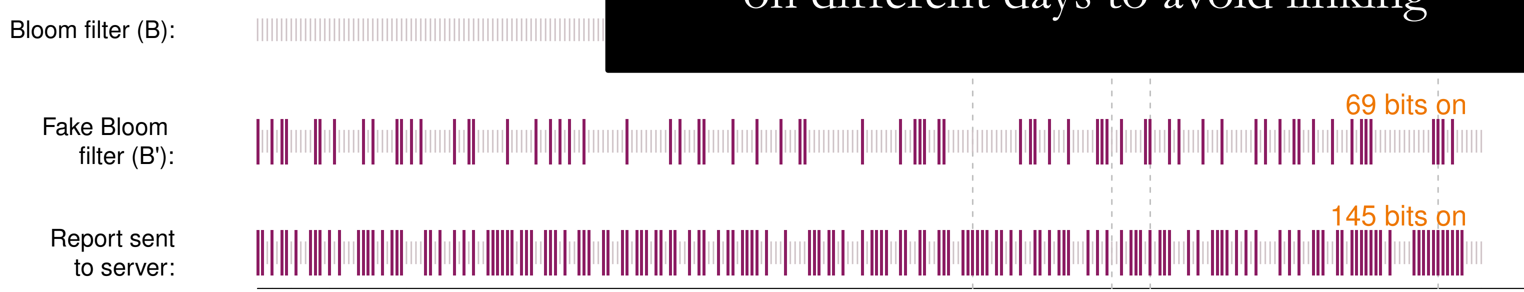
- Idea 3: Again use RR on the Fake bloom filter

Finance.com

Why randomize two times?

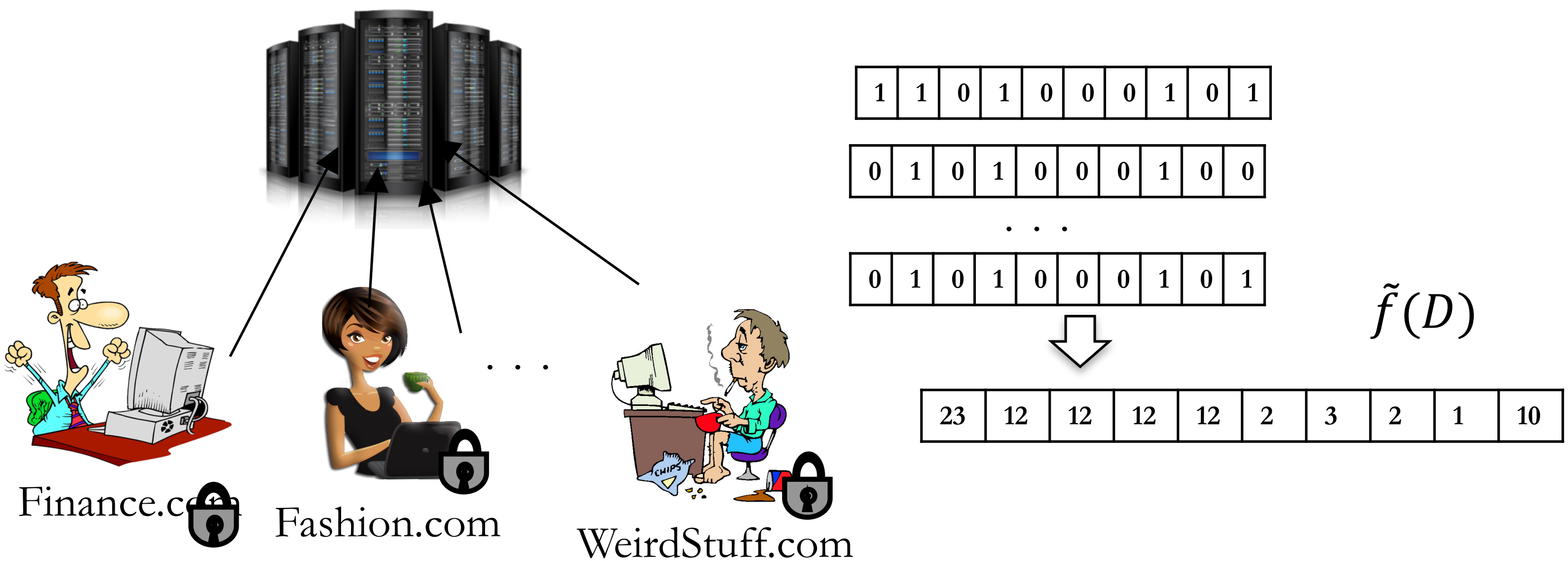
- Chrome collects information each day

- Want perturbed values to look different on different days to avoid linking



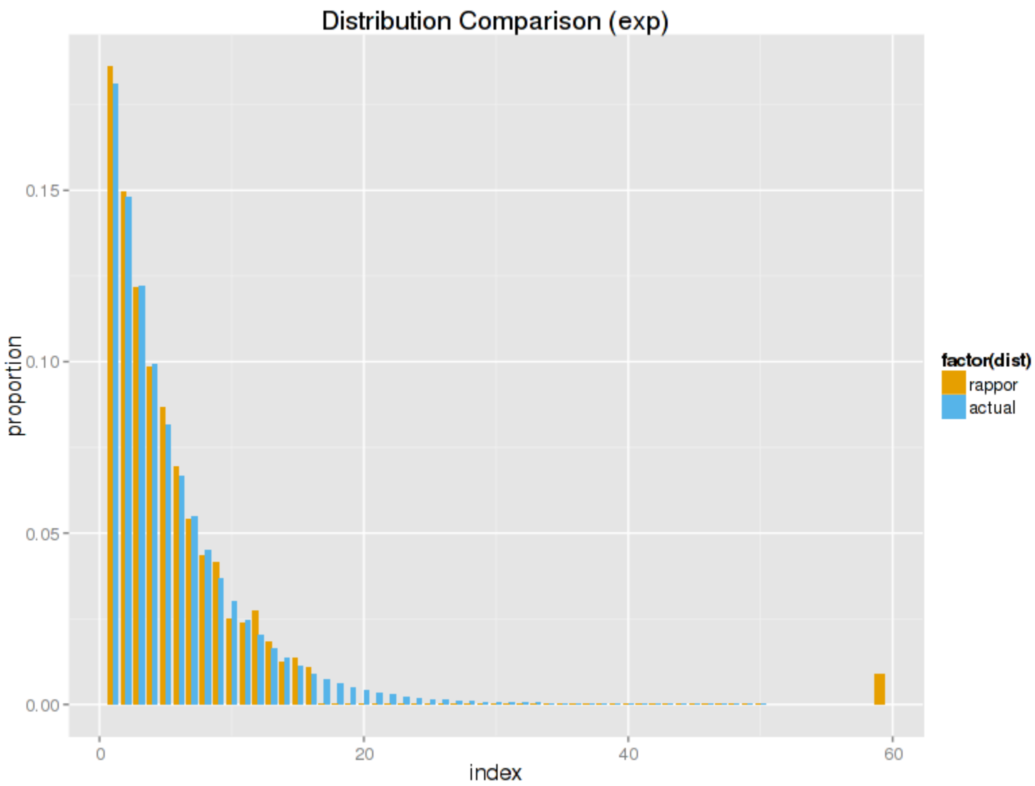
Server Report Decoding

- Step 5: estimates bit frequency from reports $\tilde{f}(D)$
- Step 6: estimate frequency of candidate strings with regression from $\tilde{f}(D)$



Evaluation

http://google.github.io/rappor/examples/report.html



Simulation Input

Number of clients	100,000
Total values reported / obfuscated	700,000
Unique values reported / obfuscated	50

RAPPOR Parameters

k	Size of Bloom filter in bits	16
h	Hash functions in Bloom filter	2
m	Number of Cohorts	64
p	Probability p	0.5
q	Probability q	0.75
f	Probability f	0.5

Other Real World Deployments

- Differentially private password Frequency lists [Blocki et al. NDSS '16]
  - release a corpus of 50 password frequency lists representing approximately 70 million **Yahoo!** users
  - varies from 8 to 0.002
- Human Mobility [Mir et al. Big Data '13]
  - synthetic data to estimate commute patterns from call detail records collected by **AT&T**
  - 1 billion records ~ 250,000 phones
- **Apple** will use DP [Greenberg. Wired Magazine '16]
  - in iOS 10 to collect data to improve QuickType and emoji suggestions, Spotlight deep link suggestions, and Lookup Hints in Notes
  - in macOS Sierra to improve autocorrect suggestions and Lookup Hints

Module 5: Privacy in the real world

- Real world deployments of differential privacy
  - OnTheMap RAPPOR
- Privacy beyond Tabular Data
  - No Free Lunch Theorem
  - Customizing differential privacy using Pufferfish

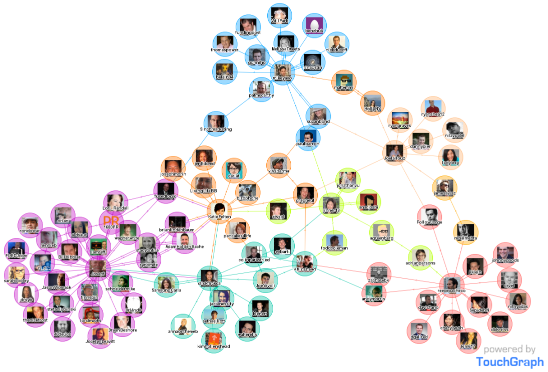
Differential Privacy & Complex Datatypes

- Defining neighboring databases
  - What is a record?
- Records can be correlated
  - Unravels privacy guarantee

Graphs

Neighboring databases … differ in one record.

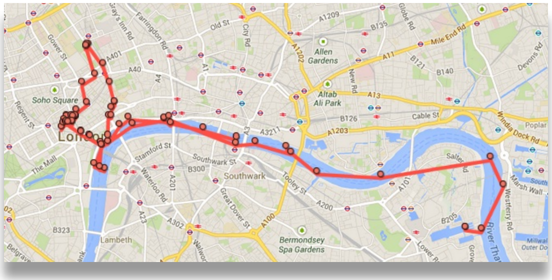
- In graphs, a record can be:
  - An edge (u,v)
  - The adjacency list of node u



Trajectories

Neighboring databases … differ in one record.

- In location trajectories, a record can be:
  - Each location in the trajectory
  - A sequence of locations spanning a window of time
  - The entire trajectory



US Census Bureau Data

- Employee
  - Age
  - Sex
  - Race & Ethnicity
  - Education
  - Home location (Census block)

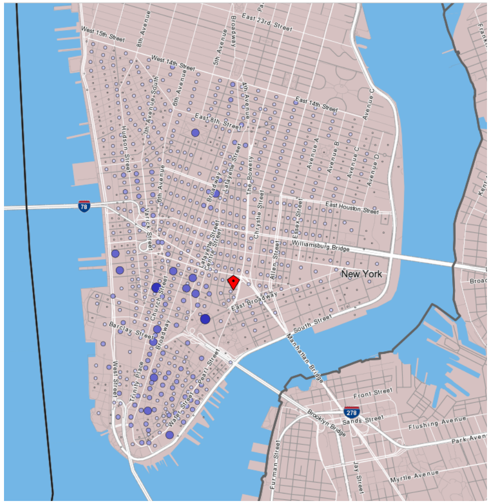
- Job
  - Start date
  - End date
  - Worker & Workplace IDs
  - Earnings

- Employer
  - Geography (Census blocks)
  - Industry
  - Ownership (Public vs Private)

US Census Bureau Data

Neighboring databases … differ in one record.

- A record can be:
  - An employee
  - An employer
  - Something else?
    - Come to talk on Thursday



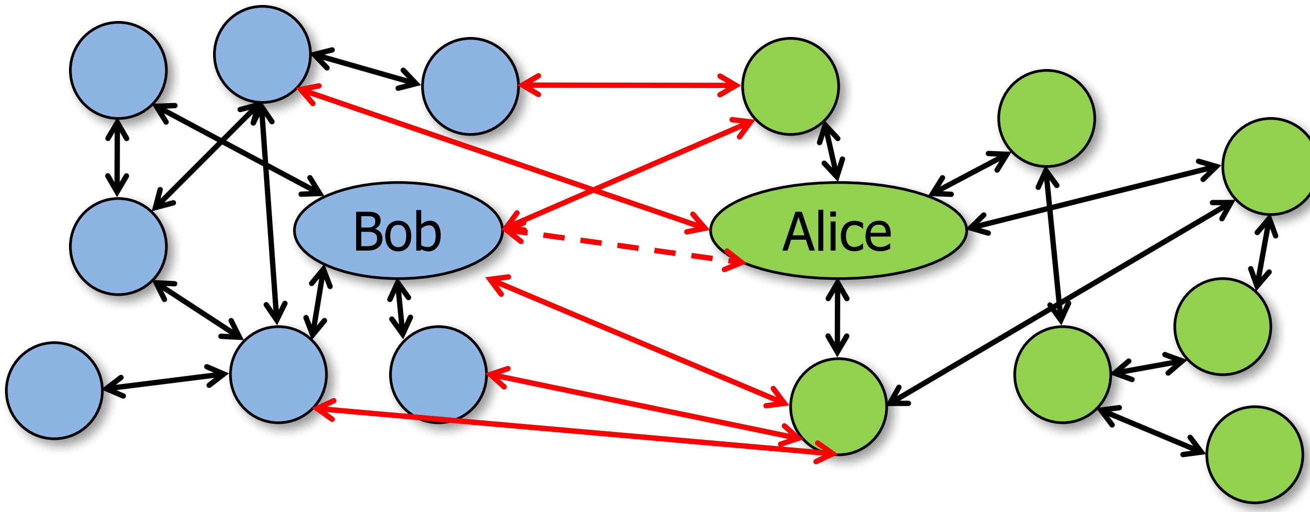
Employment in Lower Manhattan

Differential Privacy & Complex Datatypes

- Defining neighboring databases
  - What is a record?
- Records can be correlated
  - Unravels privacy guarantee

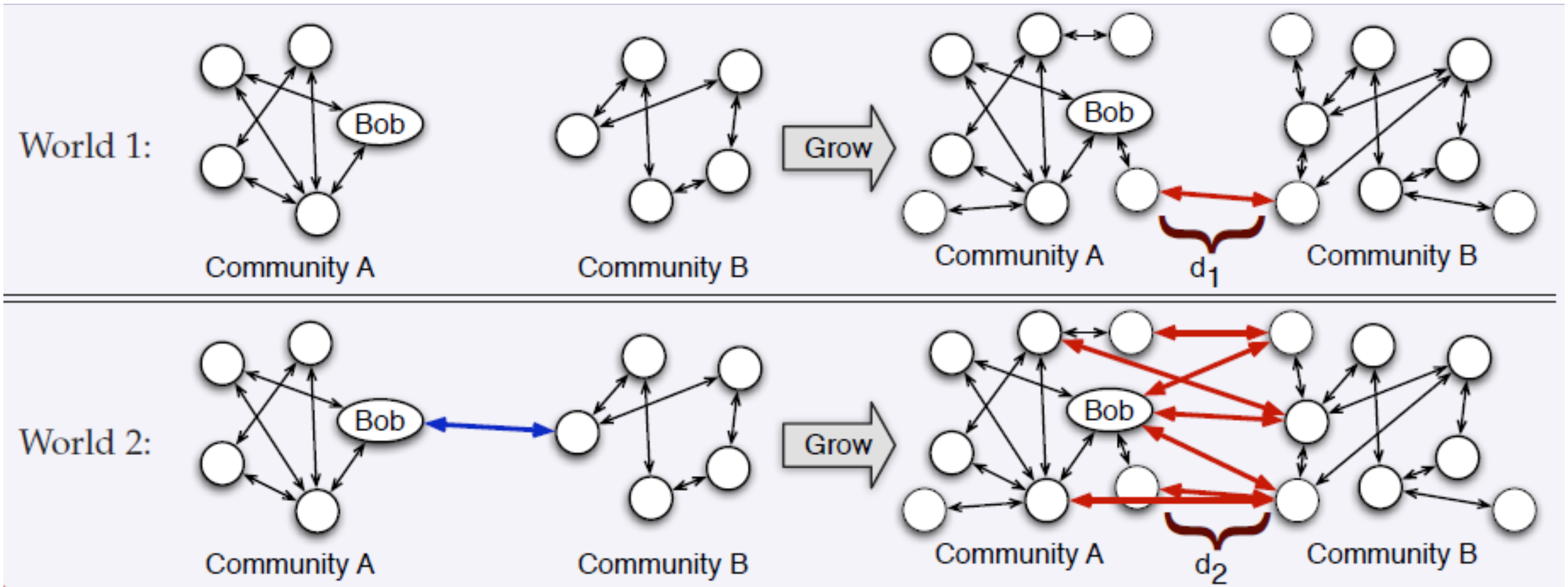
[KM11]

Correlations and DP



- Want to release the number of edges between **blue** and **green** communities.
- Should not disclose the presence/absence of Bob-Alice edge.

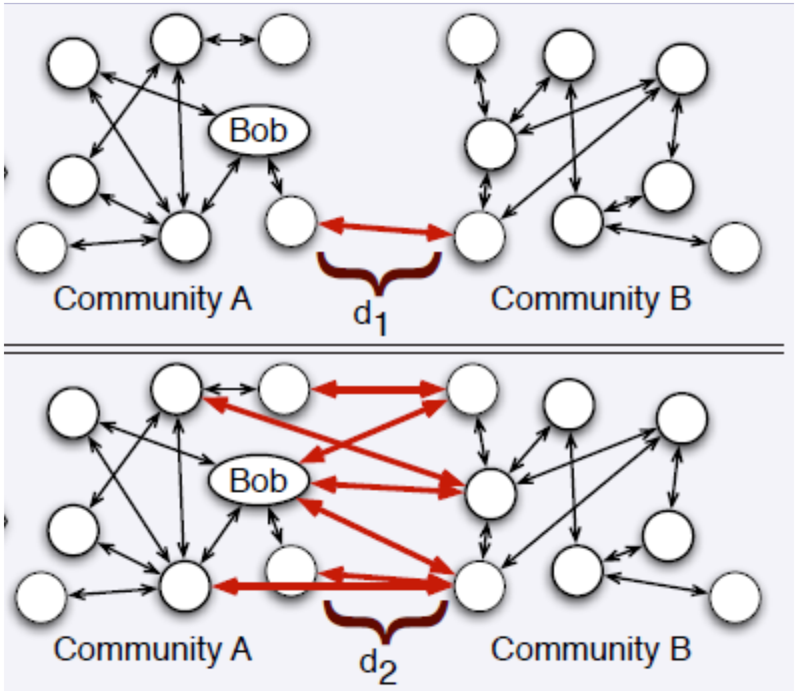
Adversary knows how social networks evolve



Depending on the social network evolution model, *(d_2-d_1)* is *linear* or even *super-linear* in the size of the network.

[KM11]

Differential privacy fails to avoid breach



Output $(d_1 + \delta)$

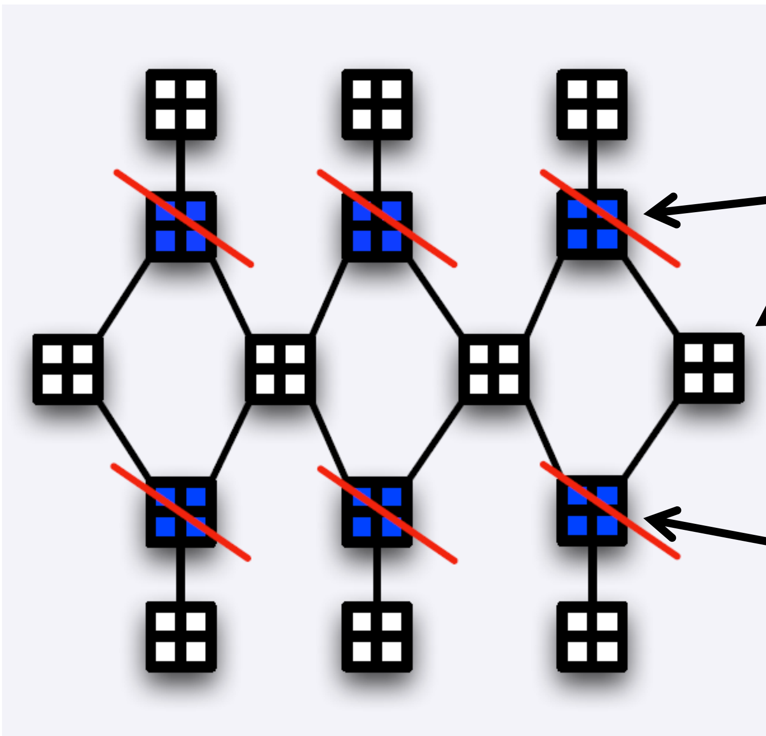
$\delta \sim \text{Laplace}(1/\varepsilon)$

Output $(d_2 + \delta)$

Adversary can distinguish between the two worlds if $d_2 - d_1$ is large.

[KM11]

Reason for Privacy Breach

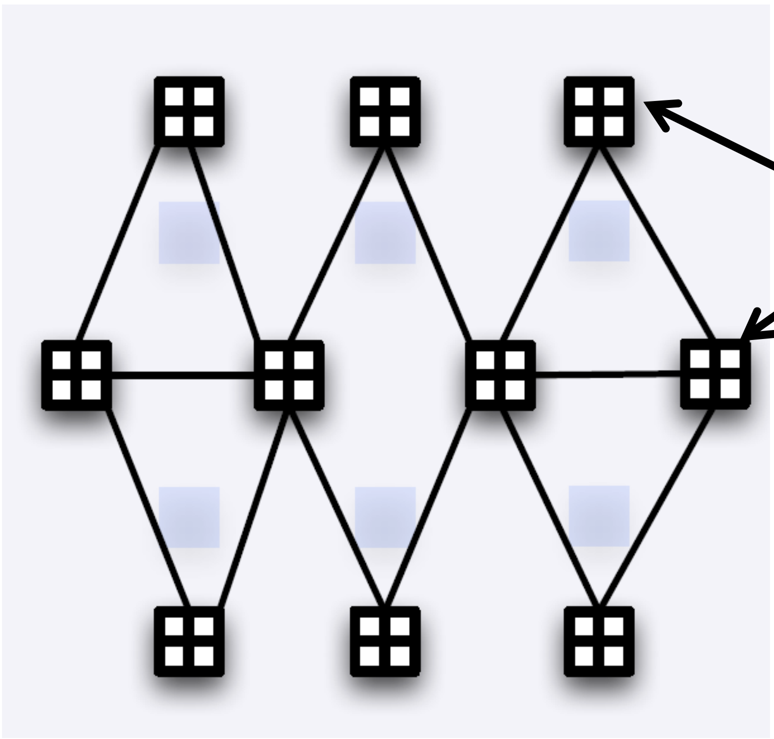


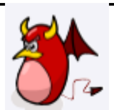
- Pairs of tables that differ in one tuple
-  cannot distinguish them

Tables that do not satisfy background knowledge

Space of all possible tables

Reason for Privacy Breach



 can distinguish between every pair of these tables based on the output

Space of all possible tables

No Free Lunch Theorem

It is not possible to guarantee *any* utility in addition to privacy, *without making assumptions about*

- the data generating distribution

[KM11]

- the background knowledge available to an adversary

[DN 10]

Need a formal theory to understand the privacy ensured by DP

[KM12, KM14]

Pufferfish

- Pufferfish (data):
  - contains tetrodotoxin (sensitive information).
- Toxin is everywhere:
  - Liver
  - Intestines
  - Skin / Muscles
- Removing all toxin = removing fish

- Chef (algorithm):
  - Processes the fish.
- Certification and license (privacy definition):
  - Rules chef must follow / restrictions on algorithm
  - Guarantees output is (relatively) safe.

- Fugu (sanitized data):
  - Tasty (high utility)
  - Minimal toxins
  - Minimal leakage of sensitive information

Pufferfish Semantics

- What is being kept secret?
- Who are the adversaries?
- How is information disclosure bounded?
  - (similar to epsilon in differential privacy)

Sensitive Information

- **Secrets**: S be a set of potentially sensitive statements
  - "individual j's record is in the data, and j has Cancer"
  - "individual j's record is not in the data"

- **Discriminative Pairs**: $S_{pairs} \subseteq S \times S$
  Mutually exclusive pairs of secrets.
  - ("Bob is in the table", "Bob is not in the table")
  - ("Bob has cancer", "Bob has diabetes")

  - Denotes an adversary's possible beliefs about a target individual.

Adversaries

- We assume a Bayesian adversary who is can be completely characterized by his/her prior information about the data
  - We do not assume computational limits

- **Data Evolution Scenarios**: set of all probability distributions that could have generated the data (… think adversary's prior).

  - *No assumptions*: All probability distributions over data instances are possible.

  - *I.I.D.*: Set of all f such that: $P(data = \{r_1, r_2, \ldots, r_k\}) = f(r_1) \times f(r_2) \times \ldots \times f(r_k)$

Information Disclosure

- Mechanism M satisfies ε-Pufferfish(S, Spairs, D), if

$$\forall\, w \in range(M)$$
$$\forall\, (s, s') \in S_{pairs}$$
$$\forall \theta \in D, s.t. \quad P(s|D), P(s'|D) \neq 0$$

$$P(M(\mathfrak{D}) = w | s, \theta) \leq e^{\varepsilon} P(M(\mathfrak{D}) = w | s', \theta)$$

Pufferfish Semantic Guarantee

$$e^{-\varepsilon} \leq \frac{P(s|M(\mathfrak{D}) = w, \theta)}{P(s'|M(\mathfrak{D}) = w, \theta)} \Big/ \frac{P(s|\theta)}{P(s'|\theta)} \leq e^{\varepsilon}$$

Posterior odds of s vs s'

Prior odds of s vs s'

Customizing Privacy

- Setup secrets and discriminative pairs based on the requirements of what must be kept secret
- Set up data generating distributions to capture correlations known to the adversary
- Pufferfish results in privacy definition that bounds the adversary's posterior and prior odds for every discriminative pair.

Advantages

- Privacy defined more generally in terms of customizable secrets rather than records
  - Better capture legal privacy policies

- Can better explore privacy-utility tradeoff by varying secrets and adversaries
  - **See application to US Census Bureau Data (Thursday 2PM DP Session)**

- Gives a deeper understanding of the protections afforded by existing privacy definition

Pufferfish & Differential Privacy

- Discriminative Pairs:
  - s_x^i: record i takes the value x
  - $s_\perp^i$: record i is not in the database
  - $S_{pairs} = \{(s_x^i, s_\perp^i) | \forall x \in dom, \forall\ record\ i\}$

- Attackers should not be able to tell whether a record is in or out of the database

Pufferfish & Differential Privacy

- Data evolution:
  - For all $\theta = [f_1, f_2, f_3, \ldots, f_k]$

$$P[Data = D|\Theta] = \prod_{r_i \in D} f_i(r_i)$$

- Adversary's prior may be any distribution that makes records **independent**

Pufferfish & Differential Privacy

- Discriminative Pairs:
  - $S_{pairs} = \{(s_x^i, s_\perp^i) | \forall x \in dom, \forall \text{ record } i\}$

- Data evolution:
  - For all θ = [$f_1, f_2, f_3, \ldots, f_k$]

$$P[Data = D|\Theta] = \prod_{r_i \in D} f_i(r_i)$$

A mechanism M satisfies differential privacy
if and only if
it satisfies Pufferfish instantiated using Spairs and {θ}

Challenges with Pufferfish

- Setting up data generating distributions are tricky
  - Adversary's knowledge is unknown
- Little work on algorithm design for Pufferfish
  - Notable Exceptions: Blowfish (next module), and Wasserstein mechanism (Thursday 2 PM DP Session)
- Not all Pufferfish definitions are "good"
  - Many do not satisfy composition

Summary

- Complex datatypes require custom privacy definitions
  - No Free Lunch theorem
  - Varied notions of neighboring databases
  - Correlations can unravel privacy ensured by DP algorithms

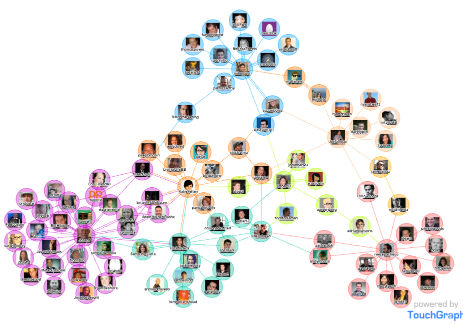
- Pufferfish is a mathematical framework for defining privacy
  - A rigorous way to customize privacy to applications
  - Helps understand semantics of privacy definitions

MODULE 6: APPLICATIONS II: NETWORK & TRAJECTORIES

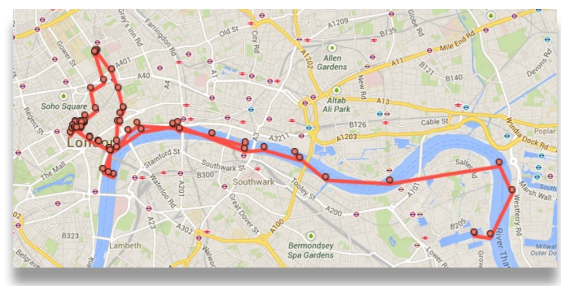
Module 6: Applications II

- Pufferfish Privacy for Non-tabular Data

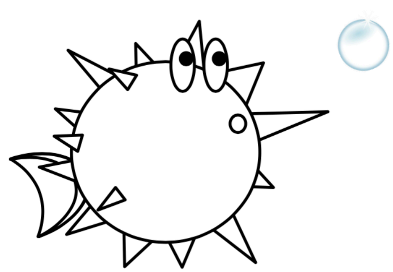
Social network



Location trajectories



- Blowfish Privacy

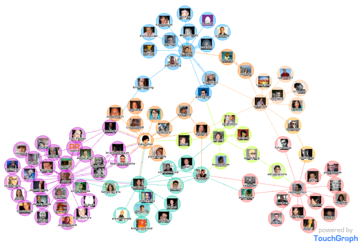


Pufferfish Semantics

- What is being kept secret?
- Who are the adversaries?
- How is information disclosure bounded?
  - (similar to epsilon in differential privacy)

Examples: Graphs

- Neighboring graphs differ in presence/absence of one edge

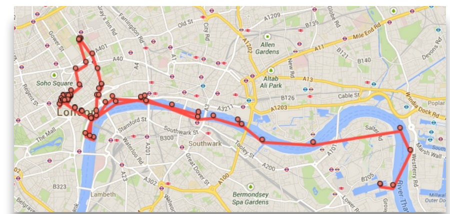


- Pufferfish meaning:
  - Data: matrix of bits
  - Secrets: whether or not an edge (u, v) is in the graph -- bit at (u, v) is 0 or 1
  - Data generating distributions: All graphs where **each edge $\boldsymbol{e}$ is independently present** with probability p_e.

- But …
  - Edges are not independent in real graphs

Examples: Location Trajectories

- Neighboring tables differ in one location (at one point of time) of an individual

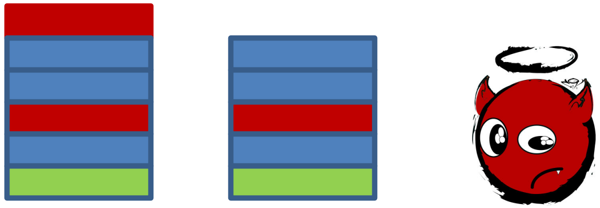


- Pufferfish meaning
  - Data: a matrix of locations
  - Secrets: Whether or not individual was at some location at some point of time
  - Data Generating Distributions: All trajectories where an individual's **location at some time is independent of all other locations** …

- But …
  - Current location depends on previous locations…

Common Themes

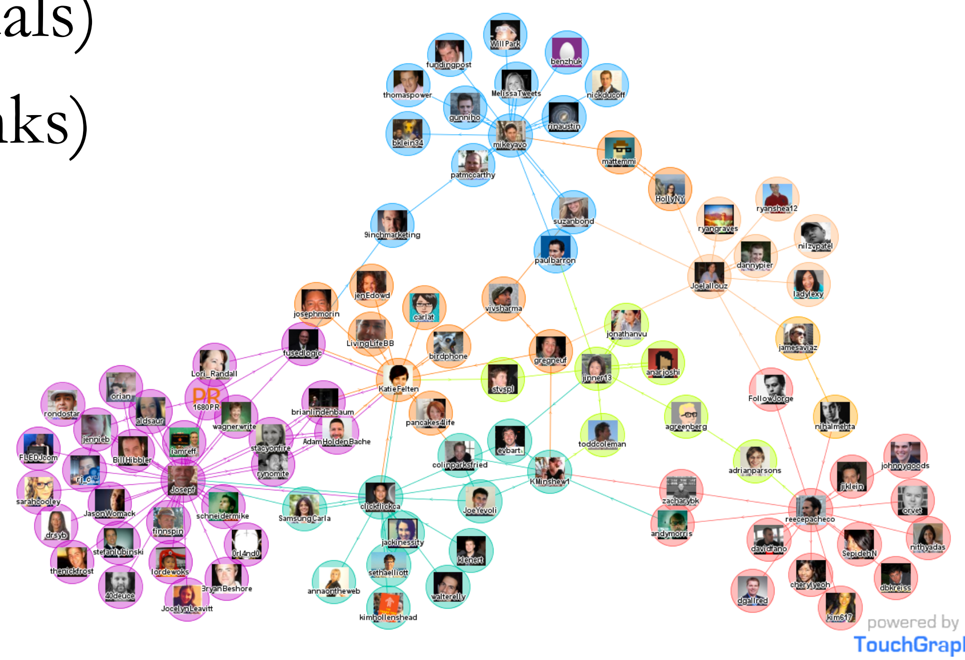
- What are secrets and neighboring datasets for different applications?



- Correlations between protected objects requires further redefinition of privacy
- New privacy definitions requires new algorithm design
- Many open questions

Social Network

- Represented using a graph $G(V, E)$
  - V: node set (individuals)
  - E: edge set (social links)



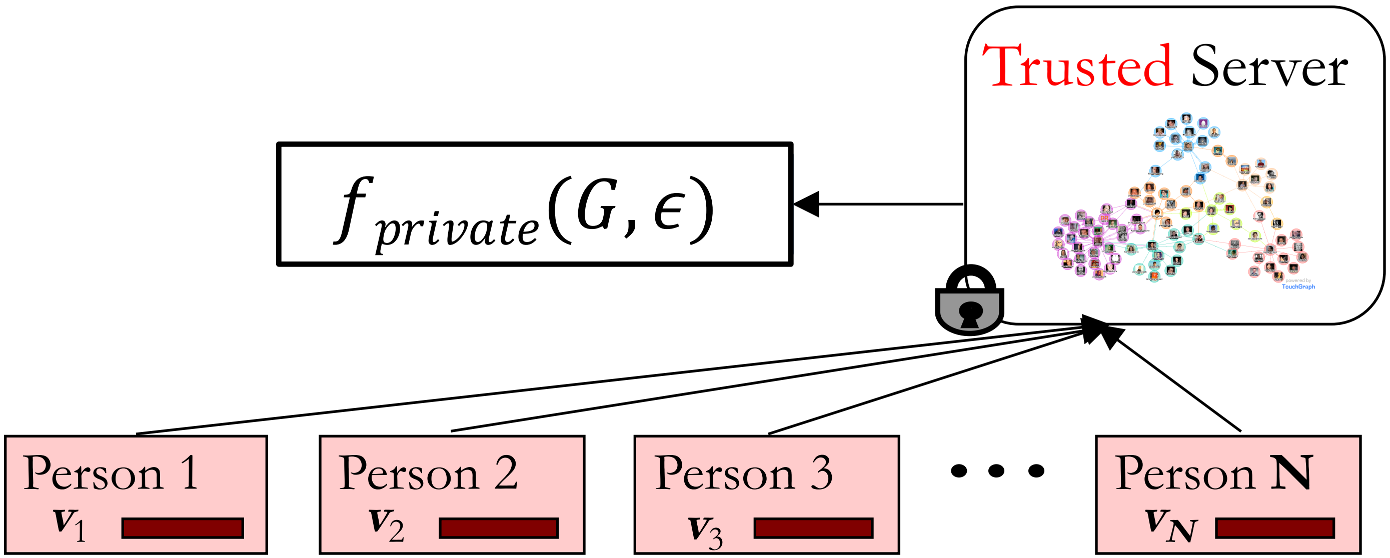
Social Network

- Attacks on graph anonymization

“it is possible for an adversary to learn whether **edges exist** or not between specific targeted pairs of nodes.” [BDK07]

“a third of the **users** on both **Twitter** and **Flickr**, can be **re-identified** in the anonymous Twitter graph with only a 12% error rate.” [NS09]

Private Analysis of Social Network



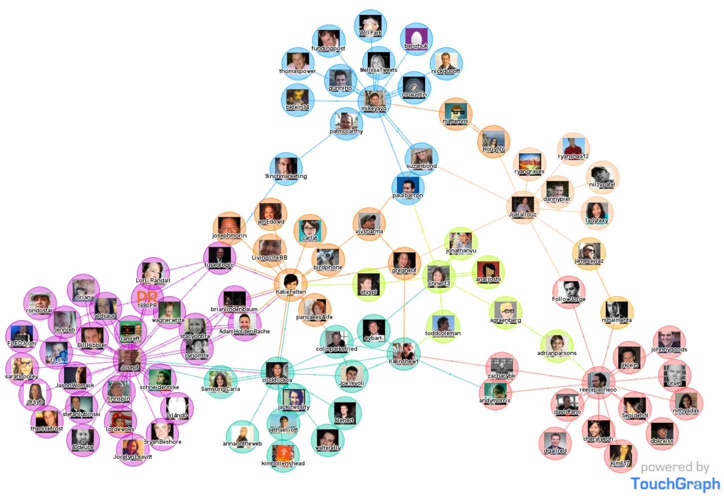
Differential Privacy: $f_{private}$ is ϵ-differentially private if for all **neighbors** G, G' and output S:

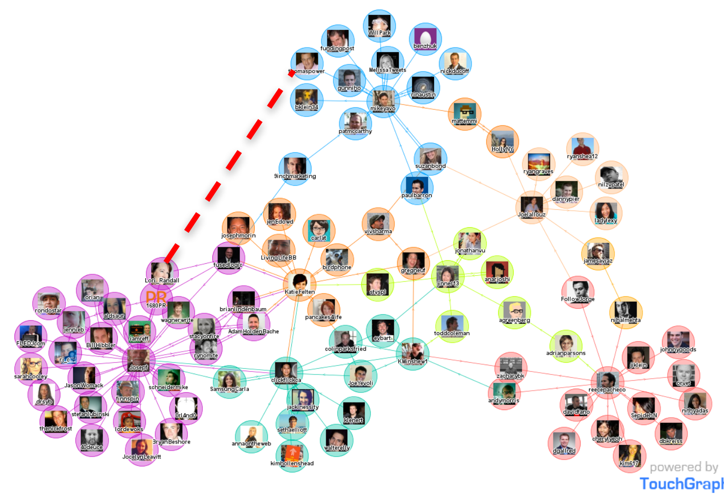
$$Pr[f_{private}(G, \epsilon) \in S] \leq e^{\epsilon} \Pr[f_{private}(G', \epsilon) \in S]$$

Variants of DP for Social Network

- Edge Differential Privacy

Secret: social links between individuals



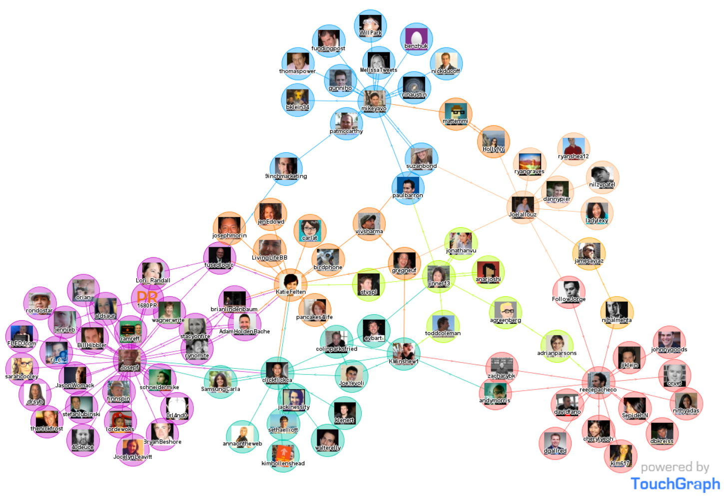


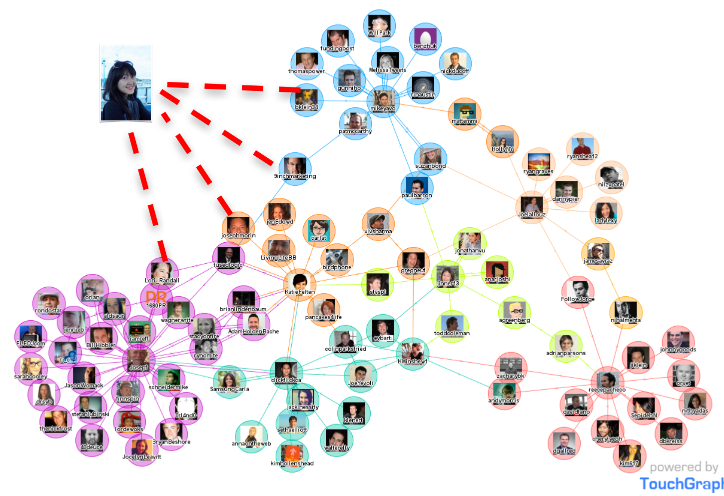
Two graphs are **neighbors** if they differ in the presence of one edge

Variants of DP for Social Network

- Node Differential Privacy

Secret: presence of an individual





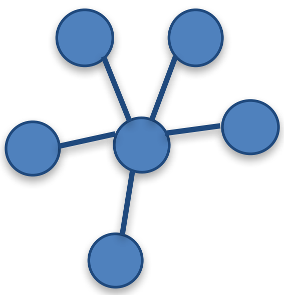
Two graphs are **neighbors** if one can be obtained by another by adding or removing a node and all its edges

Examples for Social Network Statistics

- Degree distribution $D(G)$
- Number of edges
- Counts of small subgraphs
  e.g triangles, k-triangles, k-stars, etc.
- Cut
- Distance to nearest graph with a certain property
- Joint degree distribution

Examples for Social Network Statistics

- Degree distribution $D(G)$

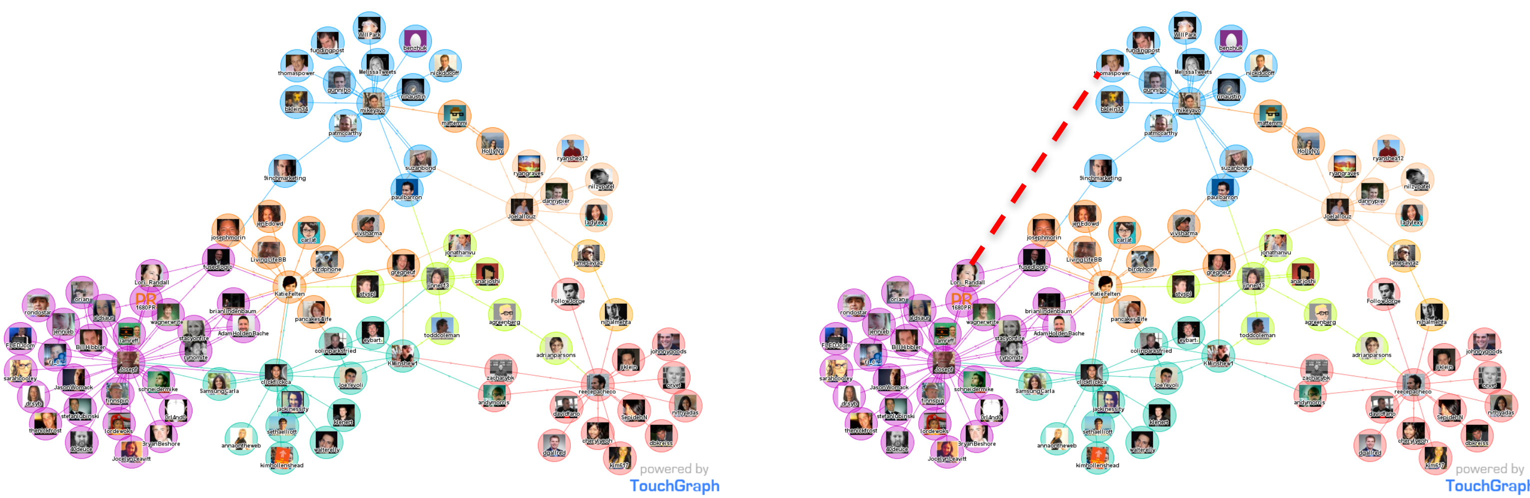


Degree	0	1	2	3	4	5
Frequency	0	5	0	0	0	1

$$D(G) = [0,5,0,0,0,1]$$

Global Sensitivity of Degree Distribution

- What is the global sensitivity of the degree distribution of $G(V, E)$, where $|V| = n$ under Edge differential privacy?



Global Sensitivity of Degree Distribution

- What is the global sensitivity of the degree distribution of $G(V,E)$, where $|V| = n$ under Edge differential privacy?

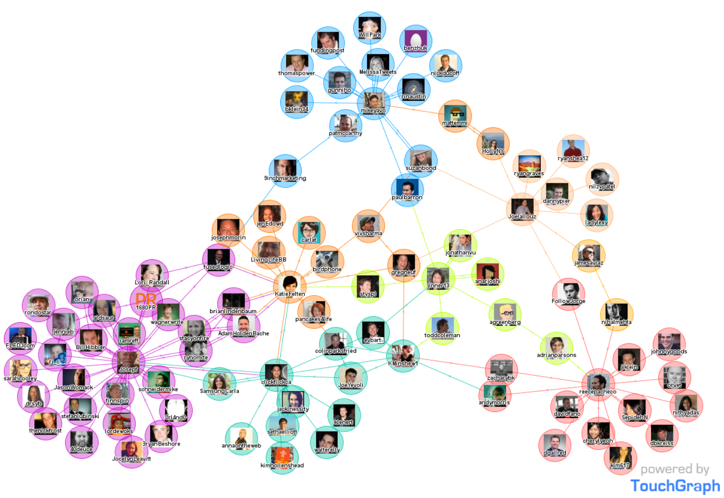
Answer: 4

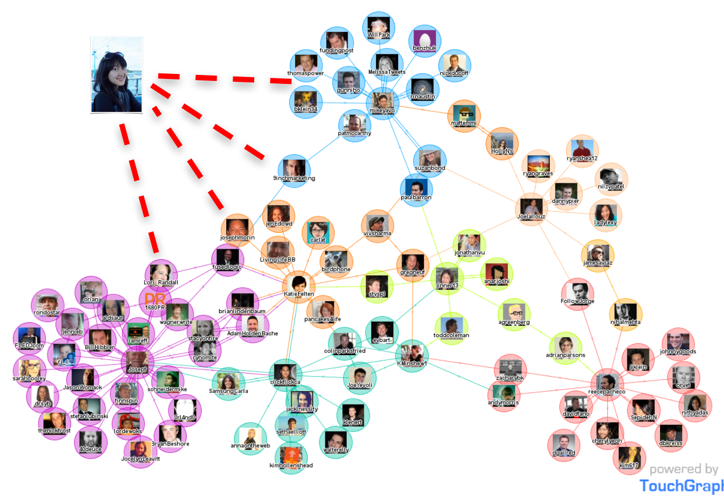
Remove edge (i,j), the changes in degree frequency

Degree	…	d_i-1	d_i	…	d_j-1	d_j	…
Frequency	…	+1	-1	…	+1	-1	…

Global Sensitivity of Degree Distribution (Exercise)

- What is the global sensitivity of the degree distribution of $G(V, E)$, where $|V| = n$ under Node differential privacy?



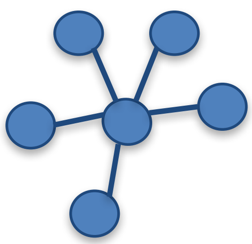


Global Sensitivity of Degree Distribution (Exercise)

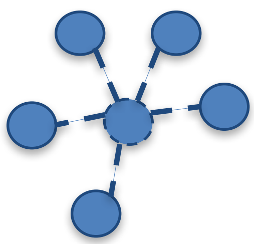
- What is the global sensitivity of the degree distribution of $G(V,E)$, where $|V| = n$ under Node differential privacy?

Answer: 2n-1

Highly Sensitive!!→ Too much noise



$D(G) = [0,5,0,0,0,1]$



$D(G') = [5,0,0,0,0,0]$

Approach to Highly Sensitive Queries

Key idea:

- Projection G on $\boldsymbol{\theta}$**-degree-bounded graphs** G_θ
- Answer queries on G_θ instead of G

$$\widetilde{D(G)} = D(G_\theta) + \color{red}{noise}$$

- Existing approaches for degree distribution
  - Node-based truncation [KNRS13]
  - Lipschitz extensions [RS15]
  - Edge-based projection [DLL16]

How much noise?

- Answer queries on G_θ instead of G

$$\widetilde{D(G)} = D(G_\theta) + \color{red}{noise}$$

- Sensitivity
  - Node-based truncation: $2\theta \cdot \delta$
    - Smooth sensitivity approach [NRS07]
  - Lipschitz extensions: 6θ
  - Edge-based projection: $2\theta + 1$

Applicable to count
- edges
- small subgraph

[KNRS13]

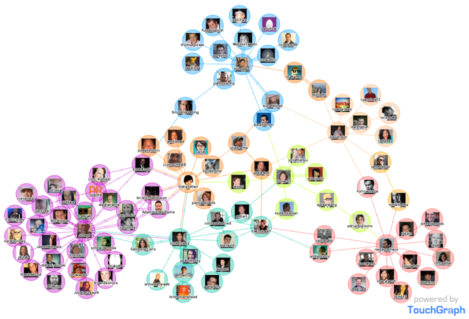
Work on Edge DP

- Degree distribution
  - Global sensitivity + Post-processing [HLM09, HRMS10, KS12, LK13]
- Small subgraph counting
  - Smooth sensitivity [NR07]
  - Ladder function [ZCPSX15]
  - Noisy sensitivity [DL09]
- Cut
  - Random projections, global sensitivity [BBDS1212]
  - Iterative updates [HR10, GRU12]
- Releasing differentially private graph
  - Exponential random graphs [LM14, KSK15]

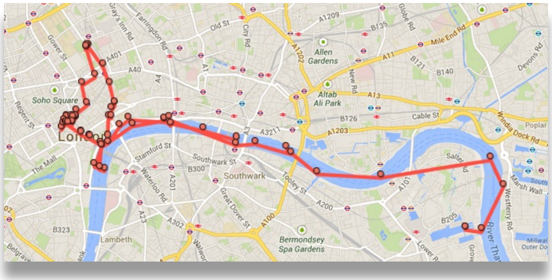
Outline of Module 6

- Pufferfish Privacy for Non-tabular Data

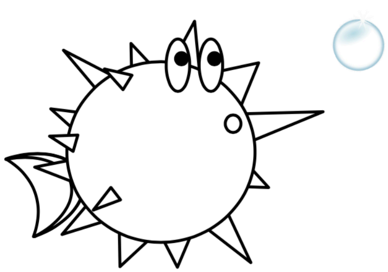
Social network



Location trajectory



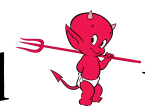
- Blowfish Privacy



Location Trajectory

High uniqueness & High predictability

[MHVB13] [SQBB10]

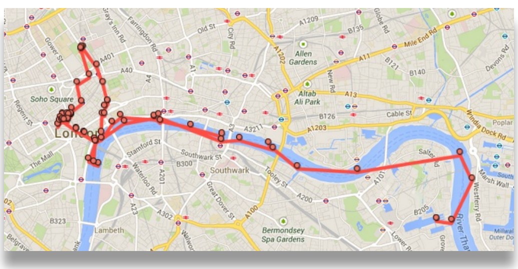
‘show me how you move and  will tell you who you are’

[GKC10]

‘geosocial service “check in” dropped from 18% to 12%’

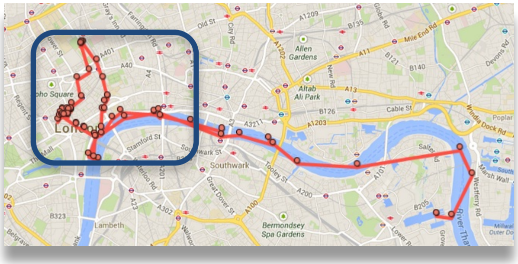
in the Pew Research Center’s Internet Project, 2013

Rich Domain for Secrets



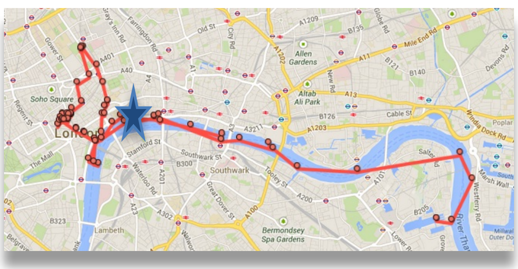
What to hide?
All properties of the individual are secret e.g. Where is home location?
Properties within a small window e.g. Did user visit home in the last hour?
Properties at a specific time e.g. Was user at work or at home at time t?
Some properties (not all) at a specific time e.g. Did user visit near home at time t ?

Rich Domain for Secrets



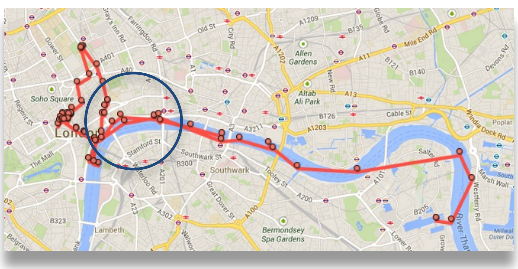
What to hide?
All properties of the individual are secret e.g. Where is home location?
Properties within a small window e.g. Did user visit home in the last hour?
Properties at a specific time e.g. Was user at work or at home at time t?
Some properties (not all) at a specific time e.g. Did user visit near home at time t ?

Rich Domain for Secrets



What to hide?
All properties of the individual are secret e.g. Where is home location?
Properties within a small window e.g. Did user visit home in the last hour?
Properties at a specific time e.g. Was user at work or at home at time t?
Some properties (not all) at a specific time e.g. Did user visit near home at time t ?

Rich Domain for Secrets



What to hide?
All properties of the individual are secret e.g. Where is home location?
Properties within a small window e.g. Did user visit home in the last hour?
Properties at a specific time e.g. Was user at work or at home at time t?
Some properties (not all) at a specific time e.g. Did user visit near home at time t ?

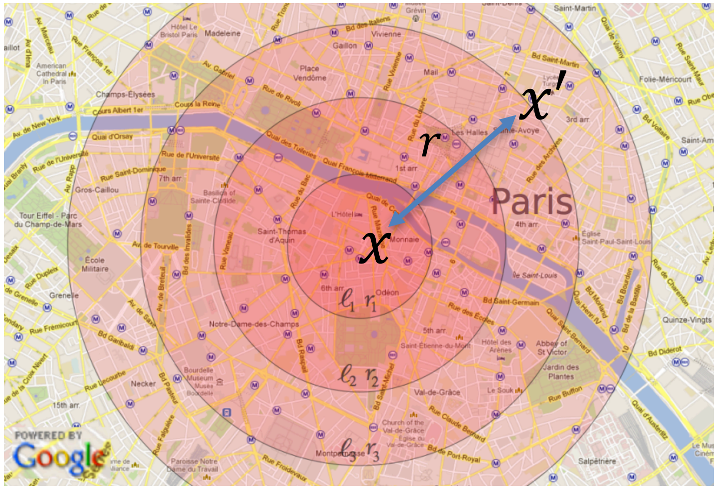
Overview of Privacy Definitions

Neighbors differ in	What to hide?
Trajectory	**All properties of the individual are secret** e.g. Where is home location?
Window	**Properties within a small window** e.g. Did user visit home in the last hour?
Event	**Properties at a specific time** e.g. Was user at work or at home at time t?
Geo-indistinguishability	**Some properties (not all) at a specific time** e.g. Did user visit near home at time t ?

Protect a Single Location

- Protect a single location

  e.g. Location-based Services (LBS) to find a restaurant
  - Not reveal the exact location
  - Revealing an approximate location is ok



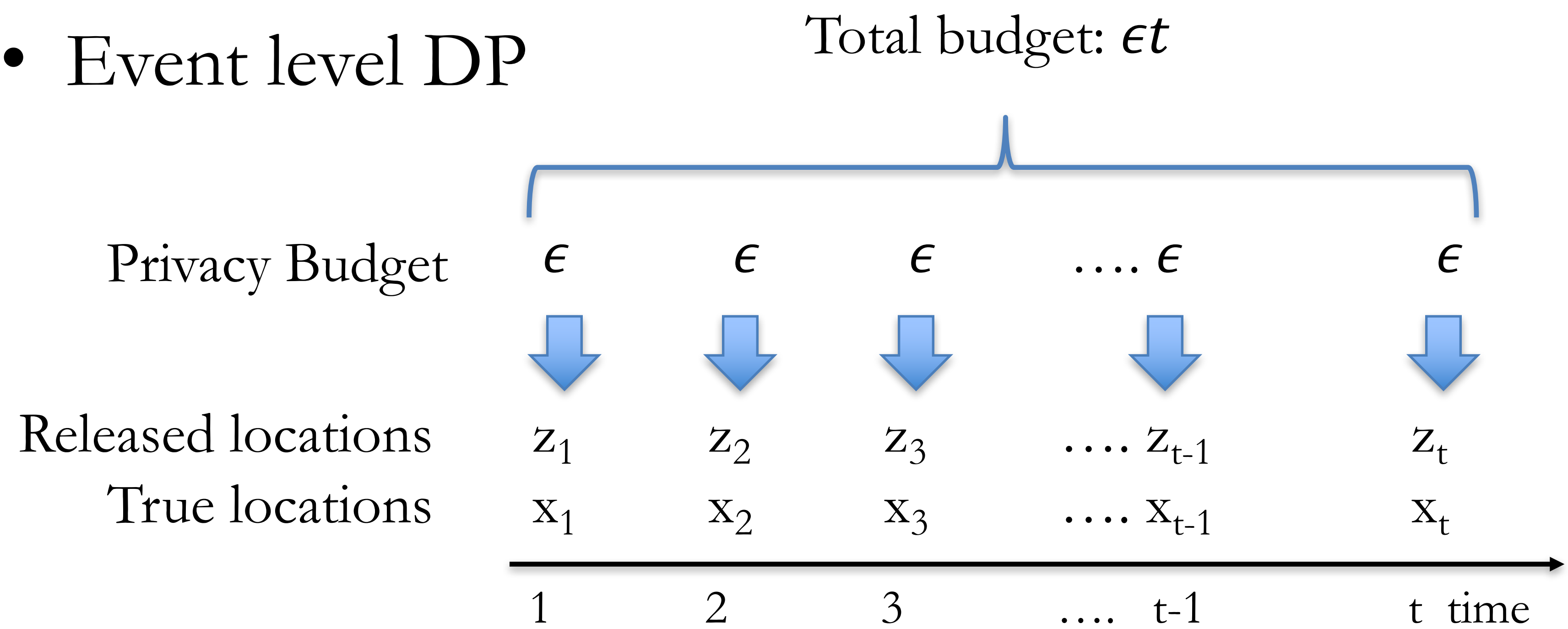
- A mechanism satisfies **ϵ-geo-indistinguishability** iff for all observations $S \subseteq Z$, for all $r > 0$, for all **neighbors** x, x' : $d(x, x') \leq r$,

[ABCP13]

$$\Pr[S|x] \leq e^{\epsilon r} Pr\ [S|x']$$

Different Levels of Protection

- Event level DP



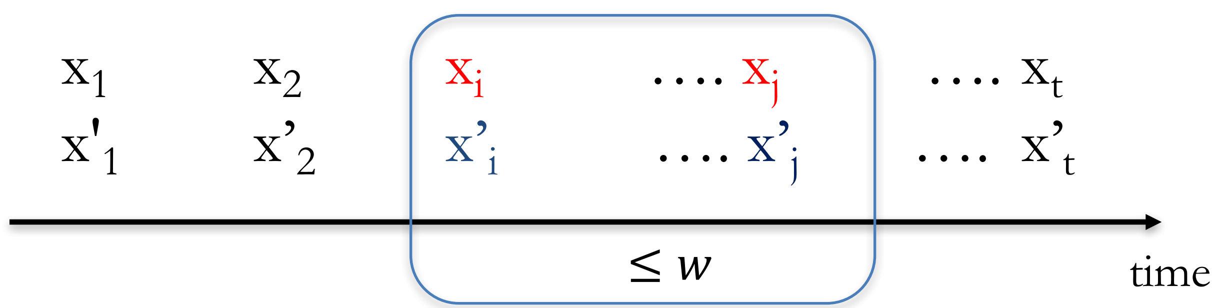
If a person staying at a location for a long time $x_1 = x_2 = \cdots = x_w$, averaging $(z_1, .., z_w)$ leaks the true location.

Different Levels of Protection

[KPXP14]

- w-event DP
  - **Neighboring stream prefix** $(x_1, x_2, .., x_t), (x'_1, x'_2 .., x'_t)$
    - For any $i < j$, if $x_i \neq x'_i$, and $x_j \neq x'_j$ then $j - i + 1 \leq w$
    - x_i and x'_i are the same or neighboring

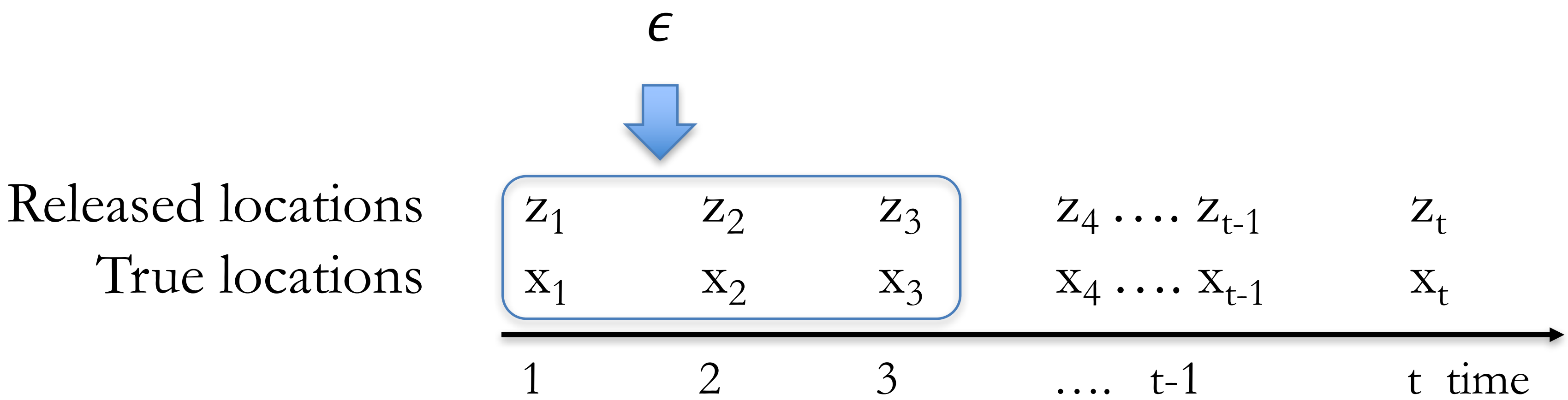
→Protect updates happening within w-event with privacy budget ϵ



Different Levels of Protection

[KPXP14]

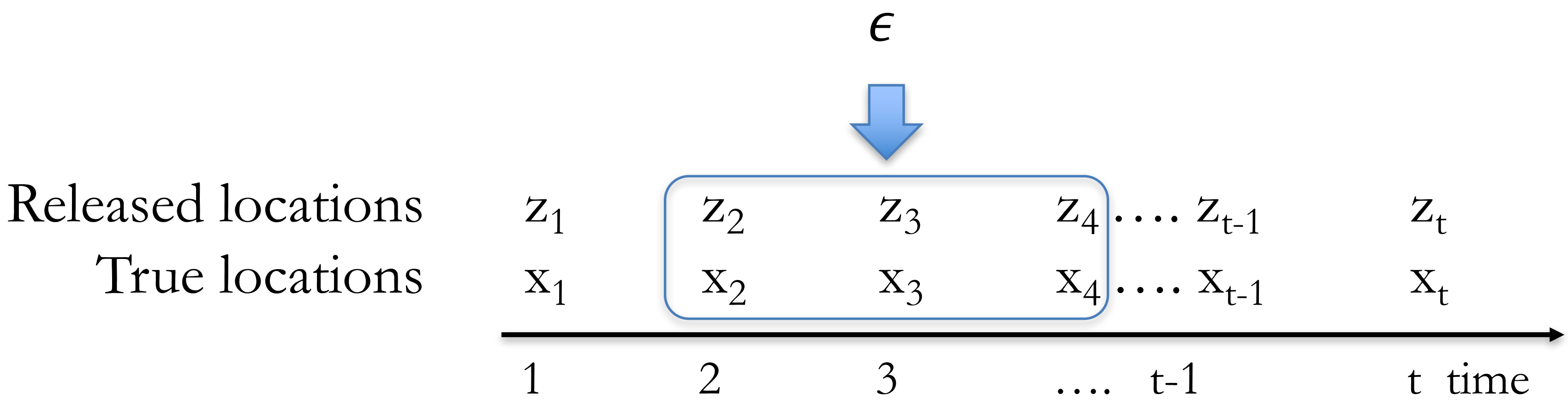
- w-event DP
  - E.g. w=3



Different Levels of Protection

[KPXP14]

- *w*-event DP
  - E.g. *w*=3



Different Levels of Protection

[KPXP14]

- w-event DP
  - Allow budget allocation strategy:
    - Adaptive assign privacy budgets to events within the same w-window
    - E.g. w=3

$$\overbrace{\frac{\epsilon}{3} \qquad \frac{\epsilon}{6} \qquad \frac{\epsilon}{2}}^{\epsilon} \qquad \frac{\epsilon}{6}$$

Released locations	z_1	z_2	z_3	$z_4 \dots z_{t-1}$	z_t
True locations	x_1	x_2	x_3	$x_4 \dots x_{t-1}$	x_t
	1	2	3	…. t-1	t time

Different Levels of Protection

[KPXP14]

- w-event DP
  - Allow budget allocation strategy:
    - Adaptive assign privacy budgets to events within the same w-window
    - E.g. w=3

$$\frac{5}{6}\epsilon \leq \epsilon$$

	$\frac{\epsilon}{3}$	$\frac{\epsilon}{6}$	$\frac{\epsilon}{2}$	$\frac{\epsilon}{6}$	
Released locations	z_1	z_2	z_3	z_4 …. z_{t-1}	z_t
True locations	x_1	x_2	x_3	x_4 …. x_{t-1}	x_t
	1	2	3	…. t-1	t time

Different Levels of Protection

- Trajectory-level DP for entire trajectory
  - Neighboring databases D_1, D_2
    - Differ in one entire trajectory
  - Release aggregate statistics for multiple trajectories

[CAC12, HCMPS15]

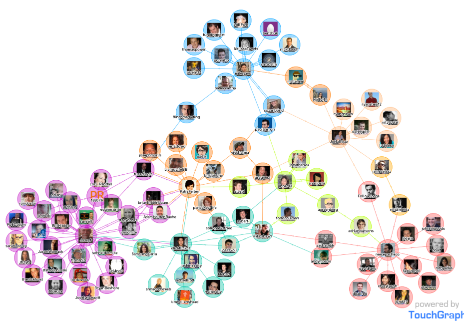
Privacy Budget ϵ

Released locations	z_1	z_2	z_3	…. z_{t-1}	z_t
True locations	x_1	x_2	x_3	…. x_{t-1}	x_t
	1	2	3	…. t-1	t time

Different Level of Privacy Protection

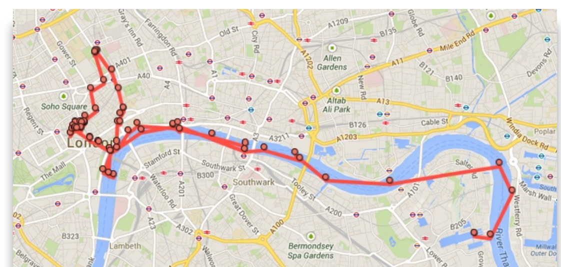
- Pufferfish Privacy for Non-tabular Data

Social network



- Edge DP
- Node DP

Location trajectory

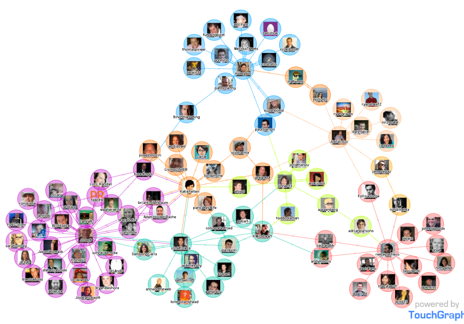


- ϵ-indistinguishability
- Event DP
- w-event DP
- Trajectory level DP

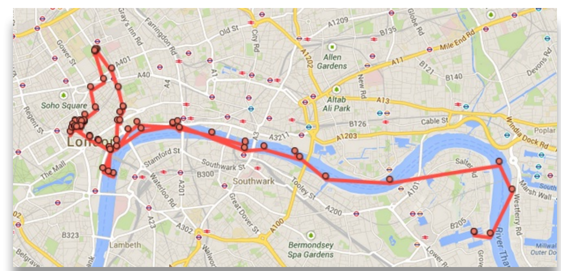
Outline of Module 6

- Pufferfish Privacy for Non-tabular Data

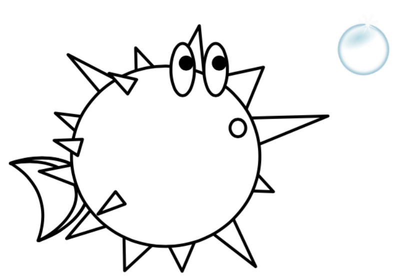
Social network



Location trajectory



- Blowfish Privacy



Blowfish Privacy [HMD14]

- Special case of Pufferfish that satisfies sequential composition

- A framework for redefining neighboring databases for complex datatypes using a *policy graph*
  - Captures many neighboring definitions
  - Handles correlations induced by constraints on database
    - Prior data releases
    - Location constraints

Blowfish

- **Differential Privacy:**
  For all outputs o, for all $|D_1 - D_2| = 1$,
  $$\Pr[A(D_1) = o] \leq e^{\epsilon}\Pr[A(D_2) = o]$$

Redefined neighbor relation

- **Blowfish Privacy**:
  For all outputs o, for all $D_1, D_2 \in \boldsymbol{N_G}$
  $$\Pr[A(D_1) = o] \leq e^{\epsilon}\Pr[A(D_2) = o]$$

Algorithm Design Simplified

[HMD16]

- Transformational equivalence between Blowfish and differential privacy
- No need to do algorithm design from scratch for each definition
- Answering queries under a Blowfish privacy policy is equivalent in error to answering transformed queries under differential privacy

Intuition

For all outputs o, for all $|D_1 - D_2| = 1$,

$$\Pr[A(D_1) = o] \leq e^{\epsilon}\Pr[A(D_2) = o]$$

Is equivalent to

For all outputs o, for all D_1, D_2

$$\Pr[A(D_1) = o] \leq e^{\epsilon\Delta(D_1,D_2)}\Pr[A(D_2) = o]$$

where $\Delta(D_1, D_2)$ is the size of symmetric difference

Definitions differ in distance metrics

- **Differential Privacy:**
  For all outputs o, for all D_1, D_2
  $\Pr[A(D_1) = o] \leq e^{\epsilon\Delta(D_1,D_2)} \Pr[A(D_2) = o]$

- **Blowfish Privacy**:
  For all outputs o, for all D_1, D_2
  $\Pr[A(D_1) = o] \leq e^{\epsilon \mathrm{d}(D_1,D_2)} \Pr[A(D_2) = o]$

Distance metric imposed by neighbor relation

Transformational equivalence …

… achieved by embedding distance imposed by neighbor definition in Blowfish to distance metric imposed by neighbors that differ in one record.

Extending Differential Privacy via Metrics

- [CEBP13] propose generalizations of differential privacy using metrics
  - Special case of Pufferfish and generalizes Blowfish

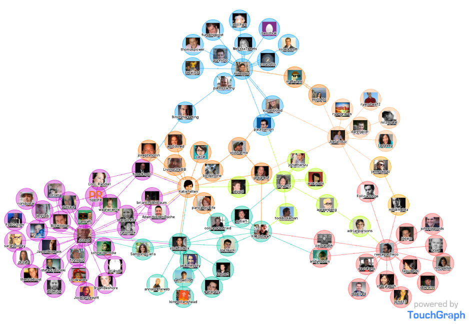
- [WSC17] use a similar intuition to derive a generalized sensitivity notion for using Laplace mechanism for Pufferfish
  - Based on Wasserstein distances
  - Computing this generalized sensitivity can be intractable
  - Examples of intractability also shown in [KM11, HMD14]

Thur 2pm DP Session

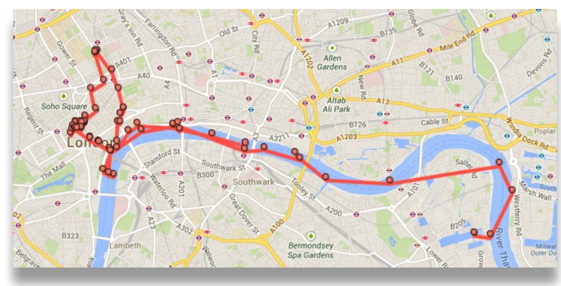
Module 6: Applications II

- Pufferfish Privacy for Non-tabular Data

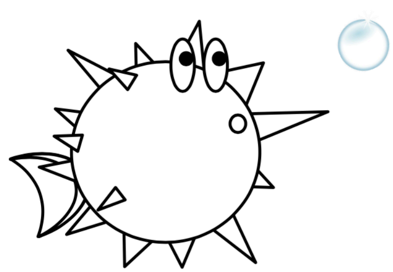
Social network



Location trajectories



- Blowfish Privacy



Open Questions

- Identify realistic policies for real world applications.
  - Is it socially acceptable to offer weaker privacy protection to high-degree nodes?
- Algorithm design under complicated constraints or correlations.
  - Correlations within both trajectories and between users, e.g. family members may share similar trajectories patterns.
  - Highly sensitive queries under constraints or correlations.
- Privacy analysis across different guarantees.

MODULE 7: SUMMARY

Module 7: Summary

- Recap of tutorial
- Five Cross-cutting ideas
- Challenges

Statistical database privacy

- Statistical database privacy is the problem of releasing aggregates while not disclosing individual records
- Privacy desiderata
  - Resilience to background knowledge
  - Composition
  - Avoid privacy by obscurity: public algorithms/implementations
- Utility desiderata
  - Accurate
  - Useful

Tutorial Summary

- Applications
  - Query answering
  - Machine learning
  - Analysis of network data
  - Trajectories
- Real-world deployments:
  - U.S. Census Bureau OnTheMap: commuting patterns
  - Google RAPPOR: browser settings
- Formal privacy definitions
  - Differential privacy, Pufferfish, Blowfish

Cross-cutting ideas

1. Higher accuracy through careful composition
   – Parallel composition, advanced composition

2. Where to inject noise?
   – On input, output, intermediate result
   – Find “information bottleneck” that has tight bound on sensitivity
   – May be dictated by application (e.g., RAPPOR)

Cross-cutting ideas

3. Lossy transformations
   - Histograms: adaptive binning
   - RAPPOR: bloom filters
   - Social networks: degree-bounded graphs
   - … results in **bias-variance tradeoffs**

4. Leverage domain knowledge
   - OnTheMap: previously published data
   - RAPPOR: heavy hitters
   - Network data: tends to be sparse
   - Histograms: smooth, sparse

Cross-cutting ideas

5. Privacy definition may be application specific
   - Differential privacy is a rigorous definition that protects individual tuples…
   - … but this may not align with semantics of application
   - In your application…
     - What are the secrets?
     - Who are the adversaries? What data correlations can they exploit?

Challenge 1: From Prototypes to Deployments

- Community needs more examples of real-world deployments

[Erlingsson et al, CCS 2014]

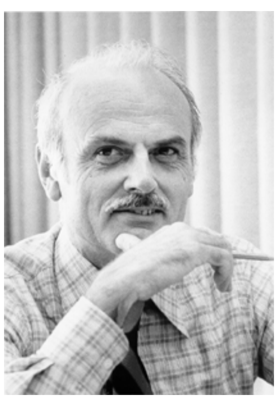
- Demonstrate usefulness in real applications
- These raise important research problems
  - Hardening against side-channel attacks [M12]
  - Matching formal privacy guarantee to needs of application

Challenge 2: From Algorithms to Systems

- Today, getting DP to work in practice requires a team of experts

- … resembles early days of database research…

“… without exception ad hoc, cumbersome, and difficult to use – they could really only be used by people having highly specialized technical skills …”



E. F. Codd on the state of databases in early 1970s

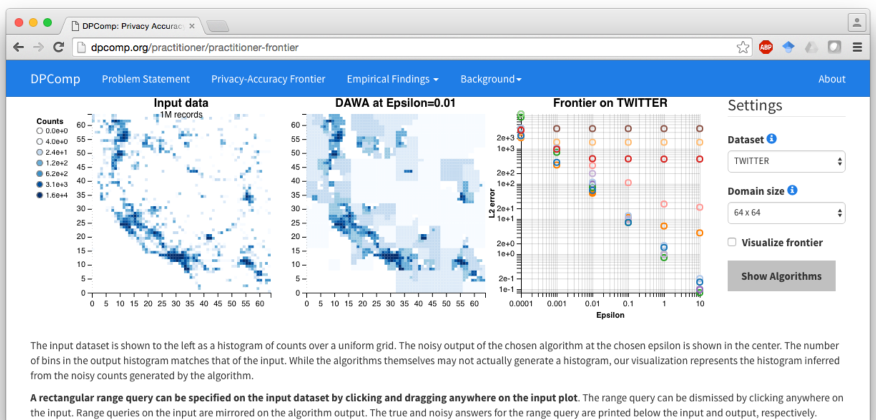
Challenge 2: From Algorithms to Systems

- Today, getting DP to work in practice requires a team of experts
- Example of systems work: Privacy Integrated Queries (PINQ) [M10]
  - Guarantees that programs satisfy privacy…
  - … but program author responsible for accuracy
- Need more research on systems…
  - Modular components
  - Automatic optimization

Challenge 3: Communicating privacy-utility tradeoffs

DPComp

SIGMOD 2016



Colgate University

Duke University

UMASS AMHERST

- Inherent tradeoff between utility and privacy
- Must be communicated to stakeholders
- Need for tunable algorithms

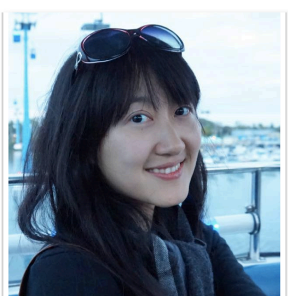
Thank you!

Ashwin Machanavajjhala
Assistant Professor, Duke University

"What does privacy mean … mathematically?"

Michael Hay
Assistant Professor, Colgate University

"Can algorithms be provably private and useful?"



Xi He
Ph.D. Candidate, Duke University

"Can privacy algorithms work in real world systems?"

Module 4 References

- [KLNR11] Kasiviswanathan, Lee, Nissim, Raskhodnikova, "What Can We Learn Privately?", SIAM Journal of Computing, 2011
- [CMS11] Chaudhuri, Monteleoni, Sarwate, "Differentially Private Empirical Risk Minimization", MLRJ 2011
- [CSC13] Song, Chaudhuri, Sarwate, "Stochastic gradient descent with differentially private updates", GlobalSIP 2013
- [BST14] Bassily, Smith, Thakurta, "Private Empirical Risk Minimization, Revisited", FOCS 2014
- [SS15] Shokri, Shmatikov, "Privacy-Preserving Deep Learning". CCS 2015
- [JKT12] Jain, Kothari, Thakurta , "Differentially Private Online Learning", COLT, 2012
- [ACG16] Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, Zhang, "Deep Learning with Differential Privacy", SIGSAC 2016
- [WLK17] Wu, Li, Kumar, Chaudhuri, Jha, Naughton, "Bolt-on Differential Privacy for Scalable Stochastic Gradient Descent-based Analytics", SIGMOD 2017
- [DRV10] Dwork, Rothblum, Vadhan. Boosting and differential privacy, FOCS 2010

Module 5 References

- [DN10] Dwork, Naor "On the Difficulties of Disclosure Protection in Statistical Databases or the Case for Differential Privacy", JPC 2010
- [KM11] Kifer, Machanavajjhala, "No Free Lunch in Data Privacy", SIGMOD 2011
- [KM12] Kifer, Machanavajjhala, "A rigorous and customization framework for privacy", PODS 2012
- [KM14] Kifer, Machanavajjhala, "Pufferfish: A framework for mathematical privacy definitions", TODS 2014
- [HMD14] He, Machanavajjhala, Ding, "Blowfish Privacy" SIGMOD 2014
- [HMD16] Haney, Machanavajjhala, Ding, "Design of Policy Aware Differentially Private Algorithms", VLDB 2016
- [CEBP13] Chatzikokolakis, Andres, Bordenabe, Palamidessi, "Broadening the scope of differential privacy using metrics", PoPETS 2013
- [WSC16] Wang, Song, Chaudhuri, "Privacy Preserving analysis of Correlated Data", Corr abs/1603.03977, 2016

Module 6 References

- [BDK07] Backstrom, Dwork, Kleinberg, "Wherefore Art Thou R3579X? Anonymized Social Networks, Hidden Patterns, and Structural Steganography", WWW 2007
- [NS09] Narayanan, Shmatikov, "De-anonymizing Social Networks." IEEE Security & Privacy 2009
- [KNRS13] Kasiviswanathan, Nissim, Raskhodnikova, Smith, "Analyzing Graphs with Node Differential Privacy", TCC 2013
- [NRS07] Nissim, Raskhodnikova, Smith. "Smooth Sensitivity and Sampling in Private Data Analysis", STOC 2007
- [RS15] Raskhodnikova, Smith, "Efficient Lipschitz Extensions for High-Dimensional Graph Statistics and Node Private Degree Distributions", arXiv:1504.07912v1, 2015
- [DLL16] Day, Li, Lyu. "Publishing Graph Degree Distribution with Node Differential Privacy", SIGMOD 2016
- [NRS07] Nissim, Raskhodnikova, Smith. "Smooth sensitivity and sampling in private data analysis", STOC 2007.
- [KRSY09] Karwa, Raskhodnikova, Smith, Yaroslavtsev, "Private Analysis of Graph Structure", VLDB 2011
- [DL09] Dwork, Lei. "Differential privacy and robust statistics", STOC 2009
- [HLMJ09] Hay, Li, Miklau, Jensen, "Accurate estimation of the degree distribution of private networks", ICDM 2009.
- [HRMS10] Hay, Rastogi, Miklau, Suciu. "Boosting the accuracy of differentially-private queries through consistency", PVLDB 2010.

Module 6 References

- [KS12] Karwa, Slavkovic. "Differentially Private Graphical Degree Sequences and Synthetic Graphs", Privacy in Statistical Databases 2012: 273-285
- [LK13] Lin, Kifer. "Information preservation in statistical privacy and bayesian estimation of unattributed histograms". SIGMOD 2013
- [GRU12] Gupta, Roth, Ullman, "Iterative Constructions and Private Data Release", TCC 2012
- [ZCPSX15] Zhang, Cormode, Procopiuc, Srivastava and Xiao, "Private Release of Graph Statistics using Ladder Functions", SIGMOD 2015
- [KSK15] Karwa, Slavkovi´c, Krivitsky. "Differentially Private Exponential Random Graphs", arXiv:1409.4696v2 2015
- [LM14] Lu. Miklau, "Exponential random graph estimation under differential privacy", KDD 2014
- [KKS15] Karwa, Krivitsky, Slavković, "Sharing Social Network Data: Differentially Private Estimation of Exponential-Family Random Graph Models", arXiv:1511.02930v1. 2015
- [BBDS12] Blocki, Blum, Datta, Sheffet, "The Johnson-Lindenstrauss Transform Itself Preserves Differential Privacy", FOCS 2012
- [HR10] Hardt, Rothblum, " A Multiplicative Weights Mechanism for Privacy-Preserving Data Analysis", FOCS 2010
- [MHVB13] Montjoye, Hidalgo, Verleysen, Blondel, " Unique in the crowd: The privacy bounds of human mobility". Sci. Rep., 3(1376), 2013.
- [SQBB10] Song, Qu, Blumm, Barabsi. " Limits of predictability in human mobility", Science, 2010

Module 6 References

- [GKC10] Gambs, Killijian, Cortez, "Show Me How You Move and I Will Tell You Who You Are", SPRINGL 2010
- [ABCP13] Andrés, Bordenabe, Chatzikokolakis, Palamidessi, "Geo-Indistinguishability: Differential Privacy for Location-Based Systems", CCS 2013
- [XX15] Xiao, Xiong. " Protecting Locations with Differential Privacy under Temporal Correlations". CCS 2015
- [KPXP] Kellaris, Papadopoulos, Xiao, Papadias, "Differentially Private Event Sequences over Infinite Streams", VLDB 2014
- [HCMPS15] He, Cormode, Machanavajjhala, Procopiuc, Srivastava, "DPT: Differentially private trajectory synthesis using hierarchical reference systems", VLDB 2015
- [CAC12] Chen, Acs, Castelluccia, "Differentially private sequential data publication via variable-length n-grams", CCS 2012.
- [HMD14] He, Machanavajjhala, Ding, "Blowfish Privacy" SIGMOD 2014
- [HMD16] Haney, Machanavajjhala, Ding, "Design of Policy Aware Differentially Private Algorithms", VLDB 2016
- [CEBP13] Chatzikokolakis, Andres, Bordenabe, Palamidessi, "Broadening the scope of differential privacy using metrics", PoPETS 2013
- [WSC17] Wang, Song, Chaudhuri, "Privacy Preserving analysis of Correlated Data", SIGMOD, 2017
- [CYXX17]]Yang Cao, Masatoshi Yoshikawa, Yonghui Xiao, Li Xiong, "Quantifying Differential Privacy under Temporal Correlations", ICDE 2017
- [KM11] Kifer, Machanavajjhala, "No Free Lunch in Data Privacy", SIGMOD 2011