Data Science and Privacy Preservation

Graham Cormode

G.Cormode@warwick.ac.uk

Schedule

- Part 1 (today): Centralized privacy models
  - The Privacy Problem
  - Syntactic Approaches to Privacy (1998 onwards)
  - (Centralized) Differential Privacy (2006 onwards)
- Part 2 (tomorrow): Local privacy models (2014 onwards)
  - Local Differential Privacy technical foundations
  - Current directions and open problems

- **Note:** This material can be quite technical and mathematical!
- Slides available from http://cormode.org/ghent

Why Privacy?

- Data subjects have inherent right and expectation of privacy
  - A lot of new data gives detailed descriptions of people's behaviour
- "Privacy" is a complex concept
  - What exactly does "privacy" mean? When does it apply?
  - Could there exist societies without a concept of privacy?
- Concretely: at collection "small print" outlines privacy rules
  - Most companies have adopted a privacy policy
  - E.g. Facebook privacy policy facebook.com/policy.php

- Significant legal framework relating to privacy
  - UN Declaration of Human Rights
  - EU General Data Protection Regulation (GDPR)
  - US: HIPAA, Video Privacy Protection, Data Protection Acts

The Privacy Problem

Regulators ratchet up pressure on Facebook over user data leak

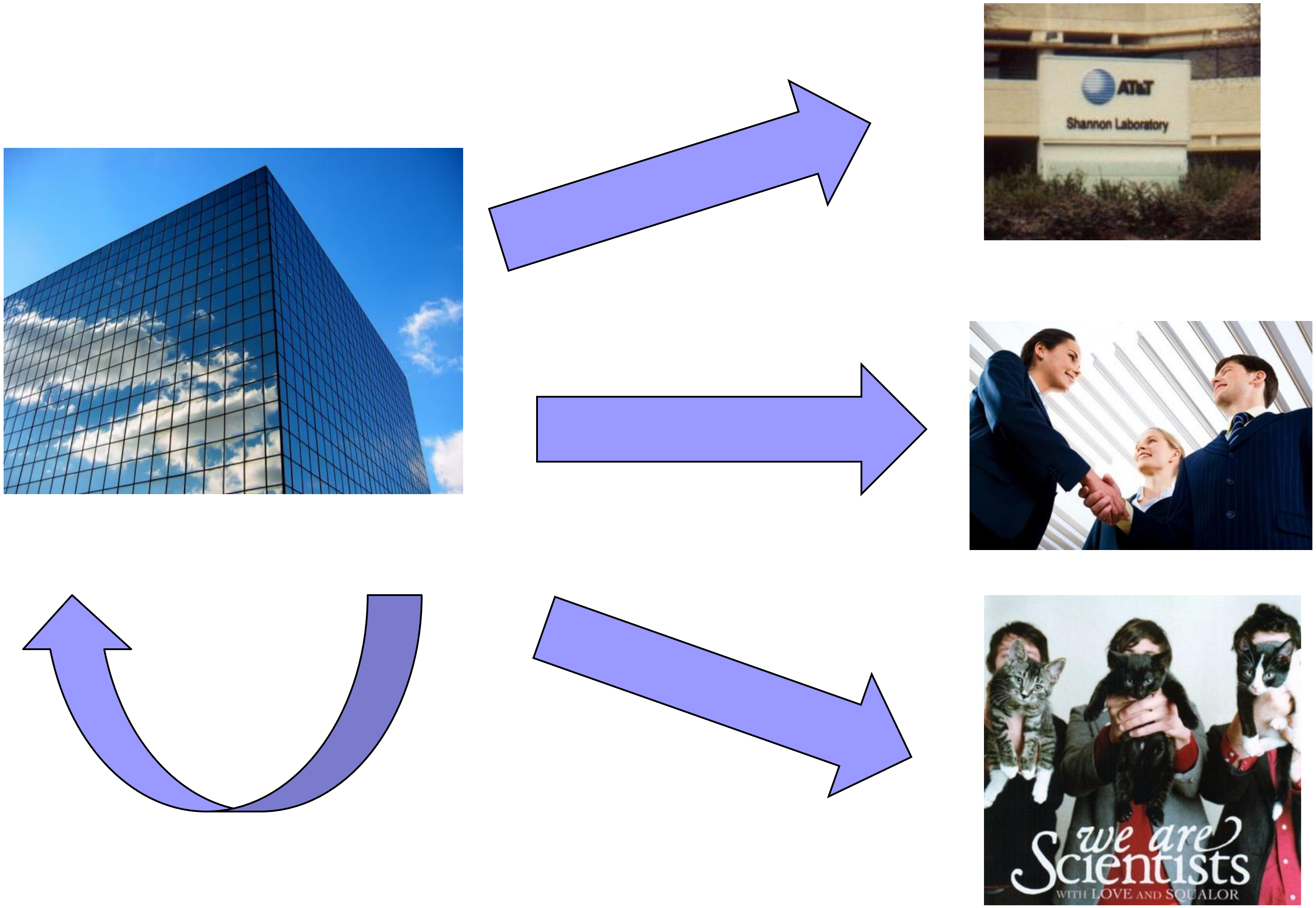
• US and UK agencies step in • Cambridge Analytica suspends Nix • Shares fall further

BBC NEWS

ht-year NHS pay cap poised to be scrapped as ministers fund 6.5% rise

- ♦ Goals for privacy in companies and cities:
  - Enable appropriate use of data while protecting customers
  - Keep CTO/minister off front page of the newspapers!
- ♦ Security is binary*: allow access to data iff you have the key
  - Encryption is robust, reliable and widely deployed
- ♦ Privacy comes in many shades:
  reveal some information, disallow unintended uses
  - Hard to control what may be inferred
  - Possible to combine with other data sources to breach privacy
  - Privacy technology is still maturing

The data release scenario

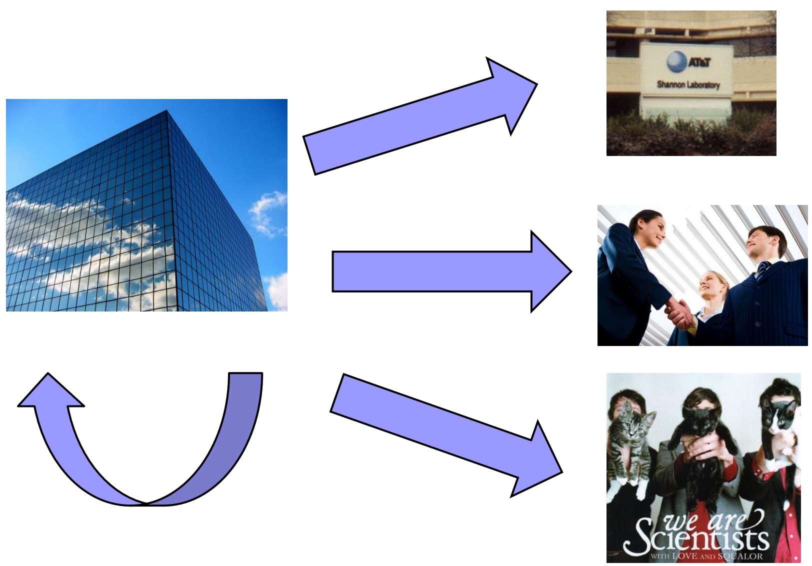


Why Anonymize?

- For Data Sharing
  - Give real(istic) data to others to study without compromising privacy of individuals in the data
  - Allows third-parties to try new analysis and mining techniques not thought of by the data owner
- For Data Retention and Usage
  - Various requirements prevent companies from retaining customer information indefinitely
  - E.g. Google progressively anonymizes IP addresses in search logs
  - Internal sharing across departments (e.g. billing $\rightarrow$ marketing)

Dimensions to consider

- How much privacy do we need?
- How much utility do we want from the anonymized data?
- How will data be accessed: as data feed, as data set, via API?



Who will use the data?

1. Permanent employees

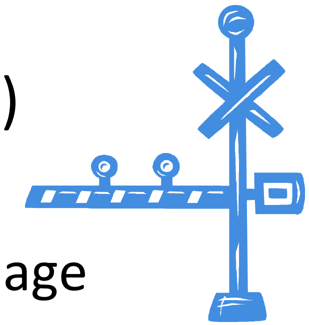
   Temporary employees
   (students, contractors)

2. External organizations

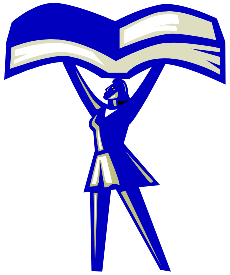
   Data purchasers

3. General Public

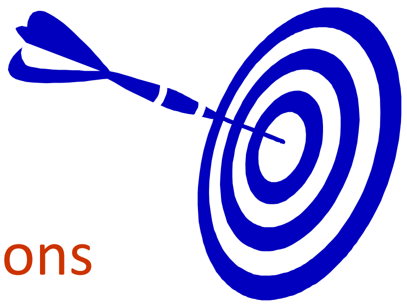
Models of Anonymization



- Interactive Model (akin to statistical databases)
  - Data owner acts as "gatekeeper" to data
  - Researchers pose queries in some agreed language
  - Gatekeeper gives an (anonymized) answer, or refuses to answer
- "Send me your code" model
  - Data owner executes code on their system and reports result
  - Cannot be sure that code is not malicious or steganographic
- Offline, aka "publish and be damned" model
  - Data owner somehow anonymizes data set
  - Publishes the results to the world, and retires
  - The model used in most real data releases

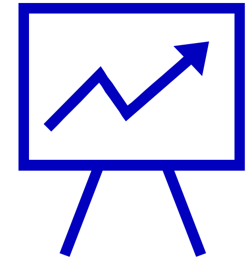


Objectives for Anonymization



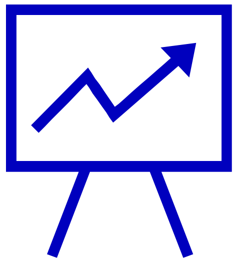
- Prevent (high confidence) inference of associations
  - Prevent inference of salary for an individual in “census”
  - Prevent inference of individual’s viewing history in “video”
  - Prevent inference of individual’s search history in “search”
  - All aim to prevent linking sensitive information to an individual
- Prevent inference of presence of an individual in the data set
  - Satisfying “presence” also satisfies “association” (not vice-versa)
  - Presence in a data set can violate privacy (eg STD clinic patients)
- Have to consider what knowledge might be known to attacker
  - Background knowledge: facts about the data set (X has salary Y)
  - Domain knowledge: broad properties of data (illness Z rare in men)

Utility



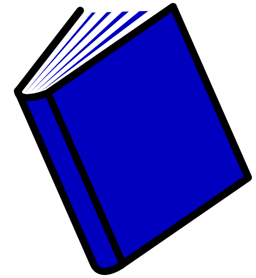
- Anonymization is meaningless if utility of data not considered
  - The empty data set has perfect privacy, but no utility
  - The original data has full utility, but no privacy
- What is "utility"? Depends what the application is…
  - For fixed query set, can look at maximum or average error
  - Problem for publishing: want to support unknown applications!
  - Need some way to quantify utility of alternate anonymizations

Measures of Utility

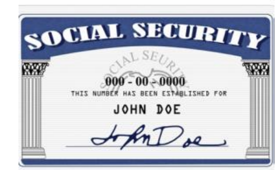


- Define a surrogate measure and try to optimize
  - Often based on the "information loss" of the anonymization
  - Simple example: number of examples deleted from a data set
- Give a guarantee for all queries in some fixed class
  - Hope the class is representative, so other uses have low distortion
  - Costly: some methods enumerate all queries, or all anonymizations
- Empirical Evaluation
  - Perform experiments with a reasonable workload on the result
  - Compare to results on original data (e.g. Netflix prize problems)
- Combinations of multiple methods
  - Optimize for some surrogate, but also evaluate on real queries

Definitions of Technical Terms

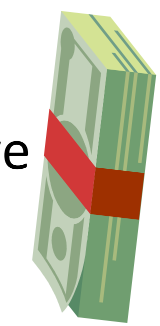


- Identifiers–uniquely identify, e.g. Social Security Number (SSN)



  - Step 0: remove all identifiers
  - Was not enough for AOL search data
- Quasi-Identifiers (QI)—such as DOB, Sex, ZIP Code

  - Enough to partially identify an individual in a dataset
  - DOB+Sex+ZIP unique for 87% of US Residents [Sweeney 02]
- Sensitive attributes (SA)—the associations we want to hide



  - Salary in the “census” example is considered sensitive
  - Not always well-defined: only some “search” queries sensitive
  - In “video”, association between user and video is sensitive
  - One SA can reveal others: bonus may identify salary…

Summary of Anonymization Motivation

- Anonymization needed for safe data sharing and retention
  - Many legal requirements apply
- Various privacy definitions possible
  - Primarily, prevent inference of sensitive information
  - Under some assumptions of background knowledge
- Utility of the anonymized data needs to be carefully studied
  - Different data types imply different classes of query

- Main focus: the publishing model with consideration of utility

Case Study: US Census

- Raw data: information about every US household
  - Who, where; age, gender, racial, income and educational data
- Why released: determine representation, planning
- How anonymized: aggregated to geographic areas (Zip code)
  - Broken down by various combinations of dimensions
  - Released in full after 72 years
  - Census 2020 will use differential privacy techniques
- Attacks: no reports of successful deanonymization so far
  - Attempts by FBI to access raw data have been rebuffed
- Consequences: greater understanding of US population
  - Affects representation, funding of civil projects
  - Rich source of data for future historians and genealogists

Case Study: Netflix Prize

- Raw data: 100M dated ratings from 480K users to 18K movies
- Why released: improve predicting ratings of unlabeled examples
- How anonymized: exact details not described by Netflix
  - All direct customer information removed
  - Only subset of full data; dates modified; some ratings deleted,
  - Movie title and year published in full
- Attacks: dataset was claimed vulnerable [Narayanan Shmatikov 08]
  - Attack links data to IMDB where same users also rated movies
  - Find matches based on similar ratings or dates in both
- Consequences: rich source of user data for researchers
  - Unclear how serious the attacks are in practice

Case Study: AOL Search Data

- Raw data: 20M search queries for 650K users from 2006
- Why released: allow researchers to understand search patterns
- How anonymized: user identifiers removed
  - All searches from same user linked by an arbitrary identifier
- Attacks: many successful attacks identified individual users
  - Ego-surfers: people typed in their own names
  - Zip codes and town names identify an area
  - NY Times identified user 4417749 as 62yr old GA widow
- Consequences: CTO resigned, two researchers fired
  - Well-intentioned effort failed due to inadequate anonymization

Exercises

- Think of a data set or data source that you are familiar with
- Is some of the data (potentially) private? Has the data already been anonymized in some way to protect privacy?
- What are the privacy implications of the raw original data being revealed? What could be discovered?
- In the data, which are the identifying attributes? Which are the quasi-identifiers? Which are the sensitive attributes?
- If all sensitive information was erased, what analyses would no longer be possible?

Working Examples

- Will study an example data set with few attributes
- "Census" data recording incomes and demographics
  - Format: **(SSN, DOB, Sex, Zip, Salary)**
    - **"Zip"** = postal code, reveals approximate region
  - Similar to UCI adult.data set (can have other attributes)

- Many other kinds of data are relevant to privacy
  - "Video" data recording movies viewed
    - Graph data—graph properties should be retained
  - "Search" data recording web searches
    - Set data—each user has different set of keywords

Tabular Data Example

- Census data recording incomes and demographics

SSN	DOB	Sex	ZIP	Salary
11-1-111	1/21/76	M	53715	50,000
22-2-222	4/13/86	F	53715	55,000
33-3-333	2/28/76	M	53703	60,000
44-4-444	1/21/76	M	53703	65,000
55-5-555	4/13/86	F	53706	70,000
66-6-666	2/28/76	F	53706	75,000

- Releasing SSN $\rightarrow$ Salary association violates individual's privacy
  - SSN is an identifier, Salary is a sensitive attribute (SA)

Tabular Data Example: De-Identification

- Census data: remove SSN to create a de-identified table

DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000

- Does the de-identified table preserve an individual's privacy?
  - Depends on what other information an attacker knows

Tabular Data Example: Linking Attack

- De-identified private data + publicly available data

DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000

SSN	DOB
11-1-111	1/21/76
33-3-333	2/28/76

- Cannot uniquely identify either individual's salary
  - DOB is a quasi-identifier (QI)

Tabular Data Example: Linking Attack

- De-identified private data + publicly available data

DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000

SSN	DOB	Sex
11-1-111	1/21/76	M
33-3-333	2/28/76	M

- Uniquely identified one individual's salary, but not the other's
  - DOB, Sex are quasi-identifiers (QI)

Tabular Data Example: Linking Attack

- De-identified private data + publicly available data

DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000

SSN	DOB	Sex	ZIP
11-1-111	1/21/76	M	53715
33-3-333	2/28/76	M	53703

- Uniquely identified both individuals' salaries
  - [DOB, Sex, ZIP] is unique for lots of US residents [Sweeney 02]

Tabular Data Example: Anonymization

- Anonymization through tuple suppression

DOB	Sex	ZIP	Salary
*	*	*	*
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
*	*	*	*
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000

SSN	DOB	Sex	ZIP
11-1-111	1/21/76	M	53715

- Cannot link to private table even with knowledge of QI values
  - Missing tuples could take any value from the space of all tuples
  - Introduces a lot of uncertainty

Tabular Data Example: Anonymization

- Anonymization through QI attribute generalization

DOB	Sex	ZIP	Salary
1/21/76	M	537**	50,000
4/13/86	F	537**	55,000
2/28/76	*	537**	60,000
1/21/76	M	537**	65,000
4/13/86	F	537**	70,000
2/28/76	*	537**	75,000

SSN	DOB	Sex	ZIP
11-1-111	1/21/76	M	53715
33-3-333	2/28/76	M	53703

- Cannot uniquely identify tuple with knowledge of QI values
  - More precise form of uncertainty than tuple suppression
  - E.g., ZIP = 537** → ZIP ∈ {53700, …, 53799}

Tabular Data Example: Anonymization

- Anonymization through sensitive attribute (SA) permutation

DOB	Sex	ZIP	Salary
1/21/76	M	53715	55,000
4/13/86	F	53715	50,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	75,000
2/28/76	F	53706	70,000

SSN	DOB	Sex	ZIP
11-1-111	1/21/76	M	53715
33-3-333	2/28/76	M	53703

- Can uniquely identify tuple, but uncertainty about SA value
  - Much more precise form of uncertainty than generalization

Tabular Data Example: Anonymization

- Anonymization through sensitive attribute (SA) perturbation

DOB	Sex	ZIP	Salary
1/21/76	M	53715	60,000
4/13/86	F	53715	45,000
2/28/76	M	53703	60,000
1/21/76	M	53703	55,000
4/13/86	F	53706	80,000
2/28/76	F	53706	75,000

SSN	DOB	Sex	ZIP
11-1-111	1/21/76	M	53715
33-3-333	2/28/76	M	53703

- Can uniquely identify tuple, but get “noisy” SA value

k-Anonymization [Samarati, Sweeney 98]

- k-anonymity: Table T satisfies k-anonymity wrt quasi-identifier QI iff each tuple in (the multiset) T[QI] appears at least k times
  - Protects against "linking attack"
- k-anonymization: Table T' is a k-anonymization of T if T' is a generalization/suppression of T, and T' satisfies k-anonymity

DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000

→

DOB	Sex	ZIP	Salary
1/21/76	M	537**	50,000
4/13/86	F	537**	55,000
2/28/76	*	537**	60,000
1/21/76	M	537**	65,000
4/13/86	F	537**	70,000
2/28/76	*	537**	75,000

k-Anonymization and Uncertainty

- Intuition: A k-anonymized table T' represents the set of all "possible world" tables T_i s.t. T' is a k-anonymization of T_i
  - With no background knowledge, all possible worlds are equally plausible

- Query Answering
  - Queries should (implicitly) range over all possible worlds
  - Example query: what is the salary of individual (1/21/76, M, 53715)? Best guess is 57,500 (weighted average of 50,000 and 65,000)
  - Example query: what is the maximum salary of males in 53706? Could be as small as 50,000, or as big as 75,000

Computing k-Anonymizations

- Huge literature: variations depend on search space and algorithm
  - Generalization vs (tuple) suppression
  - Global (e.g., full-domain) vs local (e.g., multidimensional) recoding
  - Hierarchy-based vs partition-based (e.g., numerical attributes)

Algorithm	Model	Properties	Complexity
Samarati 01	G+TS, FD, HB	One exact, binary search	$O(2^{\|QI\|})$
Sweeney 02	G+TS, FD, HB	Exact, exhaustive	$O(2^{\|QI\|})$
Bayardo+ 05	G+TS, FD, PB	Exact, top-down	$O(2^{\|QI\|})$
LeFevre+ 05	G+TS, FD, HB	All exact, bottom-up cube	$O(2^{\|QI\|})$

Algorithm	Model	Properties	Complexity
Meyerson+ 04	S, L	NP-hard, O(k log k) approximation	$O(n^{2k})$
Aggarwal+ 05a	S, L	O(k) approximation	$O(kn^2)$
Aggarwal+ 05b	G, L, HB	O(k) approximation	$O(kn^2)$
LeFevre+ 06	G, MD, PB	Constant-factor approximation	O(n log n)

Incognito [LeFevre+ 05]

- Every full-domain generalization described by a "domain vector"
  - B0={1/21/76, 2/28/76, 4/13/86} → B1={76-86}
  - S0={M, F} → S1={*}
  - Z0={53715,53710,53706,53703}→ Z1={5371*,5370*}→ Z2={537**}

DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000

B0, S1, Z2 →

DOB	Sex	ZIP	Salary
1/21/76	*	537**	50,000
4/13/86	*	537**	55,000
2/28/76	*	537**	60,000
1/21/76	*	537**	65,000
4/13/86	*	537**	70,000
2/28/76	*	537**	75,000

Incognito [LeFevre+ 05]

- Every full-domain generalization described by a "domain vector"
  - B0={1/21/76, 2/28/76, 4/13/86} $\rightarrow$ B1={76-86}
  - S0={M, F} $\rightarrow$ S1={*}
  - Z0={53715,53710,53706,53703} $\rightarrow$ Z1={5371*,5370*} $\rightarrow$ Z2={537**}

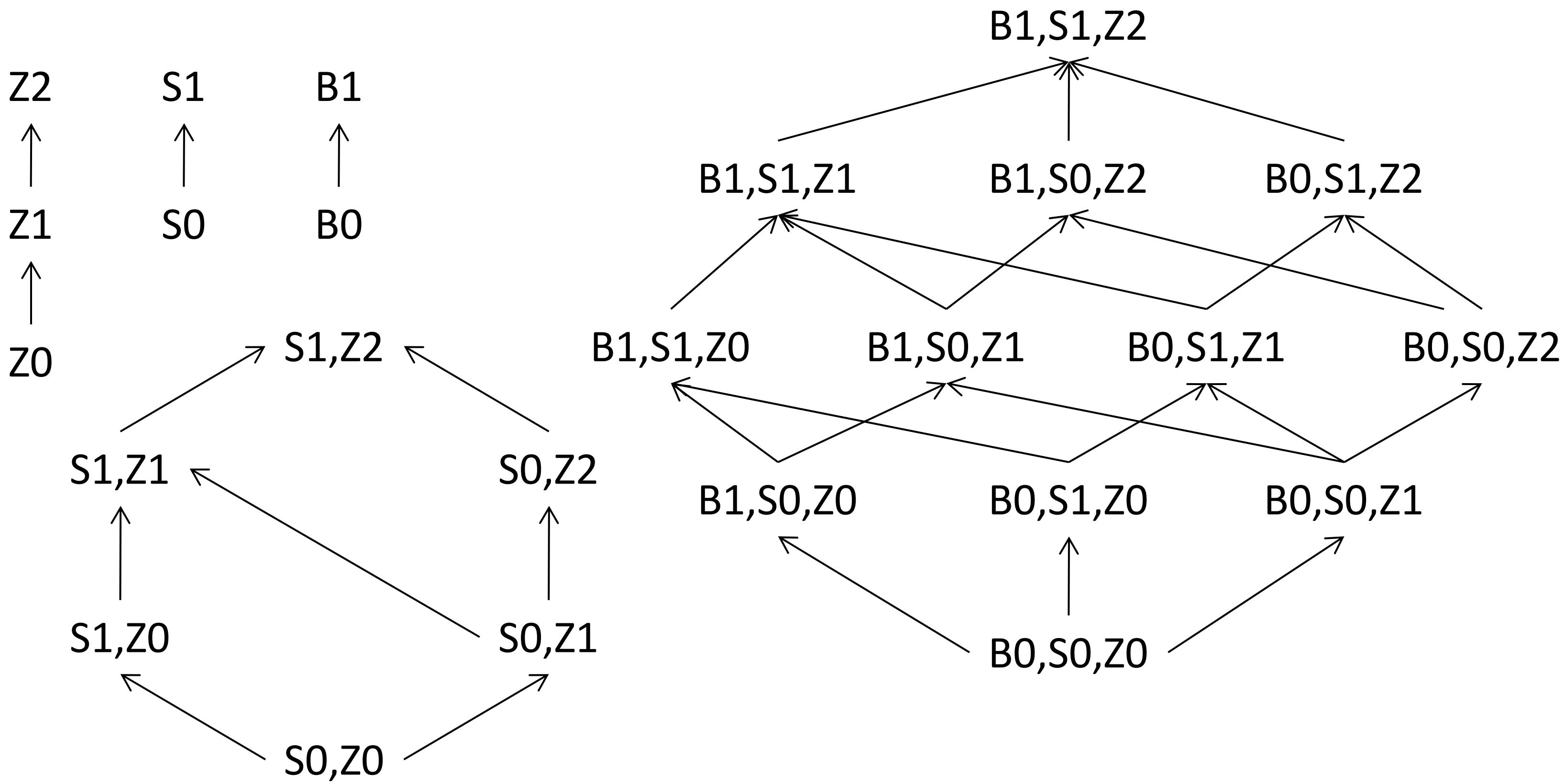
DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000

B1, S0, Z2 $\longrightarrow$

DOB	Sex	ZIP	Salary
76-86	M	537**	50,000
76-86	F	537**	55,000
76-86	M	537**	60,000
76-86	M	537**	65,000
76-86	F	537**	70,000
76-86	F	537**	75,000

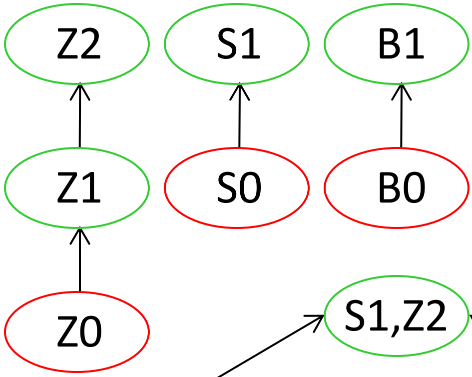
Incognito [LeFevre+ 05]

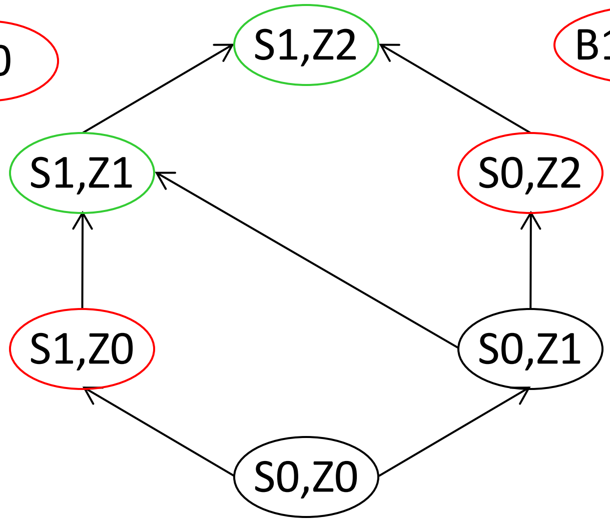
- Lattice of domain vectors

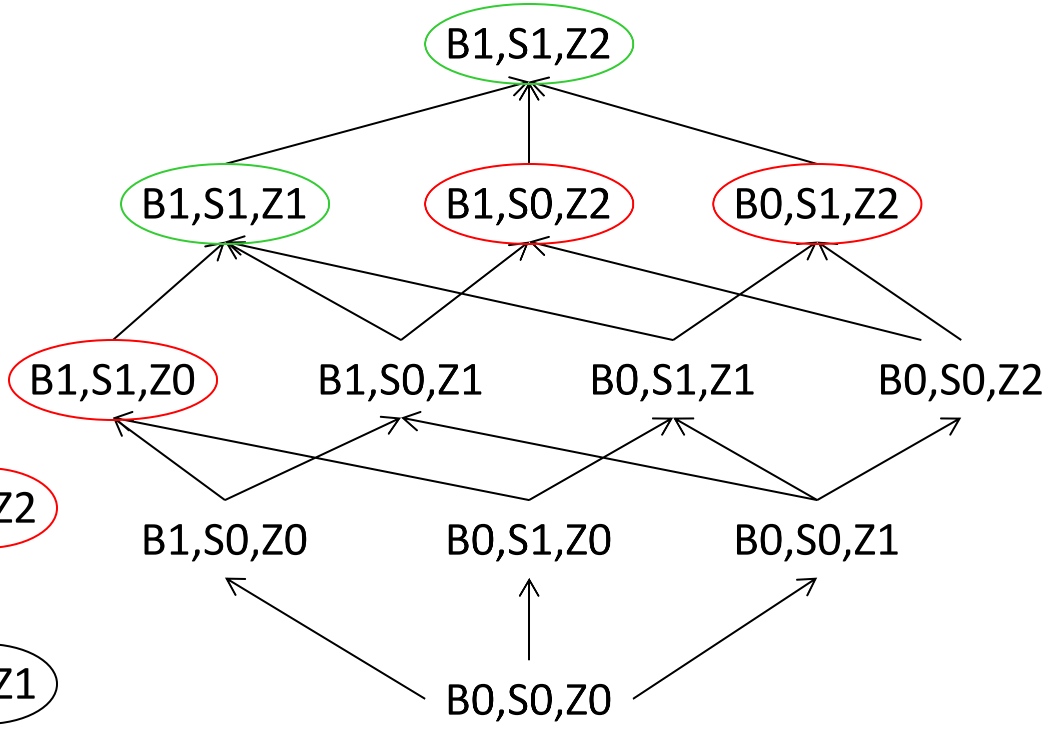


Incognito [LeFevre+ 05]

- Lattice of domain vectors







Incognito [LeFevre+ 05]

- Subset Property: If table T is k-anonymous wrt attributes Q, then T is k-anonymous wrt any set of attributes that is a subset of Q
- Generalization Property: If table T_2 is a generalization of table T_1, and T_1 is k-anonymous, then T_2 is k-anonymous
- Computes all "minimal" full-domain generalizations
  - Set of minimal full-domain generalizations forms an anti-chain
  - Can use any reasonable utility metric to choose "optimal" solution

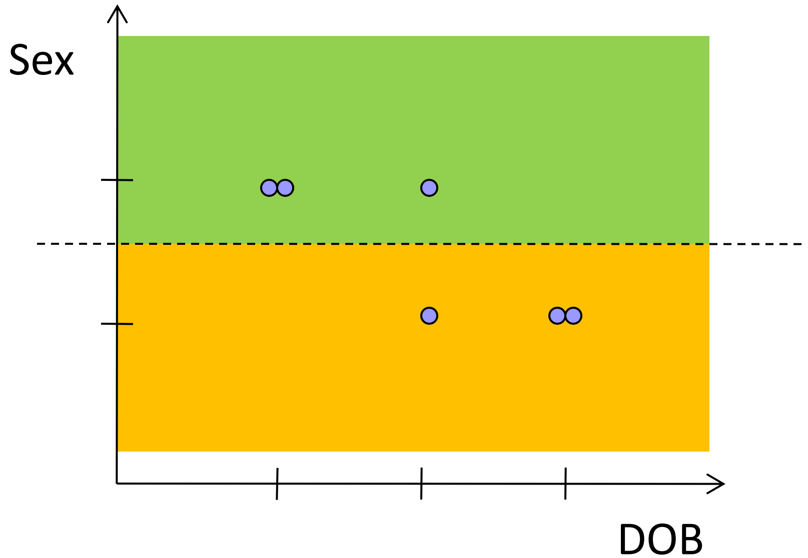
Mondrian [LeFevre+ 06]

- Computes one “good” multi-dimensional generalization
  - Uses local recoding to explore a larger search space
  - Treats all attributes as ordered, chooses partition boundaries
- Utility metrics considered in the paper
  - Discernability: sum of squares of group sizes
  - Normalized average group size = (total tuples / total groups) / k
- Efficient: greedy O(n log n) heuristic for NP-hard problem
- Quality guarantee: solution is a constant-factor approximation

Mondrian [LeFevre+ 06]

- Uses ideas from spatial kd-tree construction
  - QI tuples = points in a multi-dimensional space
  - Hyper-rectangles with ≥ k points = k-anonymous groups
  - Choose axis-parallel line to partition point-multiset at median

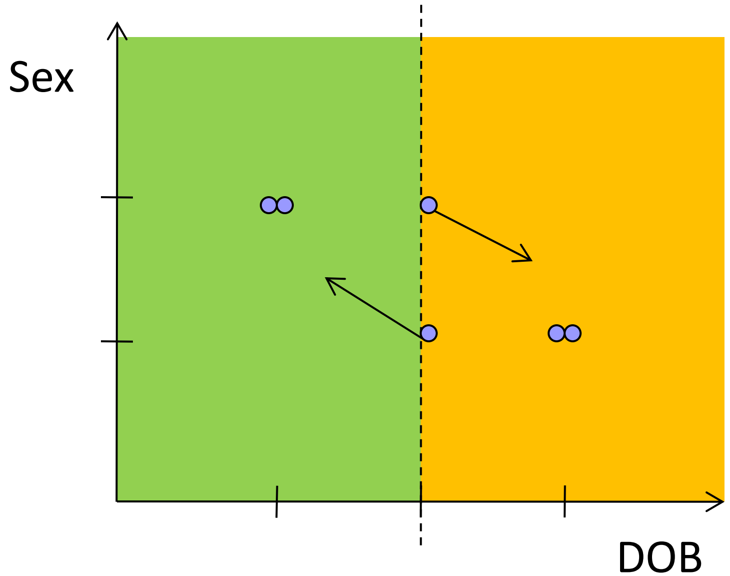
DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000



Mondrian [LeFevre+ 06]

- Uses ideas from spatial kd-tree construction
  - QI tuples = points in a multi-dimensional space
  - Hyper-rectangles with ≥ k points = k-anonymous groups
  - Choose axis-parallel line to partition point-multiset at median

DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000



Homogeneity Attack [Machanavajjhala+ 06]

- Issue: k-anonymity requires each tuple in (the multiset) T[QI] to appear ≥ k times, but does not say anything about the SA values
  - If (almost) all SA values in a QI group are equal, loss of privacy!
  - The problem is with the choice of grouping, not the data

DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	50,000
4/13/86	F	53706	55,000
2/28/76	F	53706	60,000

Not Ok! →

DOB	Sex	ZIP	Salary
1/21/76	*	537**	50,000
4/13/86	*	537**	55,000
2/28/76	*	537**	60,000
1/21/76	*	537**	50,000
4/13/86	*	537**	55,000
2/28/76	*	537**	60,000

Homogeneity Attack [Machanavajjhala+ 06]

- Issue: k-anonymity requires each tuple in (the multiset) T[QI] to appear ≥ k times, but does not say anything about the SA values
  - If (almost) all SA values in a QI group are equal, loss of privacy!
  - The problem is with the choice of grouping, not the data
  - For some groupings, no loss of privacy

DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	50,000
4/13/86	F	53706	55,000
2/28/76	F	53706	60,000

Ok! →

DOB	Sex	ZIP	Salary
76-86	*	53715	50,000
76-86	*	53715	55,000
76-86	*	53703	60,000
76-86	*	53703	50,000
76-86	*	53706	55,000
76-86	*	53706	60,000

l-Diversity [Machanavajjhala+ 06]

- *l*-Diversity Principle: a table is *l*-diverse if each of its QI groups contains at least *l* "well-represented" values for the SA

- Different definitions of *l*-diversity based on formalizing the intuition of a "well-represented" value
  - Entropy *l*-diversity: for each QI group g, entropy(g) ≥ log(*l*)
  - Recursive (c,*l*)-diversity: for each QI group g with m SA values, and r_i the i'th highest frequency, $r_1 < c\,(r_l + r_{l+1} + \ldots + r_m)$
  - Folk *l*-diversity: for each QI group g, no SA value should occur more than 1/*l* fraction of the time = Recursive(1/*l*, 1)-diversity

- Intuition: Most frequent value does not appear too often compared to the less frequent values in a QI group

Computing *l*-Diversity [Machanavajjhala+ 06]

- Key Observation: entropy *l*-diversity and recursive(c,*l*)-diversity possess the Subset Property and the Generalization Property

- Algorithm Template:
  - Take an algorithm for k-anonymity and replace the k-anonymity test for a generalized table by the *l*-diversity test
  - Easy to check based on counts of SA values in QI groups

t-Closeness [Li+ 07]

- Limitations of *l*-diversity
  - Similarity attack: SA values are distinct, but semantically similar

DOB	Sex	ZIP	Salary
1/21/76	*	537**	50,000
4/13/86	*	537**	55,000
2/28/76	*	537**	60,000
1/21/76	*	537**	50,001
4/13/86	*	537**	55,001
2/28/76	*	537**	60,001

SSN	DOB	Sex	ZIP
11-1-111	1/21/76	M	53715

- t-Closeness Principle: a table has t-closeness if in each of its QI groups, the distance between the distribution of SA values in the group and in the whole table is no more than threshold t

Answering Queries on Generalized Tables

- Observation: Generalization loses a lot of information, resulting in inaccurate aggregate analyses

- How many people were born in 1976?
  - Bounds = [1,5], selectivity estimate = 1, actual value = 4

DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000

→

DOB	Sex	ZIP	Salary
76-86	M	537**	50,000
76-86	F	537**	55,000
76-86	M	537**	60,000
76-86	M	537**	65,000
76-86	F	537**	70,000
76-86	F	537**	75,000

Answering Queries on Generalized Tables

- Observation: Generalization loses a lot of information, resulting in inaccurate aggregate analyses

- What is the average salary of people born in 1976?
  - Bounds = [50K,75K], actual value = 62.5K

DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000

→

DOB	Sex	ZIP	Salary
76-86	M	537**	50,000
76-86	F	537**	55,000
76-86	M	537**	60,000
76-86	M	537**	65,000
76-86	F	537**	70,000
76-86	F	537**	75,000

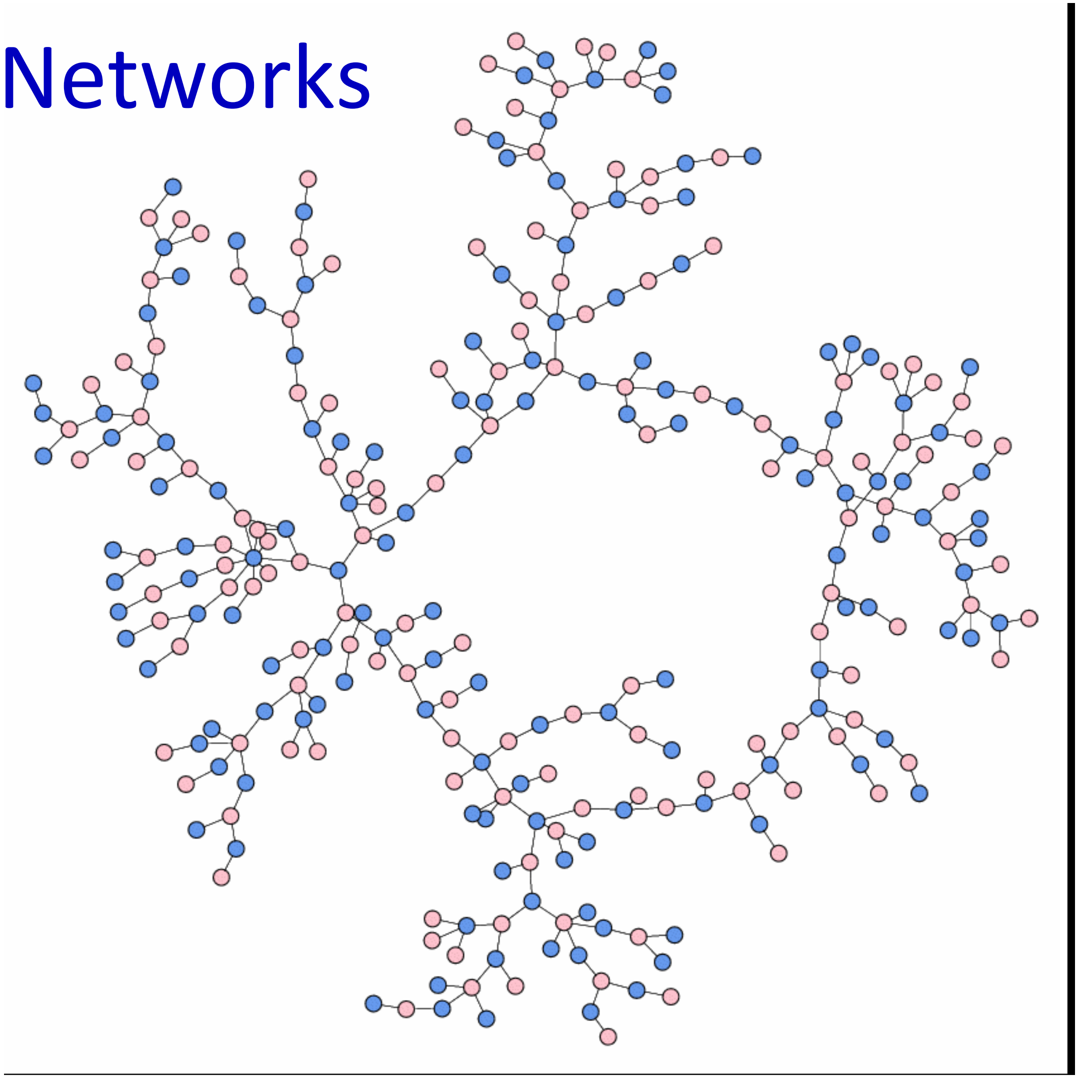
Subsequent Attacks and Developments

- Minimality Attack [Wong+ 07]:
  - Uses knowledge of anonymization algorithm to argue some possible worlds are not consistent with output
- deFinetti Attack [Kifer 09]:
  - Uses knowledge from anonymized data to argue some associations are more likely than others
- Further development:
  - Due to such attacks, work on “syntactic methods” has slowed
  - Few if any significant deployments have been reported
  - Continued interest in areas such as graph data anonymization

More to life than tables…

Recommendation Data

Social Networks



Plot from Mark Newman, based on data in
"*The structure of adolescent romantic and sexual networks*", American Journal of Sociology 110, 44-91 (2004) .
Males are red, females are blue

Location and Trajectory Data

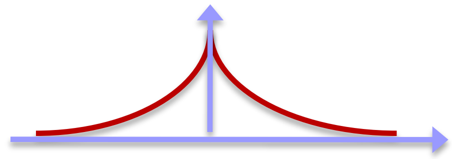
Web Search Logs

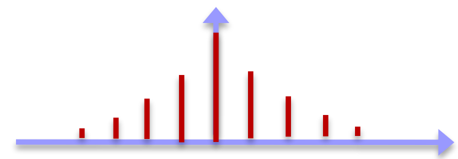
References

- [Sweeney 02] Latanya Sweeney. k-Anonymity: A Model for Protecting Privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5): 557-570 (2002)
- [Samarati Sweeney 98] Pierangela Samarati, Latanya Sweeney. Generalizing Data to Provide Anonymity when Disclosing Information. PODS 1998
- [Narayanan Shmatikov 08] Arvind Narayanan, Vitaly Shmatikov. Robust De-anonymization of Large Sparse Datasets. IEEE Symposium on Security and Privacy 2008: 111-125
- [LeFevre+ 05] Kristen LeFevre, David J. DeWitt, Raghu Ramakrishnan. Incognito: Efficient Full-Domain K-Anonymity. SIGMOD Conference 2005
- [LeFevre+ 06] Kristen LeFevre, David J. DeWitt, Raghu Ramakrishnan. Mondrian Multidimensional K-Anonymity. ICDE 2006

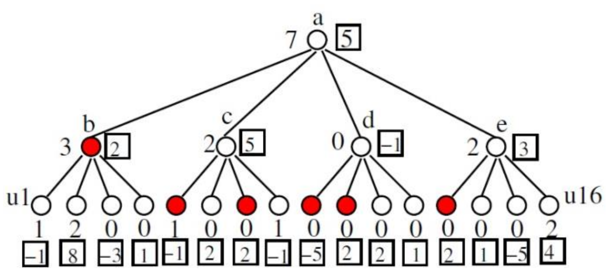
References

- [Machanavajjhala+ 06] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, Muthuramakrishnan Venkitasubramaniam. *l*-diversity: Privacy beyond k-anonymity. TKDD 1(1): 3 (2007)
- [Li+ 07] Ninghui Li, Tiancheng Li, Suresh Venkatasubramanian. Closeness: A New Privacy Measure for Data Publishing. IEEE Trans. Knowl. Data Eng. 22(7): 943-956 (2010)
- [Wong+ 07] Raymond Chi-Wing Wong, Ada Wai-Chee Fu, Ke Wang, Jian Pei. Minimality Attack in Privacy Preserving Data Publishing. VLDB 2007: 543-554
- [Kifer 09] Daniel Kifer. Attacks on privacy and deFinetti's theorem. SIGMOD Conference 2009



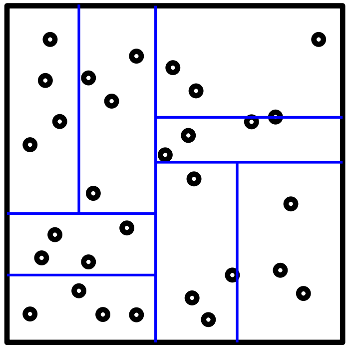


Building Blocks of Privacy: Differentially Private Mechanisms



Graham Cormode

graham@cormode.org



Differential Privacy: a new hope

- Principle: released info 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 (DP)
  - Simple recipe for some data types e.g. numeric answers
  - Simple rules allow us to reason about composition of results
  - More complex algorithms for arbitrary data (many DP mechanisms)
- Adopted and used by several organizations:
  - US Census, Common Data Project, Facebook (?), Google, Apple…

Differential Privacy Definition

The output distribution of a differentially private algorithm changes very little whether or not any individual's data is included in the input (so it's OK to contribute your data)

A randomized algorithm K satisfies ε-differential privacy if:

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

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

Neighboring datasets differ in one individual: we say $|D - D'| = 1$

Achieving Differential Privacy

- Suppose we want to output the number of left-handed people in our data set
  - Can reduce the description of the data to just the answer, n
  - Want a randomized algorithm $K(n)$ that will output an integer
  - Consider the distribution $\Pr[K(n) = m]$ for different m
- Write $\exp(\varepsilon) = \alpha$, and $\Pr[K(n) = n] = p_n$. Then:
  $\Pr[K(n) = n-1] \leq \alpha \Pr[K(n-1)=n-1] = \alpha\, p_{n-1}$
  $\Pr[K(n) = n-2] \leq \alpha \Pr[K(n-1) = n-2] \leq \alpha^2 \Pr[K(n-2)=n-2] = \alpha^2\, p_{n-2}$
  $\Pr[K(n) = n-i] \leq \alpha^i\, p_{n-i}$
  Similarly, $\Pr[K(n) = n+i] \leq \alpha^i\, p_{n+i}$

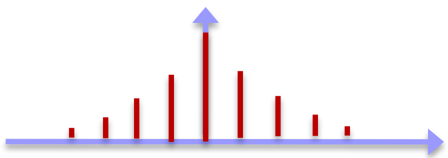
Achieving Differential Privacy

- We have $\Pr[K(n) = n\text{-}i] \leq \alpha^i\, p_{n-i}$ and $\Pr[K(n) = n+i] \leq \alpha^i\, p_{n+i}$
- Within these constraints, we want to maximize p_n
  - This maximizes the probability of returning “correct” answer
  - Means we turn the inequalities into equalities
- For simplicity, set $p_n = p$ for all n
  - Means the distribution of “shifts” is the same whatever n is
- Yields: $\Pr[K(n) = n\text{-}i] = \alpha^i\, p$ and $\Pr[K(n) = n+i] = \alpha^i\, p$
  - Sum over all shifts i:

$$p + \sum_{i=1}^{\infty} 2\alpha^i\, p = 1$$
$$p + 2p\, \alpha/(1-\alpha) = 1$$
$$p(1 - \alpha + 2\alpha)/(1-\alpha) = 1$$
$$p = (1-\alpha)/(1+\alpha)$$

Geometric Mechanism

- What does this mean?
  - For input n, output distribution is $\Pr[K(n) = m] = \alpha^{|m-n|} \cdot (1-\alpha)/(1+\alpha)$
- What does this look like?



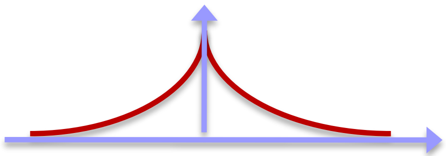
  - Symmetric geometric distribution, centered around n
  - We draw from this distribution centered around zero, and add to the true answer
  - We get the "true answer plus (symmetric geometric) noise"
- A first differentially private mechanism for outputting a count
  - We call this "the geometric mechanism"

Truncated Geometric Mechanism

- Some practical concerns:
  - This mechanism could output any value, from $-\infty$ to $+\infty$
- Solution: we can "truncate" the output of the mechanism
  - E.g. decide we will never output any value below zero, or above N
  - Any value drawn below zero is "rounded up" to zero
  - Any value drawn above N is "rounded down" to N
  - This does not affect the differential privacy properties
  - Can directly compute the closed-form probability of these outcomes
- (Truncated) geometric mechanism is unique, optimal mechanism
  - Properties proved in [Ghosh Roughgarden Sundarajaran 08]

Laplace Mechanism

- Sometimes we want to output real values instead of integers
- The Laplace Mechanism naturally generalizes Geometric



  - Add noise from a symmetric continuous distribution to true answer
  - Laplace distribution is a symmetric exponential distribution
  - Is DP for same reason as geometric: shifting the distribution changes the probability by at most a constant factor
  - PDF: $\Pr[X = x] = 1/2\lambda \exp(-|x|/\lambda)$
    Variance $= 2\lambda^2$

Sensitivity of Numeric Functions

- For more complex functions, we need to calibrate the noise to the influence an individual can have on the output
  - The (global) sensitivity of a function F is the maximum (absolute) change over all possible adjacent inputs
  - $S(F) = \max_{D, D' : |D-D'|=1} \|F(D) - F(D')\|_1$
  - Intuition: S(F) characterizes the scale of the influence of one individual, and hence how much noise we must add
- S(F) is small for many common functions
  - S(F) = 1 for COUNT
  - S(F) = 2 for HISTOGRAM
  - Bounded for other functions (MEAN, covariance matrix…)

Laplace Mechanism with Sensitivity

- Release $F(x) + Lap(S(F)/\varepsilon)$ to obtain ε-DP guarantee
  - $F(x)$ = true answer on input x
  - $Lap(\lambda)$ = noise sampled from Laplace dbn with parameter λ
  - Exercise: show this meets ε-differential privacy requirement
- Intuition on impact of parameters of differential privacy (DP):
  - Larger $S(F)$, more noise (need more noise to mask an individual)
  - Smaller ε, more noise (more noise increases privacy)
  - Expected magnitude of $|Lap(\lambda)|$ is (approx) λ

Sequential Composition

- What happens if we ask multiple questions about same data?
  - We reveal more, so the bound on ε differential privacy weakens
- Suppose we output via K_1 and K_2 with ε_1, ε_2 differential privacy:
  For any neighbouring D, D', we have

  $\Pr[\ K_1(D) = S_1\ ] \leq \exp(\varepsilon_1) \Pr[K_1(D') = S_1]$, and

  $\Pr[\ K_2(D) = S_2\ ] \leq \exp(\varepsilon_2) \Pr[K_2(D') = S_2]$

  $$\begin{aligned}\Pr[\ (K_1(D) = S_1), (K_2(D) = S_2)] &= \Pr[K_1(D)=S_1] \Pr[K_2(D) = S_2] \\ &\leq \exp(\varepsilon_1) \Pr[K_1(D') = S_1] \exp(\varepsilon_2) \Pr[K_2(D') = S_2] \\ &= \exp(\varepsilon_1 + \varepsilon_2) \Pr[(K_1(D') = S_1), (K_2(D') = S_2)]\end{aligned}$$

  - Use the fact that the noise distributions are independent
- Bottom line: result is $\varepsilon_1 + \varepsilon_2$ differentially private
  - Can reason about sequential composition by just "adding the ε's"

Parallel Composition

- Sequential composition is pessimistic
  - Assumes outputs are correlated, so privacy budget is diminished
- If the inputs are disjoint, then result is $\max(\varepsilon_1, \varepsilon_2)$ private
- Example:
  - Ask for count of people broken down by handedness, hair color

	Redhead	Blond	Brunette
Left-handed	23	35	56
Right-handed	215	360	493

  - Each cell is a disjoint set of individuals
  - So can release each cell with ε-differential privacy (parallel composition) instead of 6ε DP (sequential composition)

Exponential Mechanism

- What happens when we want to output non-numeric values?
- Exponential mechanism is most general approach
  - Captures all possible DP mechanisms
  - But ranges over all possible outputs, may not be efficient
- Requirements:
  - Input value x
  - Set of possible outputs O
  - Quality function, q, assigns "score" to possible outputs $o \in O$
    - $q(x, o)$ is bigger the "better" o is for x
  - Sensitivity of $q = S(q) = \max_{x,x',o} |q(x,o) - q(x',o)|$

Exponential Mechanism

- Sample output $o \in O$ with probability

  $$\Pr[K(x) = o] = \exp(\varepsilon\, q(x,o)) / (\sum_{o' \in O} \exp(\varepsilon q(x,o')))$$

- Result is $(2\varepsilon\, S(q))$-DP
  - Shown by considering change in numerator and denominator under change of x is at most a factor of $\exp(\varepsilon\, S(q))$
- Scalability: need to be able to draw from this distribution
- Generalizations:
  - O can be continuous, $\sum$ becomes an integral
  - Can apply a prior distribution over outputs as $P(o)$
    - We assume a uniform prior for simplicity

Exponential Mechanism Example 1: Count

- Suppose input is a count n, we want to output (noisy) n
  - Outputs O = all integers
  - $q(n,o) = -|o-n|$
  - $S(q) = 1$
  - Then $\Pr[\,K(n) = o] = \exp(-\varepsilon\, |o-n|)/(\sum_o -\varepsilon|o-n|) = \alpha^{-|o-n|} \cdot (1-\alpha)/(1-\alpha)$
  - Simplifies to the Geometric mechanism!
- Similarly, if O = all reals, applying exponential mechanism results in the Laplace Mechanism
- Illustrates the claim that Exponential Mechanism captures all possible DP mechanisms

Exponential Mechanism, Example 2: Median

- Let $M(X)$ = median of set of values in range $[0,T]$ (e.g. median age)
- Try Laplace Mechanism: $S(M) = T$
  - There can be datasets X, X' where $M(X) = 0$, $M(X') = T$, $|X-X'|=1$
  - Consider $X = [0^n, 0, T^n]$, $X' = [0^n, T, T^n]$
  - Noise from Laplace mechanism outweighs the true answer!
- Exponential Mechanism: set $q(X,o) = -|\,\mathrm{rank}_X(o) - |X|/2\,|$
  - Define $\mathrm{rank}_X(o)$ as the number of elements in X dominated by o
  - Note, $\mathrm{rank}_X(M(X)) = |X|/2$: median has rank half
  - $S(q) = 1$: adding or removing an individual changes q by at most 1
  - Then $\Pr[\,K(X) = o] = \exp(\varepsilon\, q(X,o))/(\sum_{o' \in O} \exp(\varepsilon\, q(X,o')))$
  - Problem: Output set O could be very large, how to make efficient?

Exponential Mechanism, Example 2: Median

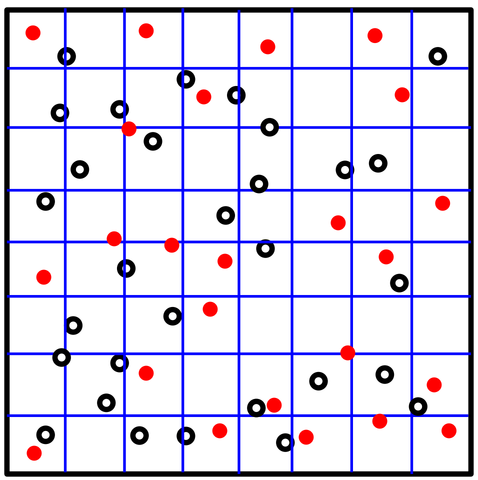
- Observation: for many values of o, $q(X, o)$ is the same:
  - Index X in sorted order so $x_1 \le x_2 \le x_3 \le \ldots \le x_n$
  - Then for any $x_i \le o < o' \le x_{i+1}$, $\text{rank}_X(o) = \text{rank}_X(o')$
  - Hence $q(X,o) = q(X,o')$
- Break possible outputs into ranges:
  - $O_0 = [0,x_1]$ $\quad$ $O_1 = [x_1, x_2]$ $\quad$ … $\quad$ $O_n = [x_n, T]$
  - Pick range O_j with probability proportional to $|O_j|\exp(\varepsilon q(X,O_j))$
  - Pick output $o \in O_j$ uniformly from the range
  - Time cost is proportional to number of ranges n (after sorting X)
- Similar tricks make exponential mechanism practical elsewhere

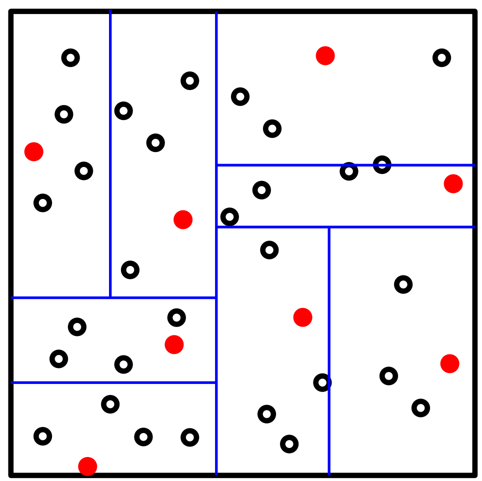
Recap

- Have developed a number of building blocks for DP:
  - Geometric and Laplace mechanism for numeric functions
  - Exponential mechanism for sampling from arbitrary sets
- And "cement" to glue things together:
  - Parallel and sequential composition theorems
- With these blocks and cement, can build a lot
  - Many papers arrive from careful combination of these tools!
- Useful fact: any post-processing of DP output remains DP
  - (so long as you don't access the original data again)
  - Helps reason about privacy of data release processes

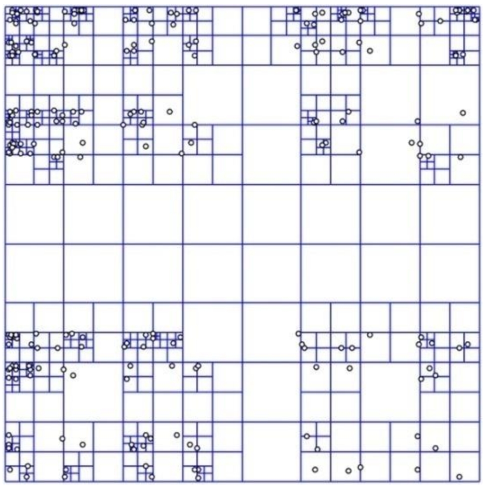
Case Study: Sparse Spatial Data

- Consider location data of many individuals
  - Some dense areas (towns and cities), some sparse (rural)
- Applying DP naively simply generates noise
  - lay down a fine grid, signal overwhelmed by noise
- Instead: compact regions with sufficient number of points

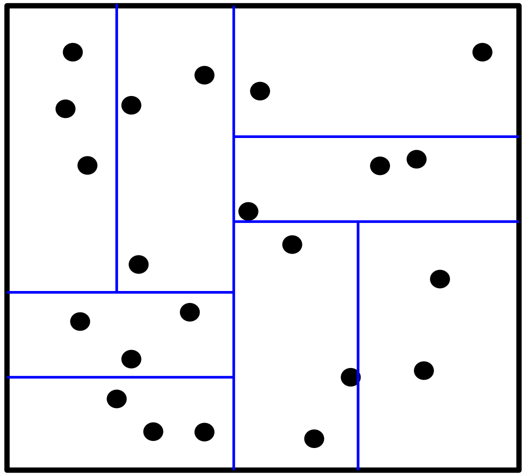




Private Spatial decompositions



quadtree



kd-tree

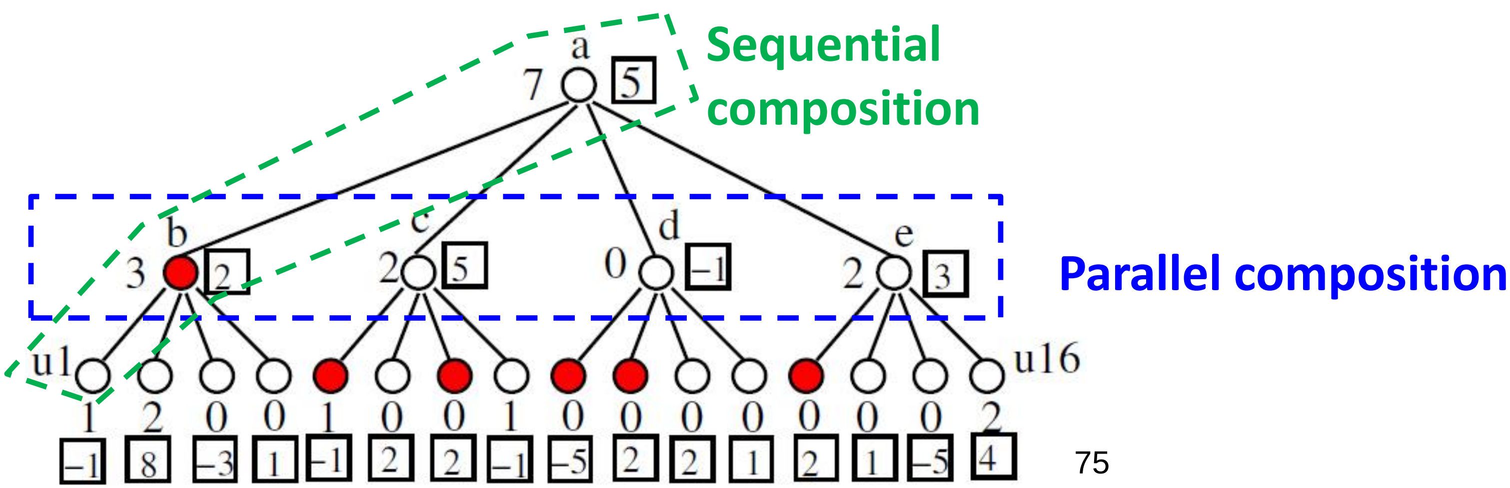
- Build: adapt existing methods to have differential privacy
- Release: a private description of data distribution (in the form of bounding boxes and noisy counts)

Building a Private kd-tree

- Process to build a private kd-tree
  - Input: maximum height *h*, minimum leaf size *L*, data set
  - Choose dimension to split
  - Get (private) median in this dimension
  - Create child nodes and add noise to the counts
  - Recurse until we hit some stopping condition, e.g.:
    - Max height is reached
    - (Noisy) count of this node less than *L*
    - Budget along the root-leaf path has used up
- The entire PSD satisfies DP by the composition property

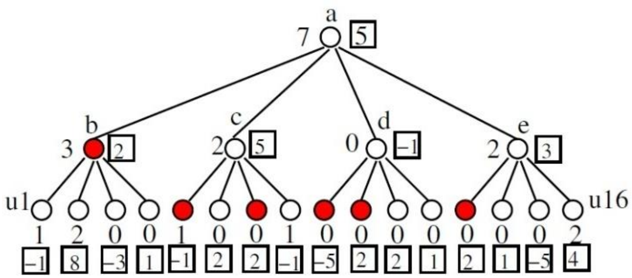
Building PSDs – privacy budget allocation

- Data owner specifies a total budget ε reflecting the level of anonymization desired
- Budget is split between medians and counts
  - Tradeoff accuracy of division with accuracy of counts
- Budget is split across levels of the tree
  - Privacy budget used along any root-leaf path should total ε



Privacy budget allocation

- How to set an ε_i for each level?
  - Compute the number of nodes touched by a 'typical' query
  - Minimize variance of such queries
  - Optimization: min $\sum_i 2^{h-i} / \varepsilon_i^2$ s.t. $\sum_i \varepsilon_i = \varepsilon$
  - Solved by $\varepsilon_i \propto (2^{(h-i)})^{1/3}\varepsilon$: more to leaves
  - Total error (variance) goes as $2^h/\varepsilon^2$



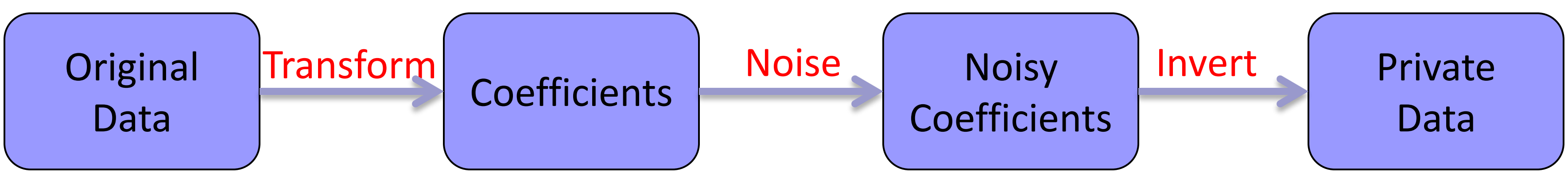
- Tradeoff between noise error and spatial uncertainty
  - Reducing h drops the noise error
  - But lower h increases the size of leaves, more uncertainty

Post-processing of noisy counts

- Can do additional post-processing of the noisy counts
  - To improve query accuracy and achieve consistency
- Intuition: we have count estimate for a node and for its children
  - Combine these independent estimates to get better accuracy
  - Make consistent with some true set of leaf counts
- Formulate as a linear system in n unknowns [Hay et al 10]
  - Avoid explicitly solving the system
  - Expresses optimal estimate for node v in terms of estimates of ancestors and noisy counts in subtree of v
  - Use the tree-structure to solve in three passes over the tree
  - Linear time to find optimal, consistent estimates

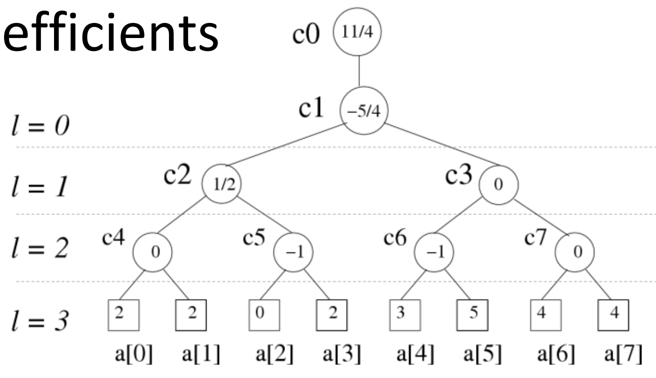
Data Transformations

- Can think of trees as a 'data-dependent' transform of input
- Can apply other data transformations
- General idea:
  - Apply transform of data
  - Add noise in the transformed space (based on sensitivity)
  - Publish noisy coefficients, or invert transform (post-processing)
- Goal: pick a transform that preserves good properties of data
  - And which has low sensitivity, so noise does not corrupt



Wavelet Transform

- Haar wavelet transform commonly used to approximate data
  - Any 1D range is expressed using 2log n coefficients
  - Each input point affects log n coefficients
  - Is a linear, orthonormal transform



(a)

- Can add noise to wavelet coefficients
  - Treat input as a 1D histogram of counts
  - Bounded sensitivity: each individual affects coefficients by O(1)
  - Can transform noisy coefficients back to get noisy histogram
- Range queries are answered well in this model
  - Each range query picks up noise (variance) $O(\log^3 n / \varepsilon^2)$
  - Directly adding noise to input would give noise $O(n / \varepsilon^2)$

Other Transforms

Many other transforms can be applied within DP

- (Discrete) Fourier Transform: also bounded sensitivity
  - Often need only a fixed set of coefficients: further reduces S(F)
  - Used for representing data cube counts, time series
- Hierarchical Transforms: binary trees and quadtrees
- Randomized Transforms: sketches and compressed sensing

$$A_8 = \sqrt{\tfrac{1}{8}}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

$$\begin{pmatrix} \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Local Sensitivity

- A common fallacy: using local sensitivity instead of global
  - Global sensitivity $S(F) = \max_{x,x' : |x-x'|=1} \|F(x)-F(x')\|_1$
  - Local sensitivity $S(F,x) = \max_{x' : |x-x'|=1} \|F(x)-F(x')\|_1$
  - These can be very different: local can be much smaller than global
  - It is tempting (but incorrect) to calibrate noise to local sensitivity
- Bad case for local sensitivity: Median
  - Consider $X = [0^n, 0, 0, T^{n-1}]$, $X' = [0^n, 0, T^n]$, $X'' = [0^n, T, T^n]$
  - $S(F,X) = 0$ while $S(F, X') = T$
  - Scale of the noise will reveal exactly which case we are in
- Still, there has to be something better than always using global?
  - Such bad cases seem artificial, rare

Smooth Sensitivity

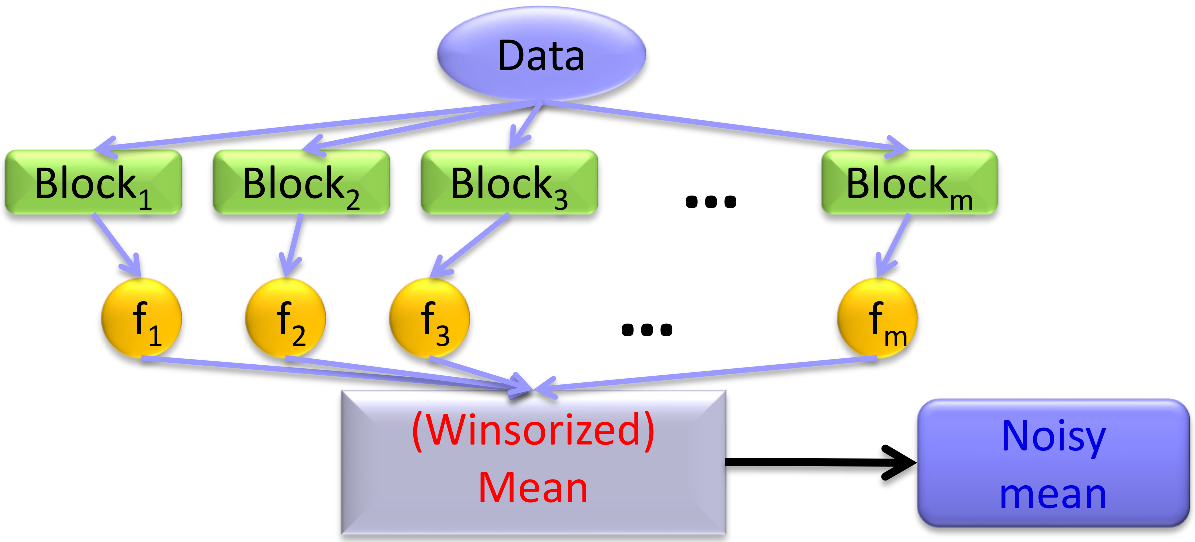
- Previous case was bad because local sensitivity was low, but "close" to a case where local sensitivity was high
- "Smooth sensitivity" combines sensitivity from all neighborhoods (based on parameter β)
  - $SS(F,x) = \max_{o \in O} LS(F,o) \exp(-\beta\ |o - x|)$
  - Contribution of output o is decayed exponentially based on distance of o from x, $|o - x|$
  - Can add Laplace noise scaled by $SS(F,x)$ to obtain (variant of) DP

Smooth Sensitivity: Example

- Consider the median function M over n items again
  - Compute the maximum change in the median for each distance d
  - LS measures when median changes from x_i to x_{i+1}
- So LS at distance d is at most $\max_{0\leq j\leq d} (x_{n/2+j} - x_{n/2+j-d-1})$
  - Largest gap that can be created by inserting/deleting at most d items
- Gives $SS(M,x) = \max_{0\leq d\leq n} \exp(-d\beta) \max_{0\leq j\leq d} (x_{n/2+j} - x_{n/2+j-d-1})$
  - Can compute in time $O(n^2)$
  - Empirically, exponential mechanism seems preferable
  - No generic process for computing smooth sensitivity

Sample-and-aggregate

- Sample-and-aggregate gives a useful template
  - Intuition: sampling is almost DP - can't be sure who is included
  - Break input into moderate number of blocks, m
  - Compute desired function on each block
  - Snap to some range [min, max] and aggregate (e.g. mean)
  - Add Laplace noise scaled by sensitivity (max-min)



Sparse Data

- Suppose we have many (overlapping) queries, most of which have a small answer, but we don't know which
  - We are only interesting in large answers (e.g. frequent itemsets)
  - Two problems: time efficiency, and "privacy efficiency"
- Time efficiency:
  - Don't want to add noise to every single zero-valued query
  - Assume we can materialize all non-zero query answers
  - Count how many are zero
  - Compute probability of noise pushing a zero-query past threshold
  - Sample from Binomial distribution how many to "upgrade"
  - Sample noisy value conditioned on passing threshold

Sparse Data – Privacy Efficiency

- Only want to pay for c queries with that exceed threshold T
  - Assume all queries have sensitivity S
- Compute noisy threshold $T' = T + Lap(2S/\varepsilon)$
- For each query, add noise $Lap(2Sc/\varepsilon)$, only output if above T'
- Result is ε-DP
  - For "suppressed" answers, probability of seeing same output is about the same as if T' was a little higher on neighboring input
  - For released answers, DP follows from Laplace mechanism
- Result is reasonably accurate: with high probability,
  - All suppressed answers are smaller than $T + \alpha$
  - All released answers have error at most α

  for parameter $\alpha(c, 1/\varepsilon, S)$, and at most c query answers $> T - \alpha$

Sparse Vector Technique

- Sparse Vector Technique allows us to save on privacy budget
  - When asking multiple questions, most of which are negative
- Setting: private input vector D, threshold T, budget ε, limit c
  - List of queries Q_i whether $Q_i(D) > T$? Sensitivity of all queries < Δ
- Initialize: count = 0, $\rho = Lap(2\ \Delta/\varepsilon)$
- For each query i
  - Local noise $v_i = Lap(4c\ \Delta\ /\varepsilon)$
  - If $Q_i(D) + v_i \geq T + \rho$ then
    - output "over threshold", increment count, abort if count ≥ c
  - Else, output "under threshold"

Sparse Vector Technique

- Optimization: can choose how to split budget between local noise ν_i and global noise ρ
  - Give more to $\nu_{i,}$ because of the factor of c
- Can easily have a different threshold for each query
- Caution needed:
  multiple incorrect versions of SVT have been published!
  - They neglected to use cutoff limit c, or applied noise incorrectly
- If we know all Q_i in advance, can use EM to sample from them
  - Empirically, more accurate than SVT in practice!

Multiplicative weights [Hardt et al 12]

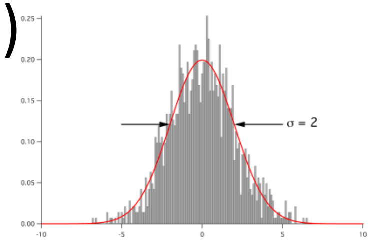
- The idea of "multiplicative weights" widely used in optimization
  - Up-weight 'good' answers, down-weight 'poor' answers
  - Applied to output of DP mechanism
- Set-up:
  - (Private) input, represented as vector D with n entries
  - Q, set of queries over x (matrix)
  - T, bound on number of iterations
  - Output: ε-DP vector A so that $Q(A) \approx Q(D)$

Multiplicative Weights Algorithm

- Initialize vector A_0 to assign uniform weight for each value
- For $i=1$ to T:
  - Exponential Mechanism $(\varepsilon/2T)$ to sample j prop. to $|Q_j(A_i) - Q_j(D)|$
    - Try to find query with large error
  - Laplace Mechanism to estimate $\Delta = (Q_j(A) - Q_j(D)) + Lap(2T/\varepsilon)$
    - Error in the selected query
  - Set $A_i = A_{i-1} . \exp(\Delta\, Q_j(D)/2n)$, normalize so that A_i is a distribution
    - (Noisily) reward good answers, penalize poor answers
- Output $A = \text{average}_i\ nA_i$ — or just output A_n
  - Privacy follows via sequential composition of EM and LM steps
  - Accuracy (should) improve in each iteration, up to log iterations

Differential privacy for data release

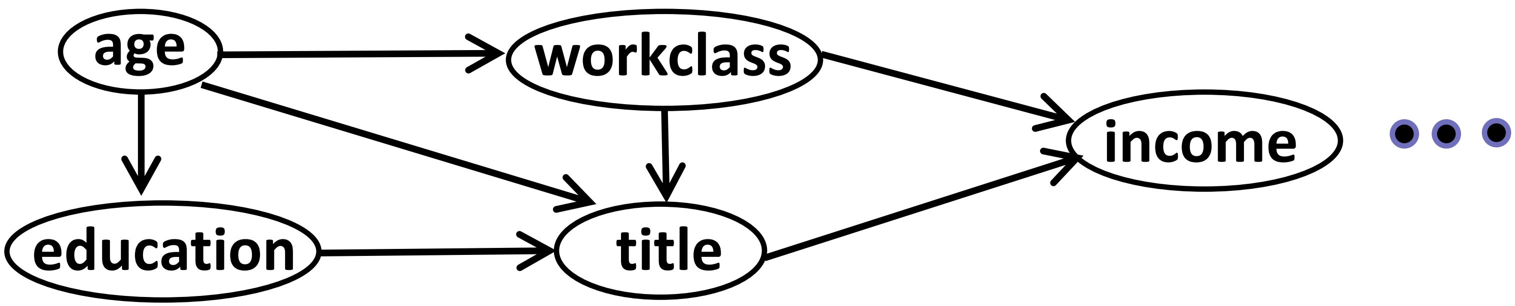
- Differential privacy is an attractive model for data release
  - Achieve a fairly robust statistical guarantee over outputs
- Problem: how to apply to data release where f(x) = x?
  - Trying to use global sensitivity does not work well
- General recipe: find a model for the data (e.g. PSDs)
  - Choose and release the model parameters under DP



- A new tradeoff in picking suitable models
  - Must be robust to privacy noise, as well as fit the data
  - Each parameter should depend only weakly on any input item
  - Need different models for different types of data
- Next 3 (biased) examples of recent work following this outline

Example 1: PrivBayes [Zhang et al. 14]

- Directly materializing tabular data: low signal, high noise
- Use a **Bayesian network** to approximate the full-dimensional distribution by lower-dimensional ones:



$$
\begin{aligned}
\mathbf{Pr[H]} \approx\ & \mathbf{Pr[age]} \cdot \mathbf{Pr[education|age]} \cdot \mathbf{Pr[workclass|age]} \cdot \\
& \mathbf{Pr[title|age,education,workclass]} \cdot \mathbf{Pr[income|workclass,title]} \cdot \\
& \mathbf{Pr[marital\ status|age,income]} \cdots
\end{aligned}
$$

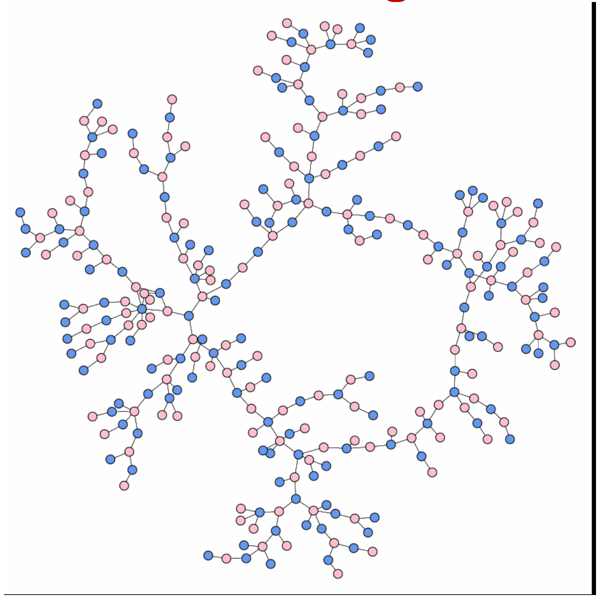
low-dimensional distributions: high signal-to-noise

PrivBayes (SIGMOD14)

- **STEP 1:** Choose a suitable Bayesian Network BN
  - in a differentially private way
  - sample (via exponential mechanism) edges in the network
  - design surrogate quality function with low sensitivity
- **STEP 2:** Compute distributions implied by edges of BN
  - straightforward to do under differential privacy (Laplace)
- **STEP 3:** Generate synthetic data by sampling from the BN
  - post-processing: no privacy issues
- Evaluate utility of synthetic data for variety of different tasks
  - performs well for multiple tasks (classification, regression)

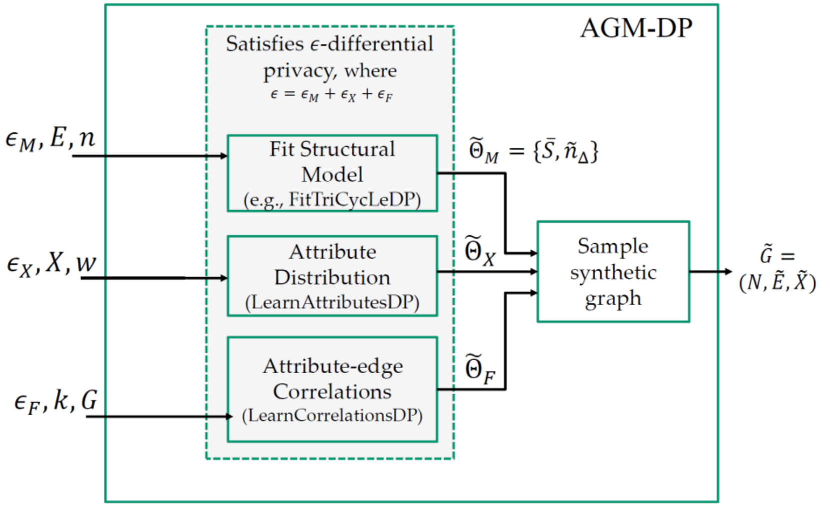
Example 2: Graph Data

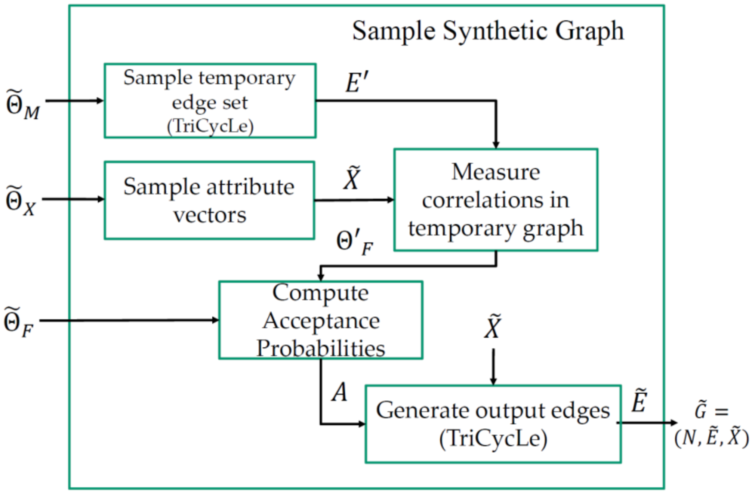
- Releasing graph structured data remains a big challenge
  - Each individual (node) can have a big impact on graph structure
- Most current work focuses on releasing graph statistics
  - Counts of small subgraphs like stars, triangles, cliques etc.
  - These counts are parameters for graph models
  - Sensitivity of these counts is large: one edge can change a lot



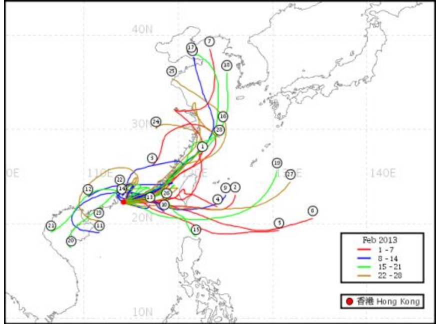
Attributed Graph Data [Jorgensen et al. 16]

- Real graphs (e.g. social networks) have attributes
  - Different types of node, different types of edge
- Define graph models that have attribute distributions
  - Capture real graph structure e.g. number of triangles
- Learn parameters from input graphs (under differential privacy)
- Sample "realistic" graphs from the learned model

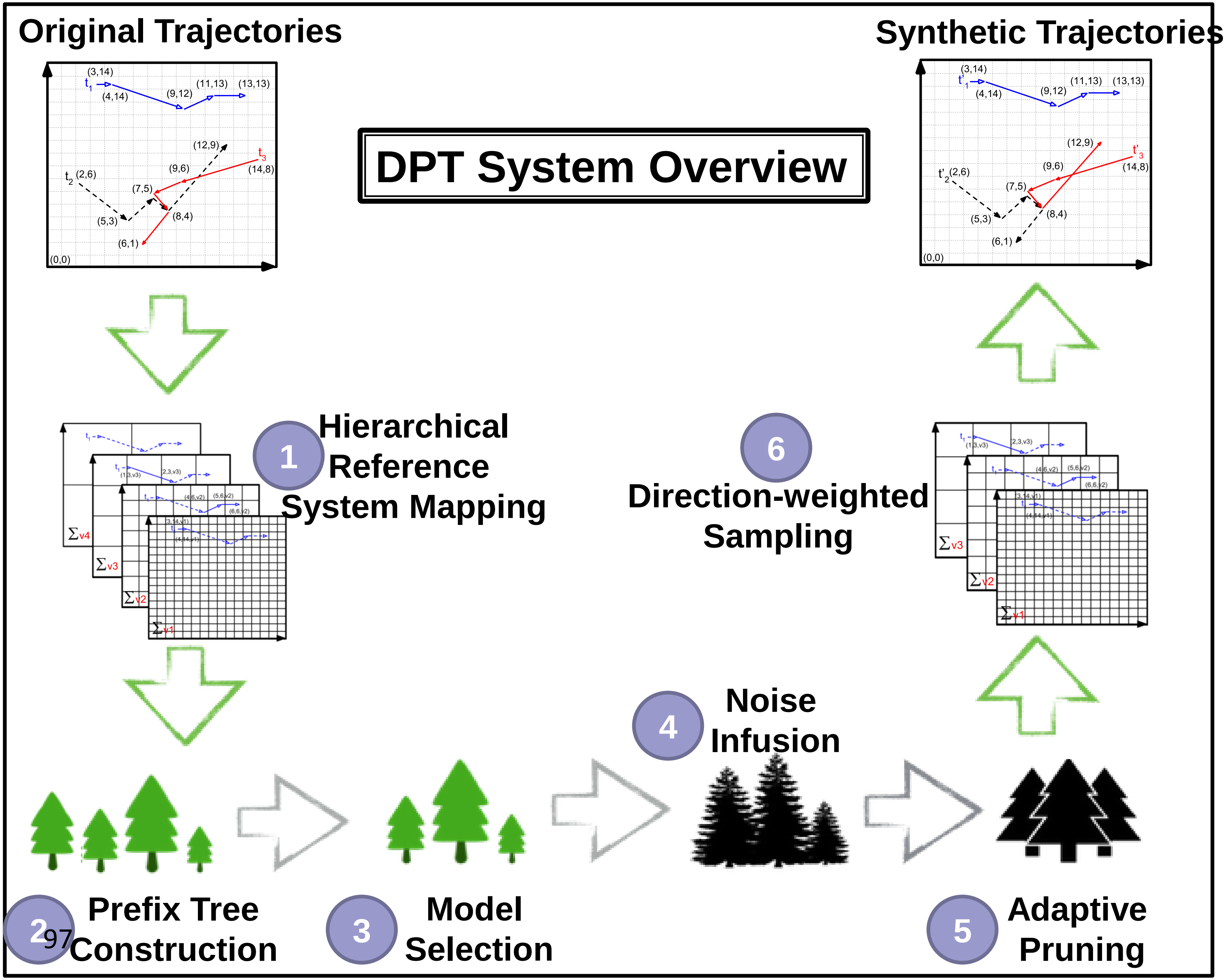




Example 3: Trajectory Data



- More and more location and mobility data available
  - From GPS enabled devices, approximate location from wifi/phone
- Location and movements are very sensitive!
- Location and movements are very identifying!
  - Easy to identify 'work' and 'home' locations from traces
  - 4 random points identify 95% of individuals [Montjoye et al 2013]
- Aim for Differentially Private Trajectories [He et al. 15]
  - Find a model that works for trajectory data
  - Based on Markov models at multiple resolutions


Original Trajectories
Synthetic Trajectories
DPT System Overview
1
Hierarchical Reference System Mapping
6
Direction-weighted Sampling
4
Noise Infusion
2
Prefix Tree Construction
3
Model Selection
5
Adaptive Pruning

Other topics

- Huge amount of work in DP across theory, security, DB…
- Many topics not touched on in this tutorial:
  - Connections to game theory and auction design
  - Mining primitives: regression, clustering, frequent itemsets
  - Efforts in programming languages and systems to support DP
  - Variant definitions: (ε, δ)-DP, other privacy/adversary models
  - Lower bounds for privacy (what is not possible)
  - Applications to graph data (social networks), mobility data etc.
  - Applications to machine learning: classifiers that don't leak
  - Privacy over data streams: pan-privacy and continual observation

State of Anonymization

- Data privacy and anonymization is a subject of ongoing research today
- Many unresolved challenges:
  - How can a social network release a substantial data set without revealing private connections between users?
  - How can a video website release information on viewing patterns without disclosing who watched what?
  - How can a search engine release information on search queries without revealing who searched for what?
  - How to release private information efficiently over large scale data?

Concluding Remarks

- Differential privacy can be applied effectively for data release
- Care is still needed to ensure that release is allowable
  - Can’t just apply DP and forget it: must analyze whether data release provides sufficient privacy for data subjects
- Many open problems remain:
  - Transition these techniques to tools for data release
  - Want data in same form as input: private synthetic data?
  - Allow joining anonymized data sets accurately
  - Obtain alternate (workable) privacy definitions

Thank you!

References – Basic Building Blocks

- Differential privacy, Laplace Mechanism and Sensitivity:
  - Calibrating Noise to Sensitivity in Private Data Analysis. Cynthia Dwork, Frank McSherry, Kobbi Nissim, Adam Smith. Theory of Cryptography Conference (TCC), 2006.
  - Differential Privacy. Cynthia Dwork, ICALP 2006
- Geometric Mechanism
  - Universally utility-maximizing privacy mechanisms. Arpita Ghosh, Tim Roughgarden, Mukund Sundararajan. STOC 2009
- Sequential and Parallel Composition, Median Example
  - Privacy integrated queries: an extensible platform for privacy-preserving data analysis. Frank McSherry. SIGMOD 2009.
- Exponential Mechanism
  - Mechanism Design via Differential Privacy. Frank McSherry and Kunal Talwar. FOCS, 2007

References – Applications & Transforms

- Spatial Data Application
  - Differentially private spatial decompositions. Graham Cormode, Magda Procopiuc, Entong Shen, Divesh Srivastava, and Ting Yu. In International Conference on Data Engineering (ICDE), 2012
- Data Transforms
  - Differential privacy via wavelet transforms. Xiaokui Xiao, Guozhang Wang, Johannes Gehrke, ICDE 2010
  - Privacy, accuracy, and consistency too: a holistic solution to contingency table release. Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank Mcsherry, Kunal Talwar. PODS 2007
  - Differentially Private Aggregation of Distributed Time-Series with Transformation and Encryption. Vibhor Rastogi and Suman Nath, SIGMOD 2010

References – Advanced Mechanisms

- Smooth Sensitivity, Sample and Aggregate
  - Smooth Sensitivity and Sampling in Private Data Analysis. Kobbi Nissim, Sofya Raskhodnikova and Adam Smith. STOC 07
  - GUPT: Privacy Preserving Data Analysis Made Easy. Prashanth Mohan, Abhradeep Thakurta, Elaine Shi, Dawn Song, David Culler. SIGMOD 2012
- Sparse Data Processing
  - Differentially Private Summaries for Sparse Data. Graham Cormode, Magda Procopiuc, Divesh Srivastava, and Thanh Tran. ICDT 2012
  - Understanding the Sparse Vector Technique for Differential Privacy. Min Lyu, Dong Su, Ninghui Li. PVLDB 10(6): 637-648 (2017)
- Multiplicative Weights
  - A Multiplicative Weights Mechanism for Privacy Preserving Data Analysis. Moritz Hardt and Guy Rothblum. FOCS 2010.
  - A simple and practical algorithm for differentially private data release. Moritz Hardt, Katrina Ligett, Frank McSherry. NIPS 2012

References – DP for Data Release

- PrivBayes: Private Data Release via Bayesian Networks. Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, Xiaokui Xiao. ACM Trans. Database Syst. 42(4): 25:1-25:41, 2017
- Publishing Attributed Social Graphs with Formal Privacy Guarantees. Zach Jorgensen, Ting Yu, Graham Cormode. SIGMOD 2016
- Unique in the Crowd: The privacy bounds of human mobility. Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen & Vincent D. Blondel. Scientific Reports 3, Article number 1376, 2013
- Xi He, Graham Cormode, Ashwin Machanavajjhala, Cecilia M. Procopiuc, Divesh Srivastava. DPT: Differentially Private Trajectory Synthesis Using Hierarchical Reference Systems. VLDB 2015