Data-driven concerns in privacy

Graham Cormode
graham@cormode.org

Joint work with
Magda Procopiuc (AT&T)
Entong Shen (NCSU)
Divesh Srivastava (AT&T)
Thanh Tran (UMass Amherst)
Grigory Yaroslavtsev (Penn State)
Ting Yu (NCSU)
Outline

♦ Anonymization and Privacy models
♦ Non-uniformity of data
♦ Optimizing linear queries
♦ Predictability in data
The anonymization scenario
Data-driven privacy

♦ Much interest in private data release
  – **Practical**: release of AOL, Netflix data etc.
  – **Research**: hundreds of papers

♦ In practice, many data-driven concerns arise:
  – Efficiency / practicality of algorithms as data scales
  – How to interpret privacy guarantees
  – Handling of common data features, e.g. sparsity
  – Ability to optimize for known query workload
  – Usability of output for general processing

♦ **This talk**: outline some efforts to address these issues
Differential Privacy [Dwork 06]

♦ **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
  – Simple recipe for some data types e.g. numeric answers
  – Simple rules allow us to reason about composition of results
  – More complex for arbitrary data (exponential mechanism)

♦ Adopted and used by several organizations:
  – US Census, Common Data Project, Facebook (?)
The output distribution of a differentially private algorithm changes very little whether or not any individual’s data is included in the input – so you should contribute your data.

A randomized algorithm $K$ satisfies $\varepsilon$-differential privacy if:
Given any pair of neighboring data sets, $D_1$ and $D_2$, and $S$ in $\text{Range}(K)$:

$$\Pr[K(D_1) = S] \leq e^\varepsilon \Pr[K(D_2) = S]$$
Achieving $\varepsilon$-Differential Privacy

(Global) Sensitivity of publishing:

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

E.g., count individuals satisfying property $P$: one individual changing info affects answer by at most 1; hence $s = 1$

For every value that is output:

- Add Laplacian noise, $\text{Lap}(\varepsilon/s)$:
- Or Geometric noise for discrete case:

Simple rules for composition of differentially private outputs:

Given output $O_1$ that is $\varepsilon_1$ private and $O_2$ that is $\varepsilon_2$ private

- (Sequential composition) If inputs overlap, result is $\varepsilon_1 + \varepsilon_2$ private
- (Parallel composition) If inputs disjoint, result is $\max(\varepsilon_1, \varepsilon_2)$ private
Outline

♦ Anonymization and Privacy models
♦ **Non-uniformity of data**
♦ Optimizing linear queries
♦ Predictability in data
Sparse Spatial Data [ICDE 2012]

♦ 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

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

![quadtree](image1.png) ![kd-tree](image2.png)
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:
    - 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 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 $\varepsilon$
Privacy budget allocation

♦ How to set an $\varepsilon_i$ 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
  - 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
Experimental study

- 1.63 million coordinates from US TIGER/Line dataset
  - Road intersections of US States
- Queries of different shapes, e.g. square, skinny
- Measured median relative error of 600 queries for each shape
Experimental study

- Effectiveness of geometric budget and post-processing

- Relative error reduced by up to an order of magnitude
- Most effective when limited privacy budget
Outline

♦ Anonymization and Privacy models
♦ Non-uniformity of data
♦ **Optimizing linear queries**
♦ Predictability in data
Linear queries capture many common cases for data release
- Data is represented as a vector $x$
- Want to release answers to linear combinations of entries of $x$
- E.g. contingency tables in statistics
- Model queries as matrix $Q$, want to know $y = Qx$

\[
Q = \begin{pmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
\end{pmatrix}
\]

\[
x = \begin{pmatrix}
3 \\
5 \\
7 \\
0 \\
1 \\
4 \\
9 \\
2 \\
\end{pmatrix}
\]
Answering Linear Queries

♦ Basic approach:
  – Answer each query in \( Q \) directly, and add uniform noise

♦ Basic approach is suboptimal
  – Especially when some queries overlap and others are disjoint

♦ Several opportunities for optimization:
  – Can assign different scales of noise to different queries
  – Can combine results to improve accuracy
  – Can ask different queries, and recombine to answer \( Q \)

\[
Q = \begin{pmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
\end{pmatrix}
\]
The Strategy/Recovery Approach

♦ Pick a strategy matrix $S$
  - Compute $z = Sx + v$  ➔ noise vector
    ➔ strategy on data
  - Find $R$ so that $Q = RS$
  - Return $y = Rz = Qx + Rv$ as the set of answers
  - Measure accuracy based on $\text{var}(y) = \text{var}(Rv)$

♦ Common strategies used in prior work:

  I: Identity Matrix
  C: Selected Marginals
  Q: Query Matrix
  H: Haar Wavelets
  F: Fourier Matrix
  P: Random projections
Step 1: Error Minimization

- Given $Q, R, S, \varepsilon$ want to find a set of values $\{\varepsilon_i\}$
  - Noise vector $v$ has noise in entry $i$ with variance $1/\varepsilon_i^2$
- Yields an optimization problem of the form:
  - Minimize $\sum_i b_i / \varepsilon_i^2$ (minimize variance)
  - Subject to $\sum_i |S_{i,j}| \varepsilon_i \leq \varepsilon$ (guarantee $\varepsilon$ differential privacy)
- The optimization is convex, can solve via interior point methods
  - Costly when $S$ is large
  - We seek an efficient closed form for common strategies
Grouping Approach

- We observe that many strategies $S$ can be broken into groups that behave in a symmetrical way
  - Rows in a group are disjoint (have zero inner product)
  - Non-zero values in group $i$ have same magnitude $C_i$
- All common strategies meet this grouping condition
  - Identity ($I$), Fourier ($F$), Marginals ($C$), Projections ($P$), Wavelets ($H$)
- Simplifies the optimization:
  - A single constraint over the $\varepsilon_i$’s
  - New constraint: $\sum_{\text{Groups } i} C_i \varepsilon_i = \varepsilon$
  - Closed form solution via Lagrangian

\[
\begin{pmatrix}
\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\
\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\
\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}}
\end{pmatrix}
\]
Step 2: Optimal Recovery Matrix

- Given Q, S, {ε_i}, find R so that Q = RS
  - Minimize the variance \( \text{Var}(Rz) = \text{Var}(R(Sx + v)) = \text{Var}(Rv) \)
- Find an optimal solution by adapting Least Squares method
- This finds \( x' \) as an estimate of \( x \) given \( z = Sx + v \)
  - Define \( \Sigma = \text{Cov}(z) = \text{diag}(2/\varepsilon_i^2) \) and \( U = \Sigma^{-1/2} S \)
  - OLS solution is \( x' = (U^T U)^{-1} U^T \Sigma^{-1/2} z \)
- Then \( R = Q(S^T \Sigma^{-1} S)^{-1} S^T \Sigma^{-1} \)
- Result: \( y = Rz = Qx' \) is consistent—corresponds to queries on \( x' \)
  - \( R \) minimizes the variance
  - Special case: \( S \) is orthonormal basis \( (S^T = S^{-1}) \) then \( R = QS^T \)
Overall Process

- **Ideal version:** given query matrix $Q$, compute strategy $S$, recovery $R$ and noise budget $\{\epsilon_i\}$ to minimize $\text{Var}(y)$
  - Not practical: sets up a rank-constrained SDP
  - Follow the 2-step process instead
- Given query matrix $Q$ decomposed into $Q=(RS)$, compute optimal noise budgets $\{\epsilon_i\}$ to minimize $\text{Var}(y)$ (Step 1)
- Given query matrix $Q$, strategy $S$ and noise budgets $\{\epsilon_i\}$, compute new recovery matrix $R$ to minimize $\text{Var}(y)$ (Step 2)
- Fairly fast (matrix multiplications and inversions)
  - Faster when $S$ is e.g. Fourier, since can use FFT
Experimental Study

- Used two real data sets:
  - ADULT data – census data on 32K individuals
  - NLTCS data – binary data on 21K individuals

- Tried a variety of query workloads $Q$ over these
  - Based on low-degree $k$-way marginals

- Compared the original and optimized strategies for:
  - Original queries, $Q / Q^+$
  - Fourier strategy $F/F^+$ [Barak et al. 07]
  - Clustered sets of marginals $C/C^+$ [Bing et al. 11]
  - Identity basis $I$
Experimental Results

**ADULT, 1- and 2-way marginals**

![Graph showing relative error vs. ε for ADULT data](image)

**NLTCS, 2- and 3-way marginals**

![Graph showing relative error vs. ε for NLTCS data](image)

- Optimized error gives constant factor improvement
- Time cost for the optimization is negligible on this data
Outline

♦ Anonymization and Privacy models
♦ Non-uniformity of data
♦ Optimizing linear queries
♦ Predictability in data
Revisiting the privacy definition [KDD 2011]

- Differential privacy guarantees that what I learn about an individual from the released data is about the same whether or not they are in the data.
- So I can’t learn much about an individual from the released data, right?
- WRONG!
  - Will show how differentially private output can still allow us to draw accurate conclusions about individuals.
Use Machine Learning to Perform Inference

- **Key idea:** build an accurate classifier under DP
- **Data model:** target ("sensitive") attribute \( s \in SA \)
  - Think disease status, salary band, etc.
- "Observable" attributes \( t_1, t_2 \ldots t_m \)
  - Think age, gender, postal code, height etc.
- **Goal:** build a classifier that given \((t_1, t_2, \ldots t_m)_i\) predicts \( s_i \)
  - An accurate classifier reveals the private information
Building the Classifier

♦ Build a naïve Bayes classifier for \( s \):
  – Prediction is \( s' = \arg \max_{s \in SA} \Pr[s] \prod_{j=1}^{m} \Pr[t_j | s] \)

♦ Parameters are the marginal distributions
  \( \Pr[t_i | s] = \Pr[t_i \cap s]/\Pr[s] \approx |\{r \in T : t_i = t_i \cap r_s = s\}|/|\{ r \in T : r_s = s\}| \)

♦ Just need the counts \( \forall s \in SA, i, v \in T_i |\{r \in T : t_i = v \cap r_s = s\}| \)
  – Can obtain “noisy” versions of these under differential privacy
  – Noise is small compared to most counts

♦ **Minor corrections**: add 1 to counts (Laplacian correction), round up to 1 if negative due to noise
Experimental Study

Tested accuracy of predicting

- ‘occupation’ (14 options) in UCI Adult data
- ‘income’ (9 options) in UCI Internet-usage data

Clear improvement in accuracy over baseline methods

- E.g. just predict most common attribute value
High Confidence Results

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

Why does this work?
- The classifier is based on correlations between the observable attributes and the target attribute
- These are population statistics: they arise from the coarse behavior of the whole population
- One individual has almost no influence on them
- More directly, the noise added to mask an individual does not substantially change them until the noise is very large

Differential privacy is behaving as advertised
- What we learn about the individual really is the same whether they are there or not
Enabling Disclosure

♦ Should we be worried? Correlations are inherent in the data?
  – An ‘attacker’ might never be able to collect such data
  – But almost ‘for free’ they can use released “privatized” statistics and potentially compromise an individual’s privacy

♦ “If the release of the statistic $S$ makes it possible to determine the (microdata) value more accurately than without access to $S$, a disclosure has taken place” – T. Dalenius, 1977
  – DP does not prevent disclosure, even when the attacker has no other information
  – Attempts to remove correlation in data may destroy utility!
  – Urges caution when releasing data under any privacy definition
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!