CS573 Data Privacy and Security

Differential Privacy – tabular data and range queries

Li Xiong
Outline

• Tabular data and histogram/range queries
• Algorithms for low dimensional data
• Algorithms for high dimensional data
Example: cohort discovery from medical records

- Histograms
- Cohort discovery: range queries
  - Select COUNT(*) from D
    Where A1 in I1 and A2 in I2 and ... and Am in Im.
Example: statistical agencies: data publishing

- A **marginal** over attributes $A_1, ..., A_k$ reports count for each combination of attribute values.
  - aka cube, contingency table
  - E.g. 2-way marginal on EmploymentStatus and Gender
- U.S. Census Bureau statistics can typically be derived from $k$-way marginal over different combinations of available attributes
- **Hundreds** of marginals released

<table>
<thead>
<tr>
<th>Subject</th>
<th>Estimate</th>
<th>Margin of Error</th>
<th>Percent</th>
<th>Percent Margin of Error</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EMPLOYMENT STATUS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Population 18 years and over</td>
<td>6.678</td>
<td>±/361</td>
<td>5.878</td>
<td>(X)</td>
</tr>
<tr>
<td>In labor force</td>
<td>2.715</td>
<td>±/223</td>
<td>47.8%</td>
<td>±/3.7</td>
</tr>
<tr>
<td>Civilian labor force</td>
<td>2.715</td>
<td>±/223</td>
<td>47.8%</td>
<td>±/3.7</td>
</tr>
<tr>
<td>Employed</td>
<td>2.529</td>
<td>±/228</td>
<td>44.6%</td>
<td>±/3.6</td>
</tr>
<tr>
<td>Unemployed</td>
<td>1.050</td>
<td>±/402</td>
<td>3.5%</td>
<td>±/1.7</td>
</tr>
<tr>
<td>Armed Forces</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not in labor force</td>
<td>2.901</td>
<td>±/266</td>
<td>52.2%</td>
<td>±/3.7</td>
</tr>
<tr>
<td>Civilian labor force</td>
<td>2.715</td>
<td>±/223</td>
<td>2.715</td>
<td>(X)</td>
</tr>
<tr>
<td>Percent Unemployed</td>
<td></td>
<td></td>
<td>6.0%</td>
<td>±/3.4</td>
</tr>
<tr>
<td>Female 16 years and over</td>
<td>2.021</td>
<td>±/196</td>
<td>2.021</td>
<td>(X)</td>
</tr>
<tr>
<td>In labor force</td>
<td>1.312</td>
<td>±/140</td>
<td>44.4%</td>
<td>±/4.5</td>
</tr>
<tr>
<td>Civilian labor force</td>
<td>1.312</td>
<td>±/140</td>
<td>44.4%</td>
<td>±/4.5</td>
</tr>
<tr>
<td>Employed</td>
<td>1.245</td>
<td>±/135</td>
<td>42.6%</td>
<td>±/4.3</td>
</tr>
<tr>
<td>Own children under 6 years</td>
<td>325</td>
<td>±/117</td>
<td>325</td>
<td>(X)</td>
</tr>
<tr>
<td>All parents in family in labor force</td>
<td>241</td>
<td>±/99</td>
<td>74.2%</td>
<td>±/17.3</td>
</tr>
<tr>
<td>Own children 6 to 17 years</td>
<td>475</td>
<td>±/102</td>
<td>475</td>
<td>(X)</td>
</tr>
<tr>
<td>All parents in family in labor force</td>
<td>380</td>
<td>±/65</td>
<td>81.7%</td>
<td>±/8.6</td>
</tr>
<tr>
<td><strong>COMMUTING TO WORK</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workers 10 years and over</td>
<td>2.449</td>
<td>±/271</td>
<td>2.449</td>
<td>(X)</td>
</tr>
<tr>
<td>Car, truck, or van -- drove alone</td>
<td>1.519</td>
<td>±/176</td>
<td>92.5%</td>
<td>±/5.2</td>
</tr>
<tr>
<td>Car, truck, or van -- carpooled</td>
<td>116</td>
<td>±/45</td>
<td>4.7%</td>
<td>±/2.9</td>
</tr>
<tr>
<td>Public transportation (excluding taxi)</td>
<td>17</td>
<td>±/18</td>
<td>0.7%</td>
<td>±/0.6</td>
</tr>
<tr>
<td>Walked</td>
<td>531</td>
<td>±/116</td>
<td>21.7%</td>
<td>±/4.3</td>
</tr>
<tr>
<td>Other means</td>
<td>132</td>
<td>±/56</td>
<td>5.4%</td>
<td>±/2.4</td>
</tr>
<tr>
<td>Worked at home</td>
<td>135</td>
<td>±/44</td>
<td>5.6%</td>
<td>±/2.5</td>
</tr>
<tr>
<td>Mean travel time to work</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OCCUPATION</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Civilian employed population 16 years and over</td>
<td>2.529</td>
<td>±/258</td>
<td>2.529</td>
<td>(X)</td>
</tr>
</tbody>
</table>

https://factfinder.census.gov/
Example: range queries over spatial data

**Input: sensitive data** $D$

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Latitude</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>39.98105</td>
<td>116.30142</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>39.9424</td>
<td>116.30587</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>39.93691</td>
<td>116.33438</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>39.94354</td>
<td>116.33532</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**BeijingTaxi dataset [1]:**
4,268,780 records of (lat,lon) pairs of taxi pickup locations in Beijing, China in 1 month.

**Input: range query workload** $W$

Shown is workload of **3 range queries**

**Task:** compute answers to workload $W$ over private input $D$

Problem variant: offline vs. online

- **Offline (batch):**
  - Entire $W$ given as input, answers computed in **batch**

- **Online (adaptive):**
  - $W$ is sequence $q_1, q_2, \ldots$ that arrives online
  - **Adaptive:** analyst’s choice for $q_i$ can depend on answers $a_1, \ldots, a_{i-1}$
Important aspects of problem: Data and query complexity

• Data complexity
  – Dimensionality: number of attributes
  – Domain size: number of distinct attribute combinations
  – Many techniques specialized for low dimensional data

• Query complexity
  – Given query workload vs. no query workload
  – Classes of queries: histograms, count queries, linear queries (sum, average), median …
Solution variants: query answers vs. synthetic data

Two high-level approaches to solving problem

1. **Direct:**
   - Output of the algorithm is list of query answers

2. **Synthetic data:**
   - Algorithm constructs a *synthetic dataset* $D'$, which can be queried directly by analyst
   - Analyst can pose additional queries on $D'$ (though answers may not be accurate)
Synthetic Data: Categories of Methods

- Nonparametric methods – release empirical distributions, i.e. histograms with differential privacy

- Parametric and semi-parametric methods – learn parameters of a distribution with differential privacy
Outline

• Tabular data and histogram/range queries

• Algorithms for low dimensional data
  – Baseline
  – Partitioning algorithms: kd tree, quad tree, …
  – Transformation: Wavelet, Fourier Transform, …
  – An evaluation framework: DPBench

• Algorithms for high dimensional data
Baseline algorithm: IDENTITY

1. Discretize attribute domain into cells
2. Add noise to cell counts (Laplace mechanism) – unit histogram
3. Use noisy counts to either...
   1. Answer queries directly (assume distribution is uniform within cell)
   2. Generate synthetic data (derive distribution from counts and sample)
Baseline algorithm: IDENTITY

1. Discretize attribute domain into cells
2. Add noise to cell counts (Laplace mechanism) – unit histogram
3. Use noisy counts to either...
   1. Answer queries directly (assume distribution is uniform within cell)
   2. Generate synthetic data (derive distribution from counts and sample)

Limitations
- Granularity of discretization
  - Coarse: detail lost
  - Fine: noise overwhelms signal
- Noise accumulates: squared error grows linearly with range
Empirical benchmarks

• An evaluation framework for standardized evaluation of privacy algorithms for range queries (over 1 and 2D)
• Demo: https://www.dpcomp.org/tutorial/introduction
Data-Dependent Partitioning

- Domain-based (data-independent) partitioning does not work very well
  - Equi-width: equal bucket range
  - **Uniformity** assumption

- Data-driven partitioning
  - V-optimal: with the least frequency variance
  - Intuition: highest uniformity within each bucket
  - How to do it with differential privacy?
Histograms (review)

- Divide data into buckets and store average (sum) for each bucket
- Partitioning rules:
  - Equi-width: equal bucket range
  - Equi-depth: equal frequency
  - V-optimal: with the least *frequency variance*
An Early Attempt: DPCube [SDM 2010, ICDE 2012 demo]

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Income</th>
<th>HIV+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frank</td>
<td>42</td>
<td>30K</td>
<td>Y</td>
</tr>
<tr>
<td>Bob</td>
<td>31</td>
<td>60K</td>
<td>Y</td>
</tr>
<tr>
<td>Mary</td>
<td>28</td>
<td>20K</td>
<td>Y</td>
</tr>
</tbody>
</table>

- 1. Compute unit histogram with differential privacy
- 2. kd-tree partitioning
- 3. Compute merged bin counts with differential privacy
kd-tree based partitioning

- Choose dimension and splitting point to split (minimize variance)
- Repeat until:
  - Count of this node less than threshold
  - Variance or entropy of this node less than threshold
**DPCube** [SecureDM 2010, ICDE 2012 demo]

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Income</th>
<th>HIV+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frank</td>
<td>42</td>
<td>30K</td>
<td>Y</td>
</tr>
<tr>
<td>Bob</td>
<td>31</td>
<td>60K</td>
<td>Y</td>
</tr>
<tr>
<td>Mary</td>
<td>28</td>
<td>20K</td>
<td>Y</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Original Records**

**Sequential composition**

**Limitations:**
- DP unit histogram very noisy
- Affects the accuracy of partitioning

**DP Interface**

**DP unit Histogram**

**Multi-dimensional partitioning**

**DP V-optimal Histogram**
A Later Improvement: Private Spatial decompositions [CPSSY 12]

- **Approach**: (top down) partitioning with differential privacy
  - Quad tree and hybrid/kd-tree

![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 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 – exponential mechanism with utility function $\text{utility}(x) = \text{rank}(x) - \text{rank}(\text{median})$
  - 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 Private Spatial Decompositions – privacy budget allocation

- Budget is split between medians and counts at each node
  - 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 $\varepsilon$
  - Optimal budget allocation
  - Post processing with consistency check
Data-dependent partitioning

• Heuristics based methods
  – Kd-tree, quad-tree

• Optimal methods
  – V-optimal histogram (1D or 2D)
Data-aware/Workload-Aware Mechanism [LHMY14]

Figure 1: Overview and example execution for the DAWA mechanism.

- **Step 1:** dynamic programming based methods for optimal partitioning
- **Step 2:** matrix mechanism for optimal noise given a query workload
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
Empirical benchmarks

• An evaluation framework for standardized evaluation of privacy algorithms for range queries (over 1 and 2D)
• Demo: https://www.dpcomp.org/tutorial/introduction
• Key findings:
  – Scale/size and shape of data significantly affect algorithm error
  – In a “high signal” regime (high scale, high epsilon), simpler data independent methods such as IDENTITY works well
  – In a “low signal” regime (low scale, low epsilon), data-dependent algorithm should be considered but no guarantees
  – While no algorithm universally dominates across settings, DAWA is a competitive choice on most datasets

[HMMCZ16]
Programming Assignment and Competition: Laplace mechanism for Range queries

• Required:
  – Implement the baseline IDENTITY histogram algorithm
  – Evaluate accuracy for random set of range queries

• Optional:
  – Optimizations and enhancement

• Competition
Outline

• Tabular data and histogram/range queries

• Algorithms for low dimensional data
  – Baseline
  – Partitioning algorithms: kd tree, quad tree, …
  – An evaluation framework: DPBench

• Algorithms for high dimensional data
  – Copula functions [LXJ14]
  – Bayesian networks [ZCPSX14]
Traditional Approaches

Parametric methods

Fit the data to a distribution, make inferences about parameters

e.g. PrivacyOnTheMap

Non-parametric methods

Learn empirical distribution through histograms

e.g. PSD, Privelet, FP, P-HP
Semi-parametric modeling using Copula functions

Semi-parametric methods

Haoran Li, Li Xiong, Xiaoqian Jiang. Differentially Private Synthesization of Multi-Dimensional Data using Copula Functions, EDBT 2014
Gaussian copula vs. Gaussian distribution

Gaussian copula: models the dependence with arbitrary margins

Gaussian distribution: models the joint distribution
DPCopula–MLE

**Original data set**

<table>
<thead>
<tr>
<th>Age</th>
<th>Hours/week</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>64</td>
<td>30K</td>
</tr>
<tr>
<td>31</td>
<td>82</td>
<td>60K</td>
</tr>
<tr>
<td>28</td>
<td>40</td>
<td>20K</td>
</tr>
<tr>
<td>43</td>
<td>36</td>
<td>80K</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Step 1: Computing DP marginal Histograms**

**DP marginal histograms**

**Step 2: Computing DP correlation matrix through DP MLE (Maximum Likelihood Estimation)**

\[ \tilde{P} = \begin{bmatrix} 1 & 0.053 & 0.108 \\ 0.053 & 1 & 0.132 \\ 0.108 & 0.132 & 1 \end{bmatrix} \]

**DP correlation matrix**

**Step 3: Sampling DP synthetic data**

**DP synthetic data set**

<table>
<thead>
<tr>
<th>Age</th>
<th>Hours/week</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>64</td>
<td>30K</td>
</tr>
<tr>
<td>31</td>
<td>82</td>
<td>60K</td>
</tr>
<tr>
<td>28</td>
<td>40</td>
<td>20K</td>
</tr>
<tr>
<td>43</td>
<td>36</td>
<td>80K</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**DP dependence structure**
DPCopula-hybrid

Overview

<table>
<thead>
<tr>
<th>Age</th>
<th>Hours /week</th>
<th>Income</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>64</td>
<td>30K</td>
<td>F</td>
</tr>
<tr>
<td>31</td>
<td>82</td>
<td>60K</td>
<td>M</td>
</tr>
<tr>
<td>28</td>
<td>40</td>
<td>20K</td>
<td>F</td>
</tr>
<tr>
<td>43</td>
<td>36</td>
<td>80K</td>
<td>M</td>
</tr>
</tbody>
</table>

Gender = F

<table>
<thead>
<tr>
<th>Age</th>
<th>Hours /week</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>64</td>
<td>30K</td>
</tr>
<tr>
<td>28</td>
<td>40</td>
<td>20K</td>
</tr>
</tbody>
</table>

Gender = M

<table>
<thead>
<tr>
<th>Age</th>
<th>Hours /week</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>82</td>
<td>60K</td>
</tr>
<tr>
<td>43</td>
<td>36</td>
<td>80K</td>
</tr>
</tbody>
</table>

DPCopula

\[ \tilde{n}_1 = n_1 + \text{Lap}(1/\varepsilon_0) \]

\[ \tilde{n}_1 \]

\[ \tilde{n}_2 = n_2 + \text{Lap}(1/\varepsilon_0) \]

\[ \tilde{n}_2 \]
Datasets
- US Census data: 4 attributes, 100,000 records
- Brazil data: 8 attributes, 188,846 records
- Synthetic data

Comparison:
- PSD, Privelet+, FP, P-HP

Metrics:
Random range-count queries with random query predicates covering all attributes

\[
\text{Select } \text{COUNT}(*) \text{ from } D \\
\text{Where } A_1 \in I_1 \text{ and } A_2 \in I_2 \text{ and... and } A_m \in I_m \\
\text{For each attribute } A_i, I_i \text{ is a random interval generated from} \\
\text{the domain of } A_i.
\]

Relative error:
\[
RE(q) = \frac{|A_{\text{noisy}}(q) - A_{\text{act}}(q)|}{\max\{A_{\text{act}}(q), s\}}
\]

Absolute error:
\[
ABS(q) = |A_{\text{noisy}}(q) - A_{\text{act}}(q)|
\]
Experiment: Comparison on real datasets

- Query accuracy vs. differential privacy budget

(a) US-4D
(b) Brazil-8D
(c) Brazil-8D (absolute error)
DPCopula Limitations

- Gaussian dependence assumption
- Pair-wise attribute correlation does not scale with high dimensions
- Works well for continuous data or attributes with large domains
Outline

- Tabular data and histogram/range queries
- **Algorithms for low dimensional data**
  - Baseline
  - Partitioning algorithms: kd tree, quad tree, ...
  - An evaluation framework: DPBench
- **Algorithms for high dimensional data**
  - Copula functions [LXJ14]
  - Bayesian networks [ZCPSX14]
PrivBayes

sensitive database $D$

convert

full-dim tuple distribution

+ noise

noisy distribution

sample

synthetic database $D^*$

approximate

convert

a set of low-dim distributions

+ noise

noisy low-dim distributions

sample
Bayesian network example

\[ P(B) \]

\[
\begin{array}{cc}
t & f \\
0.001 & 0.999 \\
\end{array}
\]

Burglary

\[ P(E) \]

\[
\begin{array}{cc}
t & f \\
0.001 & 0.999 \\
\end{array}
\]

Earthquake

\[ P(A / B, E) \]

\[
\begin{array}{ccc}
B & E & t & f \\
t & t & 0.95 & 0.05 \\
t & f & 0.94 & 0.06 \\
f & t & 0.29 & 0.71 \\
f & f & 0.001 & 0.999 \\
\end{array}
\]

JohnCalls

\[ P(J | A) \]

\[
\begin{array}{cc}
A & t & f \\
t & 0.9 & 0.1 \\
f & 0.05 & 0.95 \\
\end{array}
\]

MaryCalls

\[ P(M | A) \]

\[
\begin{array}{cc}
A & t & f \\
t & 0.7 & 0.3 \\
f & 0.01 & 0.99 \\
\end{array}
\]
A 5-dimensional database:

\[
\begin{align*}
\text{Pr}[age] & \quad \text{Pr}[work | age] \\
\text{age} & \quad \text{workclass} \\
\text{Pr}[edu | age] & \quad \text{Pr}[title | work] \\
\text{education} & \quad \text{title} \\
\text{Pr}[income | work] & \\
\text{income}
\end{align*}
\]
A 5-dimensional database:

\[ \Pr[*] \approx \Pr[age] \cdot \Pr[work | age] \cdot \Pr[edu | age] \cdot \Pr[title | work] \cdot \Pr[income | work] \]
Outline of the Algorithm

- **STEP 1**: Choose a suitable Bayesian network $\mathcal{N}$
  - must in a differentially private way

- **STEP 2**: Compute conditional distributions implied by $\mathcal{N}$
  - straightforward to do under differential privacy
  - inject noise – Laplace mechanism

- **STEP 3**: Generate synthetic data by sampling from $\mathcal{N}$
  - post-processing: no privacy issues
Finding optimal 1-degree Bayesian network was solved in [Chow-Liu’68]. It is a DAG of maximum in-degree 1, and maximizes the sum of mutual information $I$ of its edges.

$\iff$ finding the maximum spanning tree, where the weight of edge $(X, Y)$ is mutual information $I(X, Y)$.

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left( \frac{p(x, y)}{p(x) p(y)} \right)$$
Entropy Relationships

\[ H(X) \]

\[ H(X|Y) \]

\[ I(X;Y) \]

\[ H(Y) \]

\[ H(Y|X) \]

\[ H(X,Y) \]
Build a 1-degree BN for database

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alan</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cykie</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>David</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Eric</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Frank</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>George</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Helen</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Ivan</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Jack</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Build a Bayesian Network

- Start from a random attribute $A$
Select next tree edge by its mutual information

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alan</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cykie</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>David</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Eric</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Frank</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>George</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Helen</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Ivan</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Jack</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Select next tree edge by its mutual information

candidates:
A → B
A → C
A → D
Build a Bayesian Network

- Select next tree edge by its mutual information
Select next tree edge by its mutual information

- $I = 0$
- $I = 0$
- $I = 0$
- $I = 0.2$
- $I = 0.4$

Candidates:
- $A \rightarrow C$
- $A \rightarrow D$
- $B \rightarrow C$
- $B \rightarrow D$
Build a Bayesian Network

- Select next tree edge by its mutual information

DONE!
Do it under Differential Privacy!

(Non-private) select the edge with maximum $I$

(Private) $I$ is data-sensitive

$\Rightarrow$ the best edge is also data-sensitive
Exponential Mechanism [FOCS’07]

Databases $D$

Edges $e$

Define $q(D, e) \rightarrow R$

How good edge $e$ is as the result of selection, given database $D$

Return $e$ with probability: $Pr[e] \propto \exp \left( \frac{\varepsilon}{2} \cdot \frac{q(D, e)}{\Delta(q)} \right)$

Where $\Delta(q) = \max_{D, D', e} \|q(D, e) - q(D', e)\|_1$
Outline of the Algorithm

- **STEP 1**: Choose a suitable Bayesian network $\mathcal{N}$
  - must in a differentially private way

- **STEP 2**: Compute conditional distributions implied by $\mathcal{N}$
  - straightforward to do under differential privacy
  - inject noise – Laplace mechanism

- **STEP 3**: Generate synthetic data by sampling from $\mathcal{N}$
  - post-processing: no privacy issues
Outline

• Tabular data and histogram/range queries

• Algorithms for low dimensional data
  – Baseline
  – Partitioning algorithms: kd tree, quad tree, …
  – Transformation: Wavelet, Fourier Transform, …

• Algorithms for high dimensional data
  – Copula functions
  – Bayesian networks
Open questions

• High dimensional data
• Robust and private algorithm selection
• Error bounds for data-dependent algorithms
References