Data Mining with Differential Privacy

Arik Friedman, Assaf Schuster
Technion – Israel Institute of Technology
What this talk is about: balancing privacy with utility

Expected error rate

Accuracy of data mining model

Weak privacy

Strong privacy

Pareto frontier

Pareto improvement
Differential Privacy [DMNS’06]

Differential privacy requires that computations be insensitive to changes in any particular individual's record. Consequently, being opted in or out of the database should make little difference. Formally:

A randomized computation \( M \) provides \( \varepsilon \)-differential privacy if for any datasets \( A \) and \( B \) with symmetric difference \( A \triangle B = 1 \) and any set of possible outcomes \( S \in \text{Range}(M) \),

\[
\Pr[M(A) \in S] \leq \Pr[M(B) \in S] \times \exp(\varepsilon).
\]

\( \approx 1 + \varepsilon \) for small \( \varepsilon \)

Worst case definition

No dependency on background knowledge

Maintains composability:

\( k + k = 1 \) possible in \( k \)-anonymity

\( \varepsilon + \varepsilon \leq 2\varepsilon \) always holds in differential privacy, enables the concept of privacy budget
Data Mining with Differential Privacy

Direct access to the data

Logistic Regression [CM’08]
Recommender systems [MM’09]
Search queries and clicks [KKMN’09]
Random forests [JPW’09]
Frequent Patterns [BLST’10]

Trust boundary

Raw data

Synthetic dataset
Data Mining with Differential Privacy

Direct access to the data

Access through interface

PINQ (Privacy Integrated Queries) [Mcsherry’09]
SuLQ framework [BDMN’05]
Median mechanism [RT’10]
Data Mining with Differential Privacy

Direct access to the data

Trust boundary

Raw data

Non-interactive Datsets [BLR’08]
Complexity Results [DNRRV’09]
Private Coresets [FFKN’09]
Contingency Tables [BCDKMT’07]
Location Histograms [MKAGV’08]

Synthetic dataset
Data Mining with Differential Privacy

Direct access to the data

Access through interface

Synthetic dataset
Laplace Mechanism
Calibrating noise to sensitivity [DMNS’06]

Given a function \( f: D \rightarrow \mathbb{P}^d \) over an arbitrary domain \( D \), the sensitivity of \( f \) is

\[
S(f) = \max_{A,B \text{ where } A \Delta B = 1} \| f(A) - f(B) \|_1.
\]

Examples:
1. Count: for \( f(D) = |D| \), \( S(f) = 1 \).
2. Sum: for \( f(D) = \sum d_i \) where \( d_i \in [0, \Lambda] \), \( S(f) = \Lambda \).

Given a function \( f: D \rightarrow \mathbb{P}^d \) over an arbitrary domain \( D \), the computation

\[
M(X) = f(X) + (\text{Laplace}(S(f)/\varepsilon))^d
\]

provides \( \varepsilon \)-differential privacy.

Examples:
1. NoisyCount(D) = \( |D| + \text{Laplace}(1/\varepsilon) \).
2. NoisySum(D) = \( \sum d_i + \text{Laplace}(\Lambda/\varepsilon) \).
Exponential Mechanism \([\text{MT’07}]\)

Let \(q: D^n \times R \rightarrow \mathbb{R}\) be a query function that, given a database \(d \in D^n\), assigns a score to each outcome \(r \in R\). Then the exponential mechanism \(M\), defined by

\[
M(d, q) = \{\text{return } r \text{ with probability } \propto \exp(\varepsilon q(d, r)/2S(q))\},
\]

maintains \(\varepsilon\)-differential privacy.

Reminder: \(S(q) = \max_{A, B \text{ where } A \Delta B = 1} \|q(A) - q(B)\|_1\)

Motivation: \(\Pr(r) \propto \exp\left(\frac{\varepsilon q(d, r)}{2S(q)}\right)\)

Example – private vote: what to order for lunch?

<table>
<thead>
<tr>
<th>Option</th>
<th>Score (votes)</th>
<th>Sensitivity=1</th>
<th>Sampling Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pizza</td>
<td>27</td>
<td></td>
<td>(\varepsilon=0) 0.25 (\varepsilon=0.1) 0.4 (\varepsilon=1) 0.88</td>
</tr>
<tr>
<td>Salad</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hamburger</td>
<td>9</td>
<td></td>
<td>(\varepsilon=0) 0.25 (\varepsilon=0.1) 0.16 (\varepsilon=1) 10^{-4}</td>
</tr>
<tr>
<td>Pie</td>
<td>0</td>
<td></td>
<td>(\varepsilon=0) 0.25 (\varepsilon=0.1) 0.11 (\varepsilon=1) 10^{-6}</td>
</tr>
</tbody>
</table>

Impact of changing a single record is within \(\pm 1\)
Decision Trees

<table>
<thead>
<tr>
<th>No.</th>
<th>Blood Pressure</th>
<th>Weight</th>
<th>Temp.</th>
<th>Cough</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Low</td>
<td>Overweight</td>
<td>High</td>
<td>False</td>
<td>Sick</td>
</tr>
<tr>
<td>2</td>
<td>Low</td>
<td>Overweight</td>
<td>High</td>
<td>True</td>
<td>Sick</td>
</tr>
<tr>
<td>3</td>
<td>Normal</td>
<td>Overweight</td>
<td>High</td>
<td>False</td>
<td>Healthy</td>
</tr>
<tr>
<td>4</td>
<td>High</td>
<td>Normal</td>
<td>High</td>
<td>False</td>
<td>Healthy</td>
</tr>
<tr>
<td>5</td>
<td>High</td>
<td>Underweight</td>
<td>Normal</td>
<td>False</td>
<td>Healthy</td>
</tr>
<tr>
<td>6</td>
<td>High</td>
<td>Underweight</td>
<td>Normal</td>
<td>True</td>
<td>Sick</td>
</tr>
<tr>
<td>7</td>
<td>Normal</td>
<td>Underweight</td>
<td>Normal</td>
<td>True</td>
<td>Healthy</td>
</tr>
<tr>
<td>8</td>
<td>Low</td>
<td>Normal</td>
<td>High</td>
<td>False</td>
<td>Sick</td>
</tr>
<tr>
<td>9</td>
<td>Low</td>
<td>Underweight</td>
<td>Normal</td>
<td>False</td>
<td>Healthy</td>
</tr>
<tr>
<td>10</td>
<td>High</td>
<td>Normal</td>
<td>Normal</td>
<td>False</td>
<td>Healthy</td>
</tr>
<tr>
<td>11</td>
<td>Low</td>
<td>Normal</td>
<td>Normal</td>
<td>False</td>
<td>Healthy</td>
</tr>
<tr>
<td>12</td>
<td>Normal</td>
<td>Normal</td>
<td>High</td>
<td>True</td>
<td>Healthy</td>
</tr>
<tr>
<td>13</td>
<td>Normal</td>
<td>Overweight</td>
<td>Normal</td>
<td>False</td>
<td>Healthy</td>
</tr>
<tr>
<td>14</td>
<td>High</td>
<td>Normal</td>
<td>High</td>
<td>True</td>
<td>Sick</td>
</tr>
</tbody>
</table>
Decision Tree Induction with ID3

[Quinlan’86]

Given a set of transactions $\mathcal{T}$ over the attributes $\mathcal{A}=(A_1, A_2, ..., A_n)$ and the class $C$:

1. If $\mathcal{A}=\emptyset$ or $\forall T \in \mathcal{T}: T[C]=c$
   
   Return a leaf labeled with majority class.

2. Pick the “best” attribute $A$.

3. Split $\mathcal{T}$ to subsets $\{T \in \mathcal{T} : T[A]=a\}$ for each $a \in A$, and apply ID3 recursively on each subset.
Decision Tree Induction with Differential Privacy

Given a dataset \( T \), Attribute set \( \mathcal{A} \), class attribute \( C \) and tree depth limit:

\[ N_T = \text{NoisyCount}_{\varepsilon'}(T) \]

if \( \mathcal{A} = \emptyset \) or \( N_T < \text{threshold} \) or reached tree depth limit

\[ \forall c \in C: N_c = \text{NoisyCount}_{\varepsilon'}(r \in T \mid r_C = c) \]

return a leaf labeled with \( \arg\max_c(N_c) \)

else

Choose an attribute \( A \in \mathcal{A} \) for splitting \( T \).

\[ \forall i \in A \text{ apply the algorithm recursively on } (T_i = \{ r \in T \mid r_A = i \}, \mathcal{A} \setminus A, C) \text{ to obtain } \text{Subtree}_i. \]

return a tree with root node labeled \( A \), and edges labeled 1 to \( |A| \) each going to the \( \text{Subtree}_i \).

1. Limit tree depth to control privacy budget
2. Use noisy counts to determine class.
3. Set threshold on instance count to control noise impact
4. Choose an attribute with noisy counts or exponential mechanism
Choosing an attribute

1. Use noisy count to approximate information gain \([\text{BDMN}'05]\)

\[
V(A) = -\sum_{j \in A} \sum_{c \in C} -N_{j,c}^A \cdot \log \frac{N_{j,c}^A}{N_j^A}
\]

\(N_j^A = \text{NoisyCount}_c(T_j)\)

\(N_{j,c}^A = \text{NoisyCount}_c(T_{j,c})\)

2. Use the exponential mechanism with a query function based on a splitting criterion:

<table>
<thead>
<tr>
<th>Splitting Criterion</th>
<th>Query function</th>
<th>Sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information gain [Q’86]</td>
<td>(q_{IG}(T, A) = -\sum_{j \in A} \sum_{c \in C} \tau_{j,c}^A \cdot \log \frac{\tau_{j,c}^A}{\tau_j^A})</td>
<td>(S(q_{IG}) = \log(</td>
</tr>
<tr>
<td>Gini Index [BFOS’84]</td>
<td>(q_{GINI}(T, A) = -\sum_{j \in A} \tau_j^A \left(1 - \sum_{c \in C} \left(\frac{\tau_{j,c}^A}{\tau_j^A}\right)^2\right))</td>
<td>(S(q_{GINI}) = 2)</td>
</tr>
<tr>
<td>Max (based on resubstitution estimate [BFOS’84])</td>
<td>(q_{\text{Max}}(T, A) = \sum_{j \in A} \left(\max_{c} (\tau_{j,c}^A)\right))</td>
<td>(S(q_{\text{Max}}) = 1)</td>
</tr>
</tbody>
</table>

Notation: \(T\) – a set of records, \(r_A\) and \(r_C\) refer to the values that record \(r \in T\) takes on the attributes \(A\) and \(C\) respectively, \(\tau_j^A = |\{r \in T : r_A = j\}|\), \(\tau_{j,c}^A = |\{r \in T : r_A = j \land r_C = c\}|\). For noisy counts substitute \(N\) for \(\tau\).
Experimental evaluation: a single split

Figure 1. A single split: synthetic dataset with 10 binary attributes and a binary class, tree depth 1, $\varepsilon=0.1$, noise rate in learning data 0.1.
Conclusions and Future Work

Classifier reaches reasonable accuracy despite privacy constraints: taking privacy consideration into account when designing the algorithm is crucial to improving accuracy.

Yet, there is plenty room for improvement:

- Better budget management
- Variance in results
  - Possible solution: forests (as in [JPW’09])
- Rapid progress in theory and mechanisms
  - Median mechanism [RT’10]
  - Wavelet transforms [XWG’10]
  - Optimizing Linear Counting queries [LHRMM’10]
  - Computational differential privacy [MPRV’09]
  - Propose-Test-Release [DL’09]
Thank you for your attention!
Applying the exponential mechanism to choose a split point for a continuous attribute:

\[ \text{att} \in [0, 12] \]
\[ \varepsilon = 1.0 \]

Splitting criterion: Max

<table>
<thead>
<tr>
<th>Range</th>
<th>Max score</th>
<th>Score proportion (for range)</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 ≤ att &lt; 2</td>
<td>3</td>
<td>( \exp(3) \times 2 = 40.2 )</td>
<td>0.063</td>
</tr>
<tr>
<td>2 ≤ att &lt; 3</td>
<td>4</td>
<td>( \exp(4) \times 1 = 54.6 )</td>
<td>0.085</td>
</tr>
<tr>
<td>3 ≤ att &lt; 5</td>
<td>5</td>
<td>( \exp(5) \times 2 = 296.8 )</td>
<td>0.467</td>
</tr>
<tr>
<td>5 ≤ att &lt; 7</td>
<td>4</td>
<td>( \exp(4) \times 2 = 109.2 )</td>
<td>0.172</td>
</tr>
<tr>
<td>7 ≤ att &lt; 10</td>
<td>3</td>
<td>( \exp(3) \times 3 = 60.3 )</td>
<td>0.095</td>
</tr>
<tr>
<td>10 ≤ att &lt; 11</td>
<td>4</td>
<td>( \exp(4) \times 1 = 54.6 )</td>
<td>0.086</td>
</tr>
<tr>
<td>11 ≤ att ≤ 12</td>
<td>3</td>
<td>( \exp(3) \times 1 = 20.1 )</td>
<td>0.032</td>
</tr>
</tbody>
</table>

The split point is sampled with the exponential mechanism in two phases:

1. The domain is divided to ranges in which the score is constant. A range is chosen by applying the exponential mechanism.
2. A point is sampled uniformly from the chosen range.

In the first stage, the probability for each range \( R_i = [a', b'] \) is given by:

\[
\frac{\int_{a'}^{b'} \exp(\varepsilon q(d, r) / 2S(q))\,dr}{\int_{a}^{b} \exp(\varepsilon q(d, r) / 2S(q))\,dr} = \frac{\exp(\varepsilon c_i) |R_i|}{\sum_{j} \exp(\varepsilon c_j) |R_j|}
\]
Experimental evaluation: deeper trees

Figure 2. Deeper trees: synthetic dataset with 7 binary attributes, 3 continuous attributes and a binary class, tree depth up to 5, $\epsilon=1.0$, no noise in learning data.
Experimental evaluation: real dataset

Figure 3. Real dataset: Adult dataset, 8 nominal attributes, 6 continuous attributes, binary class attribute, trees of depth up to 5, 45,222 samples.