Data Mining with Differential Privacy
Arik Friedman and Assal Schuster

by Slawomir Goryczka
Differential Privacy

A randomized computation $M$ provides $\epsilon$-differential privacy if for any datasets $A$ and $B$ that differ by 1 record and any set of possible outcomes $S$:

$$\Pr[M(A) \in S] \leq \Pr[M(B) \in S] \times e^\epsilon$$

- $\epsilon$ allows us to control the level of privacy, lower $\epsilon$ means stronger privacy
- Composability property: a sequence of queries that guarantee $\epsilon_i$-differential privacy each guarantees overall $\Sigma \epsilon_i$-differential privacy (queries about the same data) or $\max(\epsilon_i)$ if each query asks for different data
Ensuring differential privacy

- The sensitivity of function $f$:

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

- Given $f: D \to \mathbb{R}^d$, the computation $M$ provides $\epsilon$-differential privacy:

$$M(X) = f(X) + \left( \text{Laplace}(S(f)/\epsilon) \right)^d$$
Ensuring differential privacy (2)

- For a given database $d$ and $\varepsilon$, the quality function $q$ induces a probability distribution over the output domain, from which the exponential mechanism $M$ samples the outcome.

- $M$ maintains $\varepsilon$-differential privacy:

$$M(d, q) = \left\{ \text{return } r \text{ with probability } \propto \exp \left( \frac{\varepsilon q(d, r)}{2S(q)} \right) \right\}$$

- High scoring outcomes are favored – they are exponentially more likely to be chosen.
PINQ

- PINQ stands for **Privacy INtegrated Queries**
- It is an interface for database access that ensures differential privacy of query results
- Differential privacy is ensured by adding a noise drawn from the Laplace distribution and the exponential mechanism
- Uses composition: parallel and sequential to manage privacy budget $\varepsilon$
- But it is up to data miner to chose appropriate queries in good order to spend privacy budget wisely
Differentially private ID3
(SuLQ-based ID3)

• ID3 (predecessor of C4.5) uses information gain to build a decision tree
• Naïve approach – run ID3 on differentially private (noisy) data
• But we need to change stopping criteria!

Stop further splits if all instances have the same class or there are no instances.

Stop further splits if each class count on average is larger than the standard deviation of the noise.
Differentially private ID3 (privacy budget)

• To split data points we need to determine:
  • Number of points (count)
  • The class count (to stop splitting, in leaves)
  • Evaluate attributes (in nodes)

• How to split the \( \varepsilon \) (the privacy budget)?
  • 50% to evaluate number of instances
  • 50% to determine class counts (leaves) or evaluate attributes (in nodes)

Because the count estimates required to evaluate the information gain should be carried out for each attribute separately, the overall budget needs to be split among them.
Splitting criteria
(Differentially Private ID3)

- Rather than evaluate each attribute separately, we can do it simultaneously in one query using the exponential mechanism

- /* Informally, instead comparing noisy information gain and choosing a splitting point, we will noisy chose a point based on a quality function. */

- Thus, we can spend more privacy budget for this operation in one query and reduce the expected noise

- But,... what quality function should be chosen?

\[
M(d, q) = \left\{ \text{return } r \text{ with probability } \propto \exp \left( \frac{\epsilon q(d, r)}{2S(q)} \right) \right\}
\]
Quality functions

- **Information gain** (sensitivity = \(\log(N+1) + 1/\ln 2\))

\[
q_{IG}(\mathcal{T}, A) = V(A) = - \sum_{j \in A} \sum_{c \in C} \tau_{j,c}^A \cdot \log \frac{\tau_{j,c}^A}{\tau_j^A}
\]

- **Gini index** (sensitivity = 2)

\[
q_{Gini}(\mathcal{T}, A) = - \sum_{j \in A} \tau_{j}^A \left(1 - \sum_{c \in C} \left(\frac{\tau_{j,c}^A}{\tau_j^A}\right)^2\right)
\]

- **Max operator** (sensitivity = 1)

\[
q_{Max}(\mathcal{T}, A) = \sum_{j \in A} \left(\max_c (\tau_{j,c}^A)\right)
\]

- **Gain ratio** (unbounded sensitivity)

\[
q_{Gain}(\mathcal{T}, A) = \sum_{j \in A} \left(\frac{\tau_j^A}{\tau_j^T}\right) \cdot \log \frac{\tau_j^T}{\tau_j^A}
\]
Pruning

- Because of noise the resulting tree may contain redundant splits, and pruning may improve it.

- Error based pruning (as in C4.5), where the training set is used to evaluate the decision tree before and after pruning → biased in favor of the training set.
  
  - For a given sub-tree compare it with a case when its turned into a leaf.
  
  - It is easy to compute count of a subtree (use previous values), but what about pruned case?

  Sum up values in the tree (higher noise), ask query (spend privacy budget)?
Pruning (solution)

Two passes:

- Top-down to calibrate the total instance count in each level of the tree
- Bottom-up to aggregate the class counts and calibrates them to match the total instance counts
Continuous Attributes

- C4.5: attribute values from the training set are used to determine potential split points
- Differential privacy: cannot do the same → direct privacy violation

Use the exponential mechanism:
- A learning examples induce a probability distribution over the attribute domain
- Given a splitting criterion, split points with better scores will have higher probability to be picked.
- The domain is not discrete, but it is divided into ranges with constant scores
Continuous Attributes (2)

Idea:

- Pick a range using exponential mechanism
- Chose a splitting point with uniform distribution from the chosen range

But:

- The attribute domain has to be finite
- This calculations need to be repeated for every node in the decision tree → need some privacy budget

Alternative solution: discretize the number attributes in the beginning → lose information, but save privacy budget
Experiments (synthetic datasets)

- ID3 baseline
- DiffP-ID3 InfoGain scorer
- DiffP-ID3 Gini scorer
- DiffP-ID3 Max scorer
- SuLQ-Based ID3

Parameters:
\[ B = 0.1 \]
\[ p_{\text{noise}} = 0.1 \]

Binary attribute

Average Decision Tree Accuracy (%) vs Number of Training Samples

- 100%
- 90%
- 80%
- 70%
- 60%
- 50%
- 40%
- 30%
- 20%
- 10%
- 0%

Number of Training Samples:
- 500
- 1000
- 1500
- 2000
- 2500
- 3000
- 3500
- 4000
- 4500
- 5000
Experiments (synthetic datasets)

Continuous attribute

Average Decision Tree Accuracy (%)

Number of Training Samples

- J48 baseline
- Unpruned J48
- DiffP-C4.5 InfoGain scorer
- DiffP-C4.5 Gini scorer
- DiffP-C4.5 Max scorer

B=0.1
p_{noise}=0.1
Experiments (real datasets)

- J48 baseline
- DiffP-C4.5 InfoGain scorer
- DiffP-C4.5 Gini scorer
- DiffP-C4.5 Max scorer

Average Accuracy (%)
Future work

• A challenge: large variance in the experimental results

• Possible solutions/ideas:
  • Consider other stopping rules
  • Different tactics for budget distribution

Thank you!