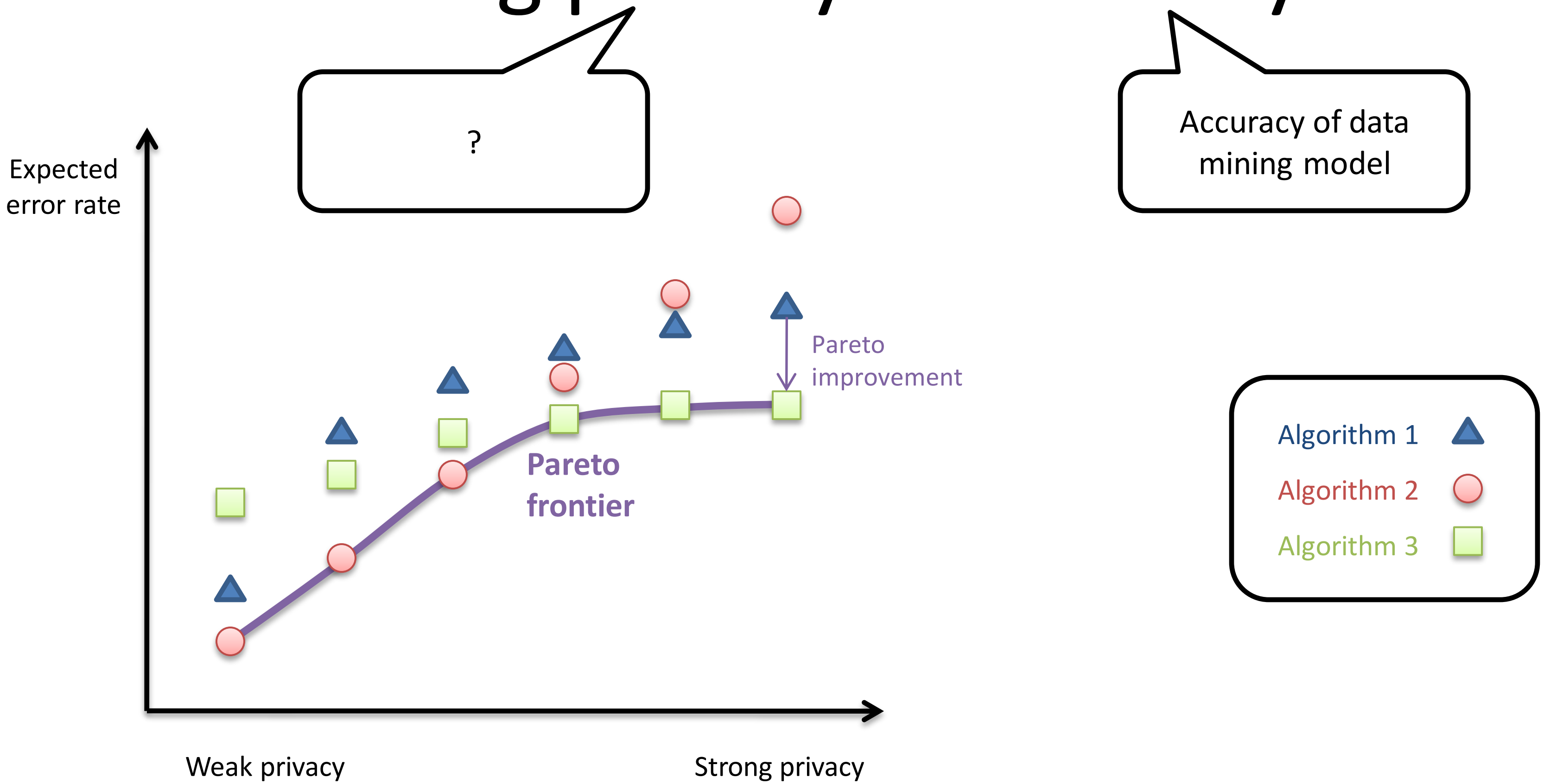
Data Mining with Differential Privacy

Arik Friedman, Assaf Schuster
Technion – Israel Institute of Technology

What this talk is about: balancing privacy with utility



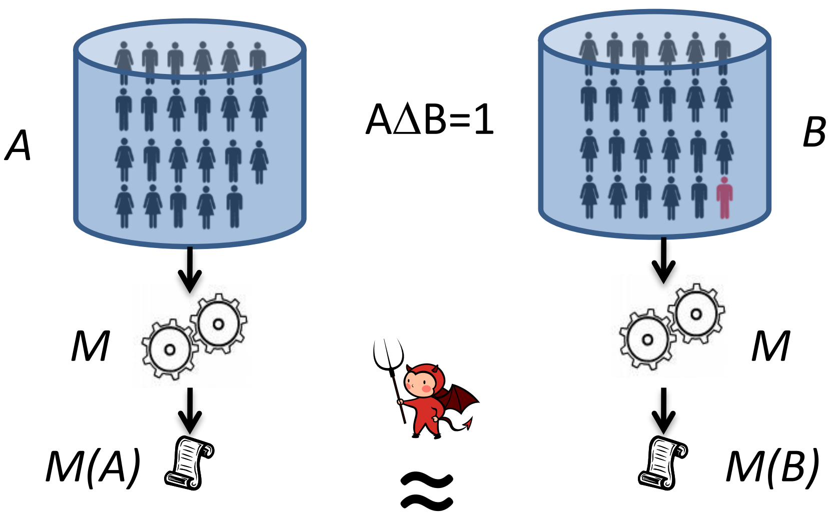
Differential Privacy [DMNS'06]

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 *ε-differential privacy* if for any datasets A and B with symmetric difference $A \Delta B=1$ and any set of possible outcomes $S \in Range(M)$,

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

≈1+ε
for small ε



⇒ Worst case definition

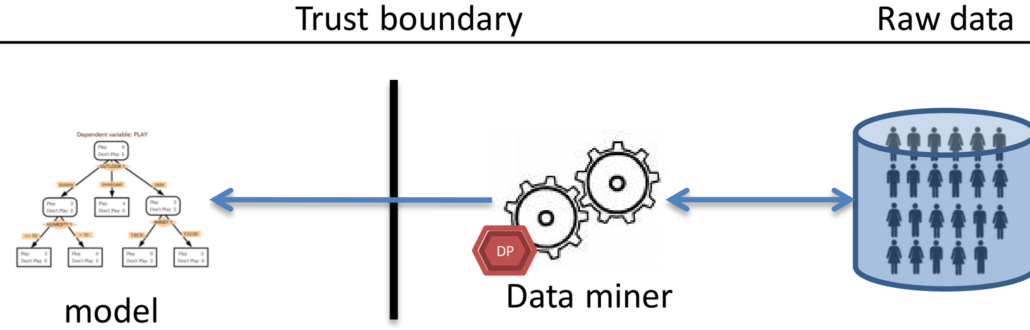
⇒ No dependency on background knowledge

⇒ Maintains composability:

k+k = 1	possible in *k*-anonymity
ε+ε ≤ 2ε	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]

Synthetic dataset

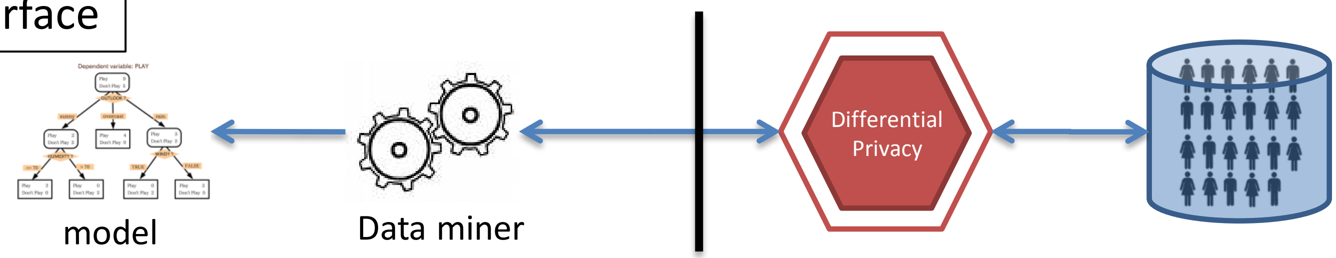
Data Mining with Differential Privacy

Trust boundary

Raw data

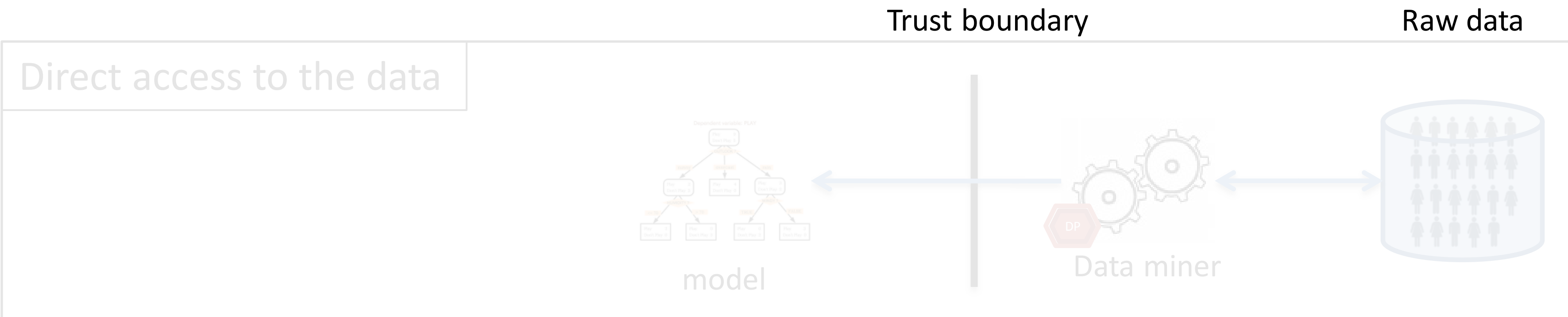
Direct access to the data

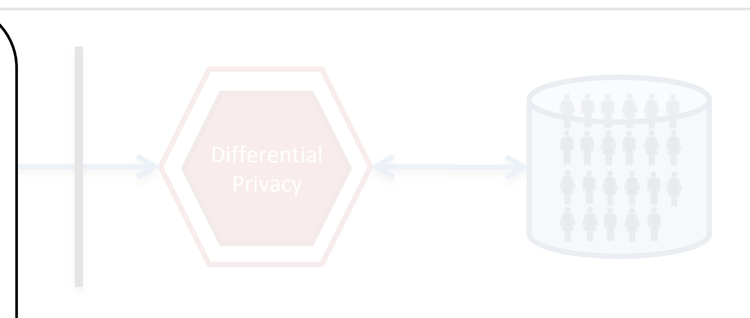
Access through interface



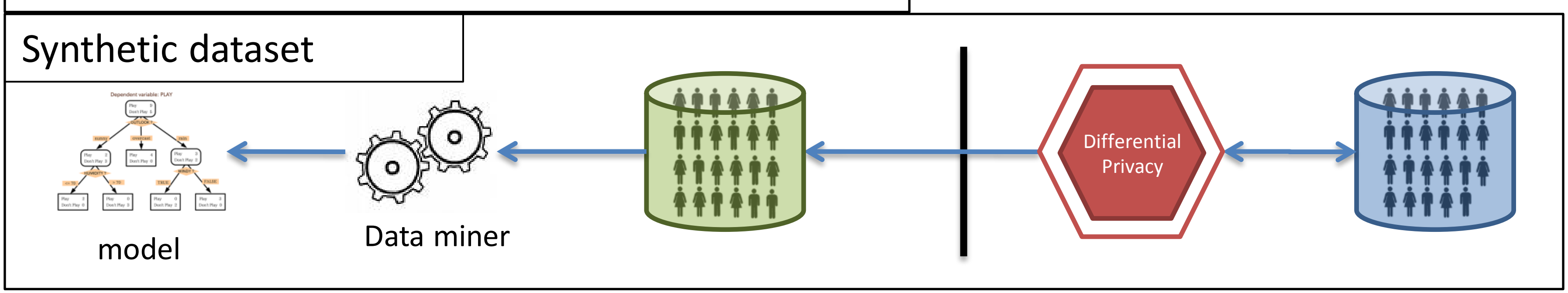
PINQ (Privacy Integrated Queries) [Mcsherry'09]
SuLQ framework [BDMN'05]
Median mechanism [RT'10]

Data Mining with Differential Privacy

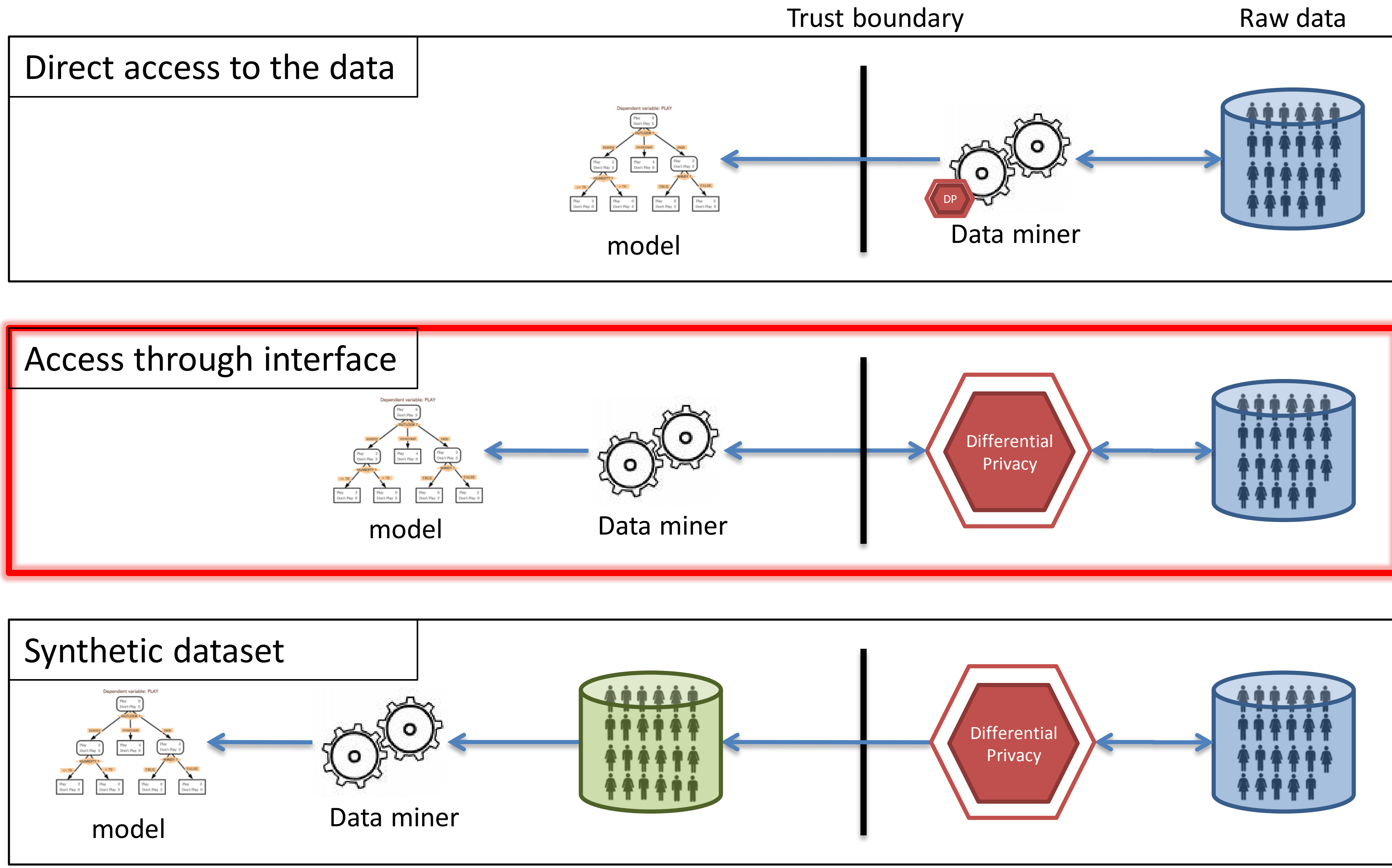




Non-interactive Datsets [BLR'08]
Complexity Results [DNRRV'09]
Private Coresets [FFKN'09]
Contingency Tables [BCDKMT'07]
Location Histograms [MKAGV'08]



Data Mining with Differential Privacy



Laplace Mechanism

Calibrating noise to sensitivity [DMNS'06]

Given a function f:$D \rightarrow \mathrm{P}^d$ 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 $f(D)=\Sigma d_i$, where $d_i \in [0,\Lambda]$, $S(f)=\Lambda$.

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

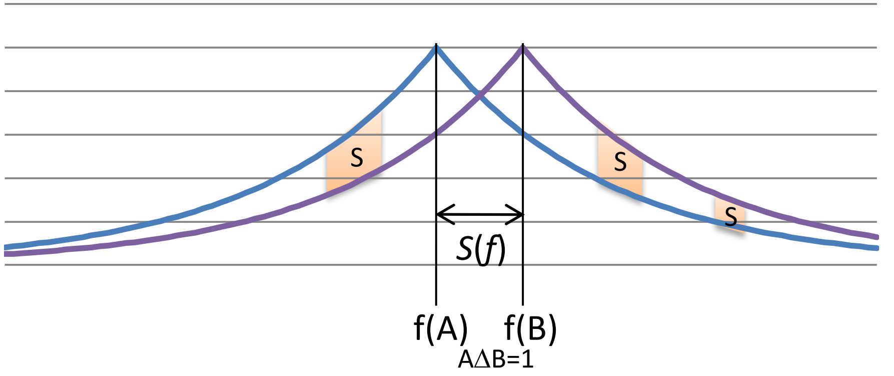
$$M(X) = f(X) + (\text{Laplace}(S(f)/\varepsilon))^d$$

provides ε-differential privacy.

Examples:

1. NoisyCount(D) = $|D|$+Laplace(1/ε).
2. NoisySum(D) = Σd_i +Laplace(Λ/ε).

Pr[M(A)∈S] ≤ Pr[M(B)∈ S] x exp(ε).



Exponential Mechanism [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

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

maintains ε-differential privacy.

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

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

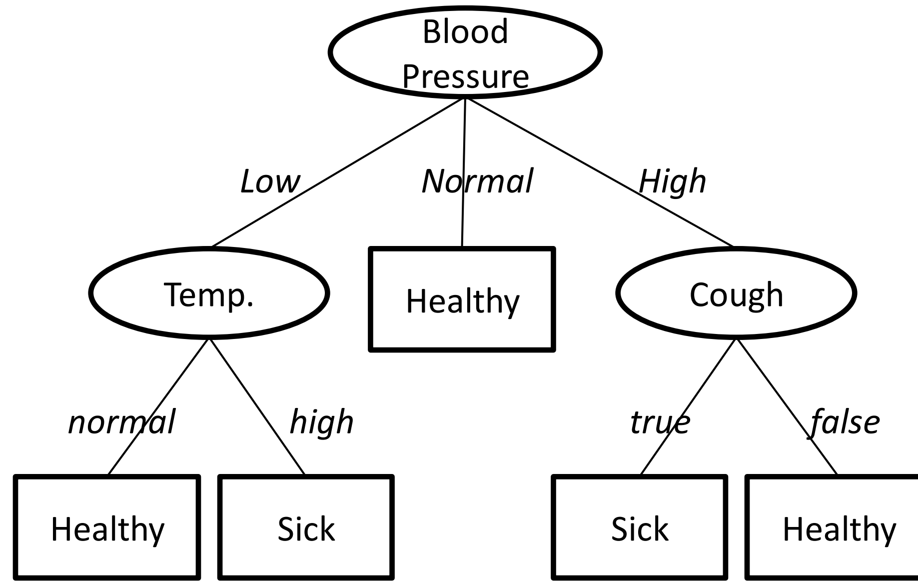
Impact of changing a single record is within ±1

Example – private vote: what to order for lunch?

Option	Score (votes) Sensitivity=1	Sampling Probability		
		ε=0	ε=0.1	ε=1
Pizza	27	0.25	0.4	0.88
Salad	23	0.25	0.33	0.12
Hamburger	9	0.25	0.16	10^{-4}
Pie	0	0.25	0.11	10^{-6}

Decision Trees

No.	Blood Pressure	Weight	Temp.	Cough	Class
1	Low	Overweight	High	False	**Sick**
2	Low	Overweight	High	True	**Sick**
3	Normal	Overweight	High	False	**Healthy**
4	High	Normal	High	False	**Healthy**
5	High	Underweight	Normal	False	**Healthy**
6	High	Underweight	Normal	True	**Sick**
7	Normal	Underweight	Normal	True	**Healthy**
8	Low	Normal	High	False	**Sick**
9	Low	Underweight	Normal	False	**Healthy**
10	High	Normal	Normal	False	**Healthy**
11	Low	Normal	Normal	False	**Healthy**
12	Normal	Normal	High	True	**Healthy**
13	Normal	Overweight	Normal	False	**Healthy**
14	High	Normal	High	True	**Sick**



Decision Tree Induction with ID3

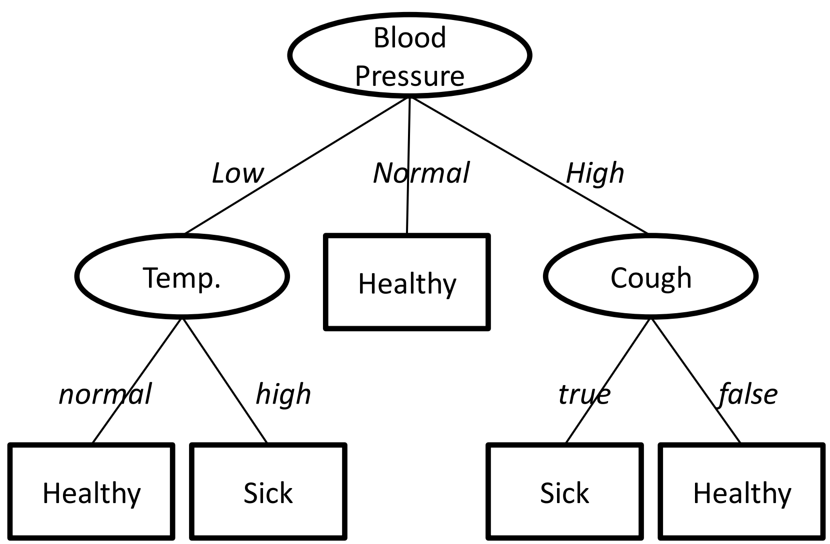
[Quinlan'86]

Given a set of transactions $\mathcal{T}$ over the attributes $\mathcal{A}=(A_1, A_2, \ldots, A_n)$ 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 $\text{argmax}_c(N_c)$

else

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

$\forall i \in A$ apply the algorithm recursively on

$(T_i = \{r \in T \mid r_A = i\}, \mathcal{A} \setminus A, C)$ to obtain Subtree_i.

return a tree with root node labeled A,

and edges labeled 1 to |A| each going to the 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

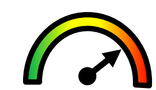
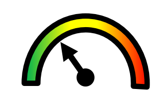
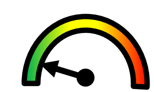
Choosing an attribute

1. Use noisy count to approximate information gain [BDMN'05]

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

$$N_j^A = NoisyCount_\varepsilon\left(\mathcal{T}_j\right)$$
$$N_{j,c}^A = NoisyCount_\varepsilon\left(\mathcal{T}_{j,c}\right)$$

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

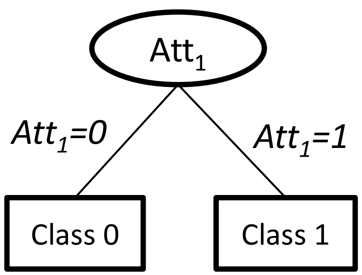
Splitting Criterion	**Query function**	**Sensitivity**	
Information gain [Q'86]	$q_{IG}(T,A) = -\sum_{j \in A} \sum_{j \in C} \tau_{j,c}^{A} \cdot \log \frac{\tau_{j,c}^{A}}{\tau_{j}^{A}}$	$S(q_{IG}) = \log(\lvert T\rvert+1)+1/\ln 2$	
Gini Index [BFOS'84]	$q_{GINI}(T,A) = -\sum_{j \in A} \tau_{j}^{A} \left(1 - \sum_{c \in C} \left(\frac{\tau_{j,c}^{A}}{\tau_{j}^{A}}\right)^2\right)$	$S(q_{GINI}) = 2$	
Max (based on resubstitution estimate [BFOS'84])	$q_{Max}(T,A) = \sum_{j \in A} \left(\max_{c}(\tau_{j,c}^{A})\right)$	$S(q_{MAX}) = 1$	

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, $\tau^A_j = \lvert\{r \in T : r_A = j\}\rvert$, $\tau^A_{j,c} = \lvert\{r \in T : r_A = j \wedge r_C = c\}\rvert$. For noisy counts substitute N for τ.

Experimental evaluation: a single split

Decision tree accuracy

(over 200 runs)



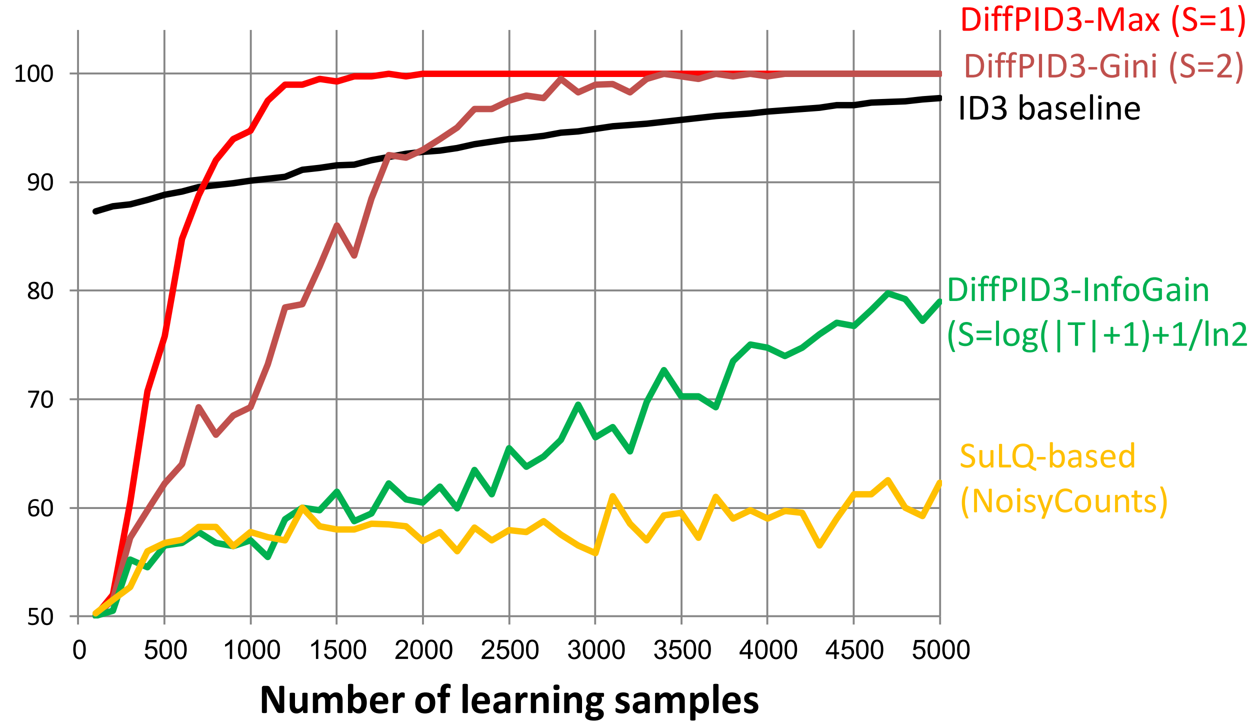
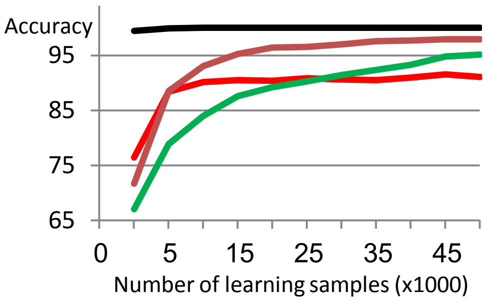


Figure 1. A single split: synthetic dataset with 10 binary attributes and a binary class, tree depth 1, ε=0.1, noise rate in learning data 0.1.

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!

Numeric attributes - example

Applying the exponential mechanism to choose a split point for a continuous attribute:

att $\in$ [0,12]

ε=1.0

Splitting criterion: Max

```
---+-+---+---+-----+-+--
   2 3   5   7    10 11
   + +   -   -     - +
```

Range	Max score	Score proportion (for range)	Probability
0 ≤ att < 2	3	exp(3)*2=40.2	0.063
2 ≤ att < 3	4	exp(4)*1=54.6	0.085
3 ≤ att < 5	5	exp(5)*2=296.8	0.467
5 ≤ att < 7	4	exp(4)*2=109.2	0.172
7 ≤ att < 10	3	exp(3)*3=60.3	0.095
10 ≤ att < 11	4	exp(4)*1=54.6	0.086
11 ≤ att ≤ 12	3	exp(3)*1=20.1	0.032

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

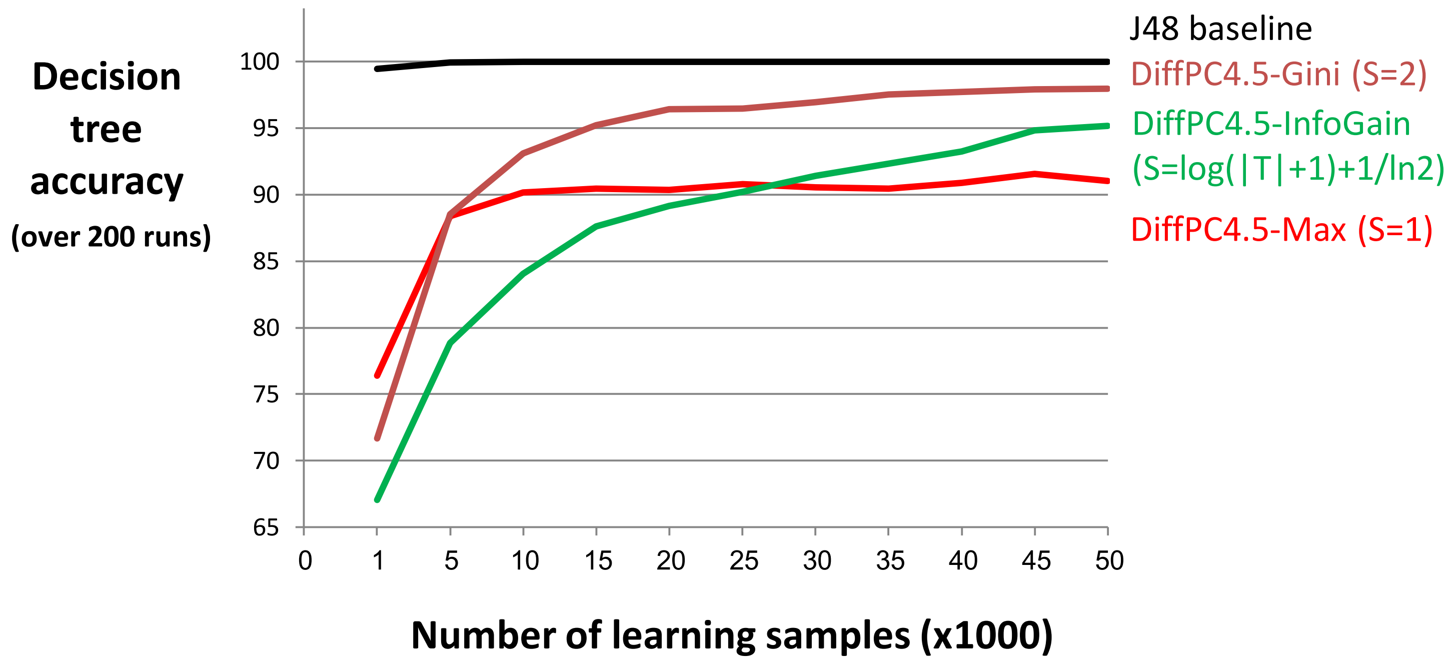


Figure 2. Deeper trees: synthetic dataset with 7 binary attributes, 3 continuous attributes and a binary class, tree depth up to 5, ε=1.0, no noise in learning data.

Experimental evaluation: real dataset

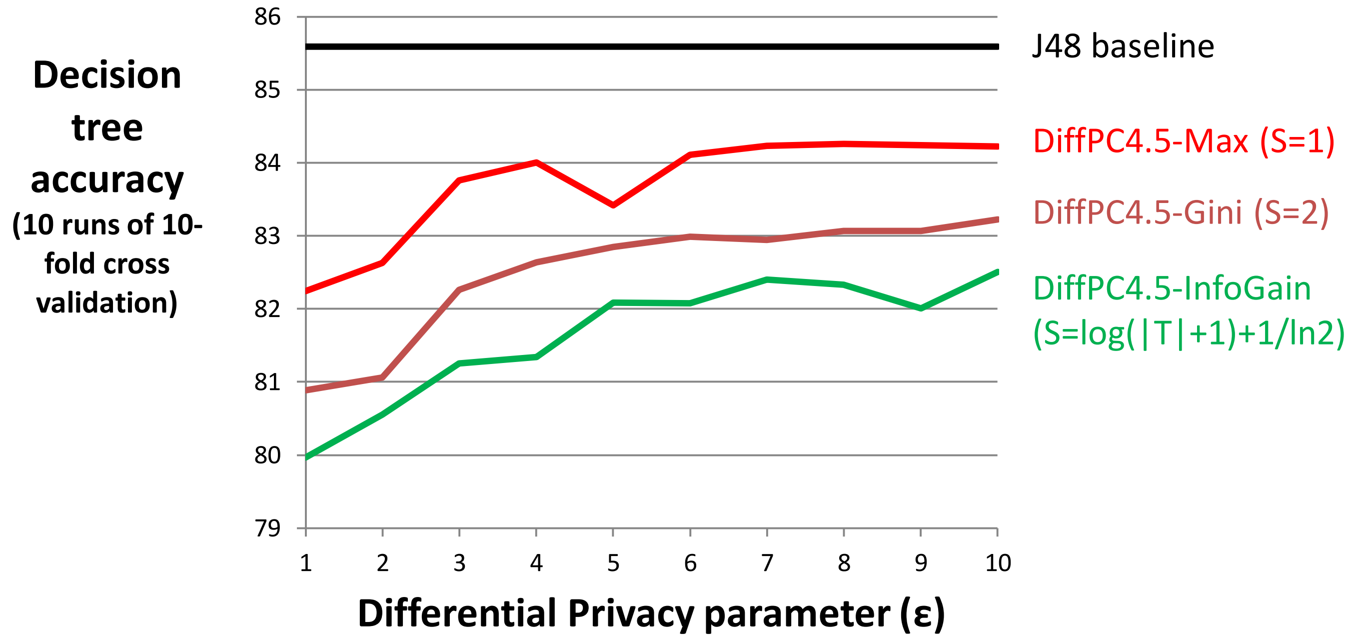


Figure 3. Real dataset: Adult dataset, 8 nominal attributes, 6 continuous attributes, binary class attribute, trees of depth up to 5, 45,222 samples.