Smooth Sensitivity for Learning Differentially-Private yet Accurate Rule Lists

Timothée Ly [1] **Julien Ferry** [2] **Marie-José Huguet** [3] **Sébastien Gambs** [4] **Ulrich Aivodji** [5]

Abstract

Differentially-private (DP) mechanisms can be embedded into the design of a machine learning algorithm to protect the resulting model against privacy leakage, although this often comes with a significant loss of accuracy. In this paper, we aim at improving this trade-off for rule lists models by establishing the smooth sensitivity of the Gini impurity and leveraging it to propose a DP greedy rule list algorithm. In particular, our theoretical analysis and experimental results demonstrate that the DP rule lists models integrating smooth sensitivity have higher accuracy that those using other DP frameworks based on global sensitivity.

1. Introduction

Machine learning models are increasingly used for high-stakes decision making tasks such as kidney exchange (Aziz et al., 2021) or recidivism prediction (Angwin et al., 2016). Because such tasks often require the use of sensitive data (*e.g.,* medical or criminal records), it is crucial to ensure that the learnt models do not leak undesired information. Another important aspect is to make sure human users can verify and trust the models' decisions. When possible, this motivates the use of inherently interpretable models (Rudin, 2019), as opposed to more complex black-boxes. However, such models are also vulnerable to privacy attacks such as membership inference (Shokri et al., 2017), in which the objective of the adversary is to infer the presence of a particular profile in the training dataset, or reconstruction attacks (Ferry et al., 2024), in which the aim of the adversary is to reconstruct the training set.

To counter this issue and protect the output of a computation over private data, Differential Privacy (DP) (Dwork et al., 2006; Dwork & Roth, 2014) has emerged as a *de facto* privacy standard. More precisely, DP aims at reconciling two antagonist purposes in privacy-preserving machine learning: extracting useful correlations from data without revealing private information about a particular individual. For instance, (Ji et al., 2014; Gong et al., 2020) have published a thorough survey on existing DP versions of classical machine learning algorithms. We can notably cite the DP versions of the Principal Component Analysis algorithm (Chaudhuri et al., 2013) and of the Stochastic Gradient Descent (Abadi et al., 2016). However, much less work has been dedicated to the DP implementations of interpretable models. Nonetheless, Fletcher & Islam (2019) have reviewed the current existing adaptations of DP to tree-based models (*i.e.*, mostly decision trees and random forests). This paper addresses one challenging future work they proposed, namely establishing the smooth sensitivity of the Gini impurity. More precisely, we first theoretically characterize the smooth sensitivity of the Gini impurity. Then, we design a DP mechanism based on smooth sensitivity with Laplace noise that we integrate into a greedy algorithm for learning rule lists models. Our experimental results show that the proposed DP mechanism incurs a lower accuracy loss than other mechanisms for a given privacy budget.

The outline of the paper is as follows. First in Section 2, we recall the background on rule lists models and DP. Afterwards in Section 3, we introduce the building blocks of our approach, namely greedy learning of rule lists, Gini impurity and smooth sensitivity. Then, in Section 4, we present our main contribution on the smooth sensitivity of the Gini impurity index as well as the DP greedy rule lists algorithm we have designed. Finally in Section 5, we empirically evaluate our proposed approach in terms of accuracy and robustness to privacy attacks before concluding in Section 6.

2. Background

In this section, we first introduce rule lists models before presenting the necessary background on differential privacy.

2.1. Rule Lists

We consider a tabular dataset $\mathcal{D}$ of n samples in which each sample s corresponds to a set of binary features and has a binary label y_s. Rule lists, originally introduced as a way to efficiently represent Boolean functions, are a common

[1]LAAS-CNRS, Telecom Paris & KTH, France [2]CIRRELT & SCALE-AI Chair in Data-Driven Supply Chains, Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Canada [3]LAAS-CNRS, Univ. Toulouse, CNRS, INSA, Toulouse France [4]Université du Québec à Montréal, Montréal, Canada [5]École de Technologie Supérieure, Montréal, Canada. Correspondence to: Timothée Ly <tly@laas.fr>, Julien Ferry <julien.ferry@polymtl.ca>.

type of interpretable models (Rivest, 1987; Angelino et al., 2017). More precisely, a rule list RL is a sequence of $K+1$ rules $(r_1, \dots, r_K, r_0) \in \mathscr{R}^{K+1}$ in which $\mathscr{R}$ is the set of possible rules (which, for instance, can be pre-mined). Any rule $r_i \in \mathscr{R}$ is composed of a Boolean assertion p_i called the antecedent and of a label prediction $q_i \in \{0,1\}$ named the consequent (*i.e.*, $r_i = p_i \rightarrow q_i$). A sample s of $\mathcal{D}$ is said to be caught by rule r_i when p_i evaluates to true for s, which leads to s being classified with label q_i. The *default rule*, $r_0 =$ True $\rightarrow q_0$ classifies any sample not caught by the previous rules to $q_0 \in \{0,1\}$ fixed. Rule Lists can be built either with an exact method such as CORELS (Angelino et al., 2018) or with heuristic approaches (Singh et al., 2021), which we specifically consider in this paper. Overall, rule lists are not extensively used in the literature, despite their advantage over decision trees in terms of compactness (Rivest, 1987).

```
if Prior-Crimes≠0 then True
else if Juvenile-Felonies ≤3 and Juvenile-Crimes≠1-3 then False
else True
```

Rule list 1. Example of a rule-list generated on the Compas dataset with our DP algorithm. The binary prediction is whether the offendent will recidivate within two years or not.

An example of a rule-list that could be used for recidivism prediction is provided in Rule List 1. Because the model is inherently interpretable, the use of any directly discriminating feature would easily be spotted (Voigt & Bussche, 2017). This is in contrast with black-box models, in which such an undesired behaviour would be more difficult - or even impossible (Merrer & Trédan, 2019) - to detect.

2.2. Differential Privacy

Differential privacy (DP) is a privacy model that provides strong privacy guarantees with respect to queries or computations performed on a database (Dwork & Roth, 2014). In particular in machine learning, DP can be integrated into the learning algorithm to ensure that the resulting model does not leak too much information with respect to the input dataset. In this context, a differentially-private learning algorithm ensures that the distribution over outputs (*i.e.*, possible models) is not impacted significantly by the addition or removal of a sample from the training set.

More formally, two datasets $\mathcal{D}$ and $\mathcal{D}'$ are said to be neighbouring if they differ at most by one sample, which we denote by $\|\mathcal{D}-\mathcal{D}'\|_1 \leq 1$ for $(\mathcal{D},\mathcal{D}') \in \mathbb{N}^{|\mathcal{X}|}$ (see Appendix B.1) in which $\mathcal{X}$ is the finite set of all possible samples in a dataset. Similarly, the number of elements in a dataset $\mathcal{D}$ is $\|\mathcal{D}\|_1$. An algorithm $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \mapsto \mathcal{Y}$ is (ε,δ)-differentially private if $\forall S \subseteq \mathcal{Y}, \forall(\mathcal{D},\mathcal{D}') \in (\mathbb{N}^{|\mathcal{X}|})^2, ||\mathcal{D}-\mathcal{D}'||_1 \leq 1$, we have: $\mathbb{P}(\mathcal{M}(\mathcal{D}) \in S) \leq \exp(\varepsilon)\mathbb{P}(\mathcal{M}(\mathcal{D}') \in S) + \delta$ (Dwork & Roth, 2014).

The parameter ε controls the level of privacy of the algorithm as it defines how much the probability of an output can vary when adding or removing a sample. Typically, $\varepsilon = 1$ is considered a reasonable value in terms of provided protection. The parameter δ can be interpreted as a probability of "total privacy failure". One possible instance of this δ-failure could be that with probability $1-\delta$, the model will behave like pure DP (*i.e.*, ε differential privacy) while with probability δ (*i.e.*, the failure probability), there will be no privacy guarantees at all. $\delta \ll \frac{1}{\|\mathcal{D}\|_1}$ is considered to be an absolute requirement since a δ of order $\mathcal{O}(\|\mathcal{D}\|_1)$ could enable the total release of some samples of the dataset.

Intuitively, differentially-private mechanisms often revolve around the idea of adding noise of a magnitude of order close to how steep the output function can change with slight variations of the input. More precisely, for a given function $f \in \mathbb{R}^k$, its *global sensitivity* precisely quantifies this aspect. The global sensitivity of f is denoted by $\Delta_p f$, in which l_p stands for the l_1 or l_2 norms. Let $f : \mathbb{N}^{|\mathcal{X}|} \mapsto \mathbb{R}^k$, its global sensitivity for any neighbouring dataset $\mathcal{D}$ and $\mathcal{D}'$ is: $\Delta_p f = \max_{\substack{\mathcal{D},\mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|} \\ ||\mathcal{D}-\mathcal{D}'||_1 = 1}} ||f(\mathcal{D}) - f(\mathcal{D}')||_p$.

One of the shortcomings of global sensitivity is that it does not take into account the position in the latent space of the points considered. One straightforward approach to achieve DP is by adding noise to the output of a given function $f \in \mathbb{R}^k$. Two common differentially-private mechanisms, the *Laplace mechanism* $\mathcal{M}^{\Delta_1}_{LAPLACE}(\mathcal{D}, f, \varepsilon)$ and the *Gaussian mechanism* $\mathcal{M}^{\Delta_2}_{GAUSS}(\mathcal{D}, f, \varepsilon, c)$ are based on this principle (Dwork & Roth, 2014). For each component of f, the amplitude of this noise $N_j^{\Delta_p}$ varies accordingly to the global sensitivity $\Delta_p f$: $\mathcal{M}^{\Delta_p}_{NOISE}(\mathcal{D}, f, \varepsilon) := f(\mathcal{D}) + (N_1^{\Delta_p}, \dots, N_k^{\Delta_p})$.

The $\mathcal{M}^{\Delta_1}_{LAPLACE}(\mathcal{D}, f, \varepsilon)$ mechanism is based on Δ_1 and on Laplace noise: $\forall i \in \{1, \dots, k\}, N_i \sim Lap(\Delta_1 f/\varepsilon)$. The probability density function of the Laplace distribution is $Lap(x \mid b) = \dfrac{1}{2b}\exp\left(\frac{-|x|}{b}\right)$ and the Laplace mechanism has been proven to be $(\varepsilon, 0)$-DP.

$\mathcal{M}^{\Delta_2}_{GAUSS}(\mathcal{D}, f, \varepsilon, c)$ was shown to satisfy (ε,δ)-DP. It uses Δ_2 for the global sensitivity together with the addition of Gaussian noise: $\forall i \in \{1, \dots, k\}, N_i \sim \mathcal{N}\left(\mu = 0, \sigma = \frac{c \cdot \Delta_2 f}{\varepsilon}\right)$ with $c^2 > 2\log(\frac{1.25}{\delta})$.

In contrast, another common and very generic mechanism, called the *Exponential mechanism* $\mathcal{M}^{\Delta u}_{EXP}(\mathcal{D}, u, \mathscr{R})$, considers the set of possible outputs valuated in a range $\mathcal{V}$ and samples one of them v with respect to their utility. More precisely, let $u : (\mathcal{D}, v) \mapsto u(\mathcal{D}, v)$ denote the utility function of element v with respect to dataset $\mathcal{D}$. The global sensitivity of the utility function is: $\Delta u = \max_{v \in \mathcal{V}} \max_{\substack{\mathcal{D}, \mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|} \\ ||\mathcal{D}-\mathcal{D}'||_1 \leq 1}} |u(\mathcal{D}, v) - u(\mathcal{D}', v)|$. $\mathcal{M}_{EXP}^{\Delta u}(\mathcal{D}, u, \mathcal{R})$ samples an element $r \in \mathcal{R}$ with probability $p \propto \exp(\frac{\varepsilon \cdot u(\mathcal{D}, r)}{2 \cdot \Delta u})$ and has been proven to satisfy $(\varepsilon, 0)$-DP (Dwork & Roth, 2014).

Finally, we will also use the *Noisy Max Report* mechanism (Dwork & Roth, 2014), $\mathcal{M}_{noisy}$, which satisfies (ε, δ)-DP, and returns: $\underset{v \in \mathcal{V}}{\operatorname{argmax}}\, \mathcal{M}_{noisy}(\mathcal{D}, u(\cdot, v), \varepsilon)$.

Among others, DP comes with two fundamental properties: the *composability* property, which enables the composition of different differentially-private mechanisms (sequentially or in parallel) and the computation of the global privacy leakage incurred, and the *post-processing* property, which ensures that DP guarantees are not affected by post-processing the output of a DP mechanism (see Appendix B.2 for more details).

3. Building Blocks

In this section, we introduce the different building blocks that are necessary for the design of our framework. We first describe the greedy algorithm as the baseline for learning rule lists models as well as the computation of the Gini impurity index. We then review the notion of smooth sensitivity, before showing how it can be used to get DP guarantees.

3.1. A Greedy Algorithm for Learning Rule Lists

Greedy algorithms are widely used for learning decision tree models. For instance, the commonly used `CART` algorithm (Breiman et al., 1984) iteratively builds a decision tree in a top-down manner, by successively selecting the feature (and split value) yielding the best information gain value according to some pre-defined criterion. While algorithms for learning rule lists in a greedy manner are far less popular than their counterparts for learning decision trees, some implementations exist in the literature. For instance, the `imodels` library (Singh et al., 2021) contains algorithms for learning different types of interpretable models, including rule lists (denoted `GreedyRL`). More precisely, `GreedyRL` iteratively calls `CART` to build a depth-one decision tree at each level of the rule list, optimizing a given information gain criterion. Just like for decision trees, greedy algorithms for building rule lists successively select the best rule $r_i = p_i \rightarrow q_i$ given some information gain criterion. Thus, at each level of the rule list being built, the `GreedyRL` algorithm iterates through all possible rules and keeps the one leading to the best information gain value.

3.2. Gini Impurity for Rule Lists

In this paper, we consider the Gini impurity index originally used in the `CART` (Breiman et al., 1984) algorithm as a measure of the information gain. In a nutshell, this index quantifies how well a rule separates the data into two categories with respect to different labels, with the value of zero being reached when the examples are perfectly separated. The algorithm stops when all the samples are classified, but other stopping criteria can be implemented such as a maximum length on the list of rules or a minimum support condition on each rule (*i.e.*, number of points left to be classified).

Consider a given rule r of a pre-existing list of rules, which means that some samples were already captured by previous rules and are not accounted for. Let $C(r) \subset \mathcal{D}$ be the subset of samples captured by rule r, in which $n_c(r)$ is the number of samples in $C(r)$ and $n_l(r)$ the number of samples not captured by rule r. For a rule list $RL = (r_1, \ldots, r_K, r_0)$, and a given position j in the sequence, let $\tilde{n}(j)$ be the number of samples not captured by previous rules $r_1 \ldots r_{j-1}$. In particular, this means that $\tilde{n}(j) = n_c(r_j) + n_l(r_j) = n - \sum_{i=1}^{j-1} n_c(r_i)$. In addition, let $\hat{y}_c(r)$ the average outcome (*i.e.* the predicted label) of the rule r, $\hat{y}_c(r) = \frac{1}{n_c(r)} \sum_{s \in C(r)} y_s$. Similarly, the average outcome of the remaining samples is $\hat{y}_l(r) = \frac{1}{n_l(r)} \sum_{s \in \mathcal{D} \setminus \left(\cup_{i=1}^{j-1} C(r_i) \,\cup\, C(r) \right)} y_s$.

The Gini impurity reduction with respect to rule r is denoted as $\mathcal{G}(r)$. It can be divided into two terms $\mathcal{G}_c(r)$ and $\mathcal{G}_l(r)$, respectively for the samples caught and the ones not caught by the rule: $\mathcal{G}(r) = \mathcal{G}_c(r) + \mathcal{G}_l(r)$ Note that we not only consider the samples caught by the rule (through $\mathcal{G}_c(r)$) but also those which are not (through $\mathcal{G}_l(r)$) as it matters for the following rules in the rule list. For binary classification, the Gini impurity reduction for a rule r at position j is given by:

$$\mathcal{G}_c(r) = \frac{n_c(r)}{\tilde{n}(j)} \left(1 - \hat{y}_c(r)^2 - (1 - \hat{y}_c(r))^2\right).$$

$$\mathcal{G}_l(r) = \frac{n_l(r)}{\tilde{n}(j)} \left(1 - \hat{y}_l(r)^2 - (1 - \hat{y}_l(r))^2\right).$$

3.3. Smooth Sensitivity

The mechanisms described in Section 2.2 rely on the notion of global sensitivity. However, some functions only display a very loose bound for their global sensitivity. For instance, the global sensitivity of the Gini impurity is 0.5, irrespective of the actual number of samples left to be classified. To address this limit, Nissim et al. (2007) have introduced the notion of the *local sensitivity* of a function $f : \mathbb{N}^{|\mathcal{X}|} \mapsto \mathbb{R}^k$ at a dataset $\mathcal{D}$, denoted $LS_f(\mathcal{D})$, as: $\max_{\substack{\mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|}: \\ ||\mathcal{D}-\mathcal{D}'||_1 = 1}} ||f(\mathcal{D}) - f(\mathcal{D}')||_1$.

However, replacing directly the global sensitivity by local sensitivity does not yield strong privacy guarantees. Thus, a more refined sensitivity notion denoted as smooth sensitivity was proposed in (Nissim et al., 2007). This notion exploits a *smooth upper bound* of $LS_f(\mathcal{D})$, denoted by $S_{f,\beta}(\mathcal{D})$, as follows. For $\beta > 0$, $S_{f,\beta}(\mathcal{D}) : \mathbb{N}^{|\mathcal{X}|} \mapsto \mathbb{R}^+$ is a β-smooth upper bound on the local sensitivity of f if it satisfies :

$$\forall \mathcal{D} \in \mathbb{N}^{|\mathcal{X}|}, \forall \mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|} \; s.t. \; ||\mathcal{D} - \mathcal{D}'||_1 = 1,$$
$$S_{f,\beta}(\mathcal{D}) \geq LS_f(\mathcal{D}) \quad \text{and} \quad S_{f,\beta}(\mathcal{D}) \leq e^{\beta} S_{f,\beta}(\mathcal{D}') \quad (1)$$

The smallest function to satisfy Equation 1 is called the *smooth sensitivity* and denoted $S^*_{f,\beta}(\mathcal{D})$:

$$\text{For } \beta > 0, S^*_{f,\beta}(\mathcal{D}) = \max_{\mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|}} LS_f(\mathcal{D}') e^{-\beta ||\mathcal{D} - \mathcal{D}'||_1}$$

Nissim et al. (2007) proposed an iterative computation of the smooth sensitivity (Lemma 3.1) considering datasets than can vary up to k samples rather than 1. Let $\mathcal{T}_k$ denote the local sensitivity of f at distance k: $\mathcal{T}_k(\mathcal{D}) = \max\left\{ LS_f(\mathcal{D}') \,\middle|\, ||\mathcal{D}' - \mathcal{D}||_1 \leq k \right\}$.

Lemma 3.1. $S^*_{f,\beta}(\mathcal{D}) = \max\left\{ e^{-\beta k} \mathcal{T}_k(\mathcal{D}) \middle| k \in \mathbb{N} \right\}$. *(proof recalled in Appendix B.3)*

As stated by (Fletcher & Islam, 2017; Zafarani & Clifton, 2020; Sun et al., 2020), smooth sensitivity is a very powerful tool to replace global sensitivity for differentially-private machine learning models. However, finding a closed form for $S^*_{f,\beta}(\mathcal{D})$ is difficult and sometimes requires to make stronger assumptions on the model. Nonetheless, two DP mechanisms were proposed by Nissim et al. (2007) based on the smooth sensitivity. The first one is based on Cauchy noise and uses a parameter γ:

$$\mathcal{M}^{S^*_{\cdot,\beta}}_{CAUCHY}(\mathcal{D}, f, \varepsilon) \; : \; \mathcal{D} \mapsto f(\mathcal{D}) + \frac{2(\gamma+1) S^*_{f,\beta}(\mathcal{D})}{\varepsilon} \cdot \eta$$

with $\beta \leq \frac{\varepsilon}{2(\gamma+1)}$, $\gamma > 1$ and $\eta \sim h(z) \propto \frac{1}{1+|z|^{\gamma}}$ the Cauchy noise. This mechanism satisfies (ε,0)-DP.

The second one uses Laplace noise and satisfies (ε, δ)-DP:

$$\mathcal{M}^{S^*_{\cdot,\beta}}_{LAPLACE}(\mathcal{D}, f, \varepsilon) \; : \; \mathcal{D} \mapsto f(\mathcal{D}) + \frac{2 \cdot S^*_{f,\beta}(\mathcal{D})}{\varepsilon}$$

with $\beta \leq \frac{\varepsilon}{2\log(2/\delta)}$ and $\eta \sim Lap(1)$, the Laplace noise. Note that in contrast to global sensitivity, adding Laplace noise within the framework of smooth sensitivity does not yield pure DP anymore but approximate one.

4. A Differentially-Private Greedy Learning Algorithm for Rule Lists

We now introduce our framework for learning differentially-private rule lists leveraging smooth sensitivity. Unlike Fletcher & Islam (2017) who integrate smooth sensitivity to determine the majority class for a leaf in a tree, we integrate it to determine the rule with the best Gini impurity. We first demonstrate how to precisely compute the smooth sensitivity of the Gini impurity before leveraging it to design a differentially-private `GreedyRL` algorithm.

4.1. Smooth Sensitivity of the Gini Impurity

The local sensitivity for the Gini impurity has been characterized in Fletcher & Islam (2015). Considering the support $\tilde{n}(j)$ of the jth rule, it is defined by:

$$LS_{\mathcal{G}}(\tilde{n}(j)) = 1 - \left(\frac{\tilde{n}(j)}{\tilde{n}(j)+1}\right)^2 - \left(\frac{1}{\tilde{n}(j)+1}\right)^2$$

Given the minimal support Λ imposed for each selection of rule, we have derived in Theorem 4.1 a method to compute the smooth sensitivity of the Gini impurity.

Theorem 4.1 (Smooth Sensitivity of the Gini impurity)**.** *Let $\Lambda \in \mathbb{N}^*$ be the given minimum support. By inverting the parameter k and the variable $\mathcal{D}$ in the function $\mathcal{T}_k(\mathcal{D})$, we define the following function :*

$$\xi_{\mathcal{D},\beta}(k) : \left| \begin{array}{ccc} \mathbb{N} & \longrightarrow & \mathbb{R}^+ \\ k & \longmapsto & e^{-k\beta} \cdot g\big[\max(\Lambda, \|\mathcal{D}\|_1 - k)\big] \end{array} \right.$$

in which

$$g : \left| \begin{array}{ccc} \mathbb{R}^+ & \longrightarrow & [0,1] \\ x & \longmapsto & 1 - \left(\frac{x}{x+1}\right)^2 - \left(\frac{1}{x+1}\right)^2 \end{array} \right.$$

The smooth sensitivity of a rule with a dataset $\mathcal{D}$ of points that remains to classify is : $S^*_{\mathcal{G},\beta}(\mathcal{D}) = \max\left[\xi_{\mathcal{D},\beta}(0),\ \xi_{\mathcal{D},\beta}(\lfloor t \rfloor),\ \xi_{\mathcal{D},\beta}(\lceil t \rceil),\ \xi_{\mathcal{D},\beta}(\|\mathcal{D}\|_1 - \Lambda)\right]$ *with* $t = \|\mathcal{D}\|_1 - \dfrac{1 - \beta - \sqrt{(1-\beta)^2 - 4\beta}}{2\beta}$ *if well defined and otherwise* 0*.*

The detailed proof is provided in Appendices A.1 and A.2 where we first prove it for $\Lambda = 1$ and generalize the proof for $\Lambda \in \mathbb{N}^*$. Crucially, recall that the smooth sensitivity is the same for any rule at a given position since we have proven that the smooth sensitivity of the Gini impurity only takes into account the number of elements left to be classified (and not how the rule captures them or not). Figure 1 gives an overview on the amount of noise one has to add to the computed Gini impurity to get a target DP guarantee, using either global or smooth sensitivity. More precisely in this figure, we display the noise distortion generated for a fixed $\varepsilon = 1$ by each DP mechanism as a function of the number of examples captured by the rule. Importantly, we observe how the use of smooth sensitivity allows to scale down the generated noise when considering more examples. This is not the case for global sensitivity, which is dataset-independent.

Many learning algorithms use a regularization parameter scaling with the length of the model to reduce overfitting (Domingos, 2012). In our case, apart from being a key factor for the smooth sensitivity, the minimum support leads to a better comprehensibility of the model (there can only be as many as $\frac{1}{\lambda}$ rules) and plays the role of the regularization parameter as it helps the model to not overfit.

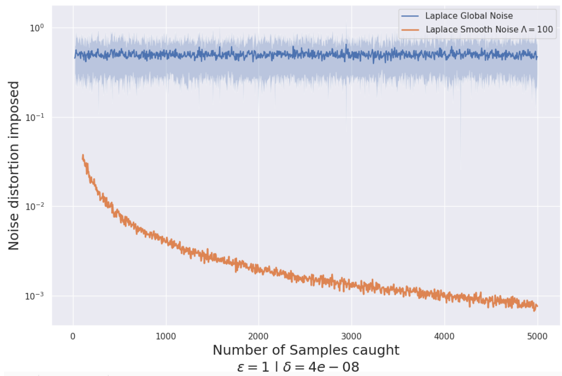


Figure 1. Comparison of the amplitude of Smooth and Global Sensitivities for the Laplace Mechanism, log scaled.

4.2. Differentially-Private Greedy Rule Lists

Our differentially-private algorithm for learning rule lists, `DP-GreedyRL`, is detailed in Algorithm 1. Note that consistent with the literature, the set of rules $\mathscr{R}$ is assumed to be publicly known and is not obtained as a computation from the data. This algorithm iteratively adds a new rule to the rule list RL. At each step, it checks whether the support in the current remaining dataset X_{rem} verifies the minimum support condition (Line 4), including the confidence threshold computed once for all (Line 2). For each rule $r \in \mathscr{R}$, its Gini impurity is computed at Line 11 and the rule $R^\star$ whose noisy Gini is the lowest is returned. $R^\star$ is then added to RL with its DP prediction $q^\star$ (Line 14) and removed from $\mathscr{R}$. The main loop is stopped when (1) the rule list reaches the maximum length, (2) the support condition is not verified anymore or (3) when adding a rule does not improve the Gini index.

Rule's prediction. In Algorithm 1, it is necessary to make the choice of the prediction for each rule differentially-private. Indeed, in the non-DP setup, the prediction is computed as the majority class among the examples caught by the rule. However, such a deterministic selection of the best prediction is not compatible with DP. For instance, consider two neighbouring datasets $\mathcal{D}$ and $\mathcal{D}'$. Let r be a rule picked from the rule list built on $\mathcal{D}$. If $\mathcal{D}'$ is $\mathcal{D}$ deprived from one element that would flip the outcome of r, the probability of observing this outcome in the built rule list is also flipped from 1 to 0 breaking any DP guarantee. Thus, the rules' predictions have to be determined using DP-protected counts. In our implementation (Algorithm 2), we use the Laplace mechanism based on the global sensitivity to compute the counts for each rule that are later used to determine the rule's prediction.

Confidence threshold for minimum support. One of the remaining issue with the proposed smooth sensitivity framework is the minimum support requirement may jeopardize the DP guarantees. For instance, consider $\mathcal{D}$ a dataset and a fixed Λ and let r a rule. Suppose that after applying rule r, the number of points remaining for classification $n_l(r)$ is exactly equal to Λ. Let also $\mathcal{D}'$ be a dataset neighbouring $\mathcal{D}$ that misses one of the samples not caught by r in $\mathcal{D}$. Then, the support of $\mathcal{D}'$ after applying rule r is strictly smaller than Λ so any rule will necessarily be discarded because it is a stopping condition. Again, this breaks any DP guarantees, as the resulting model may change significantly due to the absence of a single example in the dataset.

Algorithm 1 Approximate (ε, δ)DP-Greedy Rule List with Smoothed Sensitivity

Input: Dataset $x \in \mathbb{N}^{|\mathcal{X}|}$, Rule set $\mathscr{R}$
Parameters: Min support of the dataset λ, Max length of a rule list K, DP budget (ε, δ), Confidence $\mathcal{C}$
Output: Rule List RL (and noisy counts c_0, c_1)

1: $X_{rem} \leftarrow x$, $R_{rem} \leftarrow \mathscr{R}$, $RL \leftarrow []$, {Initialisation}
   $\Lambda \leftarrow \lfloor \|x\|_1 \times \lambda \rfloor$, Stop $\leftarrow$ False
2: $\mathcal{T} \leftarrow$ `confidence_threshold`$(\mathcal{C})$
3: **while** $RL \cdot$ `size`$() < K$ **and** $\neg$Stop **do**
4:   **if** $\mathcal{M}^{\Delta_1}_{LAPLACE}(X_{rem}, \|\cdot\|_1, \varepsilon_{node}) < \Lambda + \mathcal{T}$ **then**
5:     Stop $\leftarrow$ True
6:   **else**
7:     $\mathcal{G}_{bound} \leftarrow \mathcal{M}^{S^*_{\cdot,\beta}}_{LAPLACE}(\emptyset, \mathscr{G}_{X_{rem}}(\cdot), \varepsilon_{node}, \delta_{node})$
8:     $\mathcal{G}^\star \leftarrow \mathcal{G}_{bound}$ {no rule added gini}
9:     $R^\star \leftarrow \emptyset$, $q^\star \leftarrow$ `pred_DP`$(\emptyset, X_{rem})$
10:     **for** $r \in R_{rem}$ **do**
11:       $\mathcal{G} \leftarrow \mathcal{M}^{S^*_{\cdot,\beta}}_{LAPLACE}(r, \mathscr{G}_{X_{rem}}(\cdot), \varepsilon_{node}, \delta_{node})$
12:       **if** $\mathcal{G} < \mathcal{G}^\star$ **then**
13:         $\mathcal{G}^\star \leftarrow \mathcal{G}$, $R^\star \leftarrow r$
14:         $q^\star \leftarrow$ `pred_DP`(r, X_{rem})
15:       **end if**
16:     **end for**
17:     **if** $R^\star = \emptyset$ **then**
18:       Stop $\leftarrow$ True
19:     **else**
20:       $RL \cdot$ `append`$(R^\star, q^\star)$
21:       `update`$_{DB}(X_{rem} \leftarrow X_{rem} \setminus \mathcal{C}(R^\star))$
22:     **end if**
23:   **end if**
24: **end while**

Algorithm 2 Function `pred_DP` :

Input: Rule r, Remaining samples X_{rem}
Parameters: DP budget (ε, δ)
Output: Prediction q, (c_0 and c_1)

$c_0 \leftarrow \mathcal{M}^{\Delta_1}_{LAPLACE}(r,$ `count_0`$(X_{rem}, \cdot), \varepsilon_{node})$
$c_1 \leftarrow \mathcal{M}^{\Delta_1}_{LAPLACE}(r,$ `count_1`$(X_{rem}, \cdot), \varepsilon_{node})$
$q \leftarrow 0$ **if** $c_0 > c_1$ **else** 1

To solve this issue in the proposed algorithm, we consider a threshold for minimum support that in most cases preserve the DP guarantees. Knowing that counting queries have a global sensitivity of 1, after each split of the dataset, we add Laplace noise $\sim Lap(\frac{\Delta_1 f=1}{\varepsilon})$ to the noisy support. If the noisy support is under a given predefined threshold then we stop here and use the default classification, while otherwise we keep adding rules. To determine the threshold, assume that Λ and ε are fixed and we want a confidence $C = 0.98$. When the added noise is negative (*i.e.*, the noisy support is lower than the exact support), the algorithm does not add any rule even if the smooth sensitivity computation remains exact. However, when the noisy support is above the exact support, we need to assess how large the added noise can be. This can be done by studying the distribution of the Laplace noise to determine at what value t it will be above the confidence C. More precisely, we search for $t > 0$ such that : $\int_{-\infty}^{t} Lap(x|b)\, dx \geq C \iff t \geq -\frac{\log(2)+\log(1-C)}{\varepsilon}$

Algorithm 3 Function `confidence_threshold` :
Input: Confidence C
Parameters: DP budget (ε, δ)
Output: Threshold $\mathcal{T}$

$$\mathcal{T} = \left\lfloor -\frac{\log(2) + \log(1-C)}{\varepsilon_{node}} \right\rfloor + 1$$

The confidence threshold is $\mathcal{T} = 1 + \lfloor t \rfloor$ (Algorithm 3). For instance, with $\varepsilon = 0.1$, and $C = 0.98$, we obtain $t = \lfloor 6.733 \rfloor + 1 = 7$. This means that we can claim with a confidence of 0.98 that if the algorithm decides to add rules, then it respects the minimal support constraint. In practice, the confidence C will only apply to the later rules of the rule list when the number of samples left becomes scarce.

Privacy budget. Let (ε, δ) the total privacy budget allocated to the algorithm. Using the sequential and parallel composition for DP mechanisms, we must determine the fraction of the privacy budget to allocate per node (*i.e.*, how much privacy budget should be allocated for the choice of each rule). We will denote these quantities by ε_{node} and δ_{node}.

Let K the maximum length of a rule list. While it is common for tree-based models to display the counts for each leaf (*i.e.*, in our case for each rule), this information should also be made differentially-private. First in Line 4, the minimum support condition is verified with a global sensitivity by applying the Laplace mechanism (satisfying $(\varepsilon, 0)$-DP). Then, the computation of the Gini impurity (Line 11) is made inside the dataset for each candidate rule and only the rule corresponding to the maximum of these noisy Gini is returned to the algorithm, which is the *Noisy Max Report* mechanism that only accounts for one access. Computing the two noisy counts of the chosen rule (Algorithm 2) also counts only for one access since the sets of samples caught and not caught are disjoint, which leads to the application of the parallel composition. Finally, with sequential composition, it gives us 3 operations per node, with 2 achieving pure DP. For the default rule, only noisy counts are used and no Gini index is computed. Therefore if the counts are not displayed with the model, then the denominator is only $2K - 1$ for ε, which leads to $\varepsilon_{node} = \frac{\varepsilon}{3K-1}$ and $\delta_{node} = \frac{\delta}{K-1}$.

For a Laplace noise using the smooth sensitivity, we can use $\beta = \frac{\varepsilon_{node}}{2\log(2/\delta_{node})}$ for the β-smooth upper bound.

5. Experimental Evaluation

In this section, we assess experimentally the effect of smooth sensitivity on the resulting models' accuracy as compared to other approaches based on the global sensitivity.

5.1. Experimental settings

For our experiments, we consider three common datasets: `German Credit`, `Compas` and `Adult` in their binarized version. Sensitive attributes were removed as their use is prohibited to avoid disparate treatment. In `German Credit` (Dua & Graff, 2017) the classification task is to predict whether individuals have a good or bad credit score. Features are binarized using one-hot encoding for categorical ones and quantiles (2 bins) for numerical ones. The resulting dataset contains $1,000$ samples and we consider 49 premined rules. For `Compas` (Angwin et al., 2016), the objective is to predict whether an individual will re-offend within two years or not. Features are binarized using one-hot encoding for categorical ones and quantiles (with 5 bins) for numerical ones. The resulting dataset contains $6,150$ samples and we have 18 rules. The classification task in `Adult` (Dua & Graff, 2017) is to predict whether an individual earns more than $50,000$\$ per year. Categorical attributes are one-hot encoded and numerical ones are discretized using quantiles (3 bins). The resulting dataset contains $48,842$ samples and we use 47 rules (attributes or their negation).

In our experiments, we build upon the baseline `GreedyRL` implementation proposed by Ferry et al. (2024)[1] and further modify their code to implement our proposed DP mechanisms within the `DP-GreedyRL` algorithm[2]. For each value of ε, the test accuracy was normalized over 100 runs to account for train/test distribution (*i.e.*, train/test split of 70/30) and the randomization due to the application of DP. The value of δ was set $\frac{1}{\|\mathcal{D}\|_1^2}$ and the maximum length for rule lists was set to 5 as we empirically observed that lower values could impede the model accuracy and higher values do not substantially increase accuracy. The hyperparameters were fixed with preliminary grid search leading to $C = 0.99$,

[1] https://github.com/ferryjul/ProbabilisticDatasetsReconstruction
[2] https://gitlab.laas.fr/roc/timothee-ly/dp-greedy

$\lambda = 0.12$ for `German Credit` and $\lambda = 0.05$ for `Compas` and `Adult`. The rules are mined as conjunctions of up to two Boolean attributes or their negation as longer rules make the space exploration exponentially more time consuming. All our experiments are run on an Intel CORE I7-8700 @ 3.20GHz CPU.

5.2. Rule selection with Global Sensitivity on Gini index

At each step of Algorithm 1, the selection of the rule with the best Gini index $R^\star$ (Lines 7 and 11) is implemented by Laplace noise with smooth sensitivity. To evaluate the benefit compared to the global sensitivity, we first determine the best rule based on the global sensitivity when computing the Gini index. Thus, we implemented two versions of the proposed algorithm using global sensitivity.

Noisy Gini. The first version replaces the smooth sensivity of the Gini impurity with its global sensitivity. More precisely, the Laplace noise with global sensitivity is added to the Gini Impurity and there is no need to compute the minimum support (Line 3). Thus some privacy budget is saved during that step.

Noisy counts. The second version leverages the global sensitivity of counting queries (equal to 1) rather than using the global sensitivity of the Gini impurity which is very high. We have therefore used the noisy counts (accessed by a Laplace mechanism) of each rule to compute the Gini index. According to the post-processing property, this quantity remains differentially-private. Nonetheless, this access is not a *Noisy Max Report* mechanism anymore but a regular access to all counts for each rule. This means that the privacy budget per node needs to be further split for each rule of the ruleset $\mathscr{R}$, which leads to a factor of $1/2|\mathscr{R}|$ in the denominator.

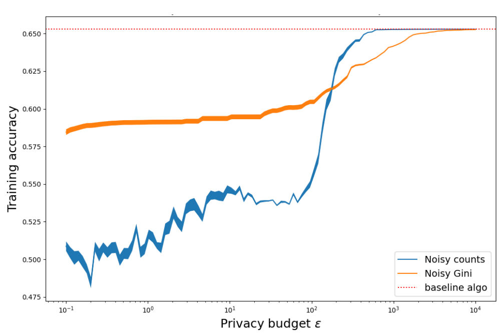


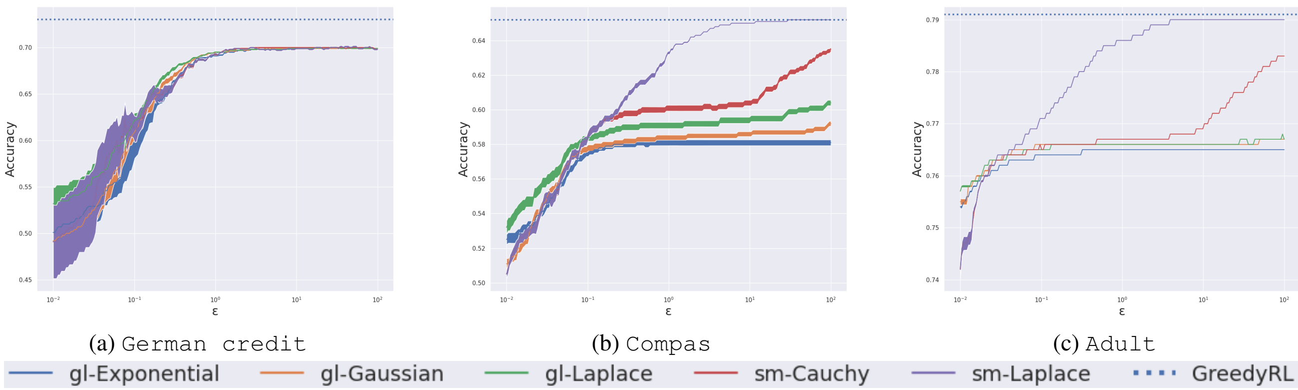
Figure 2. Comparison of Noisy counts and Noisy Gini variants applied on dataset `Compas` using global sensitivity (log-scaled)

In the experiments, we focus on the range $[0.1, 20]$ for ε. In particular, when ε goes over 20, it becomes hard to quantify how the theoretical guarantees apply on realistic settings while a value under 0.1 leads to poorly performing models. Figure 2 shows that overall a rule list model built using the Noisy Gini performs better than the model learnt based on noisy counts but is slower to reach the accuracy of the baseline model obtained with `GreedyRL`. When ε is high enough, the noise added is so low that the Gini impurity scores are ranked according to their original value hence a consistent result with `GreedyRL`. The model using only the noisy counts remains nonetheless interesting in a setting in which the mined ruleset is pre-processed beforehand to a small cardinality (*e.g.* less than a hundred) as this yields the best results of the two. Motivated by these findings, we now focus on the noisy Gini version.

5.3. Prediction Performance

We now compare the test accuracy of rule lists obtained by Algorithm 1 combined with several DP mechanism for the selection of the best rule. We consider two mechanisms based on smooth sensitivity and either Cauchy (`sm-Cauchy`) or Laplace (`sm-Laplace`) noise. We also consider two mechanisms based on global sensitivity and Gaussian (`gl-Gaussian`) or Laplace (`gl-Laplace`) noise. Finally, we implemented the Exponential mechanism using the Gini impurity as the utility function for sampling the best rule at each node.

We vary the privacy budget ε in $[0.01, 100]$. The results, averaged over the 100 runs, are displayed in Figure 3 and the test accuracy for $\varepsilon = 10$ is reported in the right part of Table 1. As shown in Figure 3, the two variants based on smooth sensitivity perform particularly well for relatively large datasets. We observe a high variance on accuracy at low ε, which is mostly due to the confidence threshold becoming exceedingly high for these privacy values as it might lead the model to output only one rule. However, this asymptotic behaviour disappears quickly especially for larger datasets. For $\varepsilon = 0.1$, the mechanisms based on smooth sensitivity either match or outperform the standard pure DP approaches. In addition, for `Compas` and `Adult`, the convergence of the approaches based on smooth sensitivity to the baseline model is very steep. In contrast, DP mechanisms based on the global sensitivity usually converge around $\varepsilon \approx 10^3$. Compared to the differentially-private random forest of Fletcher & Islam (2017), we incur at $\varepsilon = 1$ a significantly lower accuracy loss with respect to the non-private model. The Cauchy distribution has a polynomial decaying tail, which is much heavier than the exponential decaying tail of the Laplace distribution. Thus, on many random noises generated at each step of the algorithm, a few might end up far from the average amplitude, which might deteriorate significantly the accuracy. As a consequence although the smooth Cauchy mechanism provides a good alternative to DP mechanisms based on the global sensitivity, we advise to replace it by its Laplace counterpart even if the privacy guarantees provided are slightly weaker.



(a) `German credit` (b) `Compas` (c) `Adult`

Figure 3. Comparison based on the test accuracy of different DP rule list algorithms.

5.4. Robustness to Privacy Attacks

The protection provided by DP aims at hiding the contribution of any individual example to the output of a computation. Then, it is natural to evaluate it in practice using Membership Inference Attacks (MIAs) (Shokri et al., 2017), whose objective is to determine whether an individual was part of a given model's training set or not. Indeed, performing such attacks on both the original greedy rule lists and their DP counterparts, and comparing the MIA success rate, empirically quantifies the effectiveness of the DP protection. However, this approach has two main drawbacks. First, one has to select which MIA(s) to run, and different attacks can come with different success rates. Second, we implemented and used several popular attacks from the literature, and they struggled attacking even the original (non-DP) model, as reported in the Appendix C.2. An intuitive explanation lies in the simplicity of our considered models: while the output of a deep neural network is a numerical value which can virtually take any value, a rule list classifies an example using one of K rules in which K is reasonably small.

In this paper, we rather leverage the (model-agnostic) notion of *distributional overfitting* of a model, introduced by Yaghini et al. (2019). In a nutshell, it aims at quantifying how the model output distribution varies between samples inside and outside the training set. It is thus highly correlated to the vulnerability of a model to MIAs, and can be seen as an upper-bound over their success. More precisely, for $y \in \{0, 1\}$, we define: $\tau(y) = \frac{1}{2} \sum_{r \in RL} \left| \mathbb{P}[r|y, M = 1] - \mathbb{P}[r|y, M = 0] \right|$ in which $\mathbb{P}[r|y, M]$ is the probability that a sample with label y (from the training set ($M = 1$) or outside ($M = 0$)) is captured by rule $r \in RL$. The overall vulnerability of the model is given by : $V = \frac{1}{2} + \frac{1}{2} \sum_{y \in Classes} \mathbb{P}[y] \times \tau(y)$.

Intuitively, when measured on finite training and test sets, it measures how much the proportions of samples from each possible label differ among the different rules. If the model's outputs have the exact same distributions inside and outside the training set, the vulnerability is 0.5 which indicates that the expected success of a MIA is that of a random guess. We report in Table 1 the overall vulnerabilities measured on rule lists built with or without the use of DP within the greedy learning algorithm. Consistent with our preliminary observations that the greedily-built rule lists are resilient to MIAs, the vulnerabilities of both the DP and non-DP models are very low. Nevertheless, we observe that non-DP models consistently exhibit slightly higher vulnerability values (as expected), than their DP counterparts.

Table 1. Test Accuracy and Overall vulnerability of the greedy rule lists algorithm and its DP counterpart over 100 runs.

Dataset	Method	Vulnerability	Accuracy
`Compas`	`GreedyRL`	$0.507^{+} \pm 4e{-}6$	$0.660 \pm 8e{-}5$
`Compas`	`DP-GreedyRL`	$0.507^{-} \pm 4e{-}6$	$0.658 \pm 1e{-}4$
`German`	`GreedyRL`	$0.524 \pm 3e{-}5$	$0.711 \pm 5e{-}4$
`German`	`DP-GreedyRL`	$0.516 \pm 5e{-}5$	$0.683 \pm 1e{-}3$
`Adult`	`GreedyRL`	$0.502 \pm 7e{-}7$	$0.798 \pm 1e{-}5$
`Adult`	`DP-GreedyRL`	$0.502 \pm 6e{-}7$	$0.795 \pm 1e{-}5$

6. Conclusion

In this paper, we have proposed a new mechanism for DP that leverages the smooth sensitivity of the Gini impurity, directly addressing a key challenge pointed out in the literature (Fletcher & Islam, 2019). Our experiments illustrate that this new mechanism, with equivalent privacy guarantees, offers a considerable reduction of the accuracy loss compared to the differentially-private `GreedyRL` models using global sensitivity. We leave as future work the use of the Gini impurity's smooth sensitivity for the implementation of differentially-private decision trees or other interpretable machine learning models. Exploring the integration of DP on certifiably optimal learning algorithms such as `CORELS` is another promising avenue of research.

Impact Statement

Because machine learning models are increasingly used for high-stakes decision making tasks, it is crucial to ensure that their decisions can be understood by human users. Furthermore, recent texts make this a legal requirement: for instance, for a machine learning model to be compliant with the GDPR legislation, it must satisfy the right to explanation principle which states that the subject to an automatic process should have a right to obtain an explanation on the rationale behind the decision received. While the exact definition of explainability remains a point of contention in the scientific community, interpretable models are a simple yet reliable way of implementing such transparency requirements. Another crucial aspect is to protect the private data that may be used to train such machine learning models, while preserving as much as possible the final model's utility.

Our framework provides a thorough technical solution jointly handling these three aspects. More precisely, we propose novel solutions to enforce strong privacy guarantees with a limited impact on the resulting model's performances. We further demonstrate that these solutions can be used to protect inherently interpretable models which can then be safely released. On the one hand, making these interpretable models differentially-private is a significant step towards ensuring the ethical and responsible use of AI in our society. On the other hand, the technical solutions we provide (namely the use of smooth sensitivity) constitute strong privacy-preserving mechanisms which can be applied in other settings to better conciliate performance and privacy concerns.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, oct 2016.

Angelino, E., Larus-Stone, N., Alabi, D., Seltzer, M., and Rudin, C. Learning Certifiably Optimal Rule Lists. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 35–44, 2017.

Angelino, E., Larus-Stone, N., Alabi, D., Seltzer, M., and Rudin, C. Learning Certifiably Optimal Rule Lists for Categorical Data. *Journal of Machine Learning Research*, 18(234):1–78, 2018.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. propublica (2016). *ProPublica, May*, 23, 2016.

Aziz, H., Cseh, Á., Dickerson, J. P., and McElfresh, D. C. Optimal kidney exchange with immunosuppressants. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pp. 21–29, 2021.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. Classification and Regression Trees. *Biometrics*, 40:874, 1984.

Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramer, F. Membership inference attacks from first principles, 2022.

Chaudhuri, K., Sarwate, A. D., and Sinha, K. A near-optimal algorithm for differentially-private principal components. *J. Mach. Learn. Res.*, 14(1):2905–2943, 2013.

Choquette-Choo, C. A., Tramer, F., Carlini, N., and Papernot, N. Label-only membership inference attacks, 2021.

Domingos, P. A few useful things to know about machine learning. *Commun. ACM*, 55(10):78–87, 2012.

Dua, D. and Graff, C. UCI machine learning repository, 2017.

Dwork, C. and Roth, A. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.*, 9:211–407, 2014.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. D. Calibrating Noise to Sensitivity in Private Data Analysis. In *Proceedings of the Third Theory of Cryptography Conference, TCC, New York, NY, USA, March 4-7*, volume 3876, pp. 265–284, 2006.

Ferry, J., Aïvodji, U., Gambs, S., Huguet, M.-J., and Siala, M. Probabilistic Dataset Reconstruction from Interpretable Models. In *2nd IEEE Conference on Secure and Trustworthy Machine Learning*, Toronto, Canada, April 2024.

Fletcher, S. and Islam, M. A differentially private decision forest. In *Proceedings of the Thirteenth Australasian Data Mining Conference (AusDM 15)*, pp. 99–108, 2015.

Fletcher, S. and Islam, M. Z. Differentially private random decision forests using smooth sensitivity. *Expert Systems with Applications*, 78:16–31, 2017.

Fletcher, S. and Islam, M. Z. Decision Tree Classification with Differential Privacy: A Survey. *ACM Comput. Surv.*, 52(4), 2019.

Gong, M., Xie, Y., Pan, K., Feng, K., and Qin, A. K. A Survey on Differentially Private Machine Learning [Review Article]. *IEEE Comput. Intell. Mag.*, 15(2):49–64, 2020.

Ji, Z., Lipton, Z. C., and Elkan, C. Differential Privacy and Machine Learning: a Survey and Review. *CoRR*, abs/1412.7584, 2014.

Merrer, E. L. and Trédan, G. The bouncer problem: Challenges to remote explainability. *CoRR*, abs/1910.01432, 2019.

Nicolae, M.-I., Sinn, M., Tran, M. N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., Molloy, I., and Edwards, B. Adversarial robustness toolbox v1.2.0. *CoRR*, 1807.01069, 2018.

Nissim, K., Raskhodnikova, S., and Smith, A. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, pp. 75–84, June 2007.

Rivest, R. L. Learning Decision Lists. *Machine Learning*, 2:229–246, 1987.

Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, May 2019.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017.

Singh, C., Nasseri, K., Tan, Y. S., Tang, T., and Yu, B. imodels: a python package for fitting interpretable models, 2021.

Sun, L., Zhou, Y., Yu, P. S., and Xiong, C. Differentially private deep learning with smooth sensitivity. *ArXiv*, abs/2003.00505, 2020.

Voigt, P. and Bussche, A. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. 2017.

Yaghini, M., Kulynych, B., and Troncoso, C. Disparate Vulnerability: on the Unfairness of Privacy Attacks Against Machine Learning. *CoRR*, abs/1906.00389v2, 2019.

Zafarani, F. and Clifton, C. Differentially private naïve bayes classifier using smooth sensitivity. *Proceedings on Privacy Enhancing Technologies*, 2021:406 – 419, 2020.

A. Proof of the Smooth Sensitivity of the Gini Impurity (Theorem 4.1)

A.1. Case 1: for a minimum support of 1

To match with the notations used so far, we will consider a dataset $x \in \mathbb{N}^{|\mathcal{X}|}$ and suppose we take interest at the first node splitting this dataset (it is only a matter of notation), we can therefore rewrite the local sensitivity of the Gini impurity at x as:

$$LS_{\mathcal{G}}(x) = 1 - \left(\frac{\|x\|_1}{\|x\|_1 + 1} \right)^2 - \left(\frac{1}{\|x\|_1 + 1} \right)^2$$

Consider the function

$$g : \left| \begin{array}{ccc} \mathbb{R}^+ & \longrightarrow & [0, 1] \\ x & \longmapsto & 1 - \left(\frac{x}{x+1} \right)^2 - \left(\frac{1}{x+1} \right)^2 \end{array} \right.$$

g is derivable on $\mathbb{R}^+$ and $\forall x \in \mathbb{R}^+$:

$$g'(x) = -2 \times \frac{x}{x+1} \cdot \frac{x+1-x}{(x+1)^2} - 2 \times \frac{1}{x+1} \cdot \frac{-1}{(x+1)^2} = \frac{-2x}{(x+1)^3} + \frac{2}{(x+1)^3} = \frac{2(1-x)}{(x+1)^3}$$

x	0		1		$+\infty$
$g'(x)$		$+$	0	$-$	
g	0	$\nearrow$	$\frac{1}{2}$	$\searrow$	0

Note that : $LS_{\mathcal{G}} \equiv g \circ \|\cdot\|_1$

As a reminder, we are trying to determine the smooth sensitivity of the Gini impurity:

$$S^*_{\mathcal{G},\beta}(x) = \max_{k \in \mathbb{N}} \; e^{-k\beta} \mathcal{T}_k(x)$$

where

$$\mathcal{T}_k(x) = \max_{\substack{y \in \mathbb{N}^{|\mathcal{X}|} \\ ||y-x||_1 \leq k}} LS_{\mathcal{G}}(y) = \max_{\substack{y \in \mathbb{N}^{|\mathcal{X}|} \\ ||y-x||_1 \leq k}} g \circ \|y\|_1 = \max_{\substack{y \in \mathbb{N} \\ y \in [\|x\|_1 - k, \|x\|_1 + k]}} g(y)$$

We consider that $\|x\|_1 \geq 1$ as we do not build nodes when there are no samples to classify. We could eventually replace this condition by $\|x\|_1 \geq \lambda n$ where λ is the minimum support (in terms of ratio) and n the total size of the dataset.

$[\|x\|_1 - k, \|x\|_1 + k]$ is an interval with integer bounds. With the previous study of g monotonicity, this maximum is reached in $y = \max(1, \|x\|_1 - k)$.

Explanation:

- if $k \geq \|x\|_1 \geq 1$ then $1 \in [\|x\|_1 - k, \|x\|_1 + k]$ so the maximum is the global maximum of g : $1 = \max(1, \|x\|_1 - k)$
- if $k < \|x\|_1$ then $[\|x\|_1 - k, \|x\|_1 + k] \subset [1, +\infty]$ and g is monotonously decreasing on $[1, +\infty[$ so the maximum is the leftmost bound of the interval : $\|x\|_1 - k = \max(1, \|x\|_1 - k)$

$$\mathcal{T}_k(x) = g\big[\max(1, \|x\|_1 - k)\big]$$

Now that we obtained a close formula for $\mathcal{T}_k(x)$, we can determine :

$$S^*_{\mathcal{G},\beta}(x) = \max_{k \in \mathbb{N}} \; e^{-k\beta} \mathcal{T}_k(x) = \max_{k \in \mathbb{N}} \; e^{-k\beta} \cdot g\big[\max(1, \|x\|_1 - k)\big]$$

Let

$$\xi_{x,\beta}(t) : \left| \begin{array}{ccc} \mathbb{R}^+ & \longrightarrow & \mathbb{R}^+ \\ t & \longmapsto & e^{-t\beta} \cdot g\big[\max(1, \|x\|_1 - t)\big] \end{array} \right.$$

$$\xi_{x,\beta}(t) = \begin{cases} e^{-t\beta} \cdot g(1) & \text{if } t \geq \|x\|_1 - 1 \\ e^{-t\beta} \cdot g(\|x\|_1 - t) & \text{if } t \leq \|x\|_1 - 1 \end{cases}$$

$\xi_{x,\beta}$ is continuous on $\mathbb{R}^+$ and derivable on $[0, \|x\|_1 - 1[$ and $]\, \|x\|_1 - 1, +\infty[$. The monotonicity of $\xi_{x,\beta}$ is trivial for high values of t:

$$\forall t \in]\, \|x\|_1 - 1, +\infty[, \xi'_{x,\beta}(t) = -\beta \times e^{-t\beta} g(1) < 0$$

$\forall t \in [0, \|x\|_1 - 1[$,

$$\begin{aligned} \xi'_{x,\beta}(t) &= -\beta \times e^{-t\beta} g(\|x\|_1 - t) + e^{-t\beta} \times (-1) \times g'(\|x\|_1 - t) \qquad y := \|x\|_1 - t \\ &= -e^{-t\beta}\big[\beta g(y) + g'(y)\big] \\ &= -e^{-t\beta}\Big[\beta\Big(1 - \frac{y^2}{(y+1)^2} - \frac{1}{(y+1)^2}\Big) + \frac{2(1-y)}{(y+1)^3}\Big] \\ &= -e^{-t\beta} \times \frac{\beta \cdot (y+1)^3 - \beta \cdot y^2(y+1) - \beta \cdot (y+1) + 2(1-y)}{(y+1)^3} \\ &= \frac{e^{-t\beta}}{(1+y)^3} \times \Big[-\beta \cdot (y+1)^3 + \beta \cdot y^2(y+1) + \beta \cdot (y+1) - 2(1-y) \Big] \end{aligned}$$

Since $\dfrac{e^{-t\beta}}{(1+y)^3} > 0$ the sign of $\xi'_{x,\beta}(t)$ on $[0, \|x\|_1 - 1[$ is given by the polynomial $P(Y) = -\beta \cdot (Y+1)^3 + \beta \cdot Y^2(Y+1) + \beta \cdot (Y+1) - 2(1-Y) = -2\beta Y^2 + (2 - 2\beta)Y - 2$.

Let $Q := -\beta Y^2 + (1-\beta)Y - 1 = P/2$. P and Q share the same roots, we will therefore study Q. Let Δ the discriminant of polynomial Q. We associate it to the function $\Delta(\beta)$ since its value depends on β. The value of the discriminant gives whether or not the underlying function is monotonous. $\Delta(\beta) = (1-\beta)^2 - 4\beta = (\beta - 3 - 2\sqrt{2})(\beta - 3 + 2\sqrt{2})$

β	0		$\beta_1 := 3 - 2\sqrt{2}$		$\beta_2 := 3 + 2\sqrt{2}$		$+\infty$
$\Delta(\beta)$		$+$	0	$-$	0	$+$	

- For $\beta \in]3 - 2\sqrt{2}, 3 + 2\sqrt{2}[$, $\Delta(\beta) < 0$ so Q has no roots in $\mathbb{R}$ so it is negative on $\mathbb{R}$.

t	0		$\|x\|_1 - 1$		$+\infty$
$\xi'_{x,\beta}(t)$		$-$	$-$	$-$	
$\xi_{x,\beta}$	$g(\|x\|_1)$	$\searrow$	$\frac{\exp(-(\|x\|_1 - 1)\beta)}{2}$	$\searrow$	0

  In that scenario, $\boxed{S_{\mathcal{G},\beta}(x) = \xi_{x,\beta}(0) = g(\|x\|_1) = LS_{\mathcal{G}}(x)}$

- For $\beta = 3 - 2\sqrt{2}$ or $\beta = 3 + 2\sqrt{2}$, $\Delta(\beta) = 0$ so Q admits a unique root $y_0 = \frac{1-\beta}{2\beta}$ (we will ignore these two values of β as there are enough β that we can choose.)

- For $\beta \in]0, 3-2\sqrt{2}[\cup]3+2\sqrt{2}, +\infty[$, $\Delta(\beta) > 0$ so Q admits two distinct roots :

$$y_1 = \frac{1-\beta+\sqrt{(1-\beta)^2-4\beta}}{2\beta} \qquad \text{and} \qquad y_2 = \frac{1-\beta-\sqrt{(1-\beta)^2-4\beta}}{2\beta}$$

y	$-\infty$		y_2		y_1		$+\infty$
$Q(y)$		$-$	0	$+$	0	$-$	

The problem is that the roots $t_1 := \|x\|_1 - y_1$ and $t_2 := \|x\|_1 - y_2$ might overflow the interval $[0, \|x\|_1 - 1[$.

$$y_1 \underset{\beta\to0}{\sim} \frac{1}{\beta} \xrightarrow[\beta\to0]{} +\infty \qquad \text{and} \qquad y_1 \underset{\beta\to+\infty}{\sim} \frac{1}{1-\beta} \xrightarrow[\beta\to+\infty]{} 0^-$$

$$y_2 \underset{\beta\to0}{\sim} \frac{1}{(1-\beta)^2} \xrightarrow[\beta\to0]{} 1 \qquad \text{and} \qquad y_2 \underset{\beta\to+\infty}{\sim} -1$$

β	0		$3-2\sqrt{2}$		$3+2\sqrt{2}$		$+\infty$
$y_1(\beta)$	$+\infty$	$\searrow$	$3 > \cdot > 2$		$-1 < \cdot < 0$	$\nearrow$	1^-
$y_2(\beta)$	1	$\nearrow$	$3 > \cdot > 2$		$-1 < \cdot < 0$	$\searrow$	-1

Since $y \mapsto \|x\|_1 - y =: t$ is a strictly decreasing function (it is a bijection from $\mathbb{R}$ to $\mathbb{R}$) we have that $y_2 < y_1 \implies t_2 > t_1$. What we want to study is the mapping from $[y_2, y_1]$ to $[t_1, t_2]$ with respect to the domain of validity for the studied form of $\xi_{x,\beta}$.

That gives us two cases to treat:

1. $\beta \in]0, \beta_1[$. In the case that $\|x\|_1 \geq 5$ *e.g.* (which is a reasonable assumption) $\exists \beta^* \in]0, \beta_1[, \forall \beta \geq \beta^*, 0 < t_1(\beta) < \|x\|_1 - 1$ and $0 < t_2 < \|x\|_1 - 1$ (for all β in the considered interval) which gives : $0 < t_1 < t_2 < \|x\|_1 - 1$
   - So if β is too small, then the t's associated to the interval $[y_2, y_1]$ are (< 0) partly outside the domain of validity which yields

t	t_1		0		t_2		$\|x\|_1 - 1$		$+\infty$
$Q(t)$	0		$+$		0	$-$			
$\xi_{x,\beta}$			$g(\|x\|_1)$	$\nearrow$	$\xi_{x,\beta}(t_2)$	$\searrow$			0

Hence :

$$\boxed{\begin{aligned} S^*_{\mathcal{G},\beta}(x) &= \max\left[\xi_{x,\beta}(\lfloor t_2 \rfloor), \xi_{x,\beta}(\lceil t_2 \rceil)\right] \\ S^*_{\mathcal{G},\beta}(x) &= \max\left[e^{-\lfloor t_2 \rfloor \beta} g(\|x\|_1 - \lfloor t_2 \rfloor), e^{-\lceil t_2 \rceil \beta} g(\|x\|_1 - \lceil t_2 \rceil)\right] \end{aligned}}$$

– if $\beta \in]\beta^*, \beta_1[$, then all the t's associated to $[y_2, y_1]$ are in the domain of validity.

t	0		t_1		t_2		$\|x\|_1 - 1$		$+\infty$
$Q(t)$		$-$	0	$+$	0	$-$			
$\xi_{x,\beta}$	$g(\|x\|_1)$	$\searrow$	$\xi_{x,\beta}(t_1)$	$\nearrow$	$\xi_{x,\beta}(t_2)$	$\searrow$		$\searrow$	0

$$\boxed{\begin{aligned} S^*_{\mathcal{G},\beta}(x) &= \max\left[\xi_{x,\beta}(0), \xi_{x,\beta}(\lfloor t_2 \rfloor), \xi_{x,\beta}(\lceil t_2 \rceil)\right] \\ S^*_{\mathcal{G},\beta}(x) &= \max\left[g(\|x\|_1), e^{-\lfloor t_2 \rfloor \beta} g(\|x\|_1 - \lfloor t_2 \rfloor), e^{-\lceil t_2 \rceil \beta} g(\|x\|_1 - \lceil t_2 \rceil)\right] \end{aligned}}$$

2. $\beta \in]\beta_2, +\infty[$. $t_2 > \|x\|_1 - 1$ and $t_1 > \|x\|_1 - 1$ which means that the t's associated to the $[y_2, y_1]$ are ($> \|x\|_1 - 1$) all outside the domain of validity.

t	0		$\|x\|_1 - 1$		t_1		t_2		$+\infty$
$Q(t)$		$-$							
$\xi_{x,\beta}$	$g(\|x\|_1)$	$\searrow$		$\searrow$		$\searrow$		$\searrow$	0

$$\boxed{S^*_{\mathcal{G},\beta}(x) = \xi_{x,\beta}(0) = g(\|x\|_1) = LS_{\mathcal{G}}(x)}$$

A.2. Case 2: for a minimum support $\lambda \cdot n > 1$

Let $\Lambda = \lambda \cdot n$ and suppose it an integer to simplify the notations.

$$\mathcal{T}_k(x) = g\left[\max(\Lambda, \|x\|_1 - k)\right]$$

$$S^*_{\mathcal{G},\beta}(x) = \max_{k \in \mathbb{N}} \ e^{-k\beta} \mathcal{T}_k(x) = \max_{k \in \mathbb{N}} \ e^{-k\beta} \cdot g\left[\max(\Lambda, \|x\|_1 - k)\right]$$

Let

$$\xi_{x,\beta}(t) : \left| \begin{array}{ccc} \mathbb{R}^+ & \longrightarrow & \mathbb{R}^+ \\ t & \longmapsto & e^{-t\beta} \cdot g\left[\max(\Lambda, \|x\|_1 - t)\right] \end{array} \right.$$

$$\xi_{x,\beta}(t) = \begin{cases} e^{-t\beta} \cdot g(\Lambda) & \text{if } t \geq \|x\|_1 - \Lambda \\ e^{-t\beta} \cdot g(\|x\|_1 - t) & \text{if } t \leq \|x\|_1 - \Lambda \end{cases}$$

$\xi_{x,\beta}$ is continuous on $\mathbb{R}^+$ and derivable on $[0, \|x\|_1 - \Lambda[$ and $]\ \|x\|_1 - \Lambda, +\infty[$. $\xi_{x,\beta}$ derivatives remain unchanged but the bounds are shifted (from 1 to Λ). The roots y_1 and y_2 are unchanged (they solely depend on β). Instead of re-doing the case per case analysis, we will propose the following heuristic :

1. Compute t_1 and t_2 if the roots y_1 and y_2 exist

2. Compute the relative positions of t_1 and t_2 with respect to 0 and $\|x\|_1 - \Lambda$

3. We know that $\xi_{x,\beta}$ is increasing between t_1 and t_2 granted that they are in the $[0, \|x\|_1 - \Lambda[$ interval so there is an eventual max in this interval, to compare to $\xi_{x,\beta}(0)$ and $\xi_{x,\beta}(\|x\|_1 - \Lambda)$

B. Some key results for Differential Privacy

B.1. Distance between Databases

The datasets used in this article are tabular, features are $0-1$ encoded and the label is also binary. Suppose that an element of the dataset is made of m features and one label. Then, the universe of all possible elements of the dataset, denoted $\mathcal{X}$ is therefore finite of cardinality 2^{m+1}. An element a of $\mathcal{X}$ can be expanded to its tuple form as $(a_1, \dots, a_m, a_{m+1})$ where a_{m+1} is the label. We define the order relation $\preceq$ on $\mathcal{X}$. For $(a, b) \in \mathcal{X}$,

$$a \preceq b \iff \begin{cases} \exists i \in [\![1, m+1]\!], \forall k \in [\![1, i-1]\!],\ a_k \leq b_k \text{ and } a_i < b_i \\ or \\ \forall i \in [\![1, m+1]\!], a_i = b_i \end{cases}$$

$\preceq$ yields the symmetric, reflexive and transitive properties and all elements can be compared within $\mathcal{X}$ so this is a total order relation. As such, $(\mathcal{X}, \preceq)$ is a totally ordered set. We can now introduce the expanded notation for datasets. A dataset x is a collection of elements of $\mathcal{X}$ that we write as a tuple $x = (x_0, \dots, x_{|\mathcal{X}|}) \in \mathbb{N}^{|\mathcal{X}|}$ such that x_i denotes the number of elements of $\mathcal{X}$ of type i stored in the database x. The number of elements in a dataset x is given by the formula: $\|x\|_1 := \sum_{i=0}^{|\mathcal{X}|} x_i$.

With this notation, it is easy to interpret the notion of distances between dataset as the $L1$-norm of their difference. We say that two dataset x, y are adjacent if they vary only by 1 element *i.e.* $||x - y||_1 = 1$.

B.2. Composition and Post-Processing Properties

The DP-mechanisms presented above possess nice properties to use them in conjunction. DP would not yield any relevance were the entity using the privatized data able to untangle it. One cannot make a differentially-private algorithm less private in post processing. This is the guarantee provided by the *Post-Processing theorem*.

Theorem B.1 (Post-processing theorem)**.** *Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{Y}$ be an (ε, δ)- differentially private algorithm. For any function $f : \mathcal{Y} \to \mathcal{Z}$, the composition $f \circ \mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{Z}$ is (ε, δ)-DP.*

As stated above, the differentially-private mechanisms we presented all apply on a $\mathbb{R}^k$ valued function. *composition of differentially-private mechanisms* enables us to scale up from functions to algorithms. Composition tends to deteriorate the privacy guarantees but to a measurable extent. It all depends on how the composition is applied.
Sequential composition happens when when several differentially-private mechanisms, denoted $m_1, \dots, m_p$ with respective DP-coefficients $(\varepsilon_1, \dots, \varepsilon_p)$ are applied onto the same dataset x then the generated output : $(m_1(x), \dots, m_p(x))$ satisfies $(\sum_{i=1}^{p} \varepsilon_1)$-DP.

For *Parallel composition*, the differentially-private mechanisms denoted $m_1, \dots, m_p$ are applied into disjoints subsets of a given dataset $x = \amalg_{i=1}^{p} x_i$ then the generated output : $(m_1(x), \dots, m_p(x))$ satisfies $(\max_{i=1}^{p} \varepsilon_i)$-DP.

B.3. Proof of the iterative Computation Lemma of Smooth Sensitivity (Lemma 3.1, from (Nissim et al., 2007))

Let $\mathcal{D}$ and $\mathcal{D}'$ denote two datasets. Note that since : $\{\mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|} : \|\mathcal{D}' - \mathcal{D}\|_1 \leq k\} \subset \{\mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|} : \|\mathcal{D}' - \mathcal{D}\|_1 \leq k+1\}$ we have that $\forall k \in \mathbb{N}, \mathcal{T}_{k+1}(\mathcal{D}) \geq \mathcal{T}_k(\mathcal{D})$.

$$\begin{aligned}
S^*_{f,\beta}(\mathcal{D}) &= \max_{\mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|}} LS_f(\mathcal{D}') e^{-\beta||\mathcal{D}-\mathcal{D}'||_1} \\
&= \max_{k \in \{0,\dots,n\}} \max_{\substack{\mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|} \\ ||\mathcal{D}-\mathcal{D}'||_1 = k}} LS_f(\mathcal{D}') e^{-\beta||\mathcal{D}-\mathcal{D}'||_1} \\
&= \max_{k \in \{0,\dots,n\}} e^{-\beta k} \max_{\substack{\mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|} \\ ||\mathcal{D}-\mathcal{D}'||_1 = k}} LS_f(\mathcal{D}') \\
&= \max_{k \in \{0,\dots,n\}} e^{-\beta k} \mathcal{T}_k(\mathcal{D})
\end{aligned}$$

The transition from the penultimate to the final line is tricky. $\mathcal{T}_k(\mathcal{D})$ is a max over the ball of elements at distance at most k of $\mathcal{D}$, not the sphere. Note that since we are using dataset, the distance can only be an integer.

$$\begin{aligned}
\mathcal{T}_{k+1}(\mathcal{D}) &= \max(\max_{\substack{\mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|} \\ ||\mathcal{D}'-\mathcal{D}||_1 < k+1}} LS_f(\mathcal{D}'), \max_{\substack{\mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|} \\ ||\mathcal{D}'-\mathcal{D}||_1 = k+1}} LS_f(\mathcal{D}')) \\
&= \max(\max_{\substack{\mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|} \\ ||\mathcal{D}'-\mathcal{D}||_1 \leq k}} LS_f(\mathcal{D}'), \max_{\substack{\mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|} \\ ||\mathcal{D}'-\mathcal{D}||_1 = k+1}} LS_f(\mathcal{D}')) \\
&= \max(\mathcal{T}_k(\mathcal{D}), \max_{\substack{\mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|} \\ ||\mathcal{D}'-\mathcal{D}||_1 = k+1}} LS_f(\mathcal{D}'))
\end{aligned}$$

Since $\beta > 0$, $e^{-\beta k} > e^{-\beta(k+1)}$ therefore $e^{-\beta k}\mathcal{T}_k(\mathcal{D}) > \mathcal{T}_k(\mathcal{D})e^{-\beta(k+1)}$. But the quantity, $e^{-\beta k}\mathcal{T}_k(\mathcal{D})$ appears in the computation of $S^*_{f,\beta}(\mathcal{D})$ and since it is strictly greater than the left term of $\mathcal{T}_{k+1}(\mathcal{D})$ we can ignore this term and it is equivalent to compute $LS_f(\mathcal{D}')$ either on the ball or on the sphere of radius k in that case.

C. Additional Results

C.1. Additional Figures for Section 5.2

Figure 4 provides more results of the comparison between the two methods leveraging global sensitivity to output the best rule. The two plots follow the same tendencies as for dataset `compas` that is to say, the method using the global sensitivity of the Gini Impurity remains the best choice for the interval of ε considered.

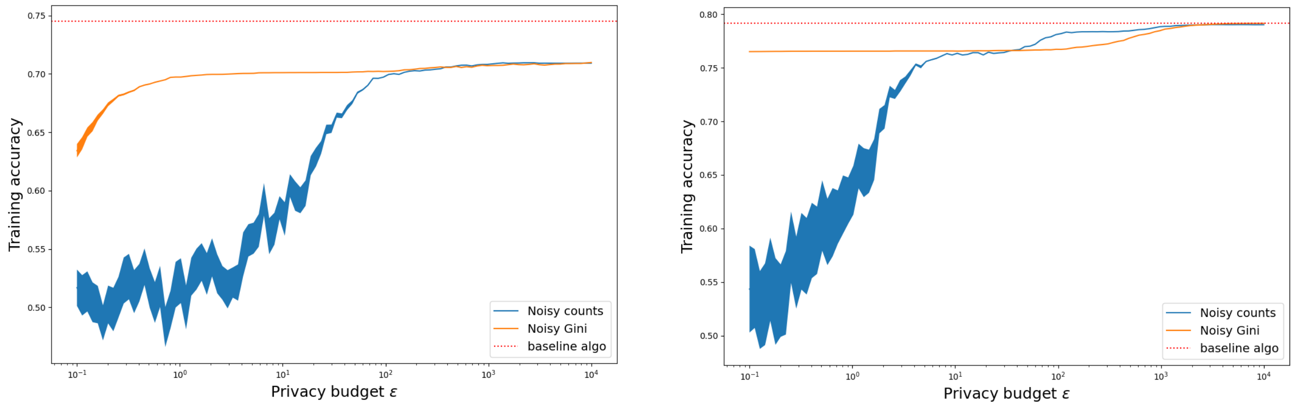
C.2. Membership Inference Attacks

Figure 5 illustrates how a dataset is split into different subsets to train a Membership Inference Attack (MIA) model. Note that the \ symbol represents the minus operation on sets.

We consider two MIAs from the popular ART toolkit [3] (Nicolae et al., 2018). The results we obtained for a black-box MIA[4] using Random Forests are presented in Figure 6. The ROC curves are displayed in log scale to highlight the results at low FPR since it is the relevant regime for Membership Inference Attacks (Carlini et al., 2022). They show that, as mentioned in Section 5.4, (even non-DP) rule lists are already resilient to MIAs. On the smallest `German credit` dataset, we observed a slightly higher distributional overfitting, which results in slightly higher TPRs at low FPR.

[3] `https://github.com/Trusted-AI/adversarial-robustness-toolbox/wiki/ART-Attacks#4-inference-attacks`

[4] `https://adversarial-robustness-toolbox.readthedocs.io/en/latest/modules/attacks/inference/membership_inference.html#membership-inference-black-box`



(a) `German credit` (b) `adult`

Figure 4. Comparison of Noisy counts and Noisy Gini variants using global sensitivity (log-scaled)

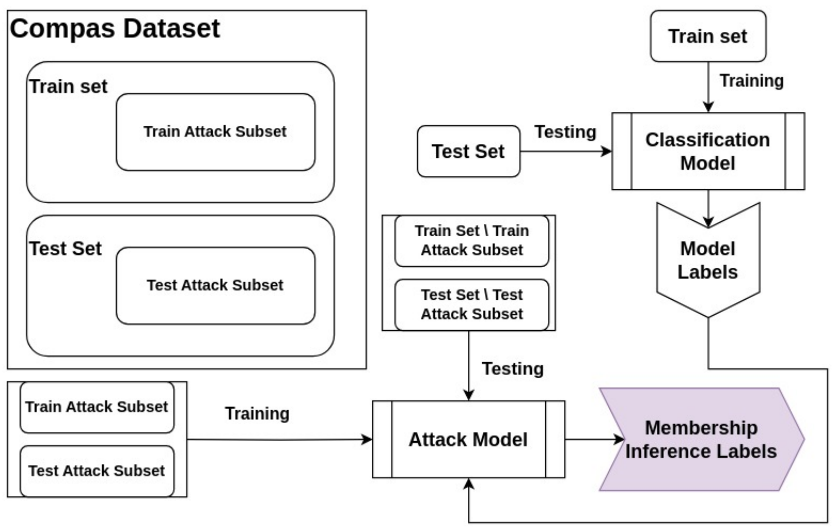
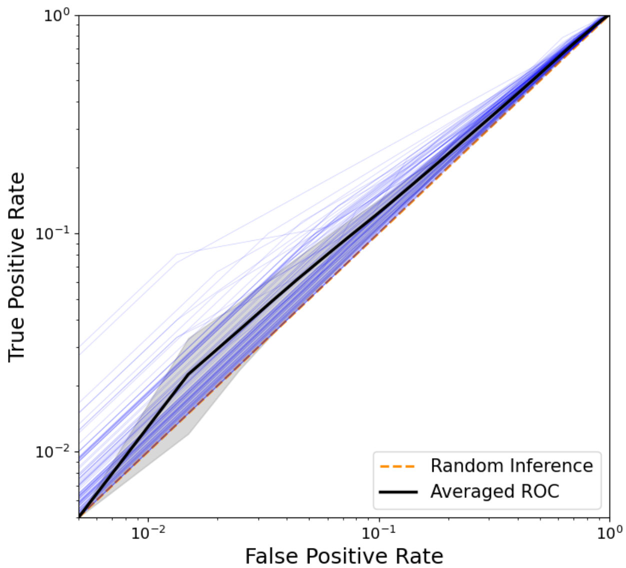


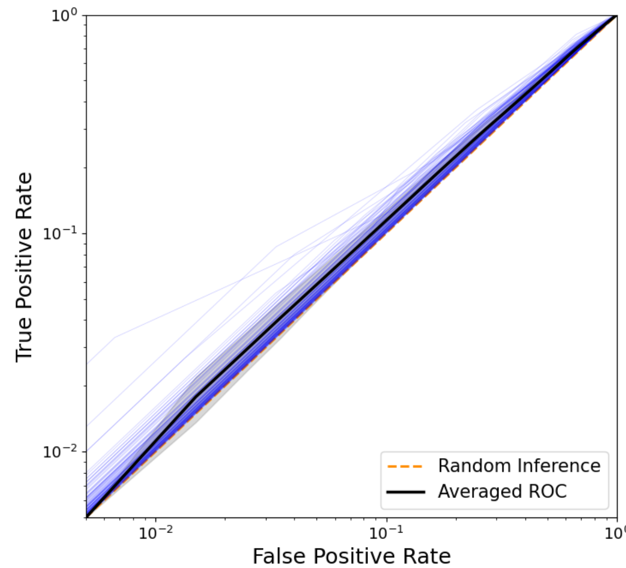
Figure 5. Pipeline of Membership Inference Attack

We also considered the Label Only Membership Inference Attack[5] (Choquette-Choo et al., 2021) but results were sub-par due to the datasets used. Indeed, the rule lists use as input binarized features whereas the attack explores the latent variables space by studying how the model output varies when the features values are tweaked. The issue here is that the model can only read features that are 0 or 1 and therefore we had to truncate the latent space exploration to the much sparser space of $\{0, 1\}^m$, making it inefficient. In addition, since the datasets are binarized, some features are actually a one-hot-encoding of a categorical feature, which means it does not make sense that several of them can be set to 1. An interesting avenue of research would be to use the latent space exploration on the non binarized features and re-apply the binarization process at each step. This is unfortunately computationally expensive and we leave it as future research.

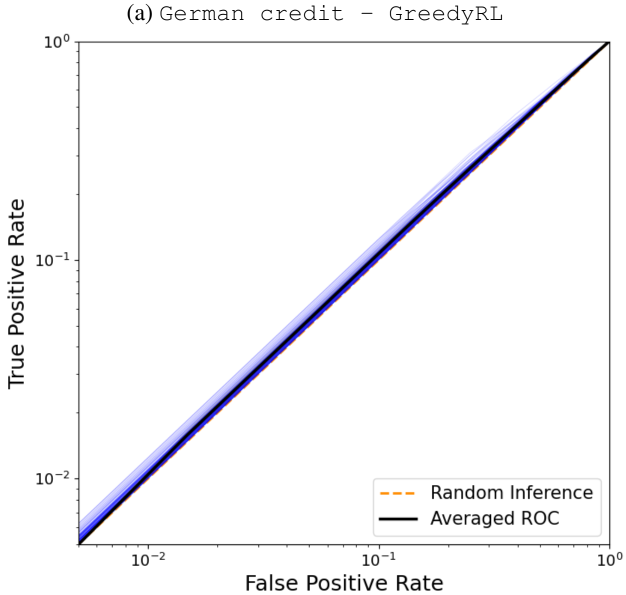
[5] `https://adversarial-robustness-toolbox.readthedocs.io/en/latest/modules/attacks/inference/membership_inference.html#membership-inference-label-only-decision-boundary`



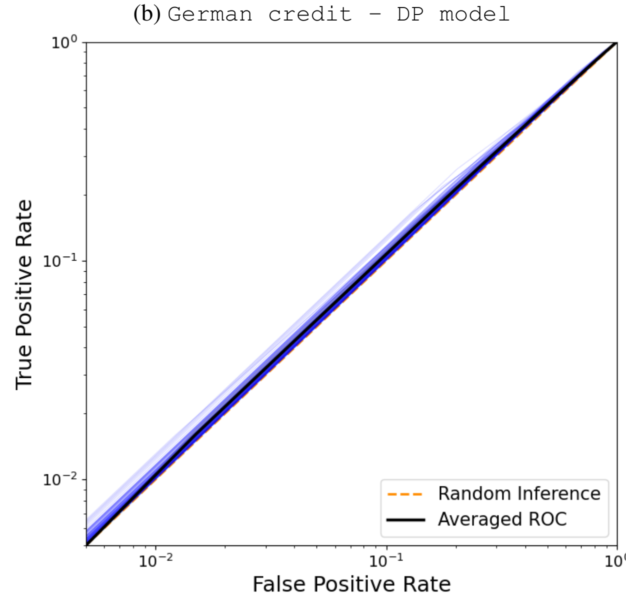
(a) `German credit - GreedyRL`



(b) `German credit - DP model`



(c) `Compas - GreedyRL`



(d) `Compas - DP model`

Figure 6. ROC Curves of Membership Inference Attacks on the DP model and on the baseline `GreedyRL`