Differentially Private Random Forests for Regression and Classification

Shorya Consul [1], **Sinead Williamson** [2, 3]

[1] Department of Electrical and Computer Engineering, The University of Texas at Austin
[2] Department of Statistics and Data Science, The University of Texas at Austin
[3] Department of Information, Risk and Operations Management, The University of Texas at Austin
shoryaconsul@utexas.edu, sinead.williamson@mccombs.utexas.edu

Abstract

Random forests are a popular method for classification and regression due to their versatility. However, this flexibility can come at the cost of user privacy, since training random forests requires multiple data queries, often on small, identifiable subsets of the training data. Privatizing these queries typically comes at a high utility cost, in large part because we are privatizing queries on small subsets of the data, which are easily corrupted by added noise. In this paper, we propose DiPriMe forests, a novel tree-based ensemble method for regression and classification problems, that ensures differential privacy while maintaining high utility. We construct trees based on a privatized version of the median value of attributes, obtained via the exponential mechanism. The use of the noisy median encourages balanced leaf nodes. This avoids the need to query very small subsets of the data, and ensures that the noise added to the parameter estimate at each leaf is not overly large. The resulting algorithm, which is appropriate for real or categorical covariates, exhibits high utility while ensuring differential privacy.

1 Introduction

The prevalence of data has been one of the key drivers of technological innovation in the last decade. The abundance of data, allied with ever-increasing computing power, has driven the rapid development of sophisticated machine learning techniques, many of which have achieved hitherto unseen levels of performance. Data collection today is pervasive, across applications and devices. This has resulted in data privacy becoming a matter of public concern.

It has long been known that querying even aggregated or perturbed data can lead to leakage of private information (Dinur and Nissim 2003), motivating the development of databases and algorithms that mitigate such privacy breaches. Differential privacy (Dwork et al. 2006) is one of the most rigorous ways of analysing and ameliorating such privacy risks. If an algorithm is ϵ-differentially private, it means we can apply a multiplicative bound to the worst-case leakage of an individual's private information. Many algorithms have been developed with this goal in mind, such as differentially private variants of linear regression (Kifer, Smith, and Thakurta 2012), k-means clustering (Huang and Liu 2018; Su et al. 2016) and expectation maximization (Park et al. 2016).

Such privacy guarantees come at a cost—modifying an algorithm to be ϵ-differentially private typically involves adding noise to any queries made by that algorithm, which will tend to negatively affect the algorithm's performance. This cost will tend to be higher when privatizing more complex algorithms that require multiple queries of the data, such as non-linear regression and classification algorithms (Smith et al. 2018; Abadi et al. 2016). One such family of non-linear regression and classification algorithms, that allows for flexible modeling but involves multiple queries, is the class of tree-based methods, such as random forests and their variants (Breiman 2001; Geurts, Ernst, and Wehenkel 2006).

Tree-based ensemble methods make minimal assumptions on the parametric forms of relationships within the data, and can be easily applied to a mixture of continuous and categorical covariates. However, building trees that capture the appropriate structure requires many queries of the data, making them challenging to privatize. Further, since the trees partition the data into arbitrarily small subsets, the noise that must be added to each subset to ensure differential privacy can quickly swamp the signal.

We propose differentially private median (DiPriMe) forests, a novel, differentially private machine learning algorithm for nonlinear regression and classification with potentially sensitive data. Rather than directly privatize queries in an existing random forest framework, we start by modifying the underlying non-private trees to be robust to the addition of noise. We note that the negative impact of added noise is greatest when there are few data points associated with certain leaf nodes, especially in regression. Splitting based on the median value at a node avoids such a scenario. The low sensitivity of the median to perturbations of the data mean that, even after privatizing the median query, we still achieve balanced leaf node occupancy, and therefore avoid overwhelming the signal at each leaf with noise.

The DiPriMe algorithm offers several advantages over existing private random forests. Because the underlying non-private algorithm is designed with privatization in mind, they achieve impressive predictive performance across a range of tasks by minimizing the negative effects of added noise. The fact that their construction explicitly depends on the distribution of the covariates facilitates the derivation of utility

bounds. Such bounds are challenging to derive for algorithms without this form of dependence, since the utility is highly dependent on the data distribution; as a result, this is to our knowledge the first utility result for a fully differentially private tree-based algorithm. And finally, unlike most existing approaches, they can easily deal with continuous or categorical covariates, without excessive additional privacy cost.

We begin by reviewing the concept of differential privacy, and discussing related approaches, in Sec. 2, before introducing DiPriMe forests in Sec. 3. We provide theoretical and empirical guarantees on both the privacy and the utility of our approach. In Sec. 4 we show that our method outperforms existing differentially private tree-based algorithms on a variety of classification and regression tasks.

2 Preliminaries

2.1 Differential Privacy

Differential privacy (DP, Dwork 2008) is a rigorous framework for limiting the amount of information that can be inferred from an individual's inclusion in a database. Formally, a randomized mechanism $\mathcal{F}$ satisfies ϵ-differential privacy for all datasets D_1 and D_2 differing on at most one element and all $\mathcal{S} \subseteq Range(\mathcal{F})$ if

$$\Pr[\mathcal{F}(D_1) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\mathcal{F}(D_2) \in \mathcal{S}]\,. \tag{1}$$

This implies that the inclusion of an individual's data can change the probability of any given outcome by at most a multiplicative factor of e^{ϵ}. A lower value of ϵ provides a stronger privacy guarantee, as it limits the effect the omission of a data point can have on the statistic.

Typically, the mechanism $\mathcal{F}$ is a randomized form of some deterministic query f. To determine the degree of randomization required to satisfy (1), we need to know the global sensitivity $\Delta(f) = \max_{D_1,D_2} \|f(D_1) - f(D_2)\|_1$, which tells us the maximum change in the outcome of the query f due to changing a single data point. Armed with the global sensitivity, we can use a number of approaches to ensure ϵ-DP; we outline the two most common mechanisms below.

- **Laplace mechanism** Starting with a deterministic query $f : \mathcal{D} \to \mathbb{R}^d$, where $\mathcal{D}$ is the space of possible data sets, we can construct an ϵ-DP query-answering mechanism that adds appropriately scaled Laplace noise, so that (Dwork, Roth et al. 2014)

$$\begin{aligned} \mathcal{F}(X) =& f(X) + (Y_1, Y_2, \ldots, Y_d) \\ Y_i \overset{iid}{\sim}& \text{ Laplace}(0, \Delta_i(f)/\epsilon), \end{aligned} \tag{2}$$

  where $\Delta_i(f)$ denotes the sensitivity of the i-th coordinate of the output. Note how the noise added scales as $1/\epsilon$ – increased privacy directly translates to increased noise variance.

- **Exponential mechanism** The Laplace mechanism assumes that our query returns values in $\mathbb{R}^d$. A more generally applicable privacy mechanism is the exponential mechanism (McSherry and Talwar 2007), which allows us to pick an outcome $r \in \mathcal{R}$, where $\mathcal{R}$ is some arbitrary space. We define a scoring function $q : \mathcal{D} \times \mathcal{R} \to \mathbb{R}$ with global sensitivity $\Delta(q)$, and a base measure μ on $\mathbb{R}$. For any dataset $D \in \mathcal{D}$, selecting an outcome r with probability

$$\Pr[\mathcal{F}(D) = r] \propto e^{\epsilon q(D,r)/2\Delta(q)} \times \mu(r) \tag{3}$$

  ensures ϵ-differential privacy. Clearly, the scoring function q should be constructed such that preferred outputs are assigned higher scores.

Often, algorithms will involve multiple queries, requiring us to account for the overall differential privacy of the composite algorithm. We can make use of two composition theorems to obtain the overall privacy level (McSherry 2009):

- **Sequential composition** tells us that a sequence of differentially private queries maintains differential privacy. Let $\mathcal{F}_i$ each provide ϵ_i-differential privacy. Then sequentially evaluating $\mathcal{F}_i(X)$ provides $(\sum_i \epsilon_i)-$differential privacy.
- **Parallel composition** tells us that the privacy guarantee for a set of queries on disjoint subsets of data is only limited by the worst-case privacy guarantee for any of the queries. If $\mathcal{F}_i$ each provide ϵ_i-differential privacy, and D_i are arbitrary disjoint subsets of the dataset D, then the sequence of $\mathcal{F}_i(X \cap D_i)$ provides $(\max_i \epsilon_i)-$differential privacy.

2.2 Differentially private tree-based methods

There have been several methods proposed in recent literature to learn ensembles of decision trees for classification in a differentially private manner (although, to the best of our knowledge, this paper is the first to consider regression); Fletcher and Islam (2019) provide an excellent survey. As we discuss in Sec. 3, most algorithms use the Laplace mechanism to privatize leaf node counts, and differ on the method of tree construction.

Most algorithms can be seen as privatized versions of random forests (Breiman 2001), which greedily choose optimal splits at internal nodes, and store sufficient statistics at the leaf nodes. One way of determining the splits in a decision tree is to choose the split with lowest entropy. Friedman and Schuster (2010) privatizes this by noising the node counts using the Laplace mechanism, and then selecting an attribute to split on using the exponential mechanism with entropy as the score. Sub-trees are iteratively constructed for each value of the attribute thus chosen. This method is built upon in the differentially private random forest (DP-RF) algorithm (Patil and Singh 2014), which builds multiple private trees using bootstrapped samples and considers multiple splitting metics. The trees are grown iteratively until they reach a specified maximum depth or the node is pure, i.e., contains instances of a single class. The differentially private decision forest (DP-DF) (Fletcher and Islam 2015) uses local sensitivity to privatize the leaf-node sufficient statistics, and incorporates a pruning method. The differentially private greedy decision forest (Xin et al. 2019) partitions the data into disjoint subsets, to reduce the overall privacy cost. (Rana, Gupta, and Venkatesh 2015) proposes a relaxation on the DP-RF approach so that the ensemble of trees preserves the variance, instead of the entire distribution of the data. This relaxation enables their method to achieve better performance, and allows for numerical covariates, at the expense of losing any claim to ϵ-differential privacy.

The above algorithms all assume categorical covariates, and at each node, there is one child node per category of the selected attribute. Continuous covariates must therefore be discretized in some manner; if this is done in a data-dependent manner, we must expend privacy budget to do so (Kotsiantis and Kanellopoulos 2006). Friedman and Schuster (2010) do propose a method of splitting continuous covariates via the exponential mechanism, but this significantly increases the privacy budget, as discussed in Fletcher and Islam (2015).

An alternative approach is to bypass the greedy splitting mechanism altogether, generating and selecting splits at random similar to the extremely randomized trees algorithm (Geurts, Ernst, and Wehenkel 2006). Jagannathan, Pillaipakkamnatt, and Wright (2009) and Bojarski et al. (2014) both use random splits to build their trees, allowing the entire privacy budget to be devoted to privatizing the leaf node sufficient statistics, and allowing categorical or continuous covariates.

3 Differentially private median forests

When designing differentially private algorithms, we wish to minimize the negative impact of privatization on the overall utility. In a tree-based setting, we need to privatize two things: determining the locations of splits, and estimating the node parameters. We consider these in turn.

Estimating leaf node parameters In a tree-based model, we estimate appropriate sufficient statistics at each leaf node—typically the mean in a regression setting, and the class counts in a classification context. In both cases, we can use the Laplace mechanism to provide appropriately privatized statistics—the approach taken by almost all existing differentially private tree algorithms. To render the count query at a given node ϵ_ℓ-DP, we can add Laplace$(0, 1/\epsilon_\ell)$ noise to each class count. In the regression context, if we have N_i data points associated with the ith leaf node, and we have bounded target $Y \leq B$, we can achieve ϵ_ℓ-DP at a single leaf node by adding Laplace$(0, 2B/N_i\epsilon_\ell)$ noise to the mean. Since the leaf nodes are disjoint, the set of all leaf-node queries is also ϵ_ℓ-DP, following the parallel composition theorem.

The utility of the resulting estimate at the ith leaf depends on two things: the value of ϵ_ℓ, and the number N_i of data points associated with the leaf node. Small values of ϵ_ℓ, and small values of N_i, both lead to the Laplace noise dominating the signal of interest. Therefore, to improve the utility of our estimates, we must either increase the per-leaf-node privacy budget, or increase the number of data points N_i. In particular, we wish to avoid the situation where the privacy budget ϵ_ℓ is less than $1/N_i$, which implies that the expected magnitude of the Laplace noise is greater than that of the signal.

Obtaining non-leaf node splits In most non-private random forests, the value at which a non-leaf node is split is determined by maximizing some score, such as the Gini index. We can privatize this by randomizing the selection procedure, an approach taken by Friedman and Schuster (2010), Patil and Singh (2014) and Fletcher and Islam (2015). In these works, the authors score attribute-specific candidate splits with one child node for each category of that attribute, and then selects an attribute according to the exponential mechanism.

Unfortunately, the design decisions made in these private algorithms are at odds with the goal of maximizing per-leaf node utility. In each case, a subtree is learned for each category of the chosen categorical covariate (or each unique value of the discretized continuous covariate), leading to more low-occupancy nodes than would be expected in a binary tree. This in turn leads to more low-occupancy leaf nodes where the added Laplace noise overwhelms the signal from the data. In addition to harming the leaf node utility, low-occupancy sub-trees are a major hindrance to good regression trees. Typically, the sensitivity of the scoring function is inversely proportional to the node count, meaning that the selection of attribute to split on will also be very noisy; sufficiently low counts would result in this selection being no better than random. Private versions of Extremely Random Trees, such as those proposed by Bojarski et al. (2014) and Jagannathan, Pillaipakkamnatt, and Wright (2009), avoid diverting budget from the leaf nodes by picking their splits entirely at random. In practice, this can sometimes improve performance, if the benefit of increased leaf node privacy budget ϵ_ℓ outweighs the benefit of chasing the optimal tree structure. However, the relatively large per-leaf node ϵ_ℓ is unfortunately paired with highly variable leaf node occupancy N_i, since splits are selected without consideration of the data distribution.

3.1 Differentially Private Median (DiPriMe) Forests

As discussed above, generating all possible splits for a given attribute, or selecting a split at random, can lead to low-occupancy nodes. In a non-private context, there is little downside to such behavior. But once we begin seeking differential privacy, low-occupancy nodes increase noise in the tree-selection process and lead to poor leaf-node utility. We therefore design an algorithm centered on creating balanced leaf nodes.

In a non-private context, if an attribute is continuous, we could achieve optimal leaf node balance by choosing to split on the median value. In a private context, we can use a differentially private estimate of the median. Let $D_i = (X_i, Y_i)$ be a numeric dataset, to be split on attribute a with bounded range $R_a = [a_L, a_U] \subset \mathbb{R}$. We score potential splits $r \in R_a$ according to $q(r) = ||X_{i,a} \cap [a_L, r)| - |X_{i,a} \cap [r, a_U]||$, noting that $q(r)$ is piecewise constant between the data points. The sensitivity of $q(r)$ is 1, so we can achieve ϵ_s-DP by selecting a bin with probability

$$Pr(r) \propto \exp\left\{-\frac{\epsilon_s}{2}\Big||X_{i,a} \cap [a_L, r)| - |X_{i,a} \cap [r, a_U]|\Big|\right\}. \tag{4}$$

If attributes are categorical, in a non-private context we can achieve optimal leaf node balance by calculating $q(C) = ||C| - |X_i \setminus C||$ for all possible splits $(C, X_i \setminus C)$ that are consistent with values of the selected attribute, and selecting the maximizing split. The sensitivity of $q(C)$ is 1, so we can

Algorithm 1 Differentially Private Median (DiPriMe) Tree

class DIPRIMETREE (i, i_{max}, k) ▷ Initialize empty tree
  if $i \leq i_{max}$ **then**
    $d \leftarrow i$
    $d_{max} \leftarrow i_{max}$
    $K \leftarrow k$
  end if
end class

procedure FITTREE(T,D,A,$\mathcal{R}_A$,B,ϵ,ρ)
  $N = |D|$
  $\epsilon_s \leftarrow \frac{\epsilon\rho}{2d_{max}}$, $\epsilon_a \leftarrow \frac{\epsilon\rho}{2d_{max}}$, $\epsilon_\ell \leftarrow \epsilon(1-\rho)$
  if $T.d = T.d_{max}$ or $A = \emptyset$ **then**
    Store privatized mean or class counts in T.
    return T
  end if
  $T.ind$, $T.val \leftarrow$ FINDSPLIT(D,A,$\mathcal{R}_A$,B, ϵ_s, $T.K$)
  $\mathcal{R}_A^L$,$\mathcal{R}_A^R$,A_L,A_R,D_L,D_R $\leftarrow$ SPLITRANGE($\mathcal{R}_A$,A, $T.ind$,$T.val$)

  $T_R \leftarrow$ DIPRIMETREE($T.d+1$, $T.d_{max}$,$T.K$)
  FITTREE(T_R,D_R,A_R,R_{A_R},B,ϵ,ρ)
  $T_L \leftarrow$ DIPRIMETREE($T.d+1$, $T.d_{max}$,$T.K$)
  FITTREE(T_L,D_L,A_L,R_{A_L},B,ϵ,ρ)
end procedure

procedure FINDSPLIT(D,A,$\mathcal{R}_A$,B,ϵ_s,K)
  $N = |D|$
  $A_S \leftarrow$ size-min$\{K, |A|\}$ subset of attributes A.
  for all $a \in A_S$ **do**
    if a is categorical **then**
      Draw subset $C_a \subset R_a$ according to (5)
    else ▷ a is a continuous attribute
      Draw split location $r \in R_a$ according to (4)
      $C_a = R_a \cap (-\infty, r)$
    end if
    $MSE_a \leftarrow$ mean squared error for chosen split
  end for
  Pick attribute $\tilde{a}$ according to (6).
  return $\tilde{a}$, $C_{\tilde{a}}$
end procedure

procedure SPLITRANGE($\mathcal{R}_A$,A, a,C_a,D)
  $\mathcal{R}_A^L, \mathcal{R}_A^R \leftarrow \mathcal{R}_A$
  $A_L, A_R \leftarrow A$
  $R_a^L \leftarrow C_a, R_a^R \leftarrow R_a \setminus C_a$
  if R_a^L cannot be further split (single category) **then**
    $A_L \leftarrow A_L \setminus a$
  end if
  if R_a^R cannot be further split (single category) **then**
    $A_R \leftarrow A_R \setminus a$
  end if
  $D^L = \{(x,y) \in D : x \in \mathcal{R}_A^L\}$
  $D^R = D \setminus D^L$
  return $\mathcal{R}_A^L$,$\mathcal{R}_A^R$,A_L,A_R,D_L,D_R
end procedure

achieve ϵ_s-DP by selecting a split $(C, X_i \backslash C)$ with probability

$$Pr(C, X_i \setminus C) \propto \exp\left\{-\frac{\epsilon_s}{2}\Big||C| - |X_i \setminus C|\Big|\right\}. \quad (5)$$

Having selected a candidate split for each attribute, we pick an attribute to split on using the exponential mechanism, using the negative mean squared error as the scoring function. The sensitivity of the mean-squared error is $4B^2/N_i$, where the target value lies in $[-B, B]$. So, for a single split, we can achieve ϵ_a-differential privacy by splitting on attribute $\tilde{a}$ with probability

$$Pr(\tilde{a}) \propto \exp\left\{-\frac{\epsilon_a N_i}{8B^2}\Big(MSE(\tilde{a})\Big)\right\}, \quad (6)$$

where $MSE(\tilde{a})$ is the mean squared error associated with that split.

We note that, unlike existing differentially private tree algorithms, the non-private version of our algorithm does *not* correspond to a commonly used tree model. Median splits are not typically used in random forests. Such splits are deterministic given the covariates, and agnostic to the target variable, thereby leading to higher errors. While Breiman (2004) considers median splits for lower branches of trees as a simplifying assumption when obtaining consistency results, he categorically states that the use of median splits will result in higher error rates than greedily learned splits. Indeed, in Sec. 4 we see that non-private median forests underperform random forests and extremely random trees in general.

Median splits are, however, successful in a private context. Having well-balanced splits offers little advantage in a non-private setting, but is critical to ensuring good performance in a differentially private setting. We find, in Sec. 4, that the resulting robustness to the detrimental impacts of privatization outweighs the benefits of using more sophisticated splitting method. Further, the privatization of the median eliminates the deterministic nature of the median splits, allowing better performance when the optimal split is far from the median.

Privacy analysis We can calculate the overall privacy budget of our algorithm using the parallel and sequential composition theorems. Since the candidate splits are on separate attributes, the total privacy cost of selecting a differentially private median for all attributes is ϵ_s, making the total privacy cost of selecting private medians for each attribute then selecting an attribute $\epsilon_s + \epsilon_a$. Since the splits at a given depth are performed on disjoint subsets of the data, the total privacy cost of building the tree is therefore $d_{max}(\epsilon_s + \epsilon_a)$ (where d_{max} is the maximum tree depth), and the total cost including privatizing the leaf nodes is $d_{max}(\epsilon_s + \epsilon_a) + \epsilon_\ell$.

As is common in tree-based algorithms, rather than use a single tree, we construct an ensemble of N_T trees. We partition our data into N_T subsets, and learn a DiPriMe tree on each subset. Partitioning has two benefits. First, it means that the overall privacy budget for the forest is still $d_{max}(\epsilon_s+\epsilon_a)+\epsilon_\ell$, due to parallel composition. Second, it encourages variation between the tree structures, allowing better exploration of the space. If we choose not to partition our data, the overall privacy budget would $N_T\,(d_{max}(\epsilon_s + \epsilon_a) + \epsilon_\ell)$.

Algorithm 2 Differentially Private Median (DiPriMe) Forest

class DIPRIMEFOREST (n_T, d_{max}, k)
  $N_T \leftarrow n_T$, $\mathcal{T} \leftarrow \{\}$
  for $i \leftarrow 1, n_T$ **do**
    $T \leftarrow$ DIPRIMETREE(0, d_{max}, k)
    $\mathcal{T} \leftarrow \mathcal{T} \cup T$
  end for
end class

procedure FITFOREST(F,D,A,$\mathcal{R}_A$,B,ϵ,ρ)
  $N_T = F.N_T$
  Partition $D = (X, Y)$ into $\{D_i\}_{i=1,\ldots,N_T}$
  $i \leftarrow 0$
  for all $T \in F.\mathcal{T}$ **do**
    $T \leftarrow$ FITTREE(T,D_i,A,$\mathcal{R}_A$,B,ϵ,ρ)
    $i \leftarrow i + 1$
  end for
end procedure

We split the overall privacy budget ϵ between the three query-specific budgets ϵ_ℓ, ϵ_s and ϵ_a. We set $\epsilon_\ell = (1-\rho)\epsilon$, and $\epsilon_s = \epsilon_a = \rho\epsilon/2d_{max}$, for some $\rho \in (0, 1)$.

We summarize the process of constructing a single DiPriMe tree in Algorithm 1. Algorithm 2 describes how we can combine multiple DiPriMe trees into a forest. We use the following notation in Algorithms 1 and 2: $D = (X, Y)$ refers to the set of data points to which the tree T is being fit. X denotes the input features and Y denotes the corresponding target values. A refers to the set of attributes that the tree can split on with $\mathcal{R}_A = \{R_a : a \in A\}$ denoting the corresponding range or categories. B is the upper bound on the absolute value of the target, i.e., $|y_i| \leq B$, ϵ is the total privacy budget for the tree, and ρ is the fraction of the privacy budget allocated to determine the median split. We include code in the supplement, and will make this public upon publication.

3.2 Utility analysis

In general, the utility of a random forest—and the change in utility due to privatizing that random forest—will depend heavily on the data distribution. Particularly, the loss of utility due to adding noise to a greedy split selection mechanism will depend on how well alternative splits capture variation in the data. This likely explains why, while utility results have been obtained for differentially private trees with random splits (Bojarski et al. 2014), to the best of our knowledge there are no existing utility guarantees for differentially private random forests with data-dependent splits.

The utility of the full DiPriMe tree algorithm, as described in Algorithm 1, also depends on the data, due to the mechanism for selecting the attribute on which to split. However, if we simplify our assumption to select the attribute at random, then we can provide bounds on the utility, relative to a non-private version of the same algorithm. In practice, we found that using the exponential mechanism to select the attribute gave greater utility than random selection on the datasets we considered.

In our analysis, we consider the regression task, and assume all covariates are continuous and that the dataset can be partitioned into equal-occupancy leaf nodes, i.e. $N = c \cdot 2^d$ for some nonnegative integer c. Both assumptions can easily be relaxed. Let Obj_i the the sum of squared errors at the ith leaf node of the non-private median tree, and $\text{Obj} = \sum_i \text{Obj}_i$ be the overall loss. Then we can bound the utility loss due to privatizing the non-leaf node splits.

Theorem 1. *Let Obj_i^* be the loss under a median tree where the splits have been privatized following* (4)*, but where the sufficient statistics at the leaf nodes are not privatized. Let N_i^* be the number of data points at the i^{th} leaf node of this tree, and let c be the number of data points at the leaf node of the non-private tree. Then, for any $t > 0$,*

$$|Obj_i - Obj_i^*| \leq 4B^2 t \text{ with probability } \min(\zeta_1, \zeta_2)$$

where $|Y| \leq B$ and

$$\zeta_1 = \frac{\gamma}{\epsilon_s^2 t^2}$$

$$\zeta_2 = \max\left\{2e^{-\epsilon_s^2 t^2/4\gamma}, 2e^{-\epsilon_s \beta t/2}\right\}$$

with $\gamma = \frac{8(1-2^{-2d})}{3}, \beta^2 = \frac{1}{2}(\sqrt{1-2/e}+1)$.

Corollary 1.1.

$$|Obj - Obj^*| \leq 2^{d+2} B^2 t \text{ with probability } \min(\zeta_1, \zeta_2)$$

We can then bound the utility loss due to privatizing both non-leaf and leaf node queries.

Theorem 2. *Let $\widetilde{Obj}$ total loss due to both privatizing median splits and privatizing leaf node means. Then, $\mathbb{E}[\widetilde{Obj}] - Obj \leq \left(B^2 t + \frac{2B^2}{\epsilon_\ell^2 (N/2^d - t)}\right) 2^{d+2}$ with probability at least $1 - \zeta$, where $\zeta = 2^{1-d}\min(\zeta_1, \zeta_2)$.*

Proofs of Theorems 1 and 2, and empirical evidence demonstrating the tightness of the bound in Theorem 1, are provided in the supplement.

4 Experiments

To consider the utility of our proposed algorithm, we look at the estimate qualities obtained across a range of regression and classification tasks, comparing against state-of-the-art private and non-private algorithms. For classification, we compare against two privatized versions of random forests (DP-DF (Fletcher and Islam 2015) and DP-RF (Patil and Singh 2014)), and a privatized version of extremely random trees (DP-ERT, (Bojarski et al. 2014)). There has been little focus on regression in the literature, limiting our comparisons for the regression task. We modified DP-ERT to estimate the mean, rather than counts, at each leaf. It is not straightforward to modify DP-DF and DP-RF in this manner, however, as their split selection mechanisms are based on the assumption of categorical targets. We created a regression analogue of DP-RF, in which splits are scored by mean squared error and means were stored instead of class counts. The hyperparameters (N_T, d_{max}) were chosen based on the dataset size to avoid low-occupancy nodes; we do not optimize these hyperparameters as this would increase the privacy cost.

4.1 Regression

We consider three datasets to benchmark our method's regression performance: the Parkinson's telemonitoring dataset ($N = 5875$) and the Appliance Energy Prediction dataset ($N = 19735$) from the UCI Machine Learning Repository (Dua and Graff 2017), and the Flight Delay dataset used by Jagannathan, Pillaipakkamnatt, and Wright (2009). The UCI datasets contain a mixture of categorical and numeric features, while the Flight Delay dataset contains only numeric features. For the purposes of computational complexity, we sampled 800,000 data instances from the Flight Delay dataset for this experiment. We bin the numeric features into 5 bins for DP-RF as it requires categorical features. For each dataset, we scaled the target variable to lie in $[0, 1]$, took 90% of the data as the training set and computed the mean squared error (MSE) over the test set. The results shown in Table 1 are for $N_T = 10$ trees in each ensemble, with the number of covariate splits to consider set to $K = 10$ for all but the DP-ERT. The private methods were run for $\epsilon = 10$ and $\rho = 0.5$. The maximum depth was 5 for the two UCI datasets and 10 for the Flight Delay dataset.

	Parkinson's	Appliances	Flight Delay
RF	2.04×10^{-2}	8.49×10^{-3}	2.11×10^{-4}
ERT	2.98×10^{-2}	9.11×10^{-3}	2.16×10^{-4}
Median	2.54×10^{-2}	9.28×10^{-3}	2.30×10^{-4}
DP-RF (2014)	2.65×10^{-1}	1.97×10^{-1}	1.40×10^{-2}
DP-ERT	3.41×10^{-2}	1.01×10^{-2}	2.34×10^{-4}
DiPriMe	3.10×10^{-2}	9.88×10^{-3}	2.37×10^{-4}

Table 1: MSEs obtained with DiPriMe, and with non-private and private tree-based ensemble methods for regression, $N_T = 10, K = 10, \epsilon = 10, \rho = 0.5$.

We see in Table 1 that DiPriMe clearly outperforms DP-ERT as a private tree-based ensemble learner for regression, even in datasets where the non-private median tree underperforms the non-private ERT algorithm. As hypothesized, we achieve better performance in the private setting by modifying our base algorithm to mitigate the impact of additional noise, even though those modifications are not helpful in the non-private setting: the benefit of additional robustness to noise outweighs the utility loss due to median splits. The effect of low-occupancy nodes is evidenced by the terrible performance of the regression analogue of DP-RF – it consistent gives an MSE at least an order of magnitude worse than any other method.

In Figure 1, we see how performance varies with privacy budget ϵ, the number of trees in the forest N_T, and the maximum tree depth d_{max}, on the Appliances dataset. We have omitted DP-RF from these figures as it vastly underperforms the other methods. We see that for most parameter settings, DiPriMe outperforms DP-ERT. Figure 1(a) illustrates the inherent trade-off between learning deeper trees and utility. Deeper trees give a finer approximation of the data, demonstrated by the decreasing MSE of the non-private methods. However, this deteriorates the utility of DiPriMe by (a) reducing the privacy budget for the split at each node (b) increasing the sensitivity of the mean at the leaf nodes as there are likely to be fewer data instances in deeper nodes. Increasing the number of trees in the ensemble results in a similar trade-off, as shown in Figure 1(b); the number of data points to learn each tree is inversely related to the number of trees in the ensemble. So, while more trees are expected to generally reduce the mean squared error, each tree has less data to learn from.

4.2 Classification

We use three datasets from the UCI Machine Learning Repository (Dua and Graff 2017)—the Banknote Authentication ($N = 1372$), Credit Card Default ($N = 30000$) and Wall-Following Robot Navigation ($N = 5456$) datasets—to compare the performance of our proposed algorithm to DP-DF, DP-RF, and DP-ERT. The Credit Card Default data contains both numeric and categorical features, while the other two datasets contain only numeric features. As DP-DF and DP-RF require categorical features, we bin the numeric features into 5 bins. We chose this data-agnostic discretization procedure to avoid leaking privacy. While alternative data-dependent discretization algorithms have been proposed in the literature (Patil and Singh 2014), such methods must be privatized and hence incur additional privacy budget.

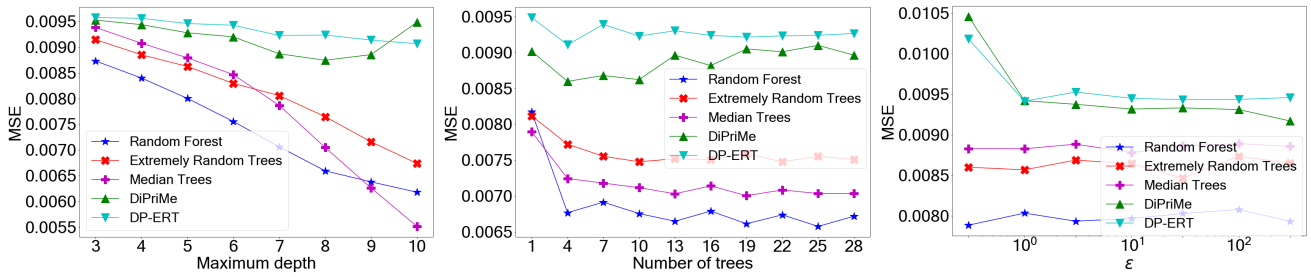
	Banknote	Credit Card	Robot
RF	0.964	0.823	0.987
ERT	0.949	0.788	0.916
Median	0.920	0.785	0.877
DP-DF (2015)	0.551	0.785	0.595
DP-RF (2014)	0.804	0.795	0.662
DP-ERT	0.642	0.664	0.620
DiPriMe	0.935	0.785	0.788

Table 2: Comparison of DiPriMe with various non-private and private tree-based ensemble methods for classification, $N_T = 10, K = 5, d_{max} = 5, \epsilon = 2, \rho = 0.5$.

Table 2 shows the classification errors obtained by each method. In the non-private setting, median forests underperform both extremely random trees and random forests. However, as we introduce privacy, we see that the performance of DiPriMe far exceeds that of DP-ERT, DP-RF and DP-DF. This can likely be attributed to (a) DiPriMe's capability to directly utilize and split on numeric features without the need for prior discretization (b) the robustness of the median construction to privatization, due to its preference for balanced node splits. As we see in Figure 2, while DiPrime has a notable loss of accuracy compared with the non-private algorithms for small values of ϵ, we get comparable performance as ϵ increases. By contrast, DP-ERT and DP-DF continue to underperform even as ϵ increases, and DP-RF requires much larger ϵ to attain comparable performance.

4.3 Balancedness of splits

Our hypothesis for the superior performance of DiPriMe is that it discourages low occupancy nodes, which in turn reduces the impact of the injected noise on selecting optimal



(a) $\epsilon = 10$, $N_T = 10$ (b) $\epsilon = 10$, $d_{max} = 8$ (c) $d_{max} = 8$, $N_T = 10$

Figure 1: Mean squared error of DiPriMe, Random Forest, Extremely Randomized Trees and DP-ERT at various values of ϵ, d_{max} and N_T on the Appliances Energy prediction dataset.

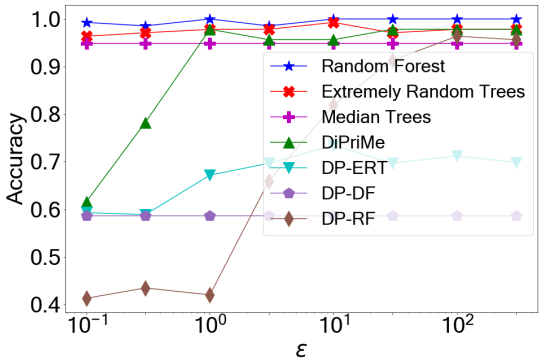


Figure 2: Performance of DiPriMe, Random Forest, Extremely Randomized Trees, DP-ERT and DP-DF at various values of ϵ for the Banknote Authentication data ($d_{max} = 5$, $N_T = 10$, $\rho = 0.5$).

splits and estimating leaf-node parameters. The claim that DiPriMe trees discourage low-occupancy nodes is borne out by a closer inspection of the ensembles of trees learned on the Robot data. Figure 3 shows that DP-ERT, DP-RF and DP-DF have very heavy-tailed distributions over the per-leaf node occupancy, with the majority of nodes having very low occupancy. This is exacerbated in the trees generated by DP-RF and DP-DF, where each split generates multiple child nodes, one for each category of the selected attribute, compared with the binary splits used by DiPriMe and DP-ERT: DP-DF and DP-RF generated an average of 290 nodes and 381 nodes respectively, compared with 63 for DiPriMe and DP-ERT. By contrast, we see that the DiPriME trees have a much larger proportion of higher occupancy nodes.

We can explore this phenomenon by comparing splits generated by DiPriMe, with those generated by DP-ERT on the Banknote Authentication dataset. Figure 4 shows that the splits generated by DP-ERT have highly imbalanced occupancy with high probability. By contrast, the median-based splits used by DiPriMe tend to assign similar occupancies to both child nodes.

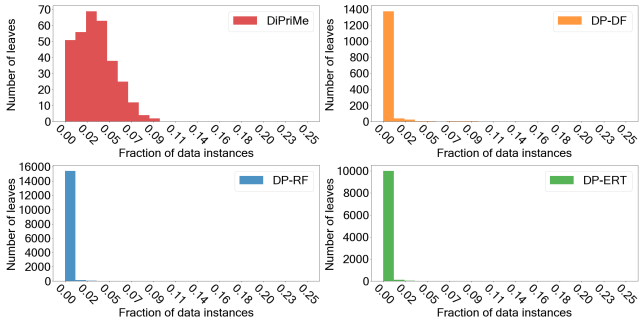


Figure 3: Histogram of fraction of instances residing at each leaf node for Robot data.

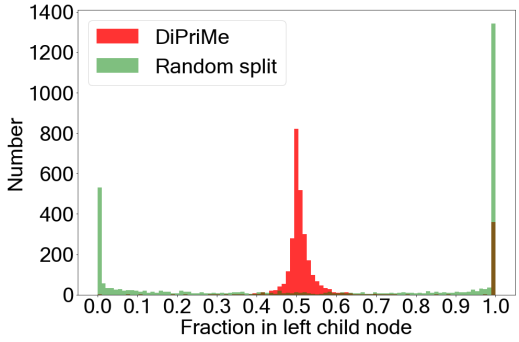


Figure 4: Fraction of instances assigned to left child, on Banknote data.

5 Discussion

We have presented a new, differentially private, tree-based method for regression and classification, based on random forests with median splits. Our algorithm can easily be used for either regression or classification, and works with both categorical and numeric covariates. Moreover, we have demonstrated, both theoretically and empirically, that our algorithm obtains impressive utility to competing methods, while maintaining the same level of differential privacy.

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308–318.

Bagdasaryan, E.; Poursaeed, O.; and Shmatikov, V. 2019. Differential Privacy Has Disparate Impact on Model Accuracy. In *Advances in Neural Information Processing Systems 32*, 15479–15488.

Blum, A.; Dwork, C.; McSherry, F.; and Nissim, K. 2005. Practical privacy: the SuLQ framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 128–138.

Bojarski, M.; Choromanska, A.; Choromanski, K.; and LeCun, Y. 2014. Differentially-and non-differentially-private random decision trees. *arXiv preprint arXiv:1410.6973* .

Breiman, L. 2001. Random forests. *Machine learning* 45(1): 5–32.

Breiman, L. 2004. Consistency for a simple model of random forests. Technical Report 670, Statistics Department, University of California at Berkeley.

Cummings, R.; Gupta, V.; Kimpara, D.; and Morgenstern, J. 2019. On the compatibility of privacy and fairness. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, 309–315.

Dinur, I.; and Nissim, K. 2003. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 202–210.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. URL http://archive.ics.uci.edu/ml.

Dwork, C. 2008. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, 1–19. Springer.

Dwork, C.; and Lei, J. 2009. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 371–380.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, 265–284. Springer.

Dwork, C.; Roth, A.; et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9(3–4): 211–407.

Fletcher, S.; and Islam, M. Z. 2015. A Differentially Private Decision Forest. In *AusDM*, 99–108.

Fletcher, S.; and Islam, M. Z. 2019. Decision tree classification with differential privacy: A survey. *ACM Computing Surveys (CSUR)* 52(4): 1–33.

Friedman, A.; and Schuster, A. 2010. Data mining with differential privacy. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 493–502.

Geurts, P.; Ernst, D.; and Wehenkel, L. 2006. Extremely randomized trees. *Machine learning* 63(1): 3–42.

Hensman, J.; Fusi, N.; and Lawrence, N. D. 2013. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835* .

Huang, Z.; and Liu, J. 2018. Optimal differentially private algorithms for k-means clustering. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, 395–408.

Jagannathan, G.; Pillaipakkamnatt, K.; and Wright, R. N. 2009. A practical differentially private random decision tree classifier. In *2009 IEEE International Conference on Data Mining Workshops*, 114–121. IEEE.

Kifer, D.; Smith, A.; and Thakurta, A. 2012. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, 25–1.

Kotsiantis, S.; and Kanellopoulos, D. 2006. Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering* 32(1): 47–58.

Lakshminarayanan, B.; Roy, D. M.; and Teh, Y. W. 2016. Mondrian forests for large-scale regression when uncertainty matters. In *Artificial Intelligence and Statistics*, 1478–1487.

Liu, J.; and Talwar, K. 2019. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 298–309.

McSherry, F.; and Talwar, K. 2007. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, 94–103. IEEE.

McSherry, F. D. 2009. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 19–30.

Mozannar, H.; Ohannessian, M. I.; and Srebro, N. 2020. Fair Learning with Private Demographic Data. *arXiv preprint arXiv:2002.11651* .

Park, M.; Foulds, J.; Chaudhuri, K.; and Welling, M. 2016. DP-EM: Differentially private expectation maximization. *arXiv preprint arXiv:1605.06995* .

Patil, A.; and Singh, S. 2014. Differential private random forest. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2623–2630. IEEE.

Rana, S.; Gupta, S. K.; and Venkatesh, S. 2015. Differentially private random forest with high utility. In *2015 IEEE International Conference on Data Mining*, 955–960. IEEE.

Roig-Solvas, B.; and Sznaier, M. 2020. Novel Tractable Bounds on the Lambert Function with Application to Maximum Determinant Problems. *arXiv preprint arXiv:2004.01115* .

Smith, M.; Álvarez, M.; Zwiessele, M.; and Lawrence, N. D. 2018. Differentially private regression with Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, 1195–1203.

Su, D.; Cao, J.; Li, N.; Bertino, E.; and Jin, H. 2016. Differentially private k-means clustering. In *Proceedings of the sixth ACM conference on data and application security and privacy*, 26–37.

Voigt, P.; and Von dem Bussche, A. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing* .

Wainwright, M. J. 2019. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press.

Xin, B.; Yang, W.; Wang, S.; and Huang, L. 2019. Differentially Private Greedy Decision Forest. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2672–2676.

Appendix

We derive sensitivities used in the main paper in Section A and provide proofs for the theorems in Section B. Algorithm 3 provides additional implementation details. Section C provides some additional experimental results
We shall use the following constants in our analysis. We assume all the target values are bounded, i.e., $|y_i| \leq B$

A Sensitivity analysis

We assume have a set $\mathcal{A}$ with data points $\{y_1, y_2, \ldots, y_N\}$ such that $|y_i| \leq B$. $\mathcal{A}|j$ denotes the set of all data points besides y_j, i.e., $\mathcal{A}|j = \{y_1, y_2, ...y_{j-1}, y_{j+1}, ..., y_N\}$.

A.1 Mean

$$\begin{aligned}
\mu_{\mathcal{A}} &= \frac{1}{N}\sum_{i=1}^{N} y_i \\
\mu_{\mathcal{A}|j} &= \frac{1}{N-1}\sum_{i=1, i\neq j}^{N} y_i \\
\left\|\mu_{\mathcal{A}} - \mu_{\mathcal{A}|j}\right\|_1 &= \left\| -\frac{1}{N(N-1)}\sum_{i=1, i\neq j}^{N} y_i + \frac{y_j}{N} \right\|_1 \\
&\leq \frac{2B}{N}
\end{aligned}$$

$\therefore \Delta(\mu_{\mathcal{A}}) = \frac{2B}{N}$

A.2 Mean squared error

The mean squared error is equivalent to the variance. Denoting the variance by σ^2,

$$\begin{aligned}
\sigma^2_{\mathcal{A}} &= \frac{1}{N}\sum_{i=1}^{N} y_i^2 - \frac{1}{N^2}\left(\sum_{i=1}^{N} y_i\right)^2 \\
\sigma^2_{\mathcal{A}|j} &= \frac{1}{N-1}\sum_{i=1, i\neq j}^{N} y_i^2 - \frac{1}{(N-1)^2}\left(\sum_{i=1, i\neq j}^{N} y_i\right)^2
\end{aligned}$$

Using the triangle inequality repeatedly, we get

$$\begin{aligned}
\left\|\sigma^2_{\mathcal{A}} - \sigma^2_{\mathcal{A}|j}\right\|_1 &\leq \left\| -\frac{1}{N(N-1)}\sum_{i=1, i\neq j}^{N} y_i^2 + \frac{2N-1}{N^2(N-1)^2}\left(\sum_{i=1, i\neq j}^{N} y_i\right)^2 \right\|_1 \\
&\quad + \left\| \frac{N-1}{N^2} y_j^2 - \frac{2y_j}{N^2}\left(\sum_{i=1, i\neq j}^{N} y_i\right) \right\|_1 \\
&\leq \left\| \frac{N-1}{N^2} B^2 \right\|_1 + \left\| \frac{N-1}{N^2} B^2 \right\|_1 + \left\| \frac{2(N-1)}{N^2} B^2 \right\|_1 \\
&\leq \frac{4B^2}{N}
\end{aligned}$$

$\therefore \Delta(\sigma^2_{\mathcal{A}}) = \frac{4B^2}{N}$

B Proofs

B.1 Theorem 1

In a depth-d median tree, there are $c = N/2^d$ data points at the i^{th} leaf node. W,l.o.g. let these be $\mathcal{A} = \{y_1, y_2, \ldots, y_c\}$. Then, the value of our objective function is

$$\begin{aligned}\text{Obj}_i &= \sum_{i=1}^{c}(y_i - \bar{y}_1)^2 \\ &= c\sigma_{\mathcal{A}}^2\end{aligned}$$

Assuming there be a subset $\mathcal{C} \subset \mathcal{A}$ of $M - \delta$ data points at node i with private median splits, our noisy objective value is

$$\text{Obj}_i^* = (c - \delta)\sigma_{\mathcal{C}}^2$$

Sensitivity of objective W.l.o.g., we take excluded points to be $\{y_1, y_2, \ldots y_\delta\}$. We assume all the points are bounded, i.e., $|y_i| \leq B$.

$$\begin{aligned}c\sigma_{\mathcal{A}}^2 &= \sum_{i=1}^{c} y_i^2 - \frac{1}{c}\left(\sum_{i=1}^{c} y_i\right)^2 \\ (c-\delta)\sigma_{\mathcal{B}}^2 &= \sum_{i=\delta+1}^{c} y_i^2 - \frac{1}{c-\delta}\left(\sum_{i=\delta+1}^{c} y_i\right)^2\end{aligned}$$

Hence,

$$\begin{aligned}\left\|c\sigma_{\mathcal{A}}^2 - (c-\delta)\sigma_{\mathcal{B}}^2\right\|_1 &= \left\|\frac{\delta}{c(c-\delta)}\left(\sum_{i=\delta+1}^{c} y_i\right)^2\right\|_1 + \left\|\sum_{i=1}^{\delta} y_i^2 - \frac{1}{c}\left(\sum_{i=1}^{\delta} y_i\right)^2\right\|_1 \\ &\quad + \left\|\frac{2}{c}\left(\sum_{i=1}^{\delta} y_i\right)\left(\sum_{i=\delta+1}^{c} y_i\right)\right\|_1 \\ &\leq \frac{\delta(c-\delta)}{c}B^2 + \delta B^2 + \frac{2\delta(c-\delta)}{c}B^2 \\ &\leq 4B^2\delta\end{aligned}$$

Returning to the utility analysis, we see that noising the median by δ leads to a maximum change of $4B^2\delta$ in the objective value at node i.
Extending this to tree of depth d, if N_i^* denotes the number of data points at the i^{th} leaf node, then

$$|N_i^* - c| \leq t \implies |\text{Obj}_i^* - \text{Obj}_i| \leq 4B^2 t$$

Corollary 1.1 follows from applying the above result over all the 2^d leaf nodes of a depth-d tree.

B.2 Theorem 2

We consider a depth-d tree. Let δ_1 be the noise added at the first level, δ_2 be the noise added at the second level, and so on. In the noised tree, at the leaf node (depth d), we will therefore have

$$\begin{aligned}N_i^* &= \frac{N}{2^d} + \sum_{j=1}^{n}\frac{\delta_j}{2^{d-j}},\ \delta_j \sim \text{Laplace}(0, 1/\epsilon_s) \\ &= \frac{N}{2^d} + \sum_{i=j}^{d}\Delta_j,\ \Delta_j \sim \text{Laplace}(0, 2^{j-d}/\epsilon_s)\end{aligned}$$

data points.
$\sum_{j=1}^{n}\Delta_j$ is unbounded, so we bound the tail probability instead, i.e., find an upper bound on $P\left(\sum_{j=1}^{d}\Delta_j \geq t\right)$. We shall use two approaches to arrive at this bound: (a) sub-exponential random variables (b) Chebyshev's inequality.

Using sub-exponential random variables

$$\begin{aligned} M_{\Delta_j}(t) &= \mathbb{E}[e^{t\Delta_j}] \\ &= \frac{1}{1-b_j^2t^2},\ |t| < \frac{1}{b_j},\ b_i = \frac{2^{j-n}}{\epsilon_s} \\ &\leq e^{2b_j^2t^2},\ \forall |t| \leq \frac{1}{\alpha_j} \end{aligned}$$

This implies that $\Delta_j \in SE(4b_j^2, \alpha_i)$ where $SE()$ denotes the class of sub-exponential random variables. To find α_j, we need to find the range of t for which

$$\begin{aligned} \frac{1}{1-b_j^2t^2} &\leq e^{2b_j^2t^2} \\ \implies b_j^2t^2 &\leq \frac{1}{2}W_0\left(\frac{-2}{e^2}\right) + 1 \end{aligned}$$

where $W_0(x)$ is the principal branch of the Lambert W function. For ease of analysis, we use the lower bound on $W_0(x)$ from (Roig-Solvas and Sznaier 2020):

$$W_0(x) \geq \sqrt{ex+1} - 1 \text{ for } 1/e \leq x \leq 0$$

to get

$$\begin{aligned} t^2 &\leq \frac{1}{2b_j^2}(\sqrt{1-2/e}+1) \\ \implies \alpha_j = \frac{b_j}{\beta},\ \beta^2 &= \frac{1}{2}(\sqrt{1-2/e}+1) \approx 0.757 \end{aligned}$$

Let $\gamma = \frac{8(1-2^{-2d})}{3}$. Then,

$$\begin{aligned} \nu_*^2 &= \frac{2\gamma}{\epsilon_s^2} \\ \frac{\nu_*^2}{\alpha_*} &= \frac{2\beta\gamma}{\epsilon_s} \end{aligned}$$

We can now bound the tail probabilities (Wainwright 2019) as

$$\begin{aligned} P\left(\left|\sum_{j=1}^{d}\Delta_i\right| \geq t\right) &\leq \begin{cases} 2e^{-t^2/(2\nu_*^2)}, & 0 \leq t \leq \frac{\nu_*^2}{\alpha_*} \\ 2e^{-t/(2\alpha_*)} & t > \frac{\nu_*^2}{\alpha_*} \end{cases} \\ &= \begin{cases} 2e^{-\epsilon_s^2t^2/(4\gamma)}, & 0 \leq t \leq \frac{2\beta\gamma}{\epsilon_s} \\ 2e^{-\epsilon_s\beta t/2} & t > \frac{2\beta\gamma}{\epsilon_s} \end{cases} \end{aligned}$$

Using Chebyshev's inequality A simpler method of bounding the tail probabilities is to use Chebyshev's inequality. This gives

$$\begin{aligned} P\left(\left|\sum_{j=1}^{n}\Delta_j\right| \geq t\right) &\leq \frac{Var\left(\sum_{j=1}^{d}\Delta_j\right)}{t^2} \\ &= \frac{\gamma}{\epsilon_s^2t^2} \end{aligned}$$

This is tighter than the sub-exponential bound for

$$\begin{aligned} t &\in \frac{\sqrt{\gamma}}{\epsilon_s}\left(2\sqrt{-W_0\left(-\frac{1}{8}\right)}, -2\sqrt{-W_{-1}\left(-\frac{1}{8}\right)}\right) \\ &\in \frac{\sqrt{\gamma}}{\epsilon_s}(0.76, 3.61) \end{aligned}$$

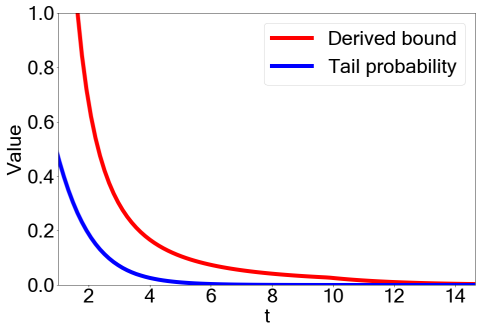


Figure 5: Tail probabilities of $\left|N_i^* - \frac{N}{2^d}\right|$ for $d = 10$ are estimated from the histogram. The red line depicts the tighter of the two bounds derived above; the blue line depicts the empirical estimate.

There is a clear dependence on $1/\epsilon_s$ in the bounds which intuitively makes sense as the noise scales by that factor.We observe that the derived bound on tail probablity of $\left|N_i^* - \frac{N}{2^d}\right|$ holds empirically from 5 with the bound becoming tighter for larger t.

Let us now use the above bound on tail probabilities to bound the effect of noising the mean of the i^{th} leaf node. We have

$$\widetilde{\text{Obj}}_i = \text{Obj}_i^* + N_i^* \rho_i^2$$

where $\rho_i \sim \text{Laplace}(0, 2B/N_i^* \epsilon_\ell)$.

The conditional expectation of the perturbation due to this noise is

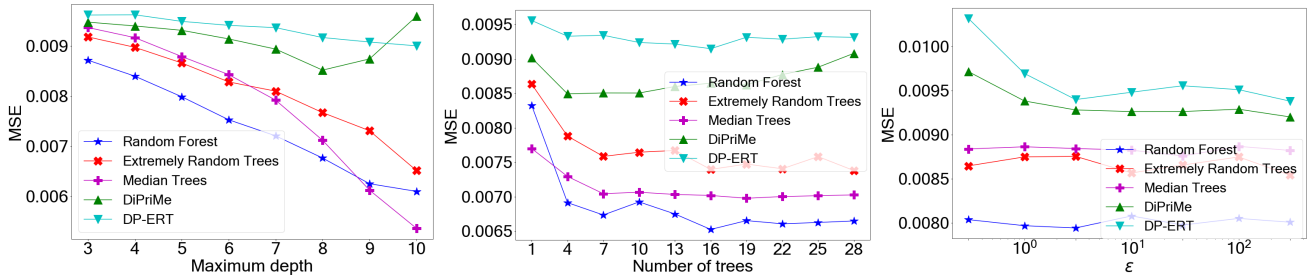
$$\mathbb{E}[N_i^* \rho_i^2 | \tilde{N}_i] = \frac{8B^2}{N_i^* \epsilon_\ell^2}$$

Note that $N_i^* \geq N/2^d - t \implies \mathbb{E}[N_i^* \rho_i^2 | N_i^*] \leq \frac{8B^2}{\epsilon_\ell^2 (N/2^d - t)}$.

As the distribution of N_i^* is symmetric about $N/2^d$, we have $\mathbb{E}[N_i^* \rho_i^2 | N_i^*] \leq \frac{8B^2}{\epsilon_\ell^2 (N/2^d - t)}$ with probability $1 - \zeta_i$, where $P\left(\left|N_i^* - \frac{N}{2^d}\right| \geq t\right) \leq 2\zeta_i$.

Applying the union bound on the leaf nodes, and setting $\zeta_i = 2^{1-d}\zeta$, we get $\sum_i \mathbb{E}[N_i^* \rho_i^2 | N_i^*] \leq \frac{2^{d+3} B^2}{\epsilon_\ell^2 (N/2^d - t)}$ with probability $1 - \zeta_i$. The result of Theorem 2 follows from combining this with that of Corollary 1.1.

C Additional results



(a) $\epsilon = 10$, $N_T = 10$ (b) $\epsilon = 10$, $d_{max} = 8$ (c) $d_{max} = 8$, $N_T = 10$

Figure 6: Mean squared error of DiPriMe without partitioning data, Random Forest, Extremely Randomized Trees and DP-ERT at various values of ϵ, d_{max} and N_T on the Appliances Energy prediction dataset.

Figure 6 displays the results for DiPriMe without paritioning the data for each tree. The trends are similar to those seen in Figure 1. An observation of note is that the optimal maximum depth for trees fit on all the data is higher than that fit on disjoint subsets

of data. This is congruent with the intuition that low-occupancy nodes are noised more heavily. Hence, trees fitted to more data can be grown deeper before suffering from a similar loss of utility. This line of reasoning leads us to believe that learning an ensemble of DiPriME trees on disjoint subsets of data will be a more powerful learner with larger amounts of training data. We see a similar trend for the task of classification (see Figure 7).

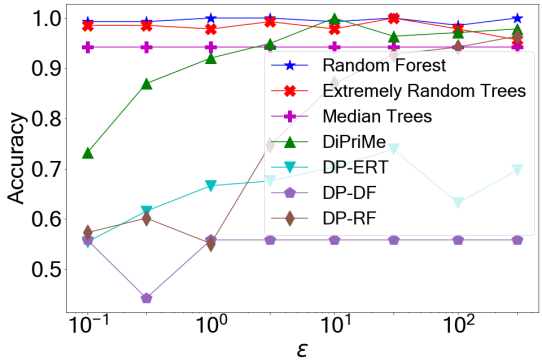
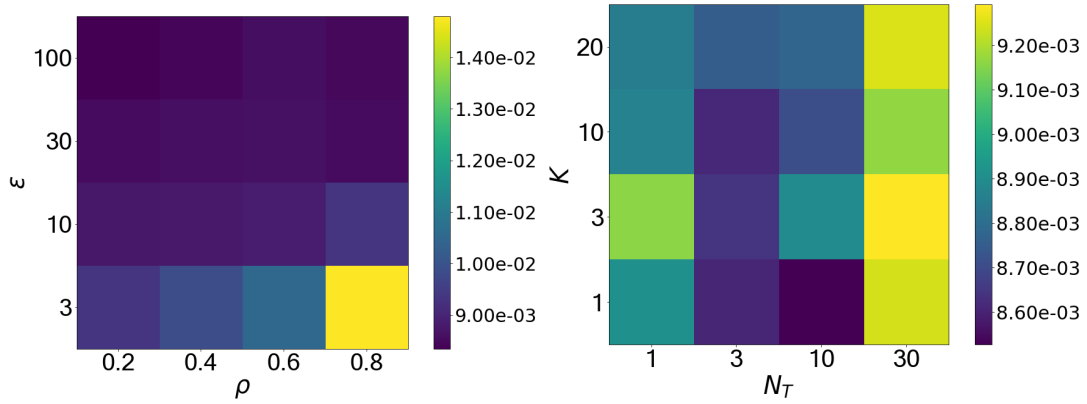


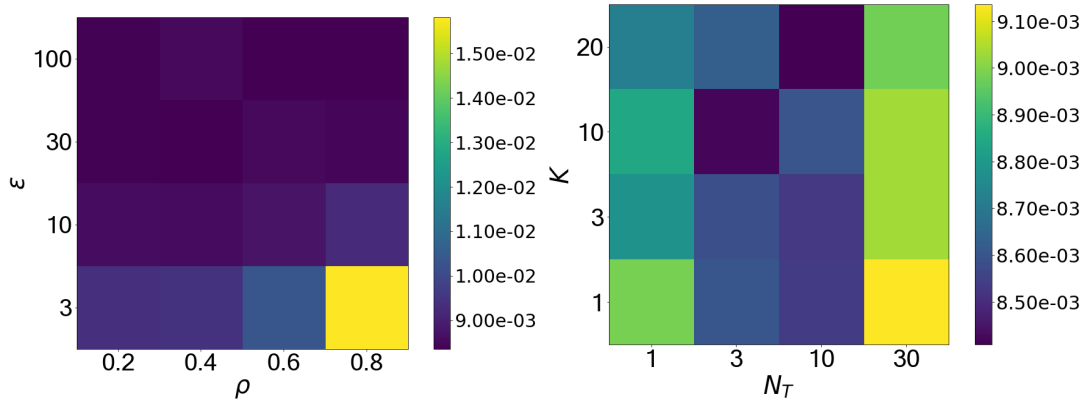
Figure 7: Performance of DiPriMe without partitioning data, Random Forest, Extremely Randomized Trees, DP-ERT and DP-DF at various values of ϵ for the Banknote Authentication data ($d_{max} = 5$, $N_T = 10$, $\rho = 0.5$).

Figure 8 once again exhibits improved performance with larger privacy budgets. It also shows that increasing N_T improves performance only to a certain limit before the increased noise reduces the utility of the DiPriMe trees. The key insight here is the importance of the hyperparameter ρ for good performance; large values of ρ leaves less privacy budget for storing the means, resulting in deterioration in the MSE. This effect is more pronounced at smaller values of ϵ as the noise scales as $1/\epsilon$. Note that data-driven hyperparameter selection has to be done privately as well (Liu and Talwar 2019). This would require additional privacy budget, i.e., a larger ϵ.



(a) $N_T = 10$, $d_{max} = 8$, $K = 10$, data partitioned

(b) $d_{max} = 8$, $\epsilon = 10$, $\rho = 0.5$, data partitioned



(c) $N_T = 10$, $d_{max} = 8$, $K = 10$, data not partitioned

(d) $d_{max} = 8$, $\epsilon = 10$, $\rho = 0.5$, data not partitioned

Figure 8: Mean squared error of DiPriMe on the Appliances Energy Prediction dataset for various hyperparameter settings