Budget Allocation on Differentially Private Decision Trees and Random Forests

A Thesis Submitted for the Degree of
Doctor of Philosophy

By

Nazanin Borhan

in

School of Software
UNIVERSITY OF TECHNOLOGY SYDNEY
AUSTRALIA
JUNE 2018

UNIVERSITY OF TECHNOLOGY SYDNEY
SCHOOL OF SOFTWARE

The undersigned hereby certify that they have read this thesis entitled "**Budget Allocation on Differentially Private Decision Trees and Random Forests**" by **Nazanin Borhan** and that in their opinions it is fully adequate, in scope and in quality, as a thesis for the degree of **Doctor of Philosophy**.

Date: JUNE 2018

Research Supervisors: _____
Paul Joseph Kennedy

_____
Robin Michael Braun

CERTIFICATE OF ORIGINAL AUTHORSHIP

Date: **JUNE 2018**

Author: **Nazanin Borhan**

Title: **Budget Allocation on Differentially Private Decision Trees and Random Forests**

Degree: **Ph.D.**

I declare that this thesis is submitted in fulfilment of the requirements for the award of Ph.D, in the School of Software, Faculty of Engineering and IT at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed
prior to publication.

Signature of Author

Acknowledgements

I owe my gratitude to all the people who have made this thesis possible and taking the opportunity to thank them has been the best part of writing this dissertation.

First, I would like to express my sincere appreciation to my supervisor, Paul J. Kennedy, for his guidance, patience and financial support over the years. Paul encouraged me to develop my research interests and afforded me a considerable freedom to explore interesting tangents in privacy-preserving data mining and I have greatly benefited from his comments and feedback. Paul, thank you for being a wonderful advisor and for always making time for me when I needed guidance.

I would like to thank my co-supervisor, Robin Michael Braun, for his time and helpful guidance. Additionally, I had the privilege of collaborating with Arik Friedman and Richard Nock, from Data61/CSIRO, on the main chapter of this dissertation (Chapter 3). Richard and Arik are brilliant researchers, and I learned a lot from working with them. Their input helped to improve the quality of this chapter significantly. Richard and Arik, thank you for your help and friendly guidance. I would also like to thank Michele Mooney from Professional Proofreading Services for proofreading this thesis.

My amazing family has shown me constant support and unconditional love, not only through my seemingly never-ending educational journey, but throughout my life. They never stopped believing in me, even when I did, and they helped me push through when I wanted to give up. I feel incredibly blessed to have such a loving and supportive family. I am especially grateful to my mother, who is without doubt the hardest working person I have ever known. She instilled in me an appreciation for education and for hard work and has shown me unconditional love and support in everything that I have done. I am also very grateful to my in-laws, for their love and support, especially during the chaotic final months of my PhD. I would also like to thank my dear friends for their friendship and for providing much-needed distractions that kept me from burning out during my final year.

I owe an enormous debt of gratitude to my husband, Steven Mastwijk, for his constant love, support and patience. Steven, thank you for being a fantastic companion, for picking up my slack at home during deadlines and for never ceasing to believe in me. You encouraged me to keep going during my lowest lows and helped me celebrate my highest points. Thank you for patiently sticking by my side all these years, I love you dearly. I am also thankful to my son, Ilai, who is the pride and joy of my life. You made me a stronger person and gave me tremendous happiness during the last year of my study.

Last, but certainly not least, I gratefully appreciate the financial assistance of Australian government and UTS university by offering me the Australian Postgraduate Award and Women in Engineering and IT Research Excellence scholarships.

To My Husband, ***Steven Mastwijk***

& Our Son, ***Ilai Kian Mastwijk***

Table of Contents

List of Tables

List of Figures

Abstract

Privacy-preserving techniques are necessary to minimize the possibility of identifying and learning sensitive information about individuals from any datasets that have been released or shared. Datasets containing sensitive information on individuals are becoming increasingly public. Although this will support data mining research, information privacy concerns need to be addressed. Many techniques have been developed to preserve the privacy of sensitive public data, with Differential Privacy (DP) being a leading method. Differential privacy, as one of the most important privacy-preserving mechanisms, typically adds noise to prevent the disclosure of individuals' sensitive data. However, adding too much noise can compromise the utility of the output from this mechanism, because the resulting predictions will be too inaccurate. Several techniques can be used to achieve differential privacy in practice, including adding Laplacian noise or Gaussian noise to the output of the computation, or adapting perturbation techniques. Each one of these techniques has been proposed to be applied to the right data mining process and applications. Finding a balance between the privacy of individuals and the utility of the results is one of the biggest challenges in differential privacy.

Differential privacy has received significant attention in the machine learning and data mining communities, in part because the individual privacy guarantee does not

fundamentally contradict the learning goal of *generalizing well* (C. Dwork & Roth, 2014). In supervised learning, where the output of the privacy preservation mechanism is a classifier and the mechanism itself involves a learning algorithm, the minimization of the *classification error* under privacy constraints is not trivial (Chaudhuri, Monteleoni, & Sarwate, 2011; Friedman & Schuster, 2010). Decision tree induction is a textbook case for such a problem. The algorithmic decisions in trees that are made with privacy considerations, have a deep impact on the accuracy of the results. An *improved* privacy-preserved decision tree algorithm, with an order of magnitude of fewer learning samples, can still achieve the same level of accuracy and privacy as the naive implementation.

Although adding Laplacian noise is still one of the main mechanisms to achieve differential privacy in decision trees, a recent line of work has shown that the exponential mechanism leads to better trade-offs between accuracy and privacy on the splitting points of top-down decision tree induction algorithms such as ID3, C4.5 and CART (Blum, Dwork, McSherry, & Nissim, 2005; Friedman & Schuster, 2010). These studies reveal a striking observation: while the splitting criterion of ID3/C4.5 guarantees a better rate of convergence in classification than CART, due to the requirement of fewer interactive queries to reach a given accuracy level, the privacy cost it incurs for a given accuracy level is comparatively larger. This observation is important because the algorithms are greedy; ID3 and C4.5 require fewer interactive queries to reach a given accuracy, but each query incurs a higher privacy cost than is the case for CART.

Many machine learning and minimization tasks, including decision trees, make use of an objective function. At each iteration of a decision tree, a parameter set is defined,

and the objective function returns some scoring value that reflects how good or bad that parameter set is. Then the parameter set is altered, and the process repeats. The process of induction is stopped when the changes in fit become acceptably small and the tree is getting closer to a local or global optima. These stopping rules comprise the rate of convergence of a tree. When an algorithm converges, it has found a parameter set meeting the optimal requirements. From the privacy standpoint, the faster the rate of convergence of a splitting function, the higher the privacy cost it incurs. This inevitably raises the question of whether some splitting functions can achieve two goals simultaneously: fast convergence rates and a small privacy cost. This motivates the research in this thesis to seek other methods to improve the spending of the privacy budget throughout the induction of the decision tree, such as tuning the allocation of the privacy budget across queries.

This thesis presents an adaptive mechanism for allocating the privacy budget, designed for any proper symmetric splitting criterion. In a nutshell, when a tree is splitting nodes, the algorithm probes for the amount of privacy budget to allocate to the split. Some data samples at a node need more privacy budget to build an accurate split, and some are not dependent on spending too much budget. The dynamic budget allocation algorithm will enhance the application of differential privacy on decision trees and forests.

This thesis focuses on the dynamic privacy budget allocation algorithm to have more control over the consumption of the budget in the data mining techniques. There are three contributions in this thesis. Contribution 1 proposes a differentially private budget allocation algorithm on decision tree induction. Contribution 2 extends this

schema on random forest algorithm. It is important to consider the adaptive allocation of the budget to the queries that need more or less privacy/utility trade-off. The algorithms are evaluated on synthetic and real datasets for decision trees and random forests. The experiments in this thesis show that the accuracy level of these algorithms are competitively high compared to state-of-the-art models. The budget allocation optimization technique has the potential to be a building block for other data mining techniques and methods. Contribution 3 of this thesis introduces a new splitting criterion for decision trees and evaluates its performance in comparison to other splitting criteria. The result of this evaluation demonstrates that the best splitting criteria for classification under the boosting framework are not necessarily the best to learn in a differentially private setting.

Notations

DP: Differential Privacy

ϵ: Privacy budget Epsilon

CART: Classification and Regression Trees

IG: Information Gain for attribute splitting

Gini Index(G): Gini Index for attribute splitting

Err Score(Err): Misclassification Error Score for attribute splitting

M-Score(M): Matsushita attribute splitting score

ID3: Iterative Dichotomiser 3

RF: Random Forest

PPDP: Privacy Preserving Data Publishing

PPDM: Privacy preserving Data Mining

ACC: Prediction Accuracy

stdev: Standard deviation from the average accuracy prediction

$M(.)$: A learning algorithm

X: A random sample

$\mathcal{D}$: A joint distribution

$\mathcal{S}$: A training set of samples

$\mathcal{S}'$: A neighboring training samples

$\mathcal{T}$: A set of classifiers

h: A (tree) classifier

C: Number of class labels of a tree

$\mathcal{Y}$: Domain of class labels of a tree

ϕ: A permissible function for any symmetric proper loss

ϕ_{IG}: Information Gain criterion

ϕ_{G}: Gini index criterion

ϕ_{Err}: Misclassification Error score

ϕ_{M}: Matsushita's loss

Δ_f: Global sensitivity of function $f(.)$

$\mathcal{L}(h)$: Set of leaves of tree h

m: Number of records in a dataset

m_ℓ: Examples that fall into leaf ℓ

m_ℓ^+: Positive examples reaching ℓ

m_ℓ^-: Negative examples reaching ℓ

d: Dimension of the observations space (number of attributes in the attribute list)

ϵ_{rem}: Remaining privacy budget

t: Adaptive budget threshold

Chapter 1

Introduction

1.1 Introduction

In big data analytics, privacy remains one of the main concerns. Current practices in privacy-preserving data publishing include developing guidelines for the usage and storage of sensitive data and also designing and implementing the process of transforming the original data, so the modified and still useful data can be safely released (Wang, Chen, Fung, & Yu, 2010; Agrawal & Srikant, 2000; Aggarwal & Philip, 2008; Mohammed, Chen, Fung, & Yu, 2011).

In the statistical analysis domain, privacy breaches can occur as soon as we consider external knowledge about any individuals in a database. Even people who do not participate in the database are at risk of privacy violation. Limiting the risk of increasing the number of these privacy breaches for a participant in a dataset is the main objective of the differential privacy (DP) model. Differential privacy (C. Dwork, McSherry, Nissim, & Smith, 2006), a cryptographically motivated definition of privacy, pioneered about a decade ago, has gained significant attention in the data mining and machine-learning communities. This mechanism is emerging as a gold standard

for protecting individuals' privacy, with a formalization of "plausible deniability", a wide applicability of the framework to many areas, and an industry appeal that has led to large-scale applications with the biggest players in the market (Simonite, 2016; Erlingsson, Pihur, & Korolova, 2014).

Several definitions of differential privacy are used in the literature (Rastogi, Hay, Miklau, & Suciu, 2009; Agrawal & Srikant, 2000). Being statistical in nature, the privacy guarantees in differential privacy are different to those based on cryptography (Vaidya & Zhu, 2006) or information theory (Sankar, Rajagopalan, & Poor, 2013). A long series of comprehensive studies beginning in 2006 attempted to introduce the concept of differential privacy (C. Dwork et al., 2006; C. Dwork, 2011a, 2008; C. Dwork & Smith., 2009; C. Dwork & Roth, 2014; G. N. R. Dwork Cynthia & Vadhan, 2010; C. Dwork, 2010; Williams & Frank, 2010). In particular, much of the earlier theoretical work on differential privacy in the literature is covered by Dwork and Smith's survey (C. Dwork & Smith., 2009). Previous research has utilized different methods and mechanisms to implement differential privacy in theory or in practice. Examples of these attempts include work on data mining (Friedman & Schuster, 2010); histograms (C. Dwork et al., 2006); contingency tables (Barak et al., 2007; Justin Thaler & Vadhan, 2012; C. Dwork, Nikolov, & Talwar, 2014); machine learning (Blum et al., 2005); regression and statistical estimation (Smith, 2011; Jain & Thakurta., 2013); clustering (Smith, 2011; Nissim, Raskhodnikova, & Smith, 2007); social network analysis (Shiva Prasad Kasiviswanathan & Smith, 2013; Jeremiah Blocki & Sheet, 2013; Niyogi, 2008); approximation algorithms (Anupam Gupta & Talwar, 2010); singular value decomposition (Hardt & Roth, 2013; Kapralov & Talwar., 2013); streaming algorithms (C. Dwork, Naor, Pitassi, & Rothblum, 2010; C. Dwork, Naor, Pitassi,

Rothblum, & Yekhanin, 2010; Darakhshan Mir & Wright, 2011) and mechanism design (Kobbi Nissim & Tennenholtz, 2012; Nissim et al., 2007).

Differential privacy addresses the paradox of learning useful information about a population while learning nothing about any individual. The utility of the results in a differentially private setting is measured by the accuracy of the statistics estimated in a privacy-preserving manner. This mechanism ensures that the statistical output of two datasets differing only by a single participant remains almost the same by adding sufficient noise to the output of a computation. Noise must be added proportionally to the number of queries to prevent the disclosure of individuals' sensitive data. However, adding too much noise can compromise the accuracy of the system's outputs. Therefore, many studies have focused on finding a trade-off between the privacy of individuals in the differential privacy mechanism and the utility of the published results.

For the purposes of this study, differential privacy is defined as a measurement of the privacy risk by a privacy budget parameter ϵ that bounds the log-likelihood ratio of the output of a private algorithm under two databases that differ only in a single individual's datum. The privacy budget ϵ is the parameter set in the differential privacy model to calculate the amount of noise that needs to be added to the results of a differentially private computation. The larger the privacy budget value, the less noise is applied to the result set, which means it is more accurate but potentially more revealing regarding privacy. Conversely, the smaller the epsilon value, the more noise is applied to the result set which means that more privacy is provided but less utility.

Initial work on differential privacy was motivated by problems in official statistics

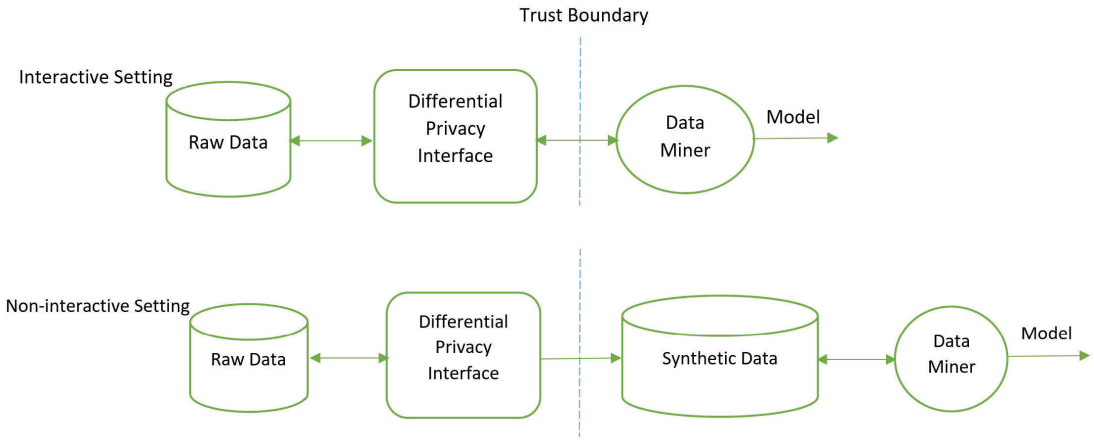

Figure 1.1: Interactive and non-interactive setting of Differentially Privacy

such as the publication of "sanitized" data tables. These works are classified as "non-interactive" approaches for differential privacy and is shown in the bottom of Figure 1.1. In this setting, the data owner prepares a differentially private version of the data and the user has full access to this data. The data owner does not care how this anonymized data will be used later or what operation will happen to it. A different approach is used in the "interactive" setting of differential privacy, in which a user poses queries to the database curator who then provides approximate answers to those queries. This setting is shown in the top of the Figure 1.1. Studies in the literature have expanded from these two settings to cover more complex data processing algorithms, such as real-time signal processing (Le & Pappas, 2012; L. Fan & Xiong, 2012); classification (Chaudhuri et al., 2011; Friedman & Schuster, 2010; Rubinstein, Bartlett, Huang, & Taft, 2012; Zhang, Zhang, Xiao, Yang, & Winslett, 2012); dimensionality reduction (Chaudhuri, Sarwate, & Sinha, 2013); and auction design (Ghosh & Roth, 2015). This thesis applies the interactive setting of differential privacy, where a privacy layer is responsible for answering the user's queries.

Differential privacy protects individuals' data while allowing the inference of general patterns from a dataset. One of the common mechanisms in DP, the *Laplace mechanism* (C. Dwork et al., 2006), involves the introduction of calibrated noise to interactive database queries. Each query incurs a privacy cost (inversely proportional to the magnitude of noise), and these costs *compose* are over a series of queries. For a computation that is comprised of a series of interactive queries, the database curator sets an overall privacy budget, and the budget can be allocated to the queries. Reducing the privacy budget means that less information on any particular individual can leak throughout the computation, but also more noise is introduced, resulting in a trade-off between privacy and accuracy.

Differentially private decision trees and Differentially private random forests play important roles in data mining applications, ranging from privacy optimization in classification to accuracy optimization in the classifier's result. A differentially private decision tree has all the characteristics of a decision tree. In addition, it adds a sufficient level of noise for all of the calculations used for generating a tree, including calculating the best attributes with which to split the node, or calculating the number of samples in the leaf nodes of the tree to label the class value. The classification results of a differentially private decision tree or a differentially private random forest tend to be accurate and fast while guaranteeing a good level of privacy.

Previous works on differentially private decision trees and forests were more focused on the accuracy of the tree's results at a certain level of privacy budget expenditure, without considering the optimization of the privacy budget allocation to minimize the amount of privacy budget spent on each level of the tree. However, saving privacy budget expenditure (ϵ) on each iteration of the tree leads to find a

better trade-off between the level of privacy and the accuracy of a differentially private tree. Overspending the privacy budget could cause relatively non-accurate trees. This problem motivates the study in this research. This thesis focuses on introducing algorithms that save the privacy budget as much as possible on every iteration of the tree without sacrificing too much of the accuracy of the resultant tree. The goal of the algorithms in this thesis are to adaptively apply the the privacy budget ϵ on the decision trees as much as is needed. The aim is to find a trade-off between privacy and the accuracy of the resultant model.

Apart from the expenditure on the privacy budget, this research goes deeper into differentially private tree induction problems. The key difference between most top-down tree induction algorithms is the choice of the splitting criterion. These splitting criteria include: the Gini criterion in CART (Breiman, Friedman, Olshen, & Stone, 1984), the Information Gain (IG, also called binary entropy) in ID3 and C4.5 (Quinlan, 2014), Matsushita's loss in Kearns and Mansour's algorithm (1999), and so on. Apart from the empirical risk, which also yields such an entropy (albeit not differentiable), when inducing a differentially private tree, Friedman and Schuster (2010) considered Gini, Information Gain and Misclassification error as splitting criteria. They demonstrated the following crucial fact: the magnitude of noise that is needed when using Information Gain is higher than that needed when using Gini (Friedman & Schuster, 2010, Appendix A). Or equivalently, for a fixed noise magnitude, Information Gain requires allocating more privacy budget than Gini. This research aims to introduce Matsushita's loss as a new differentially private splitting criterion, calculate the sensitivity of this new criterion and compare it with the other choices for attribute splitting in the literature.

1.2 Significance of Study

Using an individual's sensitive data raises ethical and technical questions about preserving their privacy in the data mining area. Modern technology which enables the generation and storage of an individual's data gives more availability and scale to this data, but the fundamental concerns about their privacy are still the same. As an example, we can refer to the government organizations that collect the individual's data for many decades, called microdata. These collections are crucial to help social planning, to be used in research and to make important social decisions. Assuring individuals that their privacy will not be violated by contributing their sensitive information can result in a better chance of receiving honest answers from them. Therefore, organizations that seek to publish individuals sensitive data or publish analyses and predictions resulting from their sensitive data need to carefully consider privacy issues. Global companies, including Apple and Google, recently started a conversation on applying differential privacy in their services. Google uses an open source implementation service called RAPPOR for telemetry, such as learning statistics about unwanted software hijacking users' settings (Erlingsson et al., 2014). Apple also uses differential privacy technology to uncover the usage of a huge volume of patterns derived from a large number of users without reducing the privacy of the individuals (Simonite, 2016). Differential privacy adds mathematical noise to a small sample of the individual's usage pattern to obscure their identity. Therefore, without the risk of violating an individual's privacy, more users will participate in making the patterns and better user behavioral patterns will emerge. This will help the users have a better experience through their services and the system will learn more information about the users' behaviors. However, even well-meaning data collection can go wrong. For

instance, Netflix ran a competition in the late 2000s to develop an improved film recommendation algorithm. To protect customer privacy, they released an anonymized version of their dataset, modified some of the movie ratings, and removed all personal information that could identify individual customers. Despite these precautions, the de-identification process on this dataset was insufficient. In a very well-known article, Narayanan and Shmatikov (2008) demonstrate that these datasets could be utilized to re-identify specific users, and even anticipate their political affiliation, if the attacker knows just a small amount of additional background information about a given user.

The data mining community is facing a new challenge in addressing the rising concerns about the privacy of individuals. Their model needs to effectively reveal the patterns within huge databases, and it is necessary to protect the privacy of individuals. In this research, the application of differential privacy on two data mining models, decision trees and random forest, is investigated. This topic has attracted attention in the literature over the last couple of years, including studies on differentially private decision trees (Blum et al., 2005; Friedman & Schuster, 2010; Mohammed et al., 2011), and differentially private random forest (Jagannathan, Pillaipakkamnatt, & Wright, 2012; Fletcher & Islam, 2015a, 2017, 2015b; Rana, Gupta, & Venkatesh, 2015). Although all of these studies are considered important aspects of the work, the adaptive usage of the privacy budget ϵ has not been studied in any of these works. The adaptive usage of the privacy budget means spending unequal quantities of the privacy budget on different queries from the tree. In all of the previous work, the privacy budget for the tree has been divided into fixed portions and injected into each query of the tree without further consideration. This means that when d queries are present, each one of them uses ϵ/d of the budget, basically assuming that all of the

queries have equal value in the success of the algorithm, or necessitating the same level of accuracy to be useful. Adaptively matching the privacy budget ϵ to use as much as is needed in each query of the tree is the aim of the algorithms in this thesis. Some queries in the algorithms can achieve their purpose with more added noise than other queries that are more sensitive to adding too much noise.

The main stakeholders of this research fall into two categories:

1. Contributors of data who are the individuals participating in a database and providing their sensitive information.

2. Organizations that use and particularly seek to publish an individual's sensitive information.

1.3 Research Questions

Based on the above research direction, the research questions of this thesis are specified as follows:

RQ1: Is it possible to propose an appropriate algorithm to allocate the privacy budget adaptively on differentially private decision trees?

RQ2: Is it possible to propose an adaptive budget allocation algorithm on differentially private random forests?

1.4 Contributions to Knowledge

This section presents the three contributions to knowledge claimed in this thesis based on the aforementioned two research questions.

1. An algorithm for adaptive privacy budget allocation on a differentially private decision tree.

2. An algorithm to apply the budget allocation schema on a differentially private random forest.

3. The proposal of Matsushita loss as a splitting criterion on differentially private decision trees.

These contributions to knowledge are described in detail in the following.

Contribution 1: An algorithm for adaptive privacy budget allocation on a differentially private decision tree.

This thesis proposes an algorithm to address the problem of adaptive privacy budget allocation on differentially private decision trees. Whereas prior art assigned a fixed privacy budget per query for a decision tree, the algorithm in this thesis proposes a novel approach to budget allocation, which determines the privacy parameter per query *dynamically*. This approach has several advantages over the state-of-the-art: it allows flexibility in the number of queries (*e.g.*, with respect to the depth of the induced trees), it avoids spending excessive amounts of the privacy budget to choose between attributes that have similar loss, and it allocates the surplus budget to noise-sensitive queries. Finally, it reduces the variance in query responses, allowing algorithms with consistent accuracy levels, despite the introduced noise. The algorithm is validated using synthetic and real-world benchmark datasets and shows competitively high accuracy results in comparison to the prior work. The first contribution corresponds to RQ1 and is addressed in Chapter 3 of this thesis.

Contribution 2: An algorithm to apply the budget allocation schema on differentially private random forest.

This research proposes and validates a framework to create an ensemble of the budget allocation algorithm for decision trees and shape a differentially private random forest. This forest inherits the benefits of the budget allocation schema and assigns a flexible amount of the privacy budget to queries during the tree induction. The evaluation of this algorithm on synthetic and real datasets shows high accuracy in the results of the ensemble classifiers compared to the state-of-the-art models. The second contribution corresponds to RQ2 and is presented in Chapter 4 of this thesis.

Contribution 3: Propose Matsushita loss as a splitting criterion on differentially private decision trees and random forests.

This research introduces Matsushita's loss in Kearns and Mansour's algorithm (Kearns & Mansour, 1999) to apply in the differentially private setting. Matsushita's loss (Kearns & Mansour, 1999; Nock & Nielsen, 2009) is an entropy function that is known to be optimal from the boosting standpoint, but it has not received any attention in relation to differential privacy. This thesis presents this new splitting function in differentially private setting and calculates its sensitivity. The research shows that its sensitivity is larger by a factor of $O(\sqrt{m}/\log m)$ compared to Information Gain's (where m is the number of examples). The accuracy result of this function is compared with other splitting criteria in Chapters 3 and 4 on decision tree and random forest. The third contribution is presented in Chapter 3 and is used in the experiments described in Chapters 3 and 4 of this thesis.

1.5 Research Methodology

The methodology of this study starts with a comprehensive literature review of the existing approaches and techniques based on the current state of knowledge around differential privacy, utility and privacy trade-off, differentially private decision trees and differentially private random forest to identify the strengths and the gaps in these research areas. The main focus of this research is to particularly identify the studies which attempt to utilize and implement differentially private decision trees and random forests.

Identifying the datasets is the next step of the methodology using synthetic datasets and some benchmark real-world datasets with different properties and sample sizes. The differentially private algorithms from this thesis are designed to improve the trade-off between the utility and privacy of the output using these datasets.

In the next step, the proposed model is implemented using the WEKA data mining tool (Witten & Frank, 1999) with the Java programming language. In order to achieve the goals of the study, the results of the implementation are validated and analyzed. The privacy and utility of the model are the parameters on which the system needs to be evaluated. The documentation of the whole process of the study is the final step of this research methodology.

1.6 Organization of the Thesis

The rest of this thesis is organized as follows: in Chapter 2 the privacy aspects of data analytics is discussed and the relevant existing literature on differential privacy and other privacy preservation techniques is reviewed. Decision trees and random forest

algorithms and previous works on differentially private versions are also discussed.

Chapter 3 of the thesis presents the budget allocation optimization algorithm on decision trees. The application of the algorithm on a single decision tree is evaluated with the other state-of-the-art algorithms on differentially private decision trees.

Chapter 4 presents an ensemble of the differentially private decision tree with the budget allocation schema and introduces a forest of differentially private trees. The accuracy output of the forest will be evaluated and compared to the other strong study in this area.

Chapter 5 completes the thesis with a conclusion of the research findings and a short summary of the results. The limitations of the study and the possibilities for future research are discussed in this chapter.

Chapter 2

Literature Review

2.1 Introduction

This chapter reviews the state-of-the-art of privacy-preserving techniques with a special focus on differential privacy (DP) as one of the most recent and efficient models in the statistical analysis domain. Based on a review of the current literature, the trade-off between utility and privacy in differential privacy is investigated, especially when applied to the data mining area. This study focuses on two data mining techniques, the decision tree and random forest, and the application of differential privacy to these two algorithms. To motivate this research, the accuracy/privacy trade-off of these two algorithms on sensitive datasets is investigated, and the research gaps in the literature are highlighted and summarized.

2.2 Privacy Preserving Models

Privacy-preserving techniques are needed to reduce the possibility of identifying sensitive information about individuals or about any datasets which have been released

for data mining. In data publishing scenarios, data are released by the data owner to the data recipient or the public. There is always a huge demand for publishing data or to exchange it between various parties. However, the original format of the data may contain sensitive information about individuals. This sensitive information can potentially violate the privacy of individuals if it is shared with other parties without precautions being taken. Privacy-preserving data mining (PPDM) and privacy-preserving data publishing (PPDP) techniques have been proposed which present methods for using and storing this sensitive data. Guidelines have been published to restrict the actions that can be conducted on data, and to control the process of modifying the original data and safely releasing it. Previous studies on PPDP and PPDM (Wang et al., 2010; Agrawal & Srikant, 2000; Aggarwal & Philip, 2008) focus more on categorizing the different approaches and techniques for privacy-preserving data publishing and on minimizing the risk of privacy violation, especially using data mining algorithms and applications. The main aim of these studies is to research the different types of privatizing techniques and the privacy models which utilize these techniques.

When defining privacy, a major challenge is modeling the background knowledge of an adversary. When adversaries have some background knowledge about their victim, simply removing specific identifiers such as names or user IDs, which are called as quasi-identifiers (QIDs), does not preserve privacy. The values of QID attributes could be publicly available from different sources and so, to identify any individual from published data, all that is needed is to simply join the QID attributes to an external data source. Different privacy preserving models have been proposed to avoid various types of attacks on the sensitive attributes in a dataset. Some of the

influential models are k-anonymity (Sweeney, 2002), (α, k)-anonymity (Wong, Li, Fu, & Wang, 2006), l-diversity (Machanavajjhala, Kifer, Gehrke, & Venkitasubramaniam, 2007), t-closeness (Li, Li, & Venkatasubramanian, 2007), personalized privacy (Xiao & Tao, 2006), δ-Presence (Nergiz, Atzori, & Clifton, 2007), (d, Y)-privacy (Rastogi, Suciu, & Hong, 2007), distributional privacy (Blum, Ligett, & Roth, 2013) and (ϵ, δ)-differential privacy (C. Dwork, 2011a), etc. Here we explain these techniques and describe some of their advantages and disadvantages.

k-Anonymity (Sweeney, 2002): The k-anonymity privacy model is one way to limit the effect of attacks on sensitive attributes. The requirement of this privacy model is that any individual in the data should not be identifiable from a group smaller than k based on their quasi-identifier. This means that if one record in the dataset has some value QID, at least $k - 1$ other records also have the same QID values, so the minimum group size of QID in this approach is at least k records. The k-anonymity model guarantees that an attacker will not be able to link private information to groups of less than k individuals, because every combination of public attribute values in the published data appears in at least k rows. By hiding the record of a victim in a large group of records with the same QID, k-anonymity can prevent record linkage attacks. However, the attacker may still be able to conduct a successful linking attack if he can obtain more background information about the target victim. The attacker can also associate the victim with their confidential value if most records in a group have similar sensitive attribute values (Wang et al., 2010). The choice of QID in this model remains an open issue, the general idea being is to diminish the correlation between QID attributes and sensitive attributes.

(α, k)-Anonymity (Wong et al., 2006): The concept of confidence measurement

in k-anonymity is introduced by Wong et al. This approach requires every QID attribute in a dataset to be shared by at least k records and the confidence measurements between each QID and the sensitive attributes should be always smaller than a threshold α, which is assigned by the data publisher. However, the sensitive values may be skewed so this approach can lead to high distortion.

l-Diversity (Machanavajjhala et al., 2007): As introduced by Machanavajjhala et al., l-diversity has been proposed to overcome the vulnerability of k-anonymity against background knowledge attacks when the attacker has additional information about the victim. In an attempt to remove the correlation between the QID attributes and the sensitive attributes, the l-diversity model requires that every QID group should contain at least l number of "well-represented" values for sensitive attributes. This ensures that in each QID group, there are at least l distinct values for the sensitive attribute. This model also has some limitations. It implicitly assumes that the frequencies of the different values of a confidential attribute are similar, therefore each sensitive attribute takes values uniformly over its domain. In real-world scenarios when this is not the case, achieving l-diversity may cause a huge loss of data utility (Wang et al., 2010).

t-Closeness (Li et al., 2007): Li et al. observed that l-diversity does not prevent attribute linkage attacks when the overall distribution of a sensitive attribute is skewed. They introduced the t-closeness model to overcome skewness attacks. The requirement of this privacy model is to keep the distribution of a sensitive attribute in any group of QIDs as close as possible to the distribution of the attribute in the overall table. t-Closeness still has some limitations, in particular, it is not possible to specify different protection levels for different sensitive values in this model. It also

could damage the correlation between QID and sensitive attributes significantly, because it requires the distribution of sensitive values to be the same in all QID groups, and this will result in a loss of data utility (Wang et al., 2010).

Personalized Privacy (Xiao & Tao, 2006). Personalized privacy allows each record owner to determine their own personal privacy level. Every sensitive attribute in this model has a taxonomy tree and each record owner of the tree specifies a guarding node in it. The record owner's personal level of privacy is violated, if within the subtree of their guarding node, an attacker can derive any domain sensitive value with a probability greater than a certain threshold. This probability value is called the breach probability. However, in practice, how each individual record owner can establish and choose their guarding node is unclear.

δ-Presence (Nergiz et al., 2007): This privacy model suggests that inside a given range δ, the probability of deriving the occurrence of any potential victim's record is bounded. The δ-presence model can discursively avoid record and attribute linkage attacks, because if the attacker has at most $\delta\%$ assurance that the prospective victim's record is present in the published table, then the probability of obtaining the sensitive attribute by running an effective linkage attack is at most $\delta\%$. Although δ-presence is a comparatively "safe" model, it assumes that the data publisher has access to the exact same external knowledge as the attacker and this may not be a true assumption in practice.

(d, Y)-Privacy (Rastogi et al., 2007): This model is a probabilistic privacy definition based on the prior and posterior probability of the presence of a record of a victim in the data table, before and after examining the published table. While most earlier works on privacy preserving models do not have a formal guarantee on privacy and

information utility, the difference of the prior and posterior probabilities in (d, Y)-privacy is bounded and a provable guarantee on privacy and utility is provided in this model. In this definition, Y shows the formal guarantee of the difference between the prior and posterior probability, and d indicates that this privacy model is designed to protect only against attacks that are d-independent. The d-independent attack is from an attacker whose prior belief is smaller than d for all records. Rastogi et al. demonstrate that only when the prior belief is smaller than d, a reasonable trade-off between privacy and utility is achievable. However, this privacy model was created to protect only against attacks that are bounded by a probability distribution and this assumption is not necessarily true in some real-world applications (Machanavajjhala, Kifer, Abowd, Gehrke, & Vilhuber, 2008). In comparison, in differential privacy, the records do not have to be assumed independent, or a previous belief of an attacker is not bounded by a probability distribution.

Distributional privacy (Blum et al., 2013): The distributional privacy model was motivated by learning theory. Using this privacy model when a data table is drawn from a specific distribution, the table should only disclose information about that specific underlying distribution and nothing else. Compared to the differential privacy model, distributional privacy is a stronger privacy notion and is capable of answering all queries over a discretized domain in a concept class of polynomial VC-dimension. However, the computational cost of the algorithm is high and developing efficient algorithms for more complicated queries with this privacy model is yet to be achieved.

(ϵ, δ)-*Differential privacy (DP) (C. Dwork, 2011a, 2008)*: Differential privacy is a mathematically strict privacy model that holds no assumptions about an attacker's background knowledge. Dwork suggested comparing the privacy risk in a published

dataset with and without the data from the record owner, instead of comparing the prior and posterior probability before and after accessing the published data. A differentially private mechanism ensures that all outputs are insensitive to any individual's data and that the participant in a dataset is not at risk of being identified in a database. The mathematically firm foundation of differential privacy also guarantees that no risk can occur by joining different datasets. Based on the same intuition, the results obtained from the algorithm will not be very different if a record owner decides not to provide their information to the data publisher.

Some of the previous studies (Wei, Du, Yu, & Gu, 2009; Clifton & Tassa, 2013) focused on a comparison between different models of privacy, including differential privacy. Wei et al. (2009) compared four of the most important privacy models, k-anonymity (Sweeney, 2002), l-diversity (Machanavajjhala et al., 2007), t-closeness (Li et al., 2007) and differential privacy, and formalized the notion of privacy and utility in this context. As a result of this comparison, they showed that differential privacy is a suitable model when data are supposed to be released to any unknown third party, such as an external data analysis company, while l-diversity shows better results in releasing data to a completely safe and trusted third party. The usage of the differential privacy mechanism in different scenarios is very well-defined in this research.

Differential privacy is an area of research which seeks to provide accurate, statistical guarantees against what an attacker can derive from learning the results of the model. This privacy model has become an increasingly popular area of research since its introduction by Dwork in 2006, with many theoretical analyses and practical instantiations contributing to it. Differential privacy is the core privacy model of this

research and in the next section of this chapter, the robust definitions of this privacy model are presented and its application is discussed.

2.3 Differential Privacy

Differential privacy (DP) has been introduced in the statistical database domain (a statistical database is a database used for statistical analyses purposes, and in this kind of database it is often desirable to allow query access only to aggregate data, not to the individual records). Considering external knowledge, it is claimed that privacy breaches can occur even for people who do not participate in a statistical database (C. Dwork, 2011a). The main objective of differentially private techniques is to limit the risk of an increasing number of privacy breaches for participants of statistical databases. This mechanism ensures that the statistical outputs of two databases that differ only by a single participant are almost the same, and this is done by adding a sufficient level of noise to the outputs.

Based on the previous works in this area (Dinur & Nissim, 2003; C. Dwork & Nissim, 2004; C. Dwork et al., 2006; C. Dwork, 2008, 2010), a complete definition of the concept of differential privacy (DP) is given by Cynthia Dwork in a comprehensive work in 2011 (C. Dwork, 2011a). She gave a definition of differential privacy and the techniques that enabling its achievement. She discussed the application of the techniques and the results of their application. Differential privacy captures the notion of individual privacy by ensuring that the addition or removal of an individual's record in a database does not have a substantial impact on the output produced by the mechanism. The data owner can be assured that the presence of their data in a dataset will not change the information that can be discovered from the differentially

private method. A formal proof by C. Dwork (2011a) showed that ϵ-differential privacy is capable of providing a strong guarantee against attackers with arbitrary background knowledge. Comparing the results from a published dataset, with and without the presence of a record of a data owner, demonstrated this strong guarantee. C. Dwork (2007) proved that if the number of queries is sub-linear in n (where n is the number of records in the database), the noise to achieve differential privacy is bounded by $O(\sqrt{n})$. An overview of different research works on differential privacy can be found in the survey by C. Dwork (C. Dwork, 2008, 2011b) and in their recent book (C. Dwork & Roth, 2014). As the significance of differential privacy comes from its rigorous definition, in the next section, we give the most important definitions of differential privacy which have been applied in this research.

2.3.1 (ϵ, δ)-Differential Privacy: Basics

Differential privacy is the notion of privacy for database access mechanisms. It captures the idea of individual privacy by ensuring that the addition or removal of an individual's record in a database does not have a substantial impact on the mechanism's output. Precise definitions of differential privacy (DP) are as follows:

Definition 1. *(C. Dwork, 2011a) A randomized algorithm M gives (ϵ, δ)-differential-privacy if for all datasets D_1 and D_2 differing on, at most, one element and all $S \subset Range(M)$,*

$$Pr(M(D_1) \in S) \leq e^{\epsilon}.Pr(M(D_2) \in S) + \delta. \tag{2.1}$$

the probability is taken over from the coin tosses of M.

The algorithm M is ϵ-differentially private if it (ϵ,0)-approximates differential privacy.

The value of ϵ allows us to control the level of privacy; normally, ϵ is a small public parameter greater than 0. Lower values of the privacy parameter ϵ and $\delta \in [0, 1]$ imply a stronger guarantee of privacy for the results.

Differential privacy is achieved by adding noise to the outcome of a query. It is provided by calibrating the magnitude of noise based on the sensitivity of a function. This sensitivity indicates the maximal possible change in its value by adding or removing a single record.

Definition 2. *(C. Dwork, 2011a) The* ***global sensitivity*** $\Delta(f)$ *of a function f is the smallest number δ such that for all D_1 and D_2 which differ on at most one element, $|f(D_1) - f(D_2)| \leq \delta$.*

Theorem 1. ***Laplace Mechanism*** *(C. Dwork, 2011a) Given a function $f : D \rightarrow R^d$ over an arbitrary domain D, the computation $M(X) = f(X) + (Lap(\Delta(f)/\epsilon))^d$ provides ϵ-differential privacy (DP).*

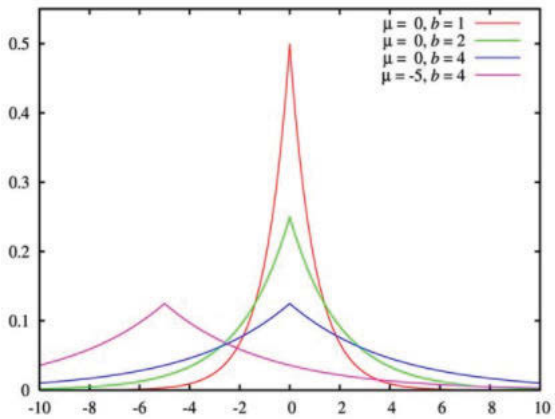

Figure 2.1: Probability density function of Laplace distribution

Let $Lap(\mu, \lambda)$ represent the Laplace distribution with a mean of μ and a standard deviation of λ. Therefore, this is a random variable with a probability density (PD) function $\frac{\lambda}{2}e^{-|x|\lambda}$. Figure 2.1 shows the probability density function of the Laplace distribution, when μ is the mean and $b = \Delta(f)/\epsilon$.

For example, we can imagine the count queries as a function over a set S, then $f(S) = |S|$ has a sensitivity of $\Delta(f) = 1$, because adding and removing a record will change the count of all the records by at most 1. Therefore, a noisy count that returns $M(S) = |S| + Lap(1/\epsilon)$ satisfies ϵ-differential privacy (DP).

Theorem 2. ***Composition Theorem*** *(C. Dwork, 2011a) The sequential application of mechanisms M_i, each giving ϵ_i-differential privacy (DP), satisfies $\sum_i \epsilon_i$-differential privacy (DP).*

In this thesis, the composition theory is particularly useful when inducing differentially private decision trees, stating that the ϵ privacy budget of the whole tree is the sum of the budget that is applied on every differentially private query on the tree.

Exponential Mechanism(McSherry & Talwar, 2007): Apart from using Laplace distribution to add noise to the results of queries, another technique for achieving differential privacy (DP) is the exponential mechanism. This mechanism has a quality function q which gives higher scores to a calculation with better outcomes. For a given database and parameter, the quality function induces a probability distribution over the output domain, from which the outcome is sampled by the exponential mechanism. High scoring outcomes are exponentially more likely to be chosen by this probability distribution while ensuring ϵ-differential privacy (DP).

Definition 3. *(McSherry & Talwar, 2007) Let $q : (D_n \times R) \mapsto \mathbb{R}$ be a quality function that, given a database $D \in D_n$, assigns a score to each outcome $r \in R$. Let*

$\Delta(q) = \max_{r,D_1 \sim D_2} ||q(D_1, r) - q(D_2, r)||_1$. *Let* M *be a mechanism for choosing an outcome* $r \in R$ *given a dataset instance* $D \in D_n$. *The mechanism* M *is then defined by*

$$M(D, q) = \left\{ \textit{returns } r \textit{ with probability} \propto \exp\left(\frac{\epsilon q(D, r)}{2\Delta(q)}\right) \right\} \tag{2.2}$$

In addition to the Laplace mechanism, this research also utilizes the exponential mechanism in the induction of differentially private trees. The scoring technique in the exponential mechanism definition makes this mechanism a very good candidate for attribute selection in decision trees (Friedman & Schuster, 2010). The best attribute to split the node receives a higher score from the exponential mechanism and this raises the probability of choosing this attribute in the induction process.

2.3.2 Properties and Benefits

Some of the properties and benefits of differential privacy are as follows (C. Dwork & Roth, 2014; C. Dwork & Pottenger, 2013):

- **Prevent linkage attacks**: Anonymization is one of the most important privacy preserving techniques previously used to avoid disclosure of sensitive attributes in a dataset. It works by removing the sensitive attributes from the attribute set and publishing the rest of the dataset. However, this technique is prone to linkage attacks which allow the discovery of the value of anonymized records by matching anonymized records with non-anonymized records from a different dataset. On the other hand, being differentially private is a property of the data access mechanism and is not related to the absence or presence of any auxiliary or background information that is available for an attacker, therefore differential privacy can neutralize linkage attacks.

- **Addresses arbitrary risks**: Differential privacy emphasizes that the addition or removal of an individual participant does not have any effect on the result of the privacy mechanism.

- **Worst-case guarantees**: Differential privacy can provide privacy for all types of datasets, even unusual databases that do not comply with possibly mistaken assumptions about the distribution from which the dataset rows are learned.

- **Quantification of privacy loss**: Privacy loss in differential privacy is measurable. Privacy loss is *quantified* by the maximum, overall $C \in Range(M)$ where M is the computation as defined in Theorem 1, and all adjacent databases D_1, D_2 of the ratio $\ln\left[\frac{Pr[M(D_1)\in C]}{Pr[M(D_2)\in C]}\right]$. $(\epsilon, 0)$-differential privacy ensures that this privacy loss is bounded by ϵ.

- **Automatic and oblivious composition**: In differential privacy, the quantification of loss also permits the cumulative privacy loss to be controlled over multiple computations. The design and analysis of complex differentially private algorithms is possible from simpler differentially private building blocks by applying composition theory on differentially private mechanisms. A simple composition theorem shows that given two differentially private databases, where one is $(\epsilon_1, 0)$-differentially private and the other is $(\epsilon_2, 0)$-differentially private, the aggregated risk encountered by participating in (or opting out of) both databases is in the worst case $(\epsilon_1 + \epsilon_2, 0)$-differentially private.

- **Group privacy**: Differential privacy allows the analysis and control of privacy loss acquired by groups, such as families. For a group of size k, an $(\epsilon, 0)$-differentially private algorithm is proven to be automatically $(k\epsilon, 0)$-differentially

private.

- **Building complex algorithms from primitives**: The quantitative, rather than binary, nature of differential privacy enables the analysis of the cumulative privacy loss of complex algorithms constructed from differentially private primitives, or "subroutines" (C. Dwork & Pottenger, 2013).

- **Closure under post-processing**: Differential private mechanisms are resistant to post-processing. A function cannot be computed on an output of a differentially private mechanism by a data analyst and make the results less differentially private. No matter what background knowledge is available about a dataset, the data analyst cannot increase the privacy risk by computing on the output that is generated by a differential private mechanism.

2.3.3 Limitations

Differential privacy comes with disadvantages and limitations, the most important of which are listed in the following. Although some of these limitations are due to the nature of differential privacy and can not be avoided, this research considers them and gives some practical solutions where possible.

- **Utility concerns**: The greatest concern in the literature is that a very low level of privacy is needed to have useful data and acceptable utility, and differential privacy actually sacrifices utility for the sake of privacy preservation. The utility of the results in a differentially private setting is measured by the accuracy of the statistics estimated in a privacy-preserving manner. As the notion of differential privacy implies a bound on utility, one of the most important challenges in this

area is investigating mechanisms to provide the best utility while guaranteeing differential privacy. In particular, in situations which should allow multiple queries in differentially private settings, noise must be added proportionally to the number of queries. However, adding too much noise can deprive the utility of the system outputs. The main focus areas of a very large range of studies in this area, including this thesis, is to consider how to evaluate the trade-offs between privacy and utility in differential privacy models.

- **Numeric-data corruption**: Dankar and El Emam (2012) addressed some of the limitations of differential privacy in their study. They mentioned the serious concerns raised in Muralidhar and Sarathy (2010) and Sarathy and Muralidhar (2011) about the application of differential privacy in numerical data, claiming that the level of noise added from the Laplace mechanism can be so large that it makes the data unusable. This thesis considers this limitation with the Laplace mechanism and attempts to explore the exponential mechanism in the tree induction process which has previously shown better results in the process (Friedman & Schuster, 2010).

- **Sensitivity calculation**: Another limitation, addressed by Muralidhar and Sarathy (Muralidhar & Sarathy, 2010; Sarathy & Muralidhar, 2011), raised a question about the sensitivity calculation and verification of differential privacy. The difficulty of calculating the sensitivity of queries in unbounded domains was emphasized. Although there can not be a general method to calculate the sensitivity of all differentially private mechanisms, this thesis proposes a general definition to calculate the sensitivity of some of the splitting criteria during the induction of differentially private decision trees as part of contribution 3.

- **Concern about the value of the privacy budget** ϵ: The value of ϵ privacy budget in differential privacy remains a significant question, thus presenting another concern in relation to differential privacy. There are no actual experiments in the literature to guide the user in choosing the right value of privacy budget ϵ in different scenarios. This thesis attempts to minimize the effect of choosing a wrong initial value for the privacy budget by introducing a budget allocation mechanism for decision trees. The budget needs to be assigned by the user and then utilized until exhausted. Even small values of privacy budget give a good level of utility in the statistical results of the trees, as shown in Chapter 3.

- **Differential privacy and health data** : Some concerns have also been expressed about using differential privacy in health data (Dankar & El Emam, 2012; Loukides, Liagouris, Gkoulalas-Divanis, & Terrovitis, 2014; Poulis, Loukides, Skiadopoulos, & Gkoulalas-Divanis, 2017). As health data, by its nature, is messy and consists of numerical and categorical data, adding Laplacian noise can harm the exact value of these data and may have a huge impact on the results derived. Therefore, the first concern is the utility of data and the fact that differential privacy generates noisy answers through limited number of queries. Health data could be very sensitive to noise, specially where any change to the values of demographics and diagnosis codes could be very important (Dankar & El Emam, 2012; Loukides et al., 2014; Poulis et al., 2017). In addition to the utility of the data, the availability of data is also important to the health analysts. The meaning and usage of data needs to be decided through a proper examination of the data by analysts to inform appropriate decisions, and in

most cases, analysts would prefer to see the actual data and not a noisy version of it. Also, the correlation between data is very important in health data. Just as the correlation between drugs, diagnoses and laboratory results is very important, if any of these data are distorted in any way by differentially private noisy techniques, the correlation has the potential to be destroyed, and the data is no longer usable. As health data analysts are accustomed to the traditional mechanism and commonly used tools, it may be hard to convince them to adopt a completely interactive and new mechanism when they do not yet understand it properly, so a non-interactive mechanism may be the most suitable approach at the current time. An example is the non-interactive framework that is adapted in the research by Mohammed, Jiang, Chen, Fung, and Ohno-Machado (2012). They presented a sanitization method to achieve differential privacy on heterogeneous data that is a combination of relational data and transaction data, in which the relational data contains raw patient data and the transaction data contains the diagnostic codes. But even in the non-interactive mechanism, a couple of drawbacks makes analysis very restrictive. In many health scenarios and analytics, actual data needs to be published without noise, thus highlighting one of the current concerns. Another important consideration is the legal framework. The health standards often do not accept any risk to health information with reasonable standards required for data. An important parameter like ϵ in differential privacy with no distinct meaning and no exact value is difficult for legal practitioners to justify. In practice, convincing data analysts and the public to use a dataset that has been transformed into many forms may be challenging. In the context of differential privacy, the parameters

need to be related to more common notions to allow easier communication to the public. For example, explaining the meaning of the value ϵ to a patient who needs to disclose their information or provide access to their data is a big challenge (Dankar & El Emam, 2012). The datasets for this thesis are selected from different areas excluding the health datasets, and dealing with the consequences of working with health data in differential privacy is left for future work on this mechanism.

2.3.4 Accessing Private Data through Interactive and Non-Interactive Settings

In differential privacy, there are two main categories of settings for accessing private data: interactive and non-interactive settings. In this section, the findings in the literature on trusting the data publisher and the methods for accessing data for both of these two settings are reviewed.

- **Non-Interactive Differential Privacy**: (Privacy-Preserving Data Publishing (PPDP)) In the non-interactive setting, the trusted data collector publishes a *sanitized* version of the collected data. Once the dataset is created, it can be published for external analysis, and the data miner is given full access to the synthetic dataset. The techniques that are used in this setting include the perturbation technique; sub-sampling; removing well-known identifiers; and releasing different types of synopses and statistics. Various categories and mechanisms are available for the non-interactive model to ensure privacy. Dankar and El Emam (2012) divided them into two categories :

1. **Noise-based mechanisms**: Blum et al. (2013) looks at the problem of differential privacy from the learning theory standpoint and releases a synthetic database that is useful for all queries in a specific class that is known ahead of time, which is non-efficient and requires the class to be known.

   Releasing histograms of noisy sanitized data is also investigated in C. Dwork (2008) which suffers from a large amount of noise that is added to the histograms with small sensitivities of one.

   Another (non-efficient) alternative, introduced by Barak et al. (2007) and later evaluated by Fienberg, Rinaldo, and Yang (2010), is to construct a positive and integral synthetic database that uses Fourier transforms to preserve all low order marginals up to a small error. This method is not suitable for large and sparse tables.

2. **Alternative mechanisms**: Releasing set-value data for counting queries with a probabilistic top-down partitioning algorithm is investigated in Chen, Mohammed, Fung, Desai, and Xiong (2011). Their algorithm starts by creating a context-free taxonomy tree and generalizing all items under one partition and then generates distinct sub-partitions based on the taxonomy tree that split the records into more specific disjoint sets. The authors did not justify the interdependence between the utility and privacy parameters in their model.

   Li, Qardaji, and Su (2011) proposed another approach in this category. In this work, a random sampling step is applied to the data followed by a generalization scheme that does not depend on the database and then suppresses all the tuples that occur less than k times. The usage of a

data-independent generalization in their model may result in poor utility. Other work in this category by Mohammed et al. (2011) is based on the generalization of the contingency table to add noise to the counts. They demonstrated that the counts become larger than the added noise because of the generalization step that increases the cell counts. Although there is no optimal generalization method that would produce relatively high enough counts in their study, their model is one of the most effective works in this category.

A data miner could prefer results that are created with a non-interactive setting of differential privacy because the synthetic dataset from the non-interactive setting can be analyzed just as if it were the original. However, initial results suggest that it is not always possible to publish a sanitized version of the dataset. Also, in order to ensure that a modified version of the dataset is usable, the synthetic dataset should be specially crafted in a manner that is suitable for the analysis to be performed on it. This thesis considers the drawbacks of this setting and uses the interactive setting in the tree induction process. However, the results of the experiments from this thesis in Chapter 3 are compared with Mohammed et al. (2011) which is one of the differentially private trees in non-interactive settings.

- **Interactive Differential Privacy**: (Privacy-Preserving Data Mining (PPDM)) In the interactive setting, the data collector provides a platform for users to make queries (at times, limited in number) about the data and to receive some noisy answers. In this setting, a privacy-preserving layer authorizes access to the data store. This privacy-preserving layer can only allow a limited number

of query operators carried out on the dataset. It audits the requested queries and controls the privacy budget. It blocks the access to the data as soon as the privacy budget is exhausted. Therefore, in order to avoid a sudden denial access to the data before the algorithm has finished executing completely, the data miner needs to plan properly in advance how to spend the privacy budget. The gain is the privacy-preserving data access layer takes care of enforcing privacy requirements, and the data miner does not need to worry about it.

In this setting, the challenge is to find a differentially private mechanism that is able to achieve the stronger privacy guarantee of ϵ-differential privacy, and

- answer random queries (any type of queries) (Blum et al., 2005; C. Dwork, 2011a),
- answer a large number of queries while providing non-trivial utility for each of the queries (Blum et al., 2013; Hardt & Talwar, 2010; Kasiviswanathan, Lee, Nissim, Raskhodnikova, & Smith, 2011),
- answer queries adaptively (C. Dwork, 2011a; Hardt & Talwar, 2010; Kasiviswanathan et al., 2011),
- is accurate and efficient (C. Dwork, 2011a; Hardt & Talwar, 2010; Jagannathan et al., 2012; Friedman & Schuster, 2010).

Most of the works on data mining using decision trees are focused on the interactive mechanisms for accessing data in differential privacy (Friedman & Schuster, 2010; Jagannathan et al., 2012; Fletcher & Islam, 2015a; Rana et al., 2015). This thesis is based on the interactive mechanism where a data miner can pose aggregate queries through a private mechanism and a database owner answers these queries in response.

2.3.5 Allocating the Privacy Budget

Every time the data is accessed in differentially private settings, a small amount of the privacy budget needs to be spent. The amount of noise that needs to be added to the output of queries is determined by the privacy parameter ϵ. This extends to all applications of differential privacy where ϵ is at the core of its definition. However, the extent of work on how to choose an appropriate value for ϵ in the real world is very limited, both for maintaining a proper amount of privacy protection for each individual in the database and for ensuring an almost equal level of efficiency for private and non-private queries. C. Dwork and Roth (2014) left these questions for the users to answer based on the user's demand and responsibility. Only two studies in the literature appear to present practical guidelines for choosing ϵ: the study of Vu and Slavkovic (2009) from the perspective of utility and the work of Hsu et al. (2014) from the perspective of privacy.

Vu and Slavkovic (2009) provided several results, within the context of clinical trials, on how to use hypothesis testing in a scenario where sensitive information is involved. They focused on the sample size of the test and showed that if privacy were not a concern, how many more records would be needed to produce the same quality of results that the test would produce. Two main factors of the hypothesis test are considered in this work: the confidence level (type I error) and the power (type II error). Scaling the number of records by the user to estimate how many samples they would need to achieve strong results for their hypothesis test is called the sample size "correction factor". The sample size correction factor equations are available in Vu and Slavkovic (2009). If a user can estimate the sample size correction factor, it is possible to use Vu and Slavkovic's work to achieve the same results in a differentially

private manner.

The other work that is presented by Hsu et al. (2014) focuses on the other aspect of the utility-privacy trade-off, that is, the privacy aspect. The first functional procedure to determine how small the privacy budget ϵ needs to be to suitably protect privacy for a given scenario is proposed by these authors. They model the problem of privacy loss in terms of money. In this model, the user who is collecting or using the sensitive data will be assigned a financial budget. The amount of risk of each individual user who is included in the data in any given scenario is assessed. By estimating the probability of an individual's privacy being maliciously breached, the user can assess the average risk per individual in the data and calculate the probable financial cost that is associated with that breach. Practical guidelines for how small or large the privacy budget ϵ can be are included in this model. The probability of a breach is directly tied to ϵ, and there are a lot of examples to show how this privacy budget/money can be spent; the output of a system can be modified with probability proportional to $\exp(\epsilon)$ by the existence of a record in the system. The user of the system can decide how many samples they want to have access to, how large their privacy budget can be and how much money they want to spend. The money can be spend on encouraging more people to participate in the study, hence collecting more data, or making payments to individuals to compensate them for a breach of their privacy.

It is a delicate process to find a balance between an algorithm that optimizes utility and the differential privacy requirements of the system and this procedure should be undertaken with great care. This thesis seeks to find this balance by analyzing the sample size at each internal node of a decision tree and heuristically deciding on the amount of allocated budget.

2.3.6 Differential Privacy and Data Mining

Data mining and machine learning are concerned with using the right features to build the right models to achieve the desired tasks. These tasks include binary and multi-class classifications, regression, clustering, descriptive modeling, etc. The conflict between data mining and differential privacy is surveyed by several works in this area. A broad overview of machine learning with differential privacy is given by Sarwate and Chaudhuri (2013). They briefly considered classification, regression, dimensionality reduction, time series, filtering and a collection of some of the basic ingredients of differential privacy, such as the exponential mechanism, robust differentially private statistics, and the differences between input, output and objective perturbation. The focus of another work by Ji, Lipton, and Elkan (2014) is on specific differentially private data mining techniques, and gave an overview of the work done with naive Bayes models, linear regression, logistic regression, linear support vector machines, kernel support vector machines, decision trees, online convex programming, clustering, feature selection, principal component analysis and statistical estimators.

Several advantages of differential privacy make it a very good candidate in data mining. The robust mathematical definition of the model makes it possible to prove if a mechanism complies with differential privacy requirements or not, and to decide which computations can or cannot be made in a differentially private framework. Also, it is independent of assumptions about the background knowledge that an adversary has about the data, so the information that is shared from different data resources in the future and past can not change the decision about the present data and its analysis. Consequently, the huge amount of personal information that is collected by

big companies in the market can be accessed for data mining with this framework.

The composability of differential privacy is another advantage of the model that is useful in data mining. When data providers let different groups of people access their data, they can be assured that there will not be any privacy breach due to conspiracy between adversarial parties or due to repetitive access by the same party. This also means that when an adversary combines two independent differentially private data releases coming from different parties, the result is still differentially private and can not violate the privacy of the participant in the published data.

One of the first steps that leads to the practical implementation of differential privacy for real-world scenarios was the introduction of the Privacy Integrated Queries platform (PINQ) (McSherry, 2009). PINQ is a programmable privacy preserving layer that stands between the data provider and the data analyst. This framework allows data analysts to have secure access to the data through the provided application interface. The data analysts using this model do not need to consider the privacy aspects of the work and they do not need to have expert knowledge in the privacy domain. Another advantage of the design of the PINQ analysis language and its careful implementation is the formal guarantee of differential privacy that is provided to any user in the platform with all the different needs and usages from the data.

The unconditional structural of the PINQ framework guarantees to considerably expand the scope for the design and deployment of privacy-preserving data analysis, especially by non-experts, because the framework does not require an analyst's expertise or diligence to be trusted. A privacy cost is associated with each interactive query in the PINQ mechanism and access to the database is blocked when a predetermined privacy budget is exhausted by the analyst. No more queries are allowed when the

initial privacy budget of the system is exhausted. Unfortunately, this means that to avoid losing access to the dataset before the analysis is finished, the analyst must carefully manage and control how the privacy budget in the framework is spent.

Iterative noise reduction, known as iReduct (Xiao, Bender, Hay, & Gehrke, 2011), is a differentially private algorithm that computes the answers to the queries with reduced relative errors. Different amounts of noise are injected into the results of different queries in the iReduct framework. In this setting, less noise is more likely to be injected into smaller values and more noise into larger values. A re-sampling technique in this algorithm applies correlated noise to the results to improve data utility. The evaluation of the framework's performance is with an instantiation of iReduct that generates marginals, that is, projections of multi-dimensional histograms onto subsets of their attributes. The main consideration of iReduct is how to add noise proportionally to fit the result of the queries.

The target of the approach is scenarios where the count queries overlap each other, which can happen, for example, when at least one tuple can be the answer to a couple of different queries. In this scenario, if less noise is added to the result of one query, to ensure privacy protection, a larger magnitude of noise will be added to another query. The idea is to first use Laplace distribution to compute a noisy answer for each query, and then, instead of using the exact answers, to use the *noisy* answers to determine the scale of noise. Adopting the iReduct framework for learning a naive Bayes classifier is a good example of applying it to a data mining problem. This approach, however, is inapplicable for a lot of problems including those studied in this thesis because of the two-step adoption of the count queries, since the count queries are concerned in a lot of scenarios, such as histograms or decision trees which

are mutually disjoint.

Hardt, Ligett, and McSherry (2012) studied the performance of linear queries that can handle a wide range of data analysis and learning algorithms. Linear queries are the equivalent of statistical queries. The algorithm in Hardt et al. (2012) called the Multiplicative Weights/Exponential Mechanism (MWEM), offers an experimental result for a differentially private data release and a combination of expert learning techniques (multiplicative weights) with an active learning component (via the exponential mechanism). The multiplicative weights approach in this algorithm maintains and corrects an approximating distribution through queries in which the approximate and true datasets differ. The exponential mechanism part of the algorithm selects the queries that are most descriptive and useful for the multiplicative weights algorithm. The authors of the MWEM algorithm claim that their algorithm matches the best recognized models and gives almost the best theoretical accuracy guarantees for differentially private data analysis with linear queries.

Another work to mention in this section is the sub-linear queries (SuLQ) framework (Blum et al., 2005), which modifies privacy analysis to real-valued functions and arbitrary row types by putting a bound on privacy noise. This work is of particular interest to this thesis as it examines the computational power of the SuLQ framework by applying it to different statistical models, such as the ID3 algorithm, and also uses principal component analysis, K-means clustering, etc. The authors show that one simple way to design a differentially private mechanism is to take the covariance matrix, adding an appropriate amount of noise to that matrix and releasing the matrix's result. They show that, for real-valued functions on arbitrary rows, to maintain a given amount of ϵ of information leakage under T queries with a probability of at

least $1-\delta$, a normally distributed noise with a mean of 0 (zero) and variance of $2T\left(\log(\frac{1}{\delta})/\epsilon^2\right)$ is sufficient. This is equal to saying that the amount of added noise is almost always $O\left(\sqrt{T\log(\frac{1}{\delta})}/\epsilon\right)$. This distribution is dependent on the number of records in the dataset: when the number of records increases, the statistical result will be more accurate. From a theoretical standpoint, the privacy budget problem can be simply eliminated for many calculations by working with a larger dataset. For example, the accuracy of a result from a large dataset with 10000 records will suffer a lot less than a smaller dataset of 1000 records, when the magnitude of added noise to a count query is as large as 100. However, this course of action may not always be achievable in practice. One reason is because collecting and accessing large amounts of information may incur higher costs than the analyst can afford. Also, in some cases, even when a large sample size is available, the studied phenomenon may apply only to a small subset of the population and this considerably limits the number of records that are available for the analysis. Therefore, the efficient use of the privacy budget is required to use the differential privacy mechanism in practice, and so far, the research community has not paid enough attention to this area.

To address this problem, the focus of this thesis is on decision tree induction as a sample data mining application. A different quality of results is achieved by using different methods for measuring a quantity with a differential privacy mechanism McSherry. This thesis studies a differential privacy querying mechanism on the decision tree induction algorithm (similar to the PINQ framework), and calculates the effect of this mechanism on the privacy costs of the model and the quality of the resulting decision trees. Based on the previous literature in this area, different methods for decision tree induction incur different privacy costs, therefore this thesis

is motivated to re-design the decision tree algorithm with privacy costs in mind. The literature review in the rest of this chapter refines and narrows the existing work on differentially private decision trees and random forests.

2.4 Decision Trees

Classification is the process of predicting a certain label for a record in a dataset. With this description, a classifier is a model that inputs unlabeled data, and using a decision-making process, outputs a predicted label. Decision trees are a non-parametric supervised learning method used for classification and a good example of a classifier (Han, Pei, & Kamber, 2011).

Decision trees are attractive to data scientists compared to other kinds of supervised learning methods. These advantages include:

- high human interpretability (Huysmans, Dejaeger, Mues, Vanthienen, & Baesens, 2011; Letham, Rudin, McCormick, Madigan, et al., 2015)
- non-parametric design (Murphy, 2012)
- relatively low computational cost (Han et al., 2011)
- the ability to discover non-linear relationships between the attributes (Han et al., 2011)
- resilience to missing values (Quinlan, 2014)
- the ability to handle both continuous and discrete data (Quinlan, 1996)
- the ability to handle non-binary labels (Quinlan, 1996)

They also have two main disadvantages: the tendency to over-fit the data if the tree is too deep and high fluctuation in the tree results with small changes in the data. However, these disadvantages can be reduced by introducing techniques to limit the depth to which the trees can grow; pruning untrustworthy leaf nodes; building an ensemble of trees instead of only one; and/or using bootstrapped data samples in each tree (Breiman, 2001). Despite some limitations, their advantages make decision trees a suitable choice for many diverse data mining applications.

Iterative Dichotomiser 3 (ID3) (Quinlan, 1986) and C4.5, both developed by J. Ross Quinlan, are the two decision tree induction algorithms used in this thesis. **Iterative Dichotomiser 3 [ID3] Algorithm** (Quinlan, 1986) is the precursor of the C4.5 algorithm and generates a decision tree based on a dataset. It begins with a dataset containing all available attributes which is associated to a root node. On each iteration of the algorithm, it goes over every unused attribute of the dataset and selects the best attribute as measured by entropy or information gain. The attribute with the largest information gain or smallest entropy value is chosen. After selecting the best attribute on which to base the splitting of the data sample, the algorithm creates subsets of the data associated with each split value of the chosen attribute. The algorithm considers only attributes that have never been selected before and continues to iterate on each subset. The algorithm stops its recursion process on a subset when any of the following stopping conditions occurs: a) when all the records in a subset belong to the same class, then a leaf node is created and the same class label is assigned to the leaf node; b) when all of the attributes are used or no more attributes are available to be selected in that node, then a leaf node is created and the most common class of the examples in the subset is assigned as the label to

the leaf node; or (c) when there are no more records available in the subset (this situation appears when there is no record in the parent set that could be a match to a specific value of the selected attribute). When the algorithm reaches any of these stopping criteria, the specific node is turned to a leaf node and labeled with the most common class of the records from the parent set. The non-terminal nodes on the constructed decision trees represent the selected attribute on which the data were split. The terminal nodes on the trees symbolize the class label of the final subset of this specific branch of the tree.

The **C4.5 Algorithm** (Quinlan, 2014) is an extension of Quinlan's ID3 algorithm. The C4.5 algorithm, at each node of the tree, chooses the attribute that splits a data sample most efficiently into groups that are enriched in one class or the other. Normalized information gain (information entropy), Gini Index or some other measurement can be the splitting criteria of this algorithm. The C4.5 algorithm has several improvements over the ID3 algorithm, as it can handle both continuous and discrete attributes even with missing values. It is also used for the concept of pruning to remove unnecessary branches from the tree, replacing them with leaf nodes.

Figure 2.2 presents an example of a general decision tree for classification. The root and internal nodes of the tree are represented by circles in this figure and the leaf nodes are denoted by squares. When the decision tree is designed for classification purposes, as in this figure, the leaf nodes hold class labels.

The concepts of depth and breadth of the tree are the two other important definitions concerning decision trees. The average depth of the tree is the average number of levels from the root node to the terminal nodes of the tree, and the average breadth of the tree is the average number of internal nodes in each level of the tree. The higher

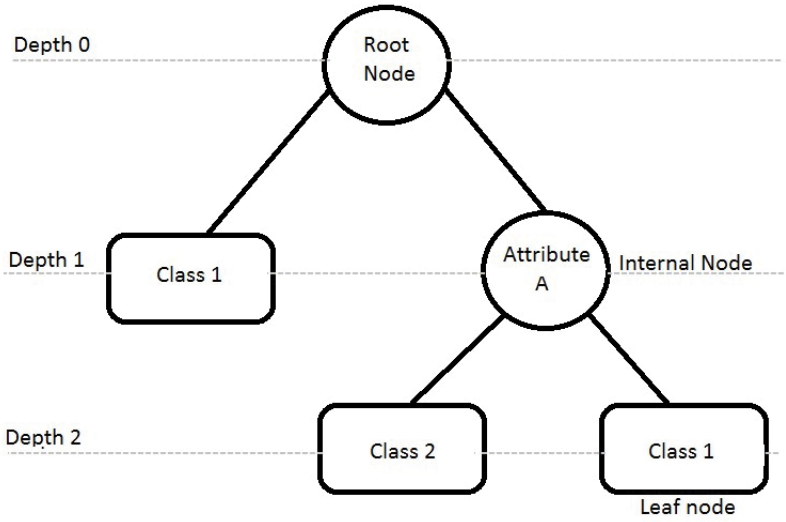

Figure 2.2: Example of a general decision tree for classification

the values of the depth and breadth of a tree, the more complex the decision tree so these two parameters are considered to be indicators of the complexity of the tree.

One set of data examples can shape many different decision trees. One difficult task is the induction of an *optimal* decision tree from a set of data (Barros, de Carvalho, & Freitas, 2015). Therefore, developing heuristics is necessary to solve the problem of inducing decision trees. In the last three decades, several developed approaches have the capability of providing reasonably accurate decision trees in a reduced amount of time and limited iterations. Within this basic tree-building procedure, two approaches can be taken that are fundamentally different: greedy approaches and random approaches. The differences between these two approaches are briefly discussed in the following subsection, however, a more thorough explanation of decision trees is given in Han et al. (2011).

2.4.1 Greedy Decision Trees

Of the tree induction approaches, the clear preference in the literature is the "top-down induction" algorithm for the tree (Barros et al., 2015). These are the algorithms that rely on a greedy style, recursive partitioning strategy for the growth of the tree. The pioneering work in the top-down induction of decision trees is Hunt's Concept Learning System (CLS) framework (Hunt, Marin, & Stone, 1966). Minimizing the cost of classifying a record is the goal of Hunt's CLS framework. In this context, cost refers to two different concepts: the computational cost for determining the value of a certain property or attribute displayed by a record, and the cost of misclassification of a record as belonging to class j when this record actually belongs to class k. At each stage of the algorithm, the CLS framework examines and calculates the space of possible decision trees on a fixed depth, chooses an action to minimize the costs in this limited space, and then moves one level down the tree.

The simplified algorithm of Hunt's work is the basis for all current top-down decision tree induction algorithms, although for some practical use cases, the assumptions of the framework are too tight. The algorithm has been improved in many ways; including relaxing the stopping criterion to avoid over-fitting an enormous decision tree or pruning a large tree. Another improvement to this algorithm is proposed to Hunt's original approach of partitioning the records with the splitting functions, which was performed by a cost-driven function. Consequent algorithms, including ID3 (Quinlan, 1986) and C4.5 (Quinlan, 2014), use information theory-based functions to partition nodes to smaller and purer subsets.

Rokach and Maimon (2005) present an algorithmic framework for decision tree top-down induction, which consists of three procedures: one for growing the tree,

one for pruning the tree and one for combining these two procedures, which is called inducing the tree. A basic strategy is to grow the tree to give each node some instances, and then prune the tree to remove unnecessary nodes which do not provide additional information for classification. Pruning is used to avoid over-fitting the trees (Mansour, 1997; Helmbold & Schapire, 1997), and it can be applied in two ways: either to stop growing the tree early before it perfectly classifies the training set (pre-pruning) or to allow the tree to perfectly classify the training set, and then post-prune the tree (post-pruning). Post-pruning is more successful in practice because it is not easy to clearly estimate how much a tree needs to be grown and when to stop growing the internal nodes of the tree (Kearns & Mansour, 1998). The tree pruning process of differentially private trees will be discussed later in Section 3.5.5.

The main issue to discuss for greedy-style trees is how to select the best combination of the attribute(s) and value(s) for splitting the nodes in the trees.

2.4.2 Selecting Splits

Choosing an attribute to split a node into smaller subsets is a major concern in the top-down induction of decision trees (Lior, 2014). Univariate decision trees (also known as axis-parallel) choose the attribute that better discriminates the input data. Based on the chosen attribute, a decision rule is generated and the input data is refined according to the outcome of this rule. The goal of multivariate decision trees is to find a combination of attributes with good discriminatory power in relation to the input data. Finding efficient approaches to rank attributes quantitatively is the concern of both these strategies (Rokach & Maimon, 2005).

Many different splitting functions have been proposed for decision trees in the

past, including information gain (Quinlan, 2014), the Gini Index score (Breiman et al., 1984), the misclassification error score (Breiman et al., 1984), and the Matsushita score (Kearns & Mansour, 1999). They all aim to maximize the discriminatory power of the child nodes in a way that is as unbiased as possible (Quinlan, 1996). The splitting criteria used in this thesis are as follows:

- **Information Gain** Information gain (Casey & Nagy, 1984; Hartmann, Varshney, Mehrotra, & Gerberich, 1982; Quinlan, 1986) belongs to the class of impurity-based criteria and is an example of a splitting function based on the basic Shannon's entropy (Shannon, Weaver, & Burks, 1951). The term *impurity* refers to the level of class separability that a split can cause between the data subsets. All the instances in a pure subset belong to the same class. The measurement values for impurity-based criteria are usually either 0 or 1. The purest subset is assigned a value of 0, which occurs when all the class values in a subset are the same. The least pure subset is assigned a value of 1, and this happens when the class values are equally distributed among all the instances in a subset.

  More formally, given a random target variable y with k discrete values in a training set of examples $\mathcal{S}$, distributed according to $P = (p_1, p_2, ..., p_k)$, an impurity-based measure is a function $f : [0, 1]^1 \rightarrow \mathbb{R}$ that satisfies the following properties (Rokach & Maimon, 2005):

  - $f(P) \geq 0$;
  - $f(P)$ is minimum if $\exists i$ such that $p_i = 1$;
  - $f(P)$ is maximum if $\forall i$, $1 \leq i \leq k$, $p_i = 1/k$;

- $f(P)$ is symmetric with respect to components of P;
- $f(P)$ is smooth (differentiable everywhere) in its range.

The entropy of a target attribute y in a training set of examples $\mathcal{S}$ can be defined as:

$$f^{entropy}(y, \mathcal{S}) = - \sum_{c_j \in dom(y)} P_{y=c_j, \mathcal{S}} \times \log_2 P_{y=c_j, \mathcal{S}}$$

Information gain is an impurity-based criterion that uses the entropy measure as the impurity measure. When each internal node is split with information gain criterion, the goal is to maximize the reduction in entropy. The information gain of a discrete attribute a_i shows how well the instance space $\mathcal{S}$ is split, based on the values of the attribute a_i:

$$\begin{aligned} f^{IG}(a_i, \mathcal{S}) &= f^{entropy}(y, \mathcal{S}) \\ &- \sum_{v_{i,j} \in dom(a_i)} P_{a_i = v_{i,j}, \mathcal{S}} \times f^{entropy}(y, \mathcal{S}_{a_i = v_{i,j}}) \end{aligned}$$

In this thesis, information gain is used to determine, in a given set of training examples, which attribute is most useful for discriminating between the classes that are learned in a differentially private manner. It quantifies the level of importance of a given attribute of the feature vectors and it is used to decide the ordering of the attributes in the nodes of a differentially private decision tree. Assigning an importance level to each of the attributes is equivalent to giving them an exponential mechanism score and calculating the probability of the highest score to guarantee a differentially private split.

- **Gini Index**

  The category of distance-based criteria for splitting a node evaluates separability, contrast or discrimination between classes. These criteria measure the distances between the distributions of class probability. A popular distance criterion, which also belongs to the class of impurity-based criteria, is the Gini Index (Breiman et al., 1984; Gelfand, Ravishankar, & Delp, 1989) and it is given by:

$$f^G(y, \mathcal{S}) = 1 - \sum_{c_j \in dom(y)} \left(P_{y=c_j, \mathcal{S}}\right)^2.$$

  The Gini Index score represents the probability of incorrectly labeling a sample when the label is randomly picked according to the distribution of class values for an attribute value.

- **Misclassification Error Score**

  Based on the re-substitution estimate described in Breiman et al. (1984), the error score corresponds to the node misclassification rate by picking the class with the highest frequency. This means that in a specific tree node, the error score chooses the attribute that minimizes the probability of misclassification, and it can be done by choosing the attribute that maximizes the total number of hits. This function is given by:

$$f^{Err}(y, \mathcal{S}) = \sum_{c_j \in dom(y)} \max(P_{y=c_j, \mathcal{S}}).$$

- **Matsushita Score**

  Kearns and Mansour (1999) introduced the Matsushita score as an optimized splitting criterion for decision trees, with this score being able to be linked to the exponential loss used by Adaptive Boosting (AdaBoost). Theoretically, this splitting measure can be used to significantly reduce the error of any learning algorithm that consistently generates classifiers in which the performance is better than in other classifiers and obtains an optimal bound for decision tree algorithms (Kearns & Mansour, 1999).

  This thesis analyzes and applies this important splitting criterion to differentially private decision trees and random forests. The Matsushita score splitting criterion is presented as follows:

$$f^{\mathrm{M}}(y, \mathcal{S}) \doteq 2\sqrt{P_{y,\mathcal{S}}\left(1 - P_{y,\mathcal{S}}\right)}$$

2.4.3 Random Decision Trees

The idea of using randomness to improve the performance of decision trees and completely remove greedy heuristics has been studied in numerous research studies (Fan, Wang, Yu, & Ma, 2003; Geurts, Ernst, & Wehenkel, 2006). In each internal node of these random decision trees, instead of using the splitting function to split the node, an attribute is randomly selected from the attribute list to split the node. Continuous attributes select the splitting point uniformly randomly from each attribute's range.

Although this approach performs very well when many random trees are used in combination, it works extremely poorly for a single tree. The computational cost of these random trees are much lower than an ensemble of greedy decision trees, because

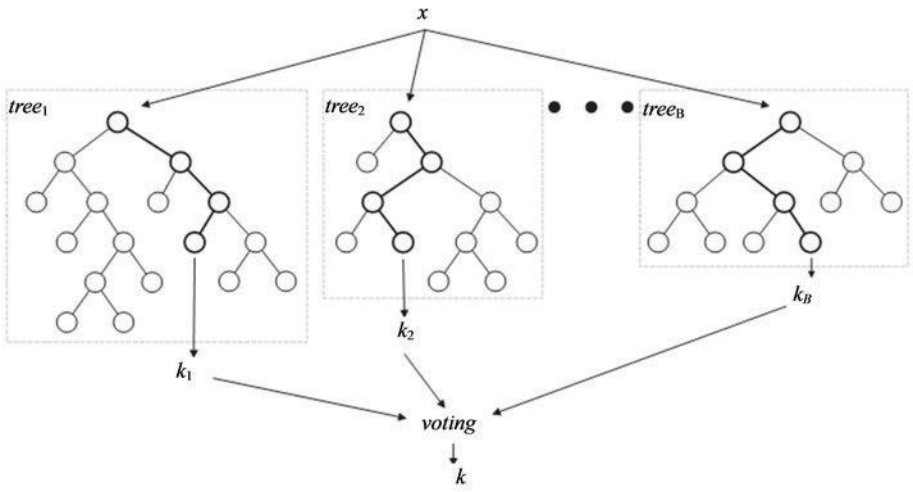

Figure 2.3: Example of a Random forest

in these trees, the calculation of the output of an objective function for every attribute in every node is avoided.

2.4.4 Random Forests

Random decision trees show their best behavior when they are involved in an ensemble to create a random forest. The random forest algorithm can be used both for classification and regression. The random forest (Breiman, 2001) is an ensemble of decision trees. Ensembles, with their divide-and-conquer approach, are used to improve the performance of classification and regression. The main principle behind ensemble methods is that a group of weak learners, which do not show accurate classification results, can come together to form a strong learner with highly accurate classification results. The random forest (see Figure 2.3) enhances this idea by combining trees, which in ensemble terms, are weak learners and the random forest is a strong learner.

This thesis also takes advantage of an ensemble of greedy-style differentially private trees to create a differentially private random forest classifier. Each tree in a random forest is built by considering a bootstrap sample set of the original training data. This process is first introduced as Bagging in Breiman (1996). This means the trees obtains a new sampling set with the replacement of instances from the original set, until all the original training data is used. Each random tree is created by using one of the bootstrap sample sets of the training data. Each random tree follows the procedure of traditional top-down induction when only a subset of attributes is allowed and favours the diversity of the ensemble. Each tree is built to its maximum depth, without applying any pruning procedure after the tree has been fully built. For a sample set, the predicted class is computed by aggregating the predictions of all of the individual decision trees with majority voting (Breiman et al., 1984).

2.4.5 Prediction Accuracy

The focus of this thesis is on classification using an individual decision tree with ID3 and C4.5 algorithms, and also a random forest from an ensemble of decision trees. After running the algorithm, the number of samples that are correctly assigned to the corresponding leaves determines the accuracy of the tree, which is calculated by $Acc_i = m_i/n$, where n is the total number of data nodes, and m_i is the number of data nodes that are correctly classified by the tree T_i. In the case of the forest, we present the average accuracy of all decision trees by $Acc = \frac{1}{|M|}\sum_{i=1}^{|M|} Acc_i$.

2.5 Differentially Private Decision Trees

When a decision tree generates output, privacy is leaked via the information that describes each node. Differentially private decision tree algorithms aim to prevent privacy leakage by spending a part of the privacy budget whenever private data is queried. The privacy budget needs to be spent when there are decisions and outputs directly based on the data. It is therefore important to identify exactly when the data is queried. The fewer times the data need to be queried, the less budget is spent, or the more budget will be available to spend per query. Sometimes, data does not necessarily need to be queried, but doing so will still be worth the privacy cost due to the better performance of the classifier arising from the query. The queries for which it should be considered compulsory will depend on whether the decision tree is built with a greedy or random approach.

If the decision tree is built greedily, this means that it uses a splitting criterion in each node to split the data, thus heuristically finding the best attribute to divide the local subset of data contained in a node. Regardless of which splitting function is used, the local data needs to be queried at least once. The other compulsory query for greedy trees involves counting leaf nodes to assign a class label. A decision tree that is built randomly does not use a splitting function in each node. Instead, attributes are picked randomly; thus, the privacy budget is not required for splitting the local nodes. The only query that the random tree needs is a query that enables the prediction of class labels for unseen data without any label. In each leaf node of the tree, this takes the form of querying the distribution of class counts for that node, with the majority class label being used as the predicted label for future records.

Like other differentially private models, the private data for differentially private trees are also accessed via interactive or non-interactive settings. In the non-interactive setting, the data owner releases an anonymized version of the data using a differentially private tree to the public and does not care how this data will be used later. However, in the interactive setting, the data itself is never publicized and it is the responsibility of the data owner to provide a reasonable access layer to the user. The data owner in the interactive setting will answer the user's queries with the results of generating a differentially private tree.

2.5.1 Differentially Private Decision Trees in the Non-Interactive Setting

Publishing a sanitized version of the data using differentially private decision trees is the aim of this group of studies. The DiffGen algorithm introduced by Mohammed et al. (2011) is the first anonymization algorithm for the non-interactive setting based on the generalization technique that preserves information for classification analysis. Their proposed solution first probabilistically generates a generalized contingency table and then adds noise to the counts. Thus, the count of each partition is typically much larger than the added noise. They mention, as a sample application, that the anonymized data can be effectively used to build a decision tree induction classifier. The DiffGen algorithm involves three key steps: selecting the candidate for specialization using the exponential mechanism, determining the split value and then publishing the noisy counts. Their algorithm shows competitively accurate results from their decision trees and therefore is used in this thesis to evaluate the algorithm in Section 3.6.2.

Zhu, Xiong, Xiang, and Zhou (2013) design another algorithm for the non-interactive data-sharing setting, named the DT-diff algorithm. They use the specialization scheme selection strategy to fully utilize the privacy budget to deal with both categorical and numerical data. The DT-Diff algorithm integrates the group-based method and differential privacy in a top-down manner. The algorithm generalizes the original dataset to a topmost general state; the general state is then specialized by narrowing it down to a more specific state through the use of a specialization which consumes the privacy budget. This process is repeated until the entire allocated budget is fully consumed. The final state of the dataset is then released for decision tree analysis. In this process, privacy is fixed by the privacy budget ϵ, and the utility is measured by the classification accuracy of the decision tree. Publishing a sanitized version of the data in hand is not practically feasible in all data mining scenarios, considering the complexity of the data, especially when it comes to synthetic datasets. The method that is applied in this thesis is to access data through an interactive setting for differentially private trees and the next section investigates the literature in this area.

2.5.2 Differentially Private Decision Trees in Interactive Setting

The number of previous research that are focused on the interactive settings of differential privacy on decision trees are more numerous than the number of studies on this subject on non-interactive settings. During the inception of differential privacy in 2005, a simple proof-of-concept decision tree algorithm was proposed by Blum et

al. (2005). They showed how a traditional non-private algorithm could be modified to achieve differential privacy by converting the concept of the splitting function to the queries that could be made differentially private. They focused on deciding which attribute should be used to split a node: the core value of a greedy decision tree algorithm which is to optimize the ability to discriminate between class labels. Specifically, by breaking the information gain into two counting queries for each attribute, they made this splitting criterion differentially private and then added the Laplace mechanism to make both of the two counting queries differentially private. Although this approach is unique and innovative, it is also costly in terms of privacy budget. If the total privacy budget of the user is ϵ, the above two queries for information gain criterion can only receive a small fraction of ϵ each. This process needs to be repeated for every attribute in the attribute schema, at every level of the tree. Considering m is the number of attributes and d is the tree depth, when this process is finished, the fraction of the total privacy budget ϵ that each query in the algorithm is alloated is equal to $\epsilon = \beta/2md$.

The first considerably strong work to discuss the application of differential privacy on decision trees is the work by Friedman and Schuster (2010). They designed a differentially private model in an interactive setting which considers the data miner as an untrusted entity that, through asking queries, can access the database using an interface. Their differentially private C4.5 algorithm, improves the previous SulQ (Blum et al., 2005) framework, and is a query interface exposed by the differentially private layer. By adding carefully calibrated noise to each query, the access layer enforces differential privacy. Fortunately, when building a greedy tree, converting a splitting function into differentially private counting queries (like SulQ framework) is

not the only approach to take. One of the major improvements made by Friedman and Schuster is, rather than submitting counting queries via the Laplace mechanism they use an exponential mechanism to submit a single query to find the best splitting attribute in each node. They proposed building a decision tree by querying the dataset two times at each node: the first time to obtain a count of the number of records in the node, and the second time to find the best attribute with which to split the records. Patil and Singh (2014) and Rana et al. (2015) also later used these two queries in their greedy tree algorithms. All the nodes, in any specific depth of the decision tree, contain disjoint subsets of the dataset and this allows them to use differentially private parallel composition theory on the disjoint subsets. Using composition theory, the privacy budget spend on the two parallel queries at each level of the tree can be summed. The portion of the privacy budget that each query submits to the dataset is equal to

$$\epsilon = \frac{\beta}{(2+n)d}. \tag{2.3}$$

The exponential mechanism chooses the output with a high probability score as the best attribute to split the records in a node. This mechanism calculates the probability for choosing any given output based on a scoring function. In the case of decision trees, this scoring function is the splitting function. The amount of noise which needs to be added to achieve differential privacy can be directly impacted by the sensitivity of these splitting functions. Therefore, each of several common splitting functions is analysed by Friedman and Schuster to gauge how sensitive it is to individual records. Friedman and Schuster found that the misclassification error score achieves the highest prediction accuracy in their experiments, despite it being the least sophisticated of the tested functions because of the low sensitivity value compared to other splitting

functions. This proves that when a function performs well in non-private scenarios, it does not necessarily continue to perform well in private scenarios when it is made differentially private. The Gini Index has a much lower level of sensitivity than the sensitivity of information gain, and it also achieved competitive results for the misclassification error score. They also tested the gain ratio splitting criterion, but due to its erratic behavior when the ratio's denominator approaches zero, they were unable to calculate its sensitivity. Quinlan's C4.5 algorithm (Quinlan, 1996) solves this problem with additional heuristics, but, ultimately, the gain ratio's sensitivity cannot be bounded, and thus it cannot be implemented with either the Laplace mechanism or the exponential mechanism. The main disadvantages of their models are in two areas: the privacy budget is allocated by assigning a fixed and equal portion of privacy budget to all the queries, and their trees come with a fixed height.

Although their study is one of the closest to the approach in this thesis, the main advantage of this thesis's approach over their work is the optimization in the allocation of the privacy budget. The algorithms in this thesis heuristically assign the best possible privacy budget to different queries of the differentially private decision trees, and thus avoid over-consuming the privacy budget. Creating trees that are not limited by a fixed height is also another advantage of this thesis over Friedman and Schuster's work.

2.6 Differentially Private Random Forests

The differentially private random forest, like the original random forest algorithm, forms each training set from a subset of chosen input features. The differentially private random forest algorithm is an ensemble method that manipulates its input

features and uses differentially private decision trees as base classifiers. In this section, the previous works on differentially private forests in the literature are reviewed to motivate the proposed differentially private random forest in Chapter 4 of this thesis.

Jagannathan et al. (2012) was the first to address the problem of constructing differentially private classifiers using random decision trees. Based on the original ID3 algorithm, they proposed the Private Random Decision Tree (PRDT) by adding noise using the Laplace distribution to each component of the leaf vectors. In their study, the structure of the tree is constructed without observing data for attribute selection at any internal node. So the splitting criteria at the internal nodes are decided entirely randomly, and because the record in the internal nodes of the tree is unseen, they cannot reveal anything about the data attributes. Only the class distribution at a leaf node is estimated from the data, so the noise needs to be added just to the leaf node to achieve differential privacy for the class labels. This allows more budget to be dedicated to the leaf node of the trees. Generating a differentially private ID3 tree using differentially private low-level queries in the first experiment in their study does not provide good privacy and good accuracy at the same time, especially for small datasets. Presenting the results of the application of their ID3 algorithm on realistic data demonstrates that the privacy parameter for each individual private sum query has to be adequately small enough to be able to achieve an acceptable privacy guarantee. Using random decision trees in another set of experiment in their study produces classifiers that have good prediction accuracy without compromising privacy, even for small datasets. Their random trees arbitrarily choose a splitting point from the attribute's domain with uniform probability. By using an experimental evaluation in Weka, these authors present the accuracy and privacy trade-off of their algorithm

on horizontally and vertically partitioned data and also when new datum arrives in an existing database. What these random trees sacrifice is a lot of the discriminatory ability created in each node by the splitting functions of greedy decision trees. This could cause problems especially when handling continuous attributes. Regardless of this fact, the overall prediction accuracy associated with an ensemble of random style trees is proved to be very similar or even better than the accuracy of an ensemble of greedy style trees. For this reason, and also due to the wide range of applications of this model in the literature as a strong benchmark, PRDT is a very good candidate to compare with the random forest algorithm proposed in this thesis. Section 4.6.3 of this thesis presents this comparison.

Continuing this line of research, the algorithm that was proposed by Bojarski, Choromanska, Choromanski, and LeCun (2014) adds noise, with a magnitude that is scaled up with the number of trees in the ensemble, to the leaf nodes of each tree. They proposed some probabilistic bounds theoretically on the prediction accuracy of a differentially private random forest, based on the number of decision trees in the ensemble and their depth. This probabilistic bound targets both the training error and the testing error. They proposed a logarithmic number of random decision trees in their algorithm. This number is relative to the number of training records that are needed to correctly classify most new test data. The number of trees in their study varies from as high as 21, or as low as one, and this is in line with the recommended number of 10 trees in the work of Jagannathan et al..

Using the same DiffP-ID3 algorithm as used by Friedman and Schuster (2010), Patil and Singh (2014) implemented random decision trees with a random number of trees and used majority voting for the resulting random forest. These authors were

the first to create a differentially private greedy style decision forest by expanding the single trees proposed by Friedman and Schuster to a forest. Their strategy to create the forest is similar to Breiman's (2001) random forest algorithm, and they used bootstrapped samples to train each differentially private tree. From a differential privacy point of view, sampling a dataset of size n with n times replacement (i.e. bootstrapping) is almost the same as using the exact same data in every tree in the ensemble, hence the privacy costs of the queries must be composed. Patil and Singh's classifiers suffer from a lack of sufficient accuracy and high variance in the results, mainly due to the small portion of the privacy budgets that is assigned to each query in the forest. Each tree in the forest will receive ϵ/T, where T is the number of trees in the ensemble, so the standard deviation of the noise will be T times bigger as well. Equally dividing the original budget into small chunks and using them to make each query of the tree differentially private is not an optimized approach to creating a differentially private random forest.

Bojarski et al. (2014) conducted an experimental and theoretical analysis and evaluation of a differentially private (and a non-differentially private) setting for random decision trees. Proposing two different algorithms for differentially private and non-differentially private random decision trees, they compared their results with majority voting, threshold averaging and probabilistic averaging algorithms. In particular, majority voting shows less sensitivity to random forest parameters such as the number of random trees and the height of the trees. They give an upper bound to the generalization error and theoretically explain how accuracy depends on the number of random trees. They consider that to correctly classify most of the data, where n is the size of the dataset, $O(\log(n))$ is the number of independently selected

random trees.

In another study by Rana et al. (2015), a weaker version of a differentially private random forest has been proposed and they show that the weak setting of differential privacy results in more accurate trees. Their work focuses on calculating how much privacy can be saved by using bootstrapped data. Since only approximations have been made, it is not the same as querying all the original number of records and the guarantee of differential privacy is not as strong. They develop worst-case scenarios for adversary models to infer about unknown data in attributes and class labels when all other aspects of the data are known to the adversary. Under these adversary models and because bootstrap sampling offers extra randomness which enables "privacy for free", they derive differentially private upper bounds on the maximum number of trees in the ensemble. They explore one potential way of finding a splitting point for continuous attributes, which requires weakening the definition of differential privacy by discretizing the attributes first. In this method, rather than spending a large amount of ϵ for a given continuous attribute, similar to the work of Friedman and Schuster, to find the optimal splitting value, they discover the average value for all the records in a node that have the class label c_1, and do the same for all the records that have the class label c_2. Halfway between these two average values is defined as the chosen splitting value. This approach increases the budget per query, although it does not maximize the discriminatory power of the child nodes. The promise of this model is to reduce the tree noise by increasing the budget per query from Friedman and Schuster's budget, which is $\epsilon = \beta/(2 + N)d$ (where N is the number of continuous attributes, and d is the tree depth), to a new division of the budget which is:

$$\epsilon = \frac{\beta}{3d} \tag{2.4}$$

To make this query weakly differentially private, they add noise using the Laplace distribution $Lap(\Delta/\epsilon)$, with the sensitivity Δ based on the smaller of the two class counts, $\Delta = 3/\min(n_{c_1}, n_{c_2})$. The continuous attributes in their work are assumed to have a normal distribution, and instead of using a smooth sensitivity or a strictly-correct global sensitivity, they estimate their sensitivity within three standard deviations, which is adequate in 99.7% of cases. Although their algorithm is not strictly differentially private, the authors instead heuristically added an amount of noise that can be considered in most scenarios "adequately private" and proposed a weakened definition of differential privacy. The assumption of this study contradicts one of the most important benefits of decision trees, which is the non-parametric independence of decision trees to their underlying data distribution.

Fletcher and Islam conducted a series of comprehensive studies on decision trees and random forests and later prepared a survey on the available models and applications for this topic (Fletcher & Islam, 2016). In this series of studies (Fletcher & Islam, 2015a, 2017, 2015b), their focus on the creation of differentially private decision trees and random forests shifted in slightly different directions. Their first work (Fletcher & Islam, 2015a) examines an application of the greedy random forest, in line with Friedman and Schuster's (2010) greedy-style trees, and like Patil and Singh's work (2014), the authors greedily generate a forest with some small changes to the details. In an attempt to reduce the noise that is induced in the tree-building process, Fletcher and Islam (2015a) explore reducing the sensitivity of the splitting functions. Instead of using the theoretical worst-case scenario to calculate a function's global sensitivity, they take advantage of the first query by Friedman and Schuster and calculate the local sensitivity of the Gini Index in the node being split. Friedman and

Schuster's first query returns the support of the node, and with this information it can be used by the user to offer a better-bounded sensitivity. Fletcher and Islam prove that the sensitivity of the Gini Index could be reduced. Reductions in sensitivity directly reduce the noise induced by mechanisms like the Laplace mechanism and the exponential mechanism. Furthermore, they return noisy counts in each leaf node of the trees instead of getting the distribution of class values. Unlike the other differentially private forests in Patil and Singh (2014) and Jagannathan et al. (2012), they reduce the number of trees from ten to four in their experiments; each one is given a different root attribute. A smaller number of trees have more privacy budget to consume on each query and result in less noisy trees, but the effect of using only four trees in the forest needs to be further investigated. In addition, to be fully differentially private, the sensitivity reduction in the Gini index proposed by Fletcher and Islam (2015a) needs to be smoothed out to account for all neighboring datasets. If the proposed sensitivity is too small, it will be rejected by the system as it would not properly provide differential privacy.

Smooth sensitivity is used in the random decision tree algorithm proposed by Fletcher and Islam (2017). Using smooth sensitivity in Nissim et al.'s (2007) framework allows all four splitting functions to add less noise than was added by Friedman and Schuster. Unlike Breimans proposed forest, which is based on bootstrap sampling with replacement and uses all data for each tree in an ensemble, Fletcher and Islam propose using sampling without replacement in each tree. This design allows the algorithm to use differential privacy's parallel composition theory on counting queries. Smooth sensitivity cannot help with counting queries: no matter what the specific data are, the output of a counting query can always change by one when one

record is added or removed. To output the majority (i.e., most frequent) label in a leaf node with the exponential mechanism, a scoring function is required. To achieve this, Fletcher and Islam (2017) propose a piecewise linear function. Each label has a score of one if the label is the most frequently occurring label in the leaf and zero otherwise. This function's global sensitivity is one, as adding or removing one record could theoretically change a score from one to zero or vice versa. By taking advantage of smooth sensitivity, Fletcher and Islam (2017) improve the accuracy of a random decision forest. They find that this setting allows their algorithm to build a forest from a much more significant number of trees; in their experiments, they used one hundred trees.

In another study, Fletcher and Islam (2015b) propose their framework to dynamically define the maximum depth of a tree, with different depths possible in different parts of the same tree. By rephrasing the problem in terms of the support signal and the noise added by the Laplace distribution, the authors define a minimum support threshold that ensures the signal-to-noise ratio (Van Drongelen, 2011) is greater than one:

$$n' = \frac{\sqrt{2}|c|}{\epsilon} \tag{2.5}$$

If the estimated support of a node is below n', the node is not further split. This prevents the tree from growing to a point where the noise added to each class count could become larger than the count itself. In addition, they allow each branch of the tree to continue growing (i.e. increase in depth) until the class counts are at risk of being overwhelmed by the noise added to them by the Laplace distribution. They estimate the support of each node similar to how Jagannathan et al. (2012) decide the maximum tree depth. The only difference is Fletcher and Islam's model can also

consider that each part of the tree can have a different number of branches. Both Jagannathan et al. (2012) and Fletcher and Islam (2015b) assume that the data is approximately uniformly divided among all branches. Based on this assumption, if the records are evenly distributed, the tree-building procedure terminates before the amount of noise overrides the class counts. By taking a heuristic approach, Fletcher and Islam (2015b) choose the number of random trees based on two factors: the number of attributes and the minimally acceptable tree size, defined as a tree with a depth of three levels. The authors develop a heuristic whereby, based on Equation 2.5, the budget assigned to each tree $\epsilon = \beta/T$ would not be so small that their minimum support threshold n' would prevent a tree from splitting at least twice. This means that the number of trees have an upper bound, so that the privacy budget can be large enough to allow for minimally acceptable tree sizes. The upper bound on the number of trees in this model is the number of attributes available in the attribute domain. In addition, they maximize tree diversity by requiring the root node of each tree to use a unique attribute with which to be split.

2.7 Research Gaps

Decision trees and random forests are the two data mining algorithms that are applied through a differential privacy setting in this thesis. This chapter reviews the current research based on these two algorithms. This thesis proposes new algorithms to address the research gaps which are identified in this chapter and can be summarized as follows.

Current differentially private decision trees in the literature do not consider a privacy budget allocation scheme on decision trees. So far, all the algorithms in

the literature assume that all queries of a differentially private tree use the same proportion of the privacy budget, and this means that all queries have an equal level of importance to the success of the algorithm, or need the same level of accuracy to be useful. The idea in this thesis is to spend uneven quantities of the privacy budget on different queries from the tree. In Chapter 3 of this thesis, the privacy budget is consumed unequally in different nodes of the differentially private tree, based on the quality and the characteristics of the data in hand.

The choice of splitting criteria on differentially private decision trees can have a significant effect on the accuracy of the results of the trees. In Chapter 3 of this thesis, a new splitting criterion which has previously never received any treatment in differential privacy is introduced and its performance is compared to other splitting functions.

The application of the budget allocation schema will be examined on an ensemble of decision trees in Chapter 4 of this thesis. The differentially private random forest algorithm in this study considers the enhancement of using an adaptive privacy budget allocation schema which has not been previously examined in the literature.

Chapter 3

Differentially Private Decision Tree

3.1 Introduction

Differential privacy has received significant attention in the machine learning community and it gives machine learning researchers a great opportunity to explore different data mining methods and algorithms in differentially private settings. In differential privacy, the privacy budget is utilized wherever private data is accessed. The budget consumption goes through the details of all the differentially private algorithms. The algorithmic foundation of decision tree induction makes them an ideal candidate for a differentially private setting.

In this chapter, the application of differential privacy on a single decision tree is studied and aligned with the contributions of the thesis and an adaptive budget allocation algorithm is introduced and evaluated on a single decision tree. This chapter of the thesis addresses Contributions 1 and 3 of this thesis, namely developing an algorithm for adaptive privacy budget allocation on a differentially private decision tree and the proposal of Matsushita loss as a splitting criterion on differentially private decision trees.

3.2 Motivations

The privacy constraints for differential privacy complicate the task of greedy tree induction: inducing each split in the decision tree incurs a privacy cost, and optimizing the allocation of the privacy budget is expensive from computational and privacy standpoints. Greedy tree induction also faces the risk of wasting privacy down the tree induction, and potentially resulting in trees that generalize poorly (Friedman & Schuster, 2010; Jagannathan et al., 2012). A heuristic proposed in Friedman and Schuster (2010) significantly alleviates wasting the privacy budget on split choices and relies on another differentially private mechanism, the *exponential mechanism* (McSherry & Talwar, 2007). Instead of selecting the top attribute according to the noisy split values as would be achieved with noisy counts (Blum et al., 2005), the *exponential mechanism* probabilistically selects an attribute based on the attributes' *true* split value. This work is the starting point of this thesis in relation to privacy.

The choice of a splitting criterion has a very significant impact on the result of a top-down induction algorithm, especially when privacy aspects are involved. The *curvature* of the splitting function (entropy function) is the key to provably unbeatable guarantees on the rate of convergence in decision tree induction (Kearns & Mansour, 1999; Nock & Nielsen, 2004). The more "concave" the entropy, the better the minimal guarantee on the rate of convergence of a tree, which is crucial for good generalization (Kearns & Mansour, 1998).[1] For example, IG guarantees better convergence rates than Gini (Kearns & Mansour, 1999; Nock & Nielsen, 2009, 2004). These works are the starting point for this thesis in relation to inducing trees.

[1]This does not contradict the no-free lunch (Nock & Nielsen, 2004, 2008; Reid & Williamson, 2010): boosting provides *lower bounds* on convergence rates.

When combining the boosting results (Kearns & Mansour, 1999) along with Friedman and Schuster's (2010) results, and when adopting the differential privacy concept of a *privacy budget*, it turns out that the choice of splitting criteria results in different accuracy levels from the trees. For example, Information Gain comparatively reduces the *number* of required decision tree splits, and therefore the number of interactive differentially private queries from the trees (Friedman & Schuster, 2010). This introduces a trade-off between learning and privacy to obtain accurate classifiers within the constraint of the overall privacy budget.

So far, to the best of our knowledge, this observation has never been formalized in the privacy vs learning context in the literature. Even in the most recent works (Fletcher & Islam, 2016), the discussion revolves around privacy and risk only. One needs to drill down into the *convergence* rate for greedy algorithms, precisely because convergence affects both generalization *and* the spending of the privacy budget. This, however, necessitates generalization of the observation of Friedman and Schuster from specific losses (*e.g.*, Gini vs. IG) to all loss functions, since many splitting criteria can be used (Devroye, Györfi, & Lugosi, 2013; Kearns & Mansour, 1999; Nock & Nielsen, 2008).

In this chapter, a formal answer is provided to this question. It is shown that the observation of Friedman and Schuster can be generalized:

The better an entropy is from the convergence standpoint, the more "expensive" it is from the privacy standpoint.

To do so, a new connection between the sensitivity of a loss function – a fundamental quantity for differential privacy – and the *perspective transform* of its entropy is made in this chapter. Perspectives are well-known objects in convex analysis

(Maréchal, 2005a, 2005b). This connection happens to be related to the curvature of the loss and therefore the convergence rates of one of these greedy minimization. The connection between the sensitivity of a loss function and its rate of convergence applies to the broad set of proper symmetric losses, without restriction. Being proper and symmetric are two basic properties of a loss function: the former states that the Bayes rule realizes the optimum of the loss, the latter that there are no class dependent misclassification costs (Nock & Nielsen, 2009; Reid & Williamson, 2010). This concept in this thesis is demonstrated on Matsushita's loss (Kearns & Mansour, 1999; Nock & Nielsen, 2009), an entropy function that is known to be optimal from the boosting standpoint, and for the first time receives treatment in differential privacy in this thesis.

Although optimizing the connection between the sensitivity of the loss functions and their convergence rate is one way to increase the accuracy of differentially private mechanisms, this research seeks more reliable methods to improve the spending of the privacy budget throughout the induction of the decision tree, such as tuning the allocation of the privacy budget across the interactive queries. Optimizing this allocation sends a strong message: significant improvements of the differentially private greedy induction of decision trees will probably require a very careful tuning of the allocation of the privacy budget across interactive queries throughout the induction of the decision tree.

The prior work allocated a fixed privacy budget per query for a decision tree, whereas this thesis proposes a novel approach to budget allocation, which determines the privacy parameter per query *dynamically*. This approach has several advantages over the state of the art: it allows flexibility in the number of queries (*e.g.*, with respect

to the depth of the induced trees), it avoids spending excessive amounts of the privacy budget to choose between attributes that have similar loss, and it allocates the surplus budget to noise-sensitive queries. Finally, it reduces the variance in query responses allowing algorithms to have consistent accuracy results despite the introduced noise.

3.3 Definitions and Basic Results

This section presents the basic concepts that are used throughout this research on learning and differential privacy, mostly based on differential privacy definitions in the prior work. The learning setting is batch supervised: X is random *observations* with associated *class* Y, sampled from a joint distribution $\mathcal{D}$ over (X, Y). Each $(x, y) \sim \mathcal{D}$ is an *example*. $\mathcal{S}$ denotes a training set of examples sampled from $\mathcal{D}$, from which a classifier $h : \mathcal{X} \to \mathcal{Y}$ is learned, where $\mathcal{X}$ and $\mathcal{Y}$ are respectively the supports of X and Y.

Without loss of generality, the number of classes are assumed to be $C = 2$, that is, $\mathcal{Y} = \{0, 1\}$ (with respect to the negative and positive classes). Classifier h belongs to a set $\mathcal{T}$, which is the set of decision trees in this research, *i.e.*, recursive domain partitioning classifiers. Learning refers to the building of some $h \in \mathcal{T}$ by minimizing a suitable *loss* function over $\mathcal{S}$ (Nock & Nielsen, 2009; Reid & Williamson, 2010).

A loss function denotes a function $l : \{0, 1\} \times [0, 1] \to \mathbb{R}^+$. Its left argument is an observed class, y, and its right argument is a local class-probability estimate $\hat{p}[y|x; h]$, typically built from a classifier h. Here it is implicitly assumed that $l(y, \hat{p})$ is lower-bounded. The expectation of l is computed over $\mathcal{S}$ and the *empirical loss* of

h is obtained:

$$L(h, \mathcal{S}) \doteq \mathbb{E}_{(x,y)\sim\mathcal{S}}[l(y, \hat{p}[y|x; h])] \ , \tag{3.1}$$

or the expectation over $\mathcal{S}$ is performed and the *true loss* of h is obtained. What choices of L are *suited* to learning? Two natural requirements of l are imposed (Nock & Nielsen, 2009; Reid & Williamson, 2010): (i) it is *proper*, *i.e.*, Bayes rule realizes the optimal true risk; (ii) it is *symmetric*, *i.e.*, there is no class-dependent misclassification cost. Remarkably, these suffice to give L a very precise shape.

Theorem 3. *For any symmetric proper loss, there exists a* ***permissible*** *function* ϕ *such that* $L(h, \mathcal{S}) = L_\phi(h, \mathcal{S})$ *with:*

$$L_\phi(h, \mathcal{S}) \doteq \mathbb{E}_{(x,y)\sim\mathcal{S}}[\phi(\hat{p}[y|x; h])], \forall h, \mathcal{S}. \tag{3.2}$$

ϕ *is permissible iff* $\mathrm{dom}\phi = \mathrm{Im}\phi = [0, 1]$, ϕ *is concave, symmetric wrt.* $x = 1/2$ *and* $\phi(0) = \phi(1) = 0, \phi(1/2) = 1$.[2]

The proof follows from Nock and Nielsen (Lemma 1, Theorem 3) (2009), Reid and Williamson (Corollary 5) (2010) and Abernethy and Frongillo (2012) (to handle non-differentiable ϕ). To simplify the definitions, a slight abuse of language in the statement of Theorem 3 is made: the loss in eq. (3.2) is in fact the *regret* of h, that is, the difference to Bayes loss (Reid & Williamson, 2010). One loss, the *0/1 loss*, gives rise to the empirical and true risks (or "*errors*"). It satisfies: $\phi(x) = \phi_{\mathrm{Err}}(x) \doteq 2\min\{x, 1 - x\}$. Thus, for any permissible ϕ, $\phi(x) \geq \phi_{\mathrm{Err}}(x)$ for all x, since ϕ is concave; hence, any suitable procedure to minimize $L_\phi(h, \mathcal{S})$ is a proxy to minimizing the empirical risk as well. The importance of Theorem 3 needs to be emphasized

[2] *This naturally extends to* $C > 2$ *classes in the multi-class multi-label setting with the Hamming loss (Schapire & Singer, 1999), where the expectation also covers classes.*

$\phi(x)$	Δ^*_ϕ
$\phi_{\mathrm{M}}(x) \doteq 2\sqrt{x(1-x)}$	$2\sqrt{m}$
$\phi_{\mathrm{IG}}(x) \doteq -x\log x - (1-x)\log(1-x)$	$\log(m+1) + \frac{1}{\ln 2}$
$\phi_{\mathrm{G}}(x) \doteq 4x(1-x)$	$4m/(m+1)$
$\phi_{\mathrm{Err}}(x) \doteq 2\min\{x, 1-x\}$	2

Table 3.1: Popular permissible ϕs and Δ^*_ϕ (Lemma 5). M = Matsushita, IG = Information Gain, G = Gini, Err = Error.

here: *any* suitable loss function has to have the form of L_ϕ, for *some* ϕ that is called an **entropy**.

Because they provide direct estimates $\hat{p}[y|x;h]$ at their leaves, decision trees are a natural fit for the optimization of any criterion of the type of L_ϕ, even in the Hamming loss setting ($C \geq 2$). It turns out that the state of the art has largely adopted, over the last thirty years, the greedy induction of a large tree by the minimization of *some* L_ϕ, in general followed by a pruning of the tree to control the true risk (Breiman et al., 1984; Quinlan, 2014; Friedman & Schuster, 2010). Table 3.1 gives examples of ϕ and is used as the reference for the loss functions and their sensitivities throughout this thesis.

Whereas the principled design of L in Theorem 3 barely gives any incentive to pick a specific ϕ against another, *boosting* provides a precise incentive (Kearns & Mansour, 1999). Boosting is a field of learning that has been very successful in showing that if one has access to a weak learner that provides poor but non-random classifiers, then sophisticated combination mechanisms exist to craft classifiers that are arbitrarily accurate. Decision trees can be viewed as such combinations where weak classifiers are the splits in the tree (Kearns & Mansour, 1999): in this case, the choice of ϕ appears to be key to rapidly reach high accuracy (1 minus the error), which is critical for good generalization. Based on Kearns and Mansour (1999), from the boosting

convergence standpoint Matsushita's loss is optimal ϕ. (ϕ_{M}).

This simple picture, which is to some extent confirmed by experiments in Dietterich, Kearns, and Mansour (1996), becomes more complex as privacy comes into play (Friedman & Schuster, 2010). *Differential privacy* essentially relies on randomized mechanisms to guarantee that *neighbor* inputs to an algorithm should not change too much its distribution of outputs (C. Dwork et al., 2006). In this context, the input to a learning algorithm $M(.)$ is a training sample (omitting additional inputs for simplicity) and two samples $\mathcal{S}$ and $\mathcal{S}'$ are neighbors, noted $\mathcal{S} \approx \mathcal{S}'$ iff they differ by at most one example. The output of $M(.)$ is a classifier $h \in \mathcal{T}$.

Definition 4. *(from (C. Dwork, 2011a)) Fix $\epsilon, \delta \geq 0$. A learning algorithm $M(.)$ is (ϵ, δ)-differentially-private if for all neighbor samples $\mathcal{S}$ and $\mathcal{S}'$ and for all $h \in \mathcal{T}$:*

$$p[M(\mathcal{S}) = h] \leq \exp(\epsilon) \cdot p[M(\mathcal{S}') = h] + \delta. \tag{3.3}$$

where the probabilities are taken over the coin flips of $M(.)$.

A simple example for obtaining the probability over the coin flips is by throwing a coin, if head shows answer honestly, but if tail shows throw the coin again and answer "Yes" if head shows, and "No" if tail shows.

The algorithm M is ϵ-differentially private if it is $(\epsilon, 0)$-differentially private. The smaller ϵ and δ, the more private the algorithm. The value of ϵ is typically smaller than one (C. Dwork, 2011a, 2011b).

Privacy comes with a price which is, in general, the noisification of M. A fundamental quantity that allows calibration of noise to the privacy parameters relies on the *global sensitivity* of a function $f(.)$, defined on the same inputs as M, which is just the maximal possible difference of f among two neighbor inputs. Assuming

$\text{Im} f \subseteq \mathbb{R}^d$, the global sensitivity of $f(.)$, Δ_f, is (C. Dwork et al., 2006):

$$\Delta_f \doteq \max_{\mathcal{S} \approx \mathcal{S}'} \|f(\mathcal{S}) - f(\mathcal{S}')\|_1 \ . \tag{3.4}$$

Differential privacy offers two standard mechanisms. The first one is the Laplace mechanism (C. Dwork et al., 2006), which adds $\text{Lap}(b)$ noise to the input, with a scale parameter $b \doteq \Delta_f / \epsilon$, and provides ϵ-differential privacy. The second one is the exponential mechanism (McSherry & Talwar, 2007). In the case where sample $\mathcal{S}$ offers several outputs $\{r : r \in \mathcal{R}\}$ for a function $f : \mathcal{R} \to \mathbb{R}$ which is a score (the higher, the better), the exponential mechanism outputs $r \in \mathcal{R}$ with probability $\propto \exp(\epsilon f(r)/(2\Delta_f))$, thereby favoring outputs with higher *true* scores while maintaining differential privacy. Finally, the *composition theorem* is used in this algorithm, which states that the sequential application of ϵ_i-differentially private mechanisms F_i ($i = 1, 2, ...$) provides $(\sum_i \epsilon_i)$-differential privacy (C. Dwork et al., 2006).

3.4 A Privacy / Boosting Trade-off

This section clarifies the connections between privacy and boosting in this thesis. The definitions and theories in this section of the thesis are based on joint work with Richard Nock from Data61.

Let's consider $C = 2$ is the number of classes here (all results easily generalize to multi-class multi-label settings). Denote $\mathcal{L}(h)$ as the set of leaves of tree $h \in \mathcal{T}$, m_ℓ the number of examples that fall into leaf ℓ ($\sum_\ell m_\ell = |\mathcal{S}| \doteq m$) and m_ℓ^+ the number of positive examples reaching ℓ. The empirical loss $L(h, \mathcal{S})$ can then be reduced to:

$$L(h, \mathcal{S}) \doteq \frac{1}{m} \sum_{\ell \in \mathcal{L}(h)} f_\phi(h, \ell, \mathcal{S}) \ , \tag{3.5}$$

with $f_\phi(h, \ell, \mathcal{S}) = m_\ell \cdot \phi(m_\ell^+/m_\ell)$. The *Global sensitivity* of $f_\phi(h, \ell, \mathcal{S})$ is

$$\Delta_\phi \doteq \sup_{\mathcal{S}' \approx \mathcal{S}} |f_\phi(h, \ell, \mathcal{S}') - f_\phi(h, \ell, \mathcal{S})| \ . \tag{3.6}$$

Suppose that the exponential mechanism is run over the splitting choices of a current leaf ℓ when inducing a tree h, to make the split choice differentially private. As shown in Freidman and Schuster (2010, Appendix A), if the candidate attributes for the split are scored by the decrease in $L(h, \mathcal{S})$, then the sensitivity used to calibrate the exponential mechanism is the one defined in eq. (3.6). Now *perspectives* in Definition 5 is introduced as a set of functions originating in convex analysis.

Definition 5. *(Maréchal, 2005a, 2005b) Given closed convex function f, the **perspective** of f, noted $\check{f}(x, y)$ is:*

$$\check{f}(x, y) \doteq \begin{cases} y \cdot f(x/y) & \text{if } y > 0 \\ f0^+(x) & \text{if } y = 0 \\ +\infty & \text{if } y < 0 \end{cases} \ , \tag{3.7}$$

where $f0^+$ is the recession function of f.

The standard textbook definition from Rockafellar (1970) are applied here. For any permissible ϕ, $-\phi$ is closed. The original requirement also includes an additional constraint, properness (not in the statistical sense of Theorem 3), which would easily be satisfied in this case (Rockafellar, 1970, pp 24). Therefore, $f_\phi(h, \ell, \mathcal{S}) = -\check{f}(m_\ell^+, m_\ell)$ with $f \doteq -\phi$. Whenever $y = 0$, $x = 0$ too ($0 \leq m_\ell^+ \leq m_\ell$), so that the only value of $f0^+$ of interest is $f0^+(0) = 0$. To alleviate notations, let's say $f_\phi(h, \ell, \mathcal{S}) = \check{\phi}(m_\ell^+, m_\ell)$, extending the perspective notation to concave functions with the convention $\check{\phi}(0, 0) = 0$. The sensitivity of f_ϕ can be completely defined from the perspective of ϕ via a quantity that is defined below.

Definition 6. *Let* $\Delta^*_\phi(m) : \mathbb{N}_* \to \mathbb{R}$ *be any function such that* $\check{\phi}(1, m+1) \leq \Delta^*_\phi(m), \forall m \in \mathbb{N}_*$.

Δ^*_ϕ (m is dropped for clarity) defines intuitively a *simple* (implementable) upper-bound on $\check{\phi}(.,.)$. The importance of Δ^*_ϕ stems from the following Theorem:

Theorem 4. *For any permissible* ϕ, $\Delta_\phi \leq \check{\phi}(1, m+1)$. *Furthermore, for any permissible* ϕ, *the bound is tight, that is, there exists* $\mathcal{S}$ *such that the inequality is an equality.*

A sketch of proof is provided in Appendix 6.1.

Below are examples for ϕ in Table 3.1 (proof in Appendix 6.2).

Lemma 5. *For the choices* $\phi \in \{\phi_{\mathrm{M}}, \phi_{\mathrm{IG}}, \phi_{\mathrm{G}}, \phi_{\mathrm{Err}}\}$, *there is a fixed* $\Delta^*_{\phi_{\mathrm{M}}} = 2\sqrt{m}$, $\Delta^*_{\phi_{\mathrm{IG}}} = \log(m+1) + \ln^{-1} 2$, $\Delta^*_{\phi_{\mathrm{G}}} = 4m/(m+1)$, $\Delta^*_{\phi_{\mathrm{Err}}} = 2$ *(*log *base-2,* ln *base-e).*

The following remarks are in place. First, the results of Friedman and Schuster are re-used for the particular choices of $\phi = \phi_{\mathrm{IG}}$ and $\phi = \phi_{\mathrm{Err}}$, if one drops the normalization constant for ϕ to be permissible. Notice that normalization does not impede privacy, as the exponential mechanism's parameters would not change if ϕ is normalized by any fixed constant. Second, the choice $\phi = \phi_{\mathrm{G}}$ from Friedman and Schuster is slightly improved. Third, there is no record of a previous treatment of $\phi = \phi_{\mathrm{M}}$ in a differentially private framework. Finally, the following Lemma shows the conflict between privacy and boosting, since the curvature of ϕ (essential for boosting, (Kearns & Mansour, 1999, Figure 3)) directly correlates with how sensitivity varies with m.

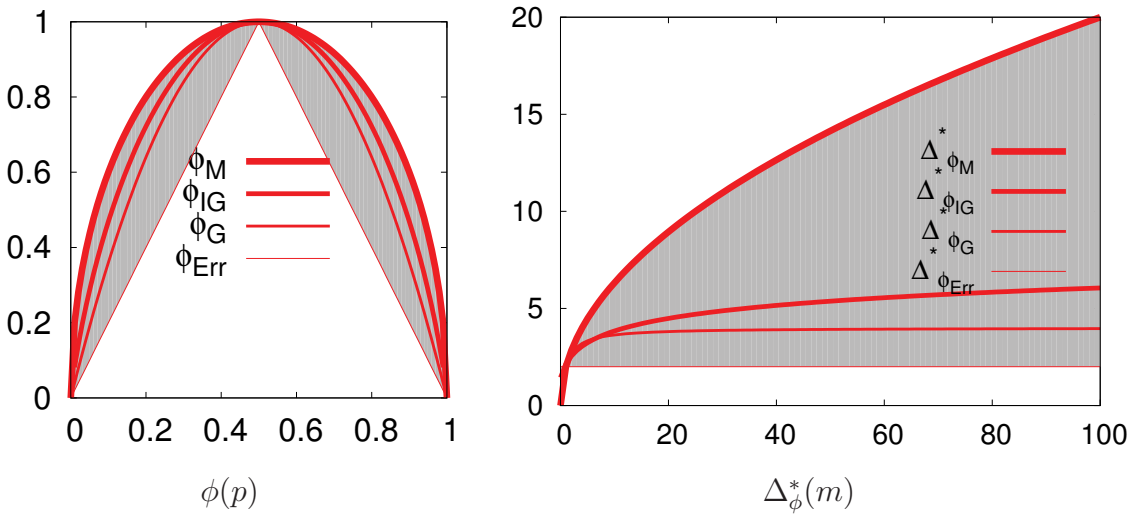

Figure 3.1: Left: plots of the four choices of permissible ϕ that is considered (Table 3.1). Right: corresponding bound on the sensitivity, Δ^*_ϕ, following Lemma 5. The shaded areas indicate where curves of proper symmetric losses within the lo-hi convergence rate bounds would typically be located.

Lemma 6. *For any twice differentiable ϕ* [3]*, for any m, there exists $a \in [0, 1/m]$ such that*

$$(\check{\phi}')(1, m+1) \;=\; (m+1)^{-2} \cdot (-\phi'')(a) \;.$$

The proof of this lemma is in Appendix 6.3.

This is illustrated in Figure 3.1, which summarizes the bounds that are used in Table 3.1 for Δ_ϕ, and clearly shows a big gap between $\Delta^*_{\phi_M}$ and the other curves.

3.5 An Adaptive Budget Allocation Algorithm

This section describes the adaptive budget allocation algorithm for differentially private decision trees and covers contribution 1 of this thesis (This section onward are

[3] *When a function is twice differentiable, it means that its second derivative is defined for all input values within its range.*

not the result of the collaborative work anymore, and are the work of the thesis's authors only). In this section, the algorithm's preliminaries are described first, and then the ADiffP-ϕ algorithm in Algorithm 1 is presented in detail.

3.5.1 Preliminaries

Consider $\mathcal{S}$ is a given set of training examples ($|\mathcal{S}| = m$). Each sample (x, y) pertains to an individual. Denote $d \doteq \dim(\mathcal{X})$ as the dimension of the observations' space. $\mathcal{X}_i \in \mathcal{X}$ denotes an attribute, and $|\mathcal{X}_i|$ its cardinality (*e.g.*, two if it is binary). It is assumed that the parameters d, m, $|\mathcal{X}_i|$ ($\mathcal{X}_i \in \mathcal{X}$) are public and do not need to be protected.

The ID3 algorithm (Quinlan, 1986) applies a greedy hill-climbing approach to constructing a decision tree. ID3 starts from the root node that keeps the complete training set, and according to the given samples in $\mathcal{S}$, it chooses the attribute that could maximize information gain, and splits the current node into new nodes. The training set can now be divided among the new nodes according to the value each sample can take on the chosen attribute. The algorithm is then be applied recursively on all the new child nodes.

The algorithm presented in Friedman and Schuster (2010), DIFFP-ID3, modifies ID3 to obtain a differentially private decision tree as follows:

(i) when nodes are split, attributes are selected using the exponential mechanism;

(ii) in the leaves, the class distribution is evaluated using the Laplace mechanism (noisy counts);

(iii) the depth of the tree is set in advance, so that a privacy budget can be allocated

in advance to each of the differentially private queries;

(iv) a differentially-private stopping condition halts the induction process if the (noisy) count of examples in a node drops below a predetermined threshold.

The stopping condition is intended to mitigate the induction of leaves where the number of examples is too small compared to the magnitude of noise added in the noisy counts.

3.5.2 The ADiffP-ϕ Algorithm

The ADiffP-ϕ Algorithm proposed in this thesis takes a similar approach to DIFFP-ID3, but extends it in the following ways: first, the depth of the tree is not fixed beforehand; instead, the induction of nodes is constrained by the remaining privacy budget, which leads to a finer grained control of structural complexity measures for generalization (Devroye et al., 2013). Second and more importantly, the algorithm assigns a portion of the privacy budget to each iteration of the decision tree induction process *dynamically*.

Algorithm 1 presents the adaptive budget allocation procedure, which can be instantiated with any permissible ϕ. The complete decision tree induction algorithm, ADiffP-ϕ, follows with a pruning of the tree and an application of the C4.5 pruning scheme using only the noisy counts conducted at the internal nodes. This does not require additional privacy budget.

ADiffP-ϕ runs recursively, and its initial input consists of the input samples $\mathcal{S}$, the privacy budget ϵ, and a parameter t that sets thresholds for adaptive budget decisions, and is discussed in more detail in the next section.

Each iteration consists of three steps:

(i) Determine the privacy budget to allocate for a split;

(ii) Check the remaining budget and return a leaf if it is insufficient;

(iii) Determine and apply a split with the exponential mechanism and induce the subtrees recursively.

Allocating budget for a split. Lines 1–9 determine how much budget to allocate for a split. Line 1 creates the list of attribute sets for the split. The number of attributes is bounded by $d = \dim(\mathcal{X})$ for the tree. Lines 2-4 split the node based on the choice of splitting function ϕ. For each candidate attribute $\mathcal{X}_i$, with values $j \in \mathcal{X}_i$, $\textbf{splits}(\mathcal{X}_i)$ is the set of splits for the attribute (*e.g.*, two for a binary variable, tunable for a real-valued variable). Let us consider $\mathcal{S}_{\mathcal{X}_i=j}$ are the samples from $\mathcal{S}$ that take the value j for attribute $\mathcal{X}_i$ ($|\mathcal{S}_{\mathcal{X}_i=j}| = m_{\mathcal{X}_i=j}$), and $\mathcal{S}^{(y)}_{\mathcal{X}_i=j}$ corresponds to the subset of those samples that also take class $y \in \mathcal{Y}$ ($|\mathcal{S}^{(y)}_{\mathcal{X}_i=j}| = m^{(y)}_{\mathcal{X}_i=j} \leq m_{\mathcal{X}_i=j}$). The perspective transforms $\Phi_{\mathcal{S},\mathcal{X}_i}$, in Line 3, is stored in the vector $\Phi_{\mathcal{S}} \in [0, \max_v \mathbb{E}_{j\sim\mathcal{X}_i}[\mathcal{S}_{\mathcal{X}_i=j}]]^d$ (Definition 5). The multi-class implementation fits to the multi-class multi-label setting (Schapire & Singer, 1999); also, the theory developed in Sections 3.3 and 3.4 applies as is.

Sort_increasing. Line 5 of the algorithm sorts the vector coordinates in increasing order. All the attributes in the attribute list, $\mathcal{X}$, are sorted increasingly, based on their splitting score values.

The delta score. $\delta_{\mathcal{S}}$ in Line 6 of the algorithm is the delta score function. The delta score assesses the entropy difference between the top candidate attributes, which is the relative difference between the top two best-scoring attributes in the attribute list ($\delta_{\mathcal{S}} \geq 0$). The unit in the denominator prevents a zero denominator.

The objective of the delta score is to tune the privacy budget spent in the differential privacy mechanisms using the following key observation: since the larger the noise in the exponential mechanism, the more likely the top splits ranking is jammed. Consequently, the algorithm chooses to spend a *small* privacy budget when the top splits look alike from the *loss* standpoint — and therefore more noise may not significantly reduce the chance of picking such a top split. In ADiffP-ϕ, this choice is triggered when the delta score is small, since in this case, at least two top splits look alike. Therefore, the algorithm fixes $\epsilon^* = \epsilon_{\min}$. If however $\delta_{\mathcal{S}}$ is large, the topmost attribute looks significantly better than all the others, and the algorithm prefers to reduce the likelihood of jamming the top ranking by locally spending a larger privacy budget, fixing [4] $\epsilon^* = 5\epsilon_{\min}$ in ADiffP-ϕ.

This tuning of the privacy budget has to be itself private, and this is ensured by using the Laplace mechanism in Line 7. The global sensitivity of $\delta_{\mathcal{S}}$ is derived in the proof of Theorem 7. The Laplace mechanism assesses the differentially private entropy difference between the top candidate attributes here, and based on this difference, the privacy budget is set to one of two levels in Line 8: either $\epsilon_{\min}$ or $5\epsilon_{\min}$. Essentially, a larger budget needs to be allocated when the difference between the top candidates is large enough to pick the top candidate with a high probability despite the noise. But spending unnecessary budget trying to choose between attributes that are expected to perform similarly needs to be avoided (the top splits look alike from the *loss* standpoint). The setting of $\epsilon_{\min}$ is discussed in more detail in the next section. Finally, Δ^*_ϕ is given in Table 3.1 for each of the choices of permissible ϕ. For other permissible ϕ, Theorem 4 and Definition 6 are used. The entropy difference can be used also to

[4] Value 5 in the multiplication is a fixed parameter and is experimentally shows a good result.

determine the privacy budget directly (i.e., $\epsilon^* \propto \delta'^{-1}_{\mathcal{S}}$), but in the experiments it is found that in practice, this approach has not led to improvements over the simpler approach with two privacy levels. The refinement of this methodology could be a subject for further research.

Noisy counts. $NoisyCount(\bullet|\epsilon^*)$ in the algorithm refers to the application of the Laplace mechanism with privacy parameter ϵ^*. In line 10, the number of records in the internal node is counted with the addition of Laplace noise to achieve differential privacy. The noisy count in this line is used to decide whether the iteration on this node continues or if it needs to stop and a leaf node needs to be created.

Stopping rule. Lines 11–16 test and apply a stopping rule, in a similar fashion to DIFFP-ID3. When the condition in line 11 of the algorithm is met, the algorithm needs to stop the iteration in this node and a leaf node needs to be created. At least ϵ_{rem} portion of the budget is needed to create a leaf node. ϵ_{rem} maintains the remaining budget for future iterations, and is updated whenever the algorithm applies a differentially private query (NoisyCounts, the exponential mechanism or the delta score calculation). $Histogram(.)$ returns a vector $s \in \mathbb{N}^{\mathcal{Y}}$ giving the per-class sample sizes in $\mathcal{S}$. s_y is its coordinate for class y, which is used for noisy counts in the node. This noisy count is used to assign a label to the leaf node.

Applying a split. If the stopping condition did not hold, lines 17–20 apply the exponential mechanism with the determined privacy parameter to select an attribute, split the samples according to the selected attribute, and then the procedure is applied recursively to each of the partitions, with the remaining privacy budget.

Now that the description of Algorithm 1 is clarified, it has to meet the main condition of this thesis; this algorithm is claimed to be differentially private and the

Algorithm 1 Procedure top-down ADiffP-$\phi(\mathcal{S}, \epsilon, t)$

Inputs: $\mathcal{S}$: learning samples; ϵ: privacy budget; t: adaptive budget threshold.
Output: ϵ_{rem}: remaining privacy budget;

1: $\mathcal{X} = \{\mathcal{X}_1, .., \mathcal{X}_d\}$
2: **for** $\mathcal{X}_i \in \mathcal{X}$ **do**
3: $\quad \Phi_{\mathcal{S},\mathcal{X}_i} \leftarrow \mathbb{E}_{j \sim \mathbf{splits}(\mathcal{X}_i), y \sim \mathcal{Y}} \left[\check{\phi} \left(m^{(y)}_{\mathcal{X}_i = j}, m_{\mathcal{X}_i = j} \right) \right]$
4: **end for**
5: $\Phi^{\uparrow} \leftarrow$ **sort_increasing**$(\Phi_{\mathcal{X}})$
6: $\delta_{\mathcal{S}} \leftarrow (\Phi^{\uparrow}_2 - \Phi^{\uparrow}_1)/(1 + \sum_{\mathcal{X}} \Phi^{\uparrow}_{\mathcal{S},\mathcal{X}_i})$ // {Comment: delta score}
7: **if** $\delta_{\mathcal{S}} + \text{Lap}\left(\frac{1}{\epsilon_{\min}} \cdot \frac{\Delta^*_\phi}{1 + (d-1) \cdot \Delta^*_\phi} \right) \leq \frac{t}{\epsilon_{\min}}$ **then**
8: $\quad \epsilon^* \leftarrow \epsilon_{\min}$ **else** $\epsilon^* \leftarrow 5\epsilon_{\min}$
9: **end if**
10: $\tilde{m}_{\mathcal{S}} \leftarrow NoisyCount(\mathcal{S}|\epsilon^*)$
11: **if** $\mathcal{X} = \emptyset$ or $\frac{\tilde{m}_{\mathcal{S}}}{C \cdot \max_{\mathcal{X}_i \in \mathcal{X}} |\mathcal{X}_i|} < \frac{t}{\epsilon^*}$ or $\epsilon_{rem} \leq 2\epsilon^*$ **then**
12: $\quad$ **if** $\epsilon_{rem} < \epsilon^*$ **then** $\epsilon^* \leftarrow \epsilon_{rem}$ **end if**
13: $\quad s \leftarrow Histogram(\mathcal{S})$
14: $\quad \forall y \in \mathcal{Y} : \tilde{m}_y \leftarrow NoisyCount(s_y|\epsilon^*)$
15: $\quad$ **return** leaf labeled with $\arg\max_y(\tilde{m}_y)$
16: **end if**
17: $\mathcal{X}^*_i \leftarrow ExpMechanism(\mathcal{X}, \Phi_{\mathcal{S}}|\epsilon^*)$
18: ϵ_{rem} is updated based on ϵ wherever the privacy budget is consumed
19: $\forall j \in \mathcal{X}^*_i$, top down ADiffP-$\phi(\mathcal{S}_{\mathcal{X}^*_i = j}, \epsilon_{rem}, t)$
20: **return** subtree $Partition(\mathcal{S}, \mathcal{X}^*_i)$

proof is as follows.

Theorem 7. *Algorithm 1 maintains ϵ-differential privacy.*

The proof is no different from that of Friedman and Schuster (2010). Plus, the update of ϵ_{rem} ensures that the algorithm does not exceed the privacy budget and hence does not violate the differential privacy guarantee. The only additional result needed relies on the main component which is different from their approach, the test on the delta score. (Line 6 of the Algorithm 1)

Lemma 8. *For any data-independent t, the test $t - \delta_{\mathcal{S}} \geq \text{Lap}(b)$, with $b \doteq (1/\epsilon) \cdot \Delta^*_\phi/(1 + (d-1) \cdot \Delta^*_\phi)$, preserves ϵ-differential privacy.*

Proof. For notational convenience, denote $\delta_{\mathcal{S}}$ as the delta score associated to sample $\mathcal{S}$. Consider two neighboring samples $\mathcal{S}_1$ and $\mathcal{S}_2$, differing by at most one example.

$$
\begin{aligned}
&|t - \delta_{\mathcal{S}_2} - (t - \delta_{\mathcal{S}_1})| \\
&= \left| \frac{\Phi^\uparrow_2(\mathcal{S}_1) - \Phi^\uparrow_1(\mathcal{S}_1)}{1 + \sum_{\mathcal{X}_i} \Phi^\uparrow_{\mathcal{X}_i}(\mathcal{S}_1)} - \frac{\Phi^\uparrow_2(\mathcal{S}_2) - \Phi^\uparrow_1(\mathcal{S}_2)}{1 + \sum_{\mathcal{X}_i} \Phi^\uparrow_{\mathcal{X}_i}(\mathcal{S}_2)} \right| && (3.8) \\
&\leq \max_{\mathcal{S} \in \{\mathcal{S}_1, \mathcal{S}_2\}} \frac{\Phi^\uparrow_2(\mathcal{S}) - \Phi^\uparrow_1(\mathcal{S})}{1 + \sum_{\mathcal{X}_i} \Phi^\uparrow_{\mathcal{X}_i}(\mathcal{S})} && (3.9) \\
&\leq \max_{\mathcal{S} \in \{\mathcal{S}_1, \mathcal{S}_2\}} \frac{\Phi^\uparrow_2(\mathcal{S}) - \Phi^\uparrow_1(\mathcal{S})}{1 + (d-1) \cdot \Phi^\uparrow_2(\mathcal{S}) + \Phi^\uparrow_1(\mathcal{S})} && (3.10) \\
&\leq \max_{\mathcal{S} \in \{\mathcal{S}_1, \mathcal{S}_2\}} \frac{\Phi^\uparrow_2(\mathcal{S})}{1 + (d-1) \cdot \Phi^\uparrow_2(\mathcal{S})} && (3.11) \\
&\leq \frac{\Delta^*_\phi}{1 + (d-1) \cdot \Delta^*_\phi} \ . && (3.12)
\end{aligned}
$$

Equation (3.8) comes from the fact that t does not depend on $\mathcal{S}_1$ nor $\mathcal{S}_2$. Ineq. (3.9) follows from $\delta_{\mathcal{S}} \geq 0$, ineq. (3.10) holds because $\Phi^\uparrow_j \geq \Phi^\uparrow_2, \forall j \geq 2$ by definition; ineq. (3.11) comes from the fact that the function on the left hand-side is decreasing with $\Phi^\uparrow_1(\mathcal{S})$. Finally, ineq. (3.12) comes from the fact that $\forall \mathcal{X}_i \in \mathcal{X}$, $\Phi^\uparrow_{\mathcal{X}_i} \leq \check{\phi}(1, m+1) \leq \Delta^*_\phi$, by definition of Δ^*_ϕ.

Hence, it follows from C. Dwork et al. (2006) that the statistic

$$
t - \delta_{\mathcal{S}} - \text{Lap}\left(\frac{1}{\epsilon} \cdot \frac{\Delta^*_\phi}{1 + (d-1) \cdot \Delta^*_\phi} \right)
$$

is ϵ-differentially private, and so is the sign test built from it in Lemma 8, as claimed.

□

3.5.3 Parameter Tuning

There are two parameters that are used throughout the algorithm and should be determined in advance, t and $\epsilon_{\min}$. The procedures for setting these parameters depend only on the public parameters (m, d, $\mathcal{Y}$) and do not require querying the samples (and spending the privacy budget).

The adaptive budget threshold parameter. This parameter determines the threshold for the privacy budget allocation. To save budget, the algorithm relies on the assignment of t using synthetic datasets adopted from Domingos and Hulten (2000) and Friedman and Schuster (2010), fixing t based *only* on the public parameters to maximize accuracy. In particular, t is calculated independently and regardless of the choice of ϵ or ϕ. The details for calculating t are in Section 3.5.4 of this chapter (based on that analysis, $t = 0.01$ is set in all the experiments).

The minimal local budget, $\epsilon_{\min}$. A key part of ADiffP-ϕ is a fixed minimal share of the total privacy budget to be spent at each iteration, $\epsilon_{\min}$. Notice that $s_* \doteq \epsilon/\epsilon_{\min}$ gives an upper-bound on the number of nodes of the tree to be induced. There is clearly a non-trivial sweet spot to fix s_* between the two extremes: if it is too small, the trees are too small to fit to the data at hand, and if it is too large, a large noise magnitude per query is created that will deteriorate the accuracy of the tree. A value of $s_* = O(\log m)$ experimentally yields good results, with the additional advantage that standard uniform convergence bounds yield with high probability an absolute difference between the empirical risk and true error that vanishes as $O(\log(m)/\sqrt{m})$ (Devroye et al., 2013). In practice, this share can be slightly modified based on the choice of the dataset. The minimal local budget is proposed as:

$$\epsilon_{\min} = ((d/|\mathcal{Y}|) \cdot \log m)^{-1} \cdot \epsilon \ . \tag{3.13}$$

3.5.4 Calculating the Threshold value t

The threshold t can be optimized using only the public parameters and does not need to query the data or use the privacy budget. In this research, the t value is a function of the training size m.

Generating Synthetic Dataset

The synthetic dataset used in this experiment is adopted from the technique introduced in Domingos and Hulten (2000) and later used in Friedman and Schuster (2010). The synthetic dataset is generated in this research with ten uniform attributes and a class attribute. First, trees with a predetermined height are generated by randomly splitting the attributes, then starting from the third level of the tree with probability p_{leaf}, the nodes are turned to leaves and a class value is picked randomly for each leaf. Finally, the points are sampled with uniform distribution and classified using the generated tree. Using this technique, training sets are created with different sizes (10000 and 15000 records are used in this experiment), and test sets are created separately and independently, with 10% of the size of the training sets.

Threshold Value

The challenge is to assign the threshold value, t, for the differentially private tree used in the comparisons in lines 6 and 10 of the Algorithm 1. Although there is no exact value defined for this threshold, by applying different varieties of synthetic datasets to this algorithm, an equivalent value for t is heuristically determined that puts a bound on the best average accuracy results.

In the experiments shown in Figure 3.2, ADiffP-ϕ is run using ϕ_{Err} as the splitting

criterion, with $p_{leaf} = 0.3$ and $p_{leaf} = 0.5$, and sample size=$\{10000, 15000\}$ records, and the accuracy results are averaged over 10-fold cross-validation. This figure employs a wide range of privacy budget ϵ and compares four different choices of threshold values $t = \{0.2, 0.1, 0.01, 0.001\}$. A bound on the threshold value is determined here, which gives the best accuracy results using different choices of privacy budgets and sample sizes. It is observed that smaller values of threshold t tend to result in better accuracy. However, there is a limit value for the threshold, which is estimated at $t = 0.01$, from which the average accuracy results do not change dramatically if the threshold t is reduced further. In the rest of the experiments in this chapter, instead of tuning the choice of t to the number of records m, this value $t = 0.01$ is always used.

3.5.5 Pruning differentially private trees

Pruning a differentially private tree must be adapted to the notion of differential privacy. Differential privacy tries to avoid adding too much noise to a smaller sample size which will create non-accurate results. So pruning techniques such as minimal cost complexity pruning (Breiman et al., 1984) and reduced error pruning (Quinlan, 2014), that are based on reducing the size of a sample data to prune the unnecessary branches are not good options for differentially private trees. Instead, like the greedy style differentially private trees in Friedman and Schuster (2010), the tree pruning approach is taken from the original C4.5 error based pruning in Quinlan (2014). Pruning is performed using the information that has been already gathered during the tree-building process, and since it does not require additional data, it reduces the impact of the noise on the resulting trees. Fletcher and Islam (2015a) also used this

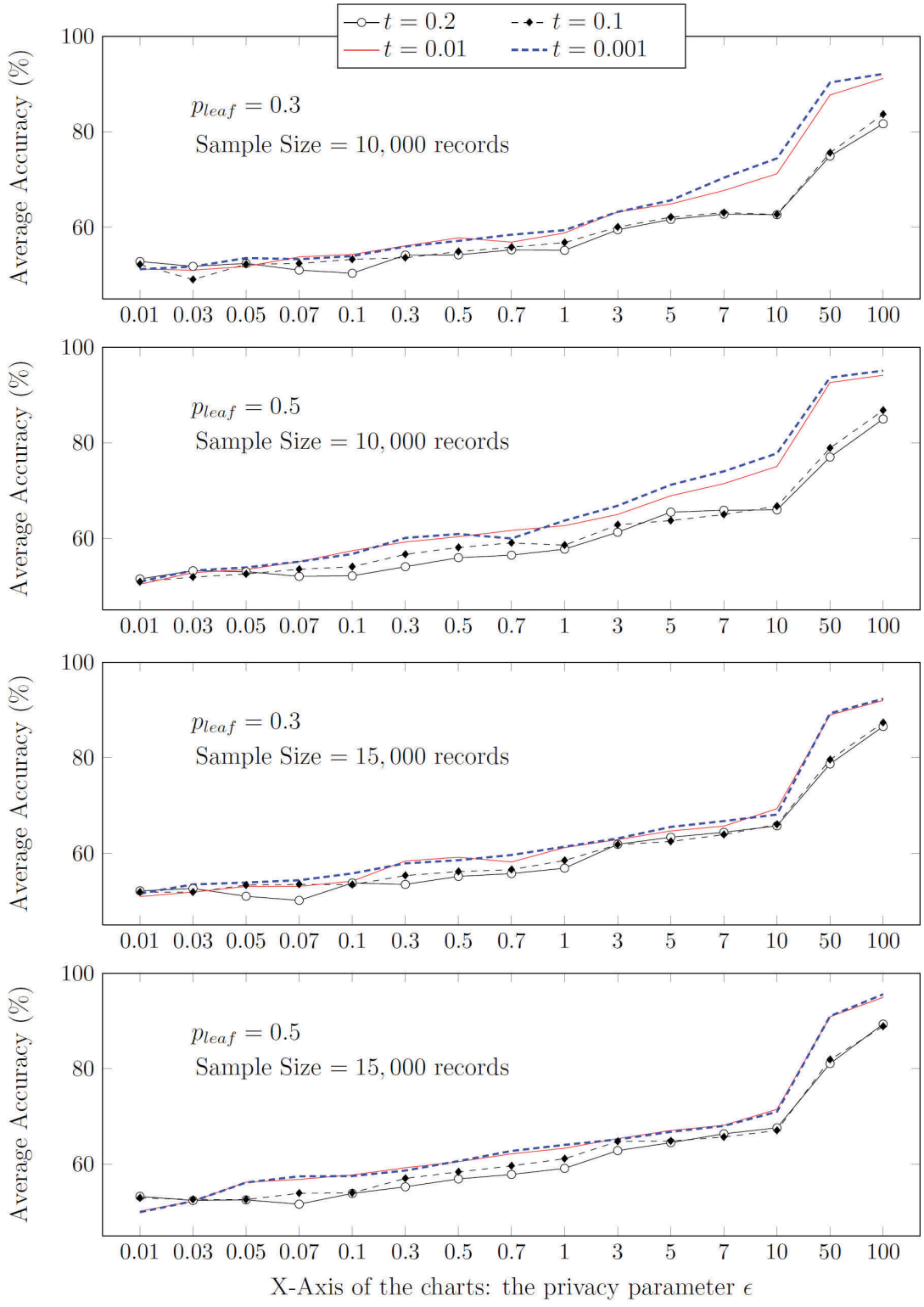

Figure 3.2: Average accuracy results of ADiffP-ϕ (using ϕ_{Err}) with the choice of four different threshold values t.

pruning process in their greedy trees with a slight change in the noise that is added to the number of child nodes.

The pruning approach for DIFFP-ID3 decision trees (Friedman & Schuster, 2010) is based on the counts of the instances of each node and the class counts at each level of the tree. Because noise is added for differential privacy, the sum of these counts does not necessarily match the data size. The pruning process is based on running two passes over the tree, a top-down pass and a bottom-up pass. The first top-down pass counts the total number of instances in each level of the tree and matches them to the count in the parent levels. The sum of all the nodes on each level of the tree should match the number of records in the whole dataset. Then, in the bottom-up pass, all class counts are aggregated and matched to the total instance counts from the first pass to make sure that the sum of the class counts in each node matches the support of that node. Finally, the instance counts from the first pass and the updated class counts from the second pass are used to measure the error rates as in C4.5 original pruning process.

3.6 Experimental Results

In this section, the experiment results of the proposed algorithm on three publicly available datasets are shown, and the ADiffP-ϕ algorithm is compared to state-of-the-art algorithms for differentially private decision trees. The algorithm is implemented in Java using the Weka data mining tool (Witten & Frank, 1999).

In these experiments, benchmark datasets previously used in Friedman and Schuster (2010) and Jagannathan et al. (2012) are considered: the Mushroom, Nursery and Adult datasets from the UCI ML Repository (Bache & Lichman, 2013). Adult has

6 continuous attributes, 8 categorical attributes and a binary class attribute. The continuous attributes are dropped and records with missing values are removed. The training and test sets are merged (m = 45,222). The mushroom dataset has m = 8,124 examples, 20 categorical attributes (after an attribute with missing values was removed) and a binary class attribute. The nursery dataset has m = 12,960 examples, 8 categorical attributes, and 5 classes.

Accuracy values shown in Figures 3.3, 3.4, and 3.5 are calculated over 10-fold cross-validation, and the error bars in these figures indicate the standard deviation of the accuracy over 10-fold cross validation.

3.6.1 Comparison of entropies with ADiffP-ϕ.

Figure 3.3 shows the results of running ADiffP-ϕ on a wide range of privacy budgets with different choices of splitting criteria from Table 3.1. The trees created using ϕ_{IG} and ϕ_{Err} splitting criteria are more accurate than the other trees. For the rest of the experiments in this section, ϕ_{IG} and ϕ_{Err} splitting criteria are used to compare the results of ADiffP-ϕ with the state-of-the-art algorithms.

3.6.2 Comparison of ADiffP-ϕ to state-of-the-art algorithms.

Figures 3.4 and 3.5 show the experiment results comparing ADiffP-ϕ to several other algorithms: DiffP-C4.5 (Friedman & Schuster, 2010), DiffGen (Mohammed et al., 2011) and SULQ-based ID3 (Friedman & Schuster, 2010) (based on the SULQ framework in Blum et al. (2005)). The results are provided in two sets: the first using ϕ_{Err} as the splitting criterion, and the second using ϕ_{IG} as the splitting criterion for the aforementioned algorithms. The results of ADiffP-ϕ, DiffP-C4.5 and SULQ-based

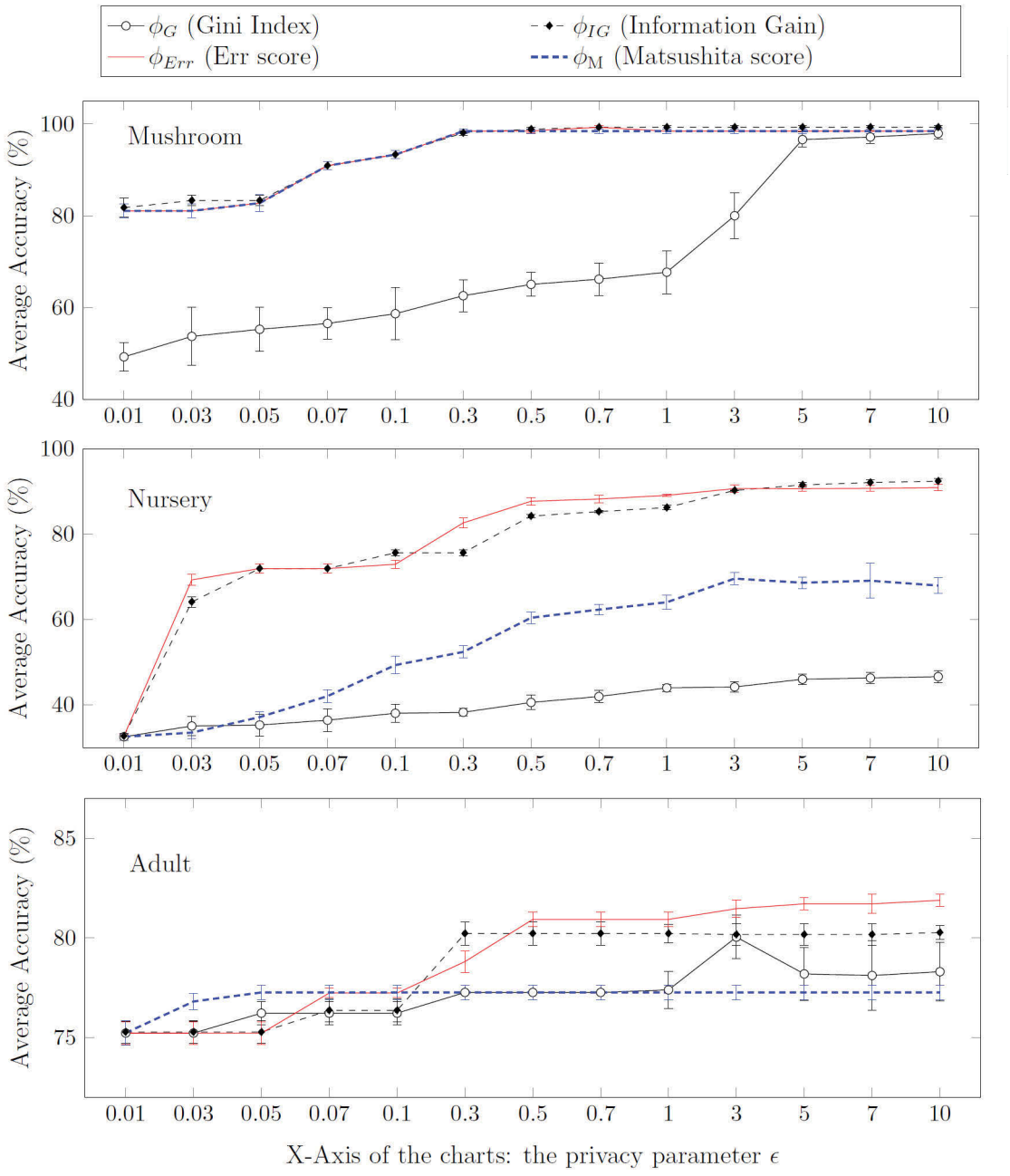

Figure 3.3: Average accuracy and standard deviation of ADiffP-ϕ over 10-fold cross validation on the Mushroom, Nursery and Adult datasets.

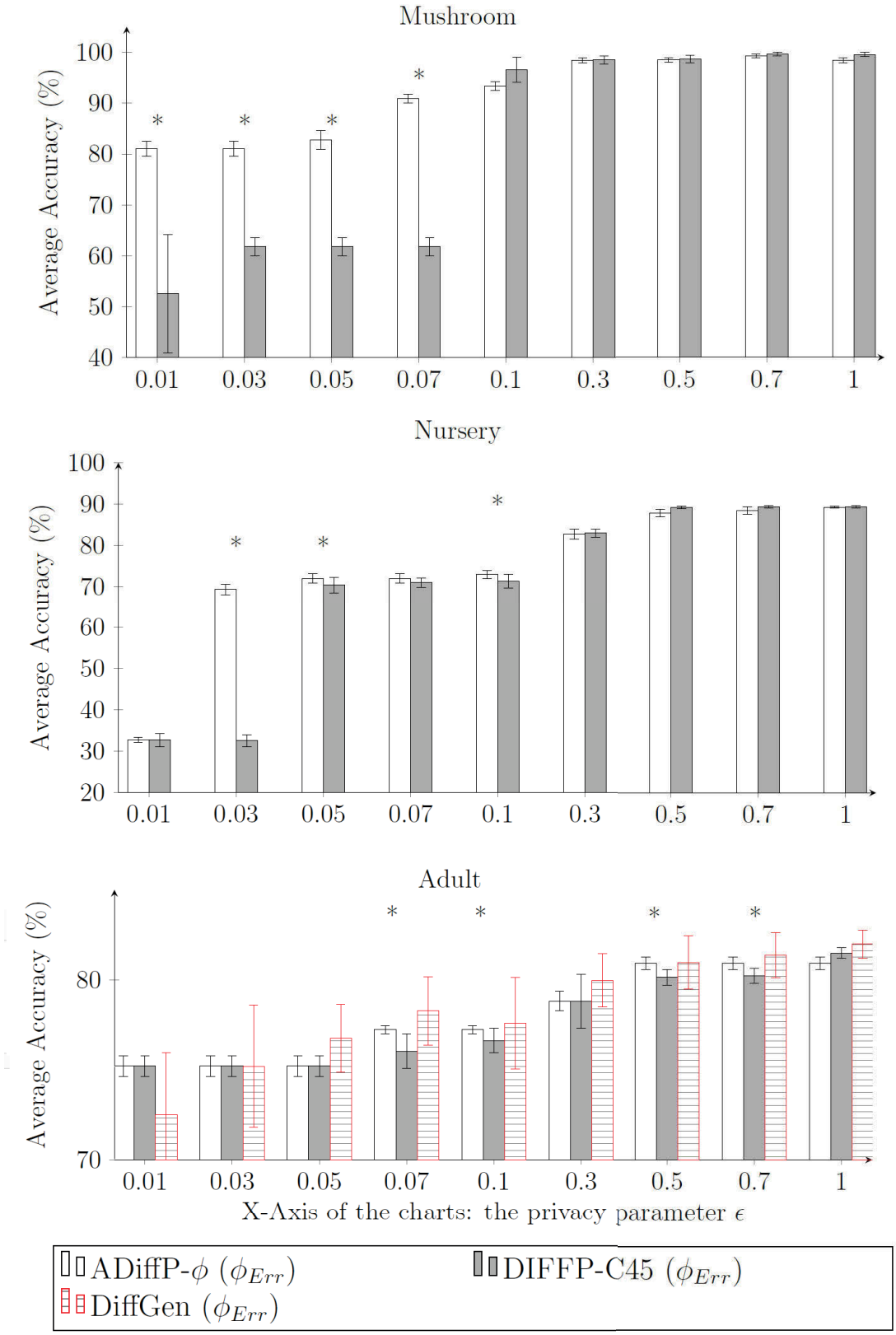

Figure 3.4: Comparing average accuracy and standard deviation of ADiffP-ϕ, DIFFP-C4.5, DiffGen and SULQ (splitting criterion: ϕ_{Err}).

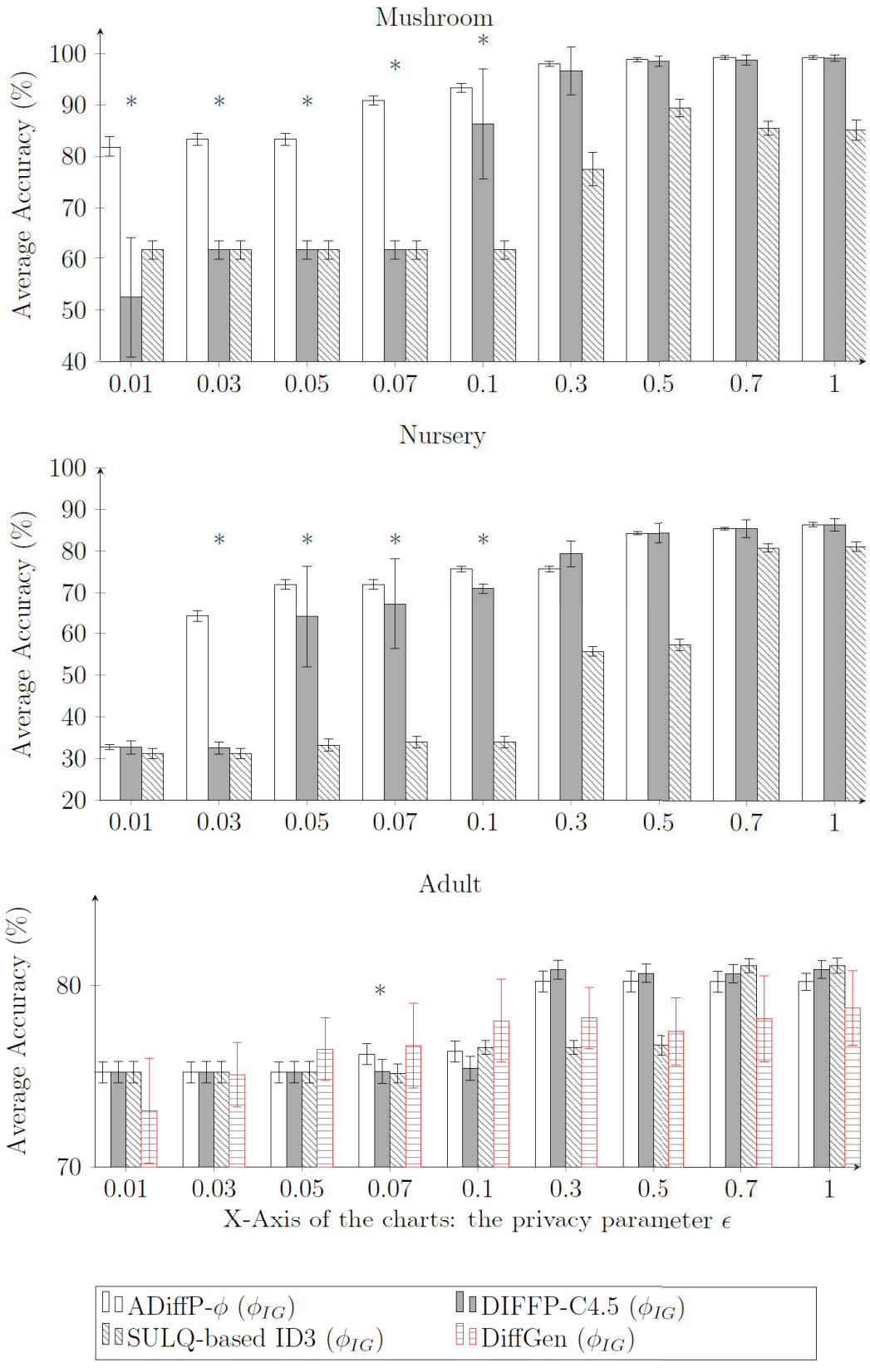

Figure 3.5: Comparing average accuracy and standard deviation of ADiffP-ϕ, DIFFP-C4.5, DiffGen and SULQ (splitting criterion: ϕ_{IG}).

ID3 are obtained by running the classifier over 10-fold cross validation. DiffP-C4.5 (Algorithm 2 in Friedman and Schuster (2010)) is used with depth five, and the same depth is used for SULQ-based ID3 (Algorithm 1 in Friedman and Schuster (2010)). For DiffGen, the settings that showed the best results in their own experiments in Mohammed et al. (2011) are followed: fix the specialization parameter to 15, and divide the Adult dataset to 2/3 training set and 1/3 test set, and then average the accuracy results over ten runs of the classifier.

In the experiments shown in Figures 3.4 and 3.5, the result of the ADiffP-ϕ algorithm are compared with the state-of-the-art algorithms using a smaller range of the privacy budgets:

$$\epsilon \in \{0.01, 0.03, 0.05, 0.07, 0.1, 0.3, 0.5, 0.7, 1\} \ .$$

The privacy budget ϵ given to a user in real-world scenarios depends on the specific use-cases: some of the well-known works in the literature (C. Dwork & Roth, 2014; C. Dwork, 2008) suggested a small value of $\epsilon = 0.01$ and some public projects (Machanavajjhala et al., 2008) used values as high as $\epsilon = 8.6$ in large scale.

The difference between Figures 3.4 and 3.5 is in the choice of ϕ. Error score, ϕ_{Err} is used as the splitting criterion in Figure 3.4 and in 3.5, Information Gain ϕ_{IG} is used as the splitting criterion. SULQ-based ID3 (Friedman & Schuster, 2010) (based on the SULQ framework of (Blum et al., 2005)) shown in another set of experiment results in Figure 3.5.

DiffGen was removed from the statistical t-test results, because it was not included in all of the experiments in Figures 3.4 and 3.5. Therefore, stars (*) in Figures 3.4 indicate area where t-test (type-I error α=0.05) shows the difference in means between algorithm 1 and DiffP-C4.5 is statistically significant. And, stars (*) in Figures 3.5

indicate area where t-test (type-I error α=0.05) shows the difference in means between algorithm 1 on one hand, and DiffP-C4.5 and SULQ-based ID3 on the other hand, is statistically significant.

This is observed with the stricter privacy settings ($\epsilon \in [0.01, 0.1]$) and on the smaller datasets, *i.e.*, Mushroom and Nursery. It demonstrates the effectiveness of ADiffP-ϕ in improving the privacy/accuracy trade-off, and seems therefore to be a good contender for strong privacy requirements, and/or when the privacy noise has to be large in smaller sample size (small ϵ, small m).

3.7 Discussion

The most significant question in this thesis is whether the dynamic budget allocation of ADiffP algorithm can result in better classification accuracy and it relies on how much of the privacy budget to spend during each step of inducing a differentially private tree. This is where the adaptive budget allocation of ADiffP-ϕ algorithm is applicable. The model can be considered as a building block for budget allocation optimization in a differentially private decision tree algorithms. Most of the current differentially private algorithms rely on sequential composition to allocate the privacy budget, by splitting the budget equally between different stages of the algorithm. Adaptive budget allocation determines what portion of the privacy budget to spend by querying the data and analyzing the data samples and then adjusting the privacy budget based on the data characteristics. The application of this method is described in a single decision tree induction procedure in this chapter and implemented using the WEKA data mining tool.

The results of the implementation of this algorithm have been validated and analyzed to meet Contribution 1 of the thesis by proposing an algorithm for adaptive privacy budget allocation on a differentially private decision tree. The system has been evaluated based on parameters such as privacy, utility and stability of the utility results. The experiments in this chapter show that adaptive allocation of the privacy budget allows the algorithm to create more accurate trees than existing ones, particularly with more sensitive data that demands stronger privacy in the output. They also show that compared to state-of-the-art models, this technique provides better standard deviation on the accuracy results of the trees with smaller values of the privacy budget.

Contribution 3 of this thesis proposed new splitting criteria in a differentially private setting for the first time in the literature. The Matsushita score in this chapter is placed as a splitting criterion for differentially private decision trees. This splitting criterion is considered as an optimized and most accurate criterion for non-private decision trees (Kearns & Mansour, 1999). However, the sensitivity of this splitting function is relatively high in differential privacy, and this makes it an expensive criterion for differentially private trees. The accuracy results of this splitting criterion are compared with other splitting functions in Figure 3.3. In this experiment, it is evident that the cheaper splitting criteria in terms of sensitivity perform better in the experiments. For example, Error score (ϕ_{Err}) with a sensitivity of 1 performed very well on splitting the nodes on the tree.

In addition to the accuracy results and stable standard deviations in the accuracy results, the ADiffP-ϕ algorithm has a couple of other advantages as well. The first advantage of this algorithm is considering the trade-off between privacy and the

convergence rate of splitting criteria, while inducing the differentially private trees. The fact that the entropy can cause extra expenses for privacy is formalized in this chapter. The trade-off is clear: better entropy from the convergence standpoint from the splitting criteria means more privacy budget to spend which can result in less accurate splits. This problem is formalized based on four splitting criteria, Information Gain (ϕ_{IG} in ADiffP-ϕ), Gini Score (ϕ_{G}), Misclassification Error score (ϕ_{Err}) and Matsushita's loss score (ϕ_{M}). This research calculates the sensitivity of these splitting criteria using one theorem (Theorem 4 and later in Lemma 6). The improvement in the sensitivity of ϕ_{G} is the result of the normalization in this theorem.

Another advantage of the algorithm is in generating trees with dynamic heights, which can reduce the impact of user error in the results due to assigning the wrong parameters to the tree. Previous differentially private trees suffer from the disadvantage of allocating a fixed depth (or level) to the tree before tree induction. Blum et al. (2005) in the SULQ framework choose the number of attributes m for the depth of the tree (in SULQ-based ID3 in Friedman and Schuster (2010) a depth of five is assigned to this tree). Friedman and Schuster (2010) in the DiffP-C4.5 algorithm choose trees with a depth of five. Mohammed et al. (2011) in the DiffGen algorithm choose trees with four levels in their experiments. In addition to enforcing a limit to the trees itself by assigning a fixed height, the portion of the privacy budget is calculated based on this fixed height, which affects all iterations in the tree. If this portion is too small, the tree might be too noisy and not accurate.

The heuristic parameters that are involved in the ADiffP-ϕ algorithm are described in detail in Section 5.2 of Chapter 5. Although some of these parameters can be theoretically optimized, the algorithm shows competitively good results and is

considered to be an excellent choice for differentially private decision tree inductions.

3.8 Conclusion

This chapter investigates the trade-off between the convergence rates of the splitting criteria and privacy in decision tree induction to address Contributions 1 and 3 of this thesis. Contribution 1 proposes *an algorithm for adaptive privacy budget allocation on a differentially private decision tree*, and Contribution 3 introduces *Matsushita loss as a splitting criterion on differentially private decision trees*.

In a non-private setting, when the convergence rate/learning of a splitting function is the only objective of the tree induction, there is a clear ordering in the accuracy of the splitting criteria. However, theoretically and empirically, this chapter shows that this is not the case when *both* differential privacy and the convergence rate are at stake. This raises the challenge of allocating the privacy budget effectively throughout the induction process to achieve the right balance in the trade-off between accuracy and privacy. This chapter provides such an example, with an algorithm that adapts the privacy budget that is allocated to each split decision. Experiments in this chapter reveal that this approach obtains better accuracy than state-of-the-art algorithms on single decision tree inductions, especially when there are strict privacy constraints and when learning over small datasets.

Decision forests generally overcome the deficiencies of single decision trees and create more stable and more accurate classifiers. Therefore, next chapter of the thesis examines generation of an ensemble of trees that is created with the algorithm from this chapter. The ensemble application of this algorithm to create differentially private forests will address Contribution 2 of this thesis. Creating this ensemble circumvents

the tree induction problem altogether, and also hinders the interpretability of the proposed model on the forest compared to the induction of a single tree.

Chapter 4

Differentially Private Random Forest

4.1 Introduction

In Chapter 3 of this thesis, a new privacy budget allocation algorithm for differentially private decision trees, called ADiffP-ϕ, is proposed. The budget allocation approach in the ADiffP-ϕ algorithm in Chapter 3 generates single differentially private decision trees with high accuracy. However, creating single trees always comes with several drawbacks, including high variance in the accuracy results, generating unbalanced trees and being dependent on the quality of the data in hand. This chapter takes advantage of applying ensemble methods on decision trees and overcomes the drawbacks of single tree inductions. Using an algorithm called RFDiffP-ϕ, this chapter examines the performance of bagging by creating a random forest classifier with single trees derived from the ADiffP-ϕ algorithm. The accuracy of the RFDiffP-ϕ algorithm is evaluated with synthetic and real datasets and is compared to the previous work in this area.

4.2 Motivations

An ensemble of random decision trees or a random forest is proposed to overcome the deficiencies of inducing single decision trees and is expected to result in better classifiers in comparison to the single trees. Every single tree is built individually when generating a decision forest, and by using random subsets of the attributes in the internal nodes of each tree. The random subset of attributes leads to a more diverse set of trees in the forest, and will lead to a better ensemble classifiers as more variety of trees are presented. Therefore, the main advantage of decision forests is the higher accuracy level in the outputs in comparison to the single trees (Breiman, 2001). However, knowledge from an ensemble of multiple trees may reveal more private information about a participant than an individual tree (Rana et al., 2015). According to Rana et al. (2015) ensemble approaches usually improves performance, however they weakens data privacy. Differential privacy benefits from a more accurate setup of random forests to generate more precise results from the classifiers, and also guarantees the desired level of privacy in the output results.

Previous work to generate differentially private random forests is extensive and is divided into two main categories: the forests that are built with greedy style tree inductions (Patil & Singh, 2014; Fletcher & Islam, 2015a; Rana et al., 2015), and the forests that are built with random style trees (Jagannathan et al., 2012; Jagannathan, Monteleoni, & Pillaipakkamnatt, 2013; Bojarski et al., 2014; Fletcher & Islam, 2015b, 2017). However, none of these techniques examines the changes in the accuracy of results by improving the privacy budget allocation on the forest classifiers. Introducing adaptive budget allocation in Chapter 3 opens an opportunity to use the enhanced budget allocation technique where an ensemble of differentially private trees is in

place. The greedy style trees that are generated by the ADiffP-ϕ algorithm have a dynamic proportion of the privacy budget per query. These trees also have dynamic depth, based on the usage of the privacy budget and the available data. These important features of a single tree, aligned with the important changes in splitting functions and their sensitivities, are examined in an ensemble setting of trees to evaluate their effectiveness in generating accurate forests.

4.3 Basic Preliminaries on generating a differentially private decision forest

The proposed algorithm in this chapter belongs to the greedy style forest category, where the greedy ADiffP-ϕ trees are used as single trees in the forest classifier and the framework adopts the bagging technique from the original Breiman (1996) framework.

Inducing a differentially private decision forest, like their non-private counterparts, has different aspects; some of them relate to generating each individual tree in the forest and others relate to the ensemble schema of the forest. Generating single differentially private decision trees is investigated in Chapter 3 of this thesis. This chapter focuses on the aspects that are specifically associated with the forest nature of the model.

4.3.1 Number of trees in the forest

One of the main parameters to set when building a forest is the number of decision trees in the forest. If the forest is made from random decision trees, the number of trees should be high enough to guarantee a good level of accuracy in the classifier

(Fan et al., 2003; Geurts et al., 2006). In the case of greedy style trees, this number does not necessarily need to be as high as with random style trees, although the forest can sometimes benefit from it. But the careful structure of the decision trees, i.e. especially if they are built from bootstrapped samples and use random subsets of the attributes in each internal node, makes them more diverse. Diversity in the trees reduces the chance of over-fitting of the ensemble classifier due to being too sensitive to small changes in the training data (Breiman, 2001). Therefore, it reduces the need to have a huge number of trees in the forest (Islam & Giggins, 2011). Furthermore, in the differentially private setting, the privacy budget needs to be allocated between all the trees in the forest and more trees means less privacy budget for each one of them, which could eventually generate less accurate ensemble classifiers. So the number of trees in the differentially private forest needs to be reasonable. This chapter relies on a set of experiments in Section 4.6 to choose the number of individual trees in the forest classifiers.

4.3.2 Privacy budget allocation on the forest

Section 3.5 shows how the privacy budget from the trees is consumed in the ADiffP-ϕ algorithm. Consequently, this budget allocation on single trees can be generalized to the forests. The overall privacy budget of the forest is divided between individual greedy style trees and is used on them until exhausted. The action of making the bootstrap samples is independent of the data, and therefore does not cost any privacy budget.

On each individual tree in the forest, the privacy budget must be consumed on the following tasks:

1. determining the budget for the tree attribute selection on each splitting point,
2. checking the stopping conditions of the tree,
3. assigning a class label to the leaf nodes, and
4. attribute selection.

This can be achieved by adding Laplace noise to count queries to perform the following tasks: budget determination for attribute selection, checking the stopping criteria of the trees and assigning the class labels to the leaf nodes. Satisfying differential privacy for the last task, attribute selection, can be achieved by applying the exponential mechanism.

The proposed algorithm selects the best attributes in the splitting points of the tree by first determining how much budget is needed to split the node (by consuming a fraction of budget), and then using the exponential mechanism to choose the best attribute to split the node. The exponential mechanism ranks the importance of each attribute and gives it a differentially private score. Then, the differentially private random forest algorithm chooses the highest-scored attribute to split the data in the internal node. The attribute selection task varies based on the choice of splitting criteria: Information Gain, Gini Index, Matsushita scorer and Error scorer. The sensitivity of each of these splitting criteria is essential to select the differentially private score of the attributes from the exponential mechanism. The details of calculating the splitting criteria and their sensitivities are given in Section 3.4 and summarized in Table 3.1.

Adding Laplacing noise ensures differential privacy when assigning the class labels on the leaf nodes of the trees. The amount of noise is determined based on the number

Algorithm 2 Procedure RFDiffP-$\phi(\mathcal{S}, \epsilon, B, t)$

Inputs: $\mathcal{S}$: learning samples; ϵ: privacy budget; B: number of bootstrap samples; t: adaptive budget threshold.

1: $\epsilon_b = \epsilon/B$
2: $\epsilon_r = 0$
3: **for** $b \in \{1, \ldots, B\}$ **do**
4: $\quad \mathcal{S}_b$: create a bootstrap sample from the training data
5: $\quad T_b$: ADiffP-$\phi(\mathcal{S}_b, \epsilon_b + \epsilon_r, t)$
6: $\quad \epsilon_r$: record the remaining privacy budget from T_b
7: **end for**
8: **return** $[T_b]_1^B$

of records at the leaf nodes and the probability distribution of the class labels. In the lower levels of the tree (that is, nearer to the root node of the tree), there are more data samples, and therefore the sensitivity of the counts will be lower, which results in less noisy outputs. However, less records remain at the leaves of the tree, so the sensitivity of the counts is higher here, resulting in more noise in the outputs. Therefore, deeper differentially private trees do not necessarily perform better than shallow trees. The RFDiffP-ϕ algorithm in this chapter uses this advantage on the forest. The privacy budget determines the depth of each tree and the adoption of the budget allocation techniques avoids unnecessary splits in the internal nodes.

4.4 The RFDiffP-ϕ Algorithm

The privacy-preserving random forest classifier, called RFDiffP-ϕ, is composed by forming an ensemble of single ADiffP-ϕ trees. Each differentially private decision tree in the ensemble is trained independently, and the final prediction of the ensemble classifier is made by aggregating the results from all of the trees.

Algorithm 2 presents the proposed differentially private random decision forest, RFDiffP-ϕ. This algorithm is based on the following steps:

In Line 1, the algorithm calculates the budget for each bootstrap sample based on the overall privacy budget assigned to the forest and the number of bootstrap samples. The user inputs the number of bootstrap data samples to use.

Lines 2-6 of the algorithm build a forest of B bootstrap samples from the training data. $\mathcal{S}_b$ in Line 3 is a bootstrap sample of the same size as the training set. Line 4 generates a differentially private tree from each of the bootstrap samples using the ADiffP-ϕ algorithm. Note that these trees use a modified version of the ADiffP-ϕ algorithm and select a random subset of the features at each splitting point. This is to guarantee diversity in the forest and avoid correlations between the trees. When some attributes are much stronger predictors than the others, they will be selected in most of the B trees and lead to correlation between trees in the ensemble. Line 1 of Algorithm 1 controls the list of attributes for each split. Here, parameter d in $\mathcal{X}_d$ is the *random* number of attributes for each split. (Note that in Chapter 3, single differentially private trees used all the attributes at each splitting point and d was equal to $\dim(\mathcal{X})$)

The privacy budget to generate each bootstrap sample is initially ϵ_b. The first tree in the ensemble receives this portion of the budget. However, if there is any budget left from the tree after induction, which is recorded as remaining budget ϵ_r in Line 5, it will be added to ϵ_b to use for the next tree in the forest. This process continues until all B trees in the forest are built.

In Line 7 of the algorithm, the ensemble forest classifier from B trees is built and returned.

The process of generating the forest in Algorithm 2 is differentially private, because the data in this algorithm is only accessed in the individual trees and all the individual trees are induced in a differentially private manner. The rest of the ensemble is independent of data and cannot leak any information, therefore it does not need differentially private treatment.

4.5 Classifying test data

The differentially private ensemble forest that was created in the previous section can now be used to classify a data point from the sampled test data. The majority vote from all the trees in the forest predicts the label of each new record in the test data. In this setting, the most commonly predicted label from all the trees in the ensemble is used to classify the new unseen records.

The classification procedure for new records is free of privacy cost, because the output that is created in a differentially private manner is used to find the majority vote, and differentially-private outputs do not incur additional privacy costs because of post-processing (C. Dwork & Roth, 2014).

4.6 Experimental Results

The RFDiffP-ϕ algorithm is implemented based on the Weka Random Forest implementation in the Java programming language (Witten & Frank, 1999). All of the default values for parameters in Weka stay the same in the implementation of this algorithm, with a couple of exceptions, including parameters I and K. The number

of bootstrap samples, which is controlled with parameter I, is modified to 10 bootstraps, to create an ensemble of ten trees in the forest. Also, in the experiments, the number of attributes for each tree is randomly assigned to make sure each internal node of the tree picks a different set of attributes and hence generates more diversity in the results. In the Weka implementation, the default parameter $K = \log_2(n) + 1$ controls the number of attributes to randomly select a particular node, where n is the number of attributes in the data. This parameter in Breiman's original RF is $\log_2(n - 1) + 1$. The parameter K can be also assigned manually in Weka. The RFDiffP-ϕ algorithm at each internal splitting node assigns a random value (always smaller than $\log_2(n) + 1$) for this parameter.

All the reported prediction accuracies in the experiments in this chapter are the average prediction accuracy results of performing ten runs of stratified 10-fold cross-validation. In another words, the 10-fold cross validation is applied on ten forests (each consisting of ten trees) in the experiments.

The experiments are based on the three classification datasets from the UCI Machine Learning Repository (Bache & Lichman, 2013): Adult, Mushroom and Nursery (details of these datasets are available in Section 3.6), and synthetic datasets (details of creating the synthetic datasets are in Section 3.5.4). All four splitting criteria from Table 3.1 are applied in the experiments. The performance of RFDiffP-ϕ is compared with the state-of-the-art in differentially private random forests and also with their non-private counterparts.

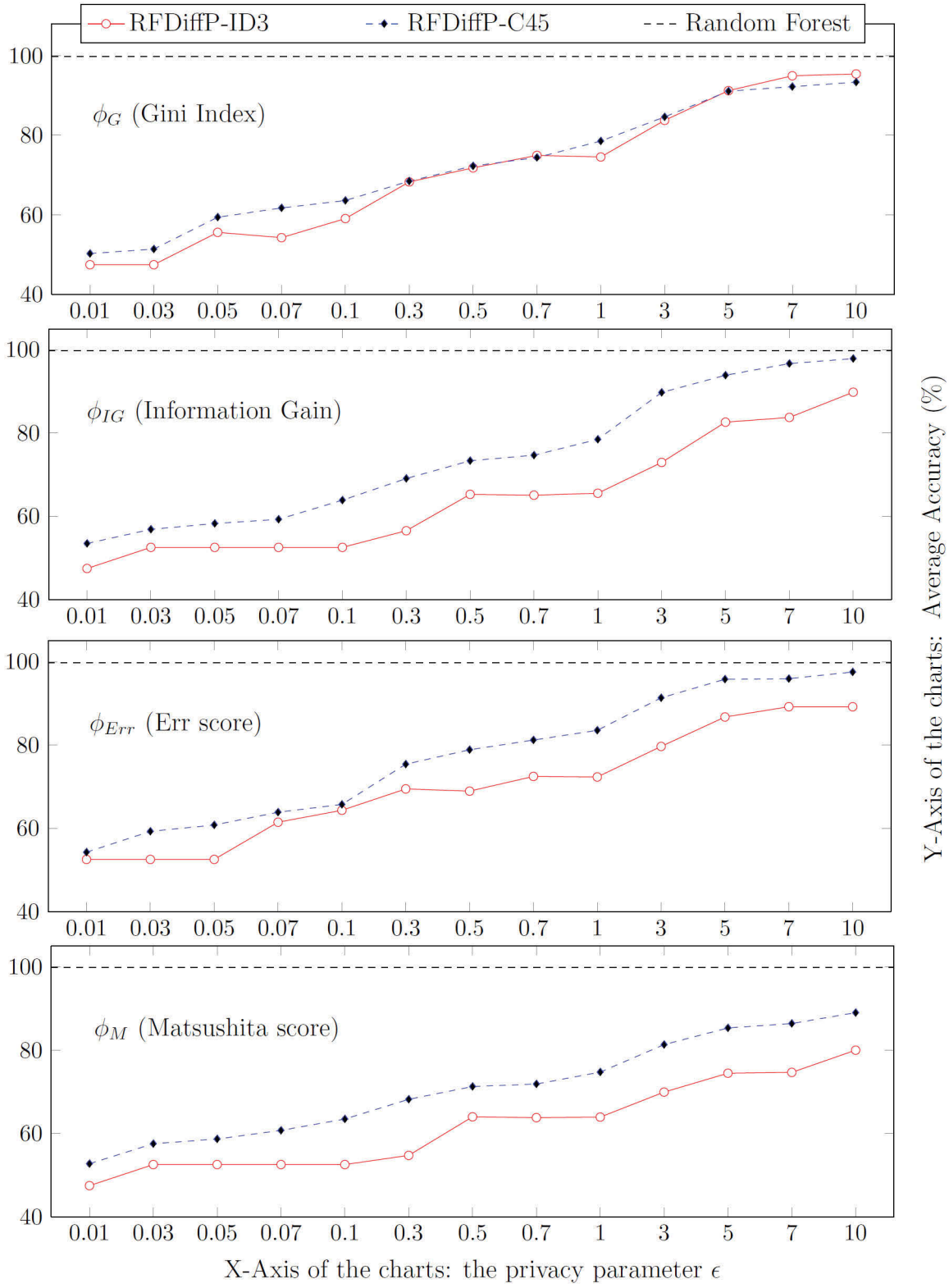

Figure 4.1: Average Accuracy of RFDiffP-ϕ on synthetic dataset, using four different scoring mechanisms and the number of trees in the forest is 10.

4.6.1 RFDiffP-ϕ results using synthetic dataset

In this section, RFDiffP-ϕ is run using synthetic datasets and the accuracy results of the algorithm with different splitting criteria are recorded.

The synthetic datasets in this section are generated with the method detailed in Section 3.5.4. The number of attribute values in the synthetic dataset is three uniform attributes, and the number of class values is two (binary class). The sample size is 15000 records. The trees to create the synthetic dataset are generated by splitting the nodes with randomly chosen attributes. From the third level of the trees, with a probability of 0.3, leaf nodes are generated and random class values are assigned to each leaf. All the 15000 records, sampling with uniform distribution, are classified using these trees and are used as the training sets. The test sets are generated independently to the training sets and contain 1500 samples.

Choosing between the C4.5 or the ID3 algorithm as the base classifier of the trees in the forest is important to the rest of the experiments in this chapter. Figure 4.1 shows the average accuracy of the RFDiffP-ϕ algorithm on four different scoring mechanisms from Table 3.1. It uses both ID3 and C4.5 as the base classifiers for the trees. Choosing C4.5 in RFDiffP-ϕ in Figure 4.1 shows better accuracy results from the forest, and because of this, the base classifier used for the rest of the experiments in this chapter is C4.5.

In addition to the base classifiers, the choice of splitting criteria in the experiments must also be considered. It is necessary to know how each of the four different splitting criteria from Table 3.1 performs with each base classifier. Between the forests that are created with the C4.5 classifiers shown in Figure 4.1, the experiments with Error Score and Information Gain show the highest accuracy results. The accuracy of the

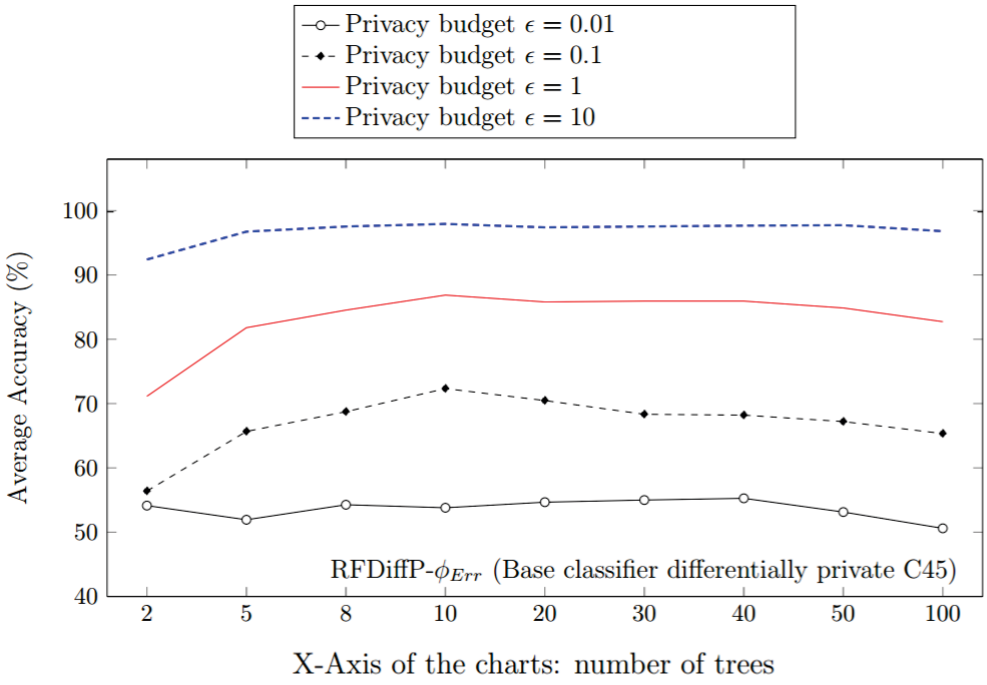

Figure 4.2: Average Accuracy of RFDiffP-ϕ_{Err} on synthetic dataset with a different number of trees.

forest classifiers from these splitting functions with larger amounts of privacy budget are very close to the optimal non-private version of them, which is shown by a black dashed line in this figure.

In another set of experiments in Figure 4.2, the RFDiffP-ϕ algorithm is run with a different number of trees and the accuracy results of the classifiers are compared. This experiment uses RFDiffP-ϕ with differentially private C4.5 trees as the base classifier and Error Score as the choice of ϕ (because this setting shows one of the best results in the previous experiments).

Four variations of the privacy budget, $\epsilon = \{0.01, 0.1, 1, 10\}$, are applied to a different number of trees, as shown in this figure. In differentially private decision forests, a large number of trees in the forest leads to allocating a lower privacy budget to each

one of them, and consequently, this means noisier results from each tree. This is the reason why the lines at the right corner of this figure – that have a higher numbers of trees in the forest – are dropping. This figure shows that the accuracy results of the forest with ten trees are high and very close to the best results that are observed from the RFDiffP-ϕ forests. Therefore, in the rest of the experiments with Algorithm 2 in this chapter, the number of trees in the forest is fixed to ten.

The aforementioned two experiments in this section show the accuracy of RFDiffP-ϕ on synthetic datasets. Based on these experiments, the number of trees in the forest is fixed to ten and C4.5 is chosen as the base classifier for the trees. These parameters are used in the experiments with the real-world datasets as detailed in the next section of this chapter.

4.6.2 RFDiffP-ϕ results using real-world datasets

In this section, RFDiffP-ϕ is evaluated on the same three real-world datasets as Section 3.6: Mushroom, Nursery and Adult from the UCI Machine Learning repository (Bache & Lichman, 2013). These three datasets are slightly modified and cleaned for these experiments: all the missing values are dropped and the continuous attributes, if any, are removed (refer to Section 3.6).

RFDiffP-ϕ in the experiment in this section creates ten forests of ten trees as classifier. The accuracy results in this experiment are averaged over ten runs of stratified 10-fold cross-validation on the forests (ten forests where each one contains ten trees). The standard deviations of the accuracy over 10-fold cross validation in the charts are reported by the error bars on the accuracy values.

Figure 4.3 compares the average accuracy of RFDiffP-ϕ with four different scoring

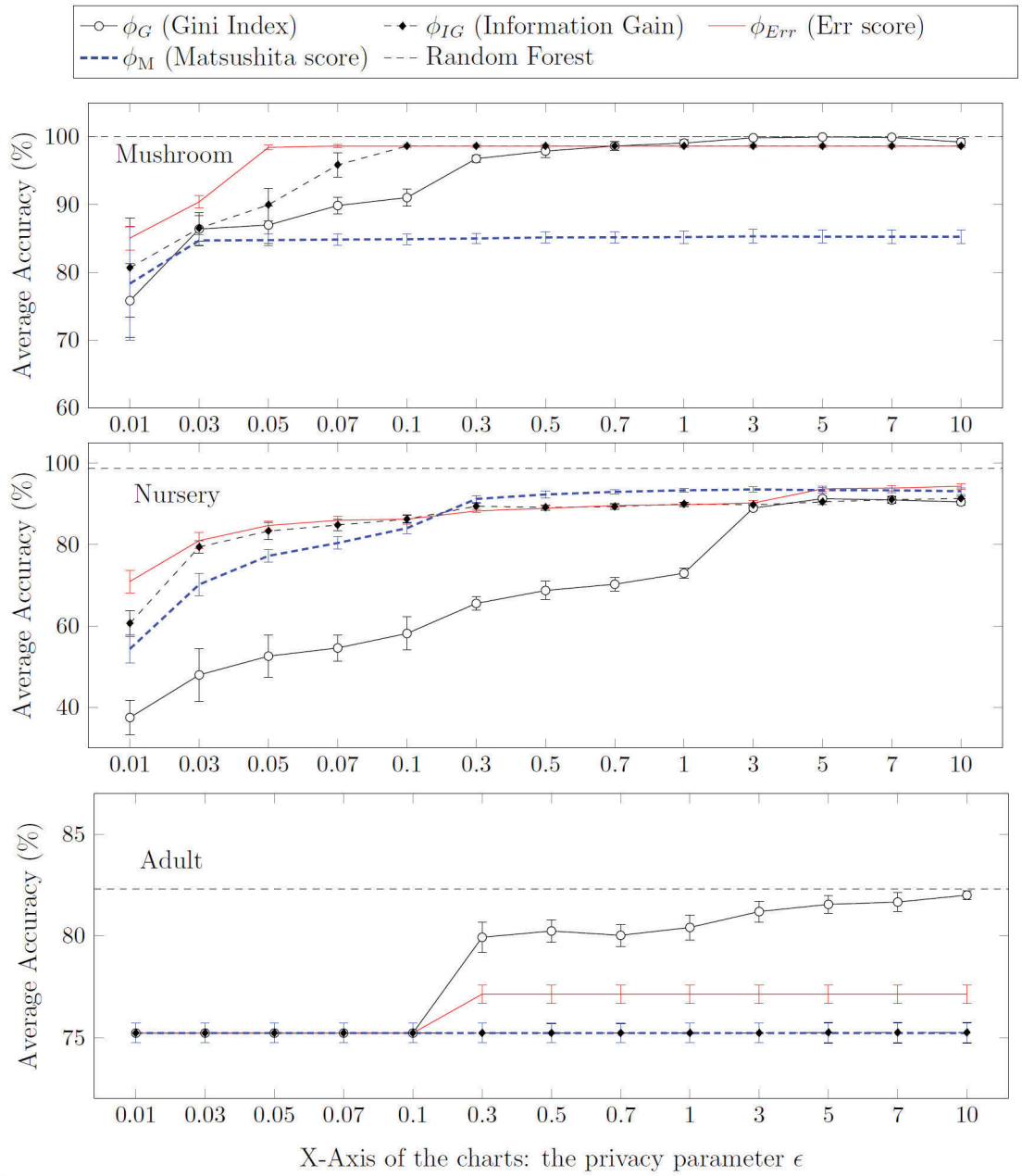

Figure 4.3: Average accuracy and standard deviation of RFDiffP-ϕ with four scoring functions.

functions as shown in Table 3.1. This experiment is similar to the experiment shown in Figure 3.3 of Chapter 3, and it shows that the accuracy of the classifiers built from the RFDiffP-ϕ algorithm is on average higher and more consistent than the ADiffP-ϕ trees. This is as expected because an ensemble classifier should normally lead to better results.

In Algorithm 2, the privacy budget is divided between the individual trees in the forest, so a small portion of the privacy budget is assigned to each tree. In this setting, the splitting functions that are more dependent on the value of the privacy budget suffer more from the low amount of privacy budget. This behavior is clear from the Adult dataset in the experiment. Consuming more privacy budget for each split causes the algorithm to quickly run out of budget and perform poorly, even when the initial budgets are increased.

None of the mentioned four scoring mechanisms in Table 3.1 performs considerably better than the others in this set of experiments. However, the accuracy results using the Misclassification Error score are the best for the Mushroom dataset and the results using the Information Gain are the best for the Adult dataset. Based on Table 3.1, the four choices of ϕ have different values and sensitivities: the sensitivity of Information Gain is $\log(m+1) + \frac{1}{\ln 2}$, Gini Index is $4m/(m+1)$, Error score sensitivity is 2 and Matsushita is $2\sqrt{m}$ (m is the number of records in the dataset). According to these sensitivity values, the success of Information Gain and Matsushita is strongly affected by the size of the datasets, and datasets with bigger sample sizes like Adult will increase the sensitivity values of these scores. Therefore, these classifiers need higher values of the privacy budget. The results shown in Figure 4.3 support this hypothesis: the Information Gain criterion on the Adult dataset with larger values of the privacy

budget provides higher accuracy results. On the other hand, the Misclassification Error score has a low sensitivity value, and therefore, as shown in this figure, works best on datasets of a smaller sample size, like Mushroom.

The accuracy of the graphs for the Matsushita score in this experiment is considerably higher than the experiments with the ADiffP-ϕ trees in Figure 3.3. Despite the high sensitivity of the Matsushita score, this experiment shows that this new differentially private classifier can be a good candidate on the ensemble classifiers, especially with smaller-sized sample sets.

The Gini index in this experiment shows comparatively good results in comparison with the other splitting criteria. It performs best with the Adult dataset and worst with the Nursery dataset. Looking more generally, in comparison with the results of the other three splitting criteria with single trees in figure 3.3, the Gini Index performs a lot better with the adaptive budget algorithm in the ensemble setting.

Generally speaking, the average accuracy results of the datasets with smaller sample sizes, like Mushroom and Nursery, are high and remain steady in this experiment, even when applying smaller privacy budgets. This shows that RFDiffP-ϕ is an excellent choice for highly sensitive datasets that require smaller amounts of the privacy budget to satisfy a tight guarantee in the privacy of the individuals.

4.6.3 Comparison of RFDiffP-ϕ with the PRDT algorithm

In this section, the RFDiffP-ϕ algorithm is compared to one of the most important differentially private random forests in the literature, Private Random Decision Tree (PRDT) (Jagannathan et al., 2012). The Adult, Nursery and Mushroom datasets were previously used in Jagannathan et al. and are considered to be good evaluation

datasets for the efficiency of PRDT. Therefore, RFDiffP-ϕ is compared to PRDT using these three datasets.

PRDT is an ensemble of random trees, randomly choosing the best attribute to split the node. When using random attribute choice on the internal nodes, privacy comes for free, and the mechanism does not spend any privacy budget on the splitting point of the trees. In return, all the privacy budget of the tree is used on the leaf nodes to predict the class labels by applying Laplace noise to obtain the noisy counts of the samples in the leaves. For the experiments in this section, the PRDT algorithm is implemented in the Weka data mining tool. The number of trees in the ensemble, as suggested by Jagannathan et al. (2012), is ten trees with a depth of five. The overall privacy budget of the forest is equally divided among these ten trees.

Gini index is well-recognized as one of the most important attribute splits for decision tree induction. In the experiments shown in Figure 4.3, ϕ_{G} shows a more consistent result using the three public datasets, and this is used as the choice of the splitting criterion in the experiment in this section.

Figure 4.4 compares the average accuracy and standard deviation of PRDT with RFDiffP-ϕ, using ϕ_{G}. A wide range of privacy budgets $\epsilon = \{0.01, 0.03, 0.05, 0.07, 0.1, 0.3, 0.5, 0.7, 1, 3, 5, 7, 10\}$ is used for this experiment. The bar charts show the average accuracy results of these two algorithms with 10-fold cross validation on ten forests (each forest contains10 trees). The small lines on top of the bar charts show the standard deviation for the average accuracy results of ten forests. The stars (*) on top of the bar charts indicate scenarios where RFDiffP-ϕ statistically significantly outperform PRDT, using a t-test, with $\alpha = 0.05$.

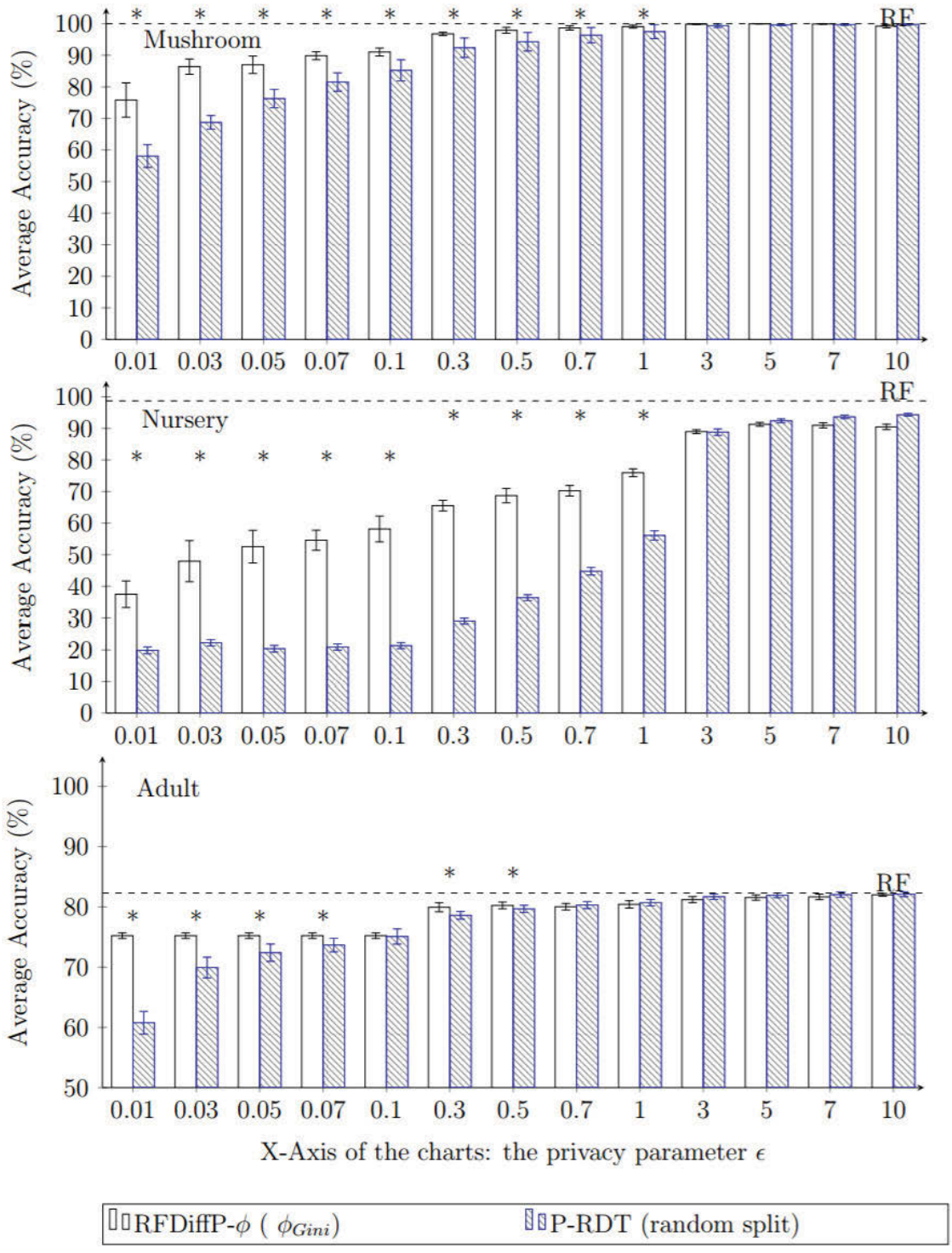

Figure 4.4: Comparison of the prediction accuracy of RFDiffP-ϕ and PRDT

The results from RFDiffP-ϕ using the Mushroom dataset outperform PRDTs results with small values for the privacy budget, and increasing the amount of ϵ improves the results. With $\epsilon > 3$, the accuracy results of RFDiffP-ϕachievethe optimum accuracy of the original non-private random forest of Breiman (2001), which is shown by the black dashed lines in Figure 4.4.

The accuracy results of the RFDiffP-ϕ algorithm on the Adult dataset are also promising. The Adult dataset is a large dataset with a binary class, and the accuracy of this algorithm using small privacy budget values is higher than PRDT, and with larger values for the privacy budget, the accuracy results approach the optimum non-private random forest results.

The Nursery dataset is relatively large with five class values. Figure 4.4 shows that the performance of the RFDiffP-ϕ algorithm is far better than that of PRDT when small values of the privacy budget are used on the Nursery dataset. However, with larger values of privacy budget, particularly when $\epsilon > 7$, although the RFDiffP-ϕ algorithm achieves very accurate results, it does not perform better than PRDT.

The experiments confirm that when ϵ is small, RFDiffP-ϕ tends to perform significantly better than PRDT. This is not the only advantage of RFDiffP-ϕ over the PRDT algorithm. The other advantage is the way the RFDiffP-ϕ algorithm selects the splitting criterion: PRDT belongs to the category of differentially private random forests that picks random attributes to split and use all the budget for noisy counts in the leaves of the trees. However, the RFDiffP-ϕ algorithm applies a greedy style of tree induction and relies on specific splitting criteria on each splitting point of the tree. The greedy style induction of RFDiffP-ϕ has the potential to further improve the results (e.g. by using more accurate splitting functions) than PRDT and other

models that rely on random style trees.

4.7 Discussion

Random forest, in this thesis, takes advantage of the ensemble paradigm and examines the performance of bagging by creating an ensemble of ADiffP-ϕ trees. This technique constructs a privacy-preserving classifier, called RFDiffP-ϕ, using ADiffP-ϕ trees. Each differentially private decision tree in the ensemble is trained independently and aggregating their results denotes the classifier's prediction.

Two sets of evaluation experiments are presented in this chapter: the first evaluates the performance of RFDiffP-ϕ on synthetic and also publicly available datasets and compares the choice of splitting criteria. Figure 4.3 shows that the average accuracy results of the smaller datasets, like Mushroom and Nursery, are high and robust over the parameter values, even when applying smaller values of privacy budgets. This shows that the RFDiffP-ϕ is an excellent choice for highly sensitive information that needs a small amount of privacy budget to strongly guarantee the privacy of the data.

The experiments on Chapter 3 of this thesis shows that in a non-private setting, the rate of convergence of Matsushita loss and Information Gain is better than Gini index and Error score. However, similar to what is observed in the previous chapter, the best splitting criteria regarding convergence rate in the differentially private settings do not necessarily show better accuracy results on the forest classifiers. The high sensitivity of Matsushita loss, ϕ_{M}, (shown in the right plot of Figure 3.1) can only be naturalized by consuming more privacy budget. Therefore, less budget remains for the rest of the tree induction and as a result, the accuracy of differentially private

forests classifiers is not satisfying. The same happens with the Information Gain criterion on forest classifiers.

The choice of Gini index in ϕ_{G} in an ensemble setting is more successful in comparison with the single trees in the previous chapter. This makes the Gini index a good candidate for use in RFDiffP-ϕ. The ϕ_{G} criterion shows the best results with bigger datasets like Adult, especially with higher values of the privacy budget.

The second set of experiments in this chapter compares RFDiffP-ϕ to one of the most important differentially private random forests in the literature, PRDT. The results show that the accuracy of RFDiffP-ϕ using the Mushroom dataset is better than PRDT with small values of the privacy budget and increasing the privacy budget improves the results. The results of RFDiffP-ϕ with the Adult dataset are also promising and it outperforms PRDT for all settings of the privacy budget.

A more significant improvement in the results of RFDiffP-ϕ can be observed using the Nursery dataset with small values for the privacy budget, which significantly outperforms PRDT. However, with larger values of the privacy budget on the Nursery dataset, PRDT performs slightly better than RFDiffP-ϕ. This behavior can be explained due to the multi-class labeling structure of the Nursery dataset (five labels for the class values). PRDT saves all the privacy budget for the leaf nodes. With a smaller privacy budget, the leaves of the classifiers do not receive enough of the budget to more accurately choose a better class value, and an accurate choice of splitting function in RFDiffP-ϕ results in better classifiers than PRDT. However, when the privacy budget is large, the budget on the leaves is large enough to perform very well in choosing between any of the five different class value labels in PRDT, and the results are slightly better than RFDiffP-ϕ.

4.8 Conclusion

In this chapter, a novel approach for a random forest algorithm under a differential privacy paradigm is presented to address Contribution 2 of this thesis, which is to propose *an algorithm to apply the budget allocation schema on a differentially private random forest*.

These random forest classifiers are built with greedy style induction of decision trees, where differential privacy is applied in both splitting points and leaves of the trees. The RFDiffP-ϕ algorithm is an ensemble of ADiffP-ϕ trees, where the characteristics of the data sample in each node of the tree are important to determine the *dynamic* allocation of the privacy budget to that node. The trees are created with dynamic heights and are automatically stopped when they meet the stopping conditions on the nodes. In the experiments using synthetically generated datasets, these classifiers show a good level of utility with all the different splitting criteria used. RFDiffP-ϕ is compared to the PRDT (Jagannathan et al., 2012) using three publicly available datasets. In the experiments with the Mushroom and Adult datasets, RFDiffP-ϕ outperforms PRDT for all values of the privacy budget. With the Nursery dataset, the RFDiffP-ϕ algorithm outperforms PRDTs results using smaller values of the privacy budget. Also, with larger values of the privacy budget for all three datasets in the experiments, the accuracy results of the RFDiffP-ϕ algorithm are very close to the non-private Breiman's (2001) Random forest.

Although some of the tree parameters, including the number of the trees in the forest, can be optimized (a more detailed explanation is given in Section 5.2 of Chapter 5), the accuracy results of the forests that are created with the RFDiffP-ϕ algorithm, especially on sensitive datasets, are high and promising.

Chapter 5

Conclusion

Recent and rapid improvements in different database sectors have persuaded more people to share their sensitive individual information with others and more companies to publish their corporate data publicly. Data mining this information has numerous usages for national and international planning and decision making. However, the privacy of the people being data mined must be strongly considered, and there is a need to find efficient data privacy mechanisms. Privacy-preserving techniques have been proposed to minimize the likelihood of recognizing an individuals private sensitive information in datasets which have been published or shared. Currently, the focus on privacy-preserving data publishing is on developing protocols for using and storing sensitive data. Researchers are looking to improve the procedure of converting the original data to a private version, and safely release modified but still useful versions of data with the public. Privacy breaches still occur, which suggests a strong need for a reliable and robust privacy model to limit the increasing risk of privacy breaches. Differential privacy focuses on how to disclose general non-specific data, but hide information about any specific participant; this is a paradox since by using

this mechanism, people can learn useful information about a population, while learning nothing about any individual in the dataset. The mechanism guarantees that the statistical results of two datasets, which are only different by a single participant, remain almost the same by adding a sufficient level of noise to the outputs of the computation. The level of noise in differential privacy is controlled by a parameter called the privacy budget, ϵ. Noise ought to be added to the number of queries proportionally to prevent the exposure of the individual's sensitive data, particularly in situations that allow multiple queries. However, adding too much noise can result in compromising the output of the system. Therefore, many studies are focusing on finding a trade-off between the privacy of individuals and the utility of the published results in differential privacy. A large body of research starting in 2006 has attempted to introduce the concept of differential privacy in a series of comprehensive studies (C. Dwork & Roth, 2014). However, in most of these works, the concept of the privacy budget and its value were incompletely determined and there are not many previous studies which focus directly on the privacy budget.

This thesis focuses on the allocation of the privacy budget on data mining techniques to have more control over its consumption. Decision trees and random forest algorithms are two data mining models that are used in this thesis because of their ability to track the consumption of the budget in the algorithmic structure of these techniques. When releasing statistical outputs from differentially private trees and differentially private forests, the accuracy of the results needs to be improved without over-consuming the privacy budget. The approaches in this thesis consider adaptive allocations of the budget on different process of tree induction, that have different needs for privacy/utility trade-off.

5.1 Addressing Research Contributions

To solve the budget allocation optimization problem in the induction of trees and forests, this thesis asked the following questions:

> **RQ1**: Is it possible to propose an appropriate algorithm to allocate the privacy budget adaptively on differentially private decision trees?
>
> A novel differentially private algorithm, namely ADiffP-ϕ, is proposed to answer this question in Chapter 3 of this thesis.
>
> **RQ2**: Is it possible to propose an adaptive budget allocation algorithm on differentially private random forests?
>
> This question is answered by proposing an algorithm, called RFDiffP-ϕ, in Chapter 4 of this thesis.

Based on the above research questions and the gaps in the knowledge, this thesis makes the three following contributions:

> **Contribution 1: An algorithm for adaptive privacy budget allocation on a differentially private decision tree.**
>
> Chapter 3 of this thesis proposes a budget allocation algorithm on differentially private decision tree inductions. The novel privacy budget allocation schema is called the ADiffP-ϕ algorithm. The basic premise of the budget allocation algorithm is to only spend as much as needed of the privacy budget for each differentially private query, thereby better controlling the privacy costs of the queries and the model. Analyzing the characteristics of the applied datasets and efficiently allocating different privacy budget values to different queries

with various sensitivities allows the algorithm to perform better than the previous models. The experiments in this chapter show that the ADiffP-ϕ algorithm generates more accurate results than state-of-the-art algorithms in decision tree induction. This algorithm also provides more consistent output by showing the accuracy results of the trees have lower standard deviation compared to the state-of-the-art models. However, increasing the accuracy of the classifiers and reducing the variance in the results are not the only good outcomes of this algorithm as it is also able to generate trees with *dynamic* heights which minimizes the impact of user error on the results due to assigning wrong parameters to the tree.

Contribution 2: An algorithm to apply the budget allocation schema on a differentially private random forest.

In Chapter 4 of this thesis, the greedy ADiffP-ϕ trees are used to build a differentially private forest, called RFDiffP-ϕ. This forest is proposed to overcome the general drawbacks of decision trees in practice. The RFDiffP-ϕ algorithm adopts the original Random Forest with bootstrap sampling from Breiman (2001). The adaptive allocation of the privacy budget in differentially private forest classifiers in RFDiffP-ϕ allows the algorithm to create more accurate forests than the previous benchmark study in the literature, especially when more sensitive data demands stronger privacy in the output (which means smaller values of privacy budgets).

Contribution 3: The proposal of Matsushita loss as a splitting criterion on differentially private decision trees.

The budget allocation algorithm closely interacts with all the process of tree inductions, including the internal nodes of the tree where splitting functions divide the data samples. On any internal node of the tree, based on the splitting criterion in use, the algorithm decides how much budget needs to be consumed to choose the best attribute to divide the data sample. The choice of splitting criteria is important because of their different functions and sensitivities in differential privacy. This thesis introduces a new splitting function in the differentially private setting, Matsushita loss (Kearns & Mansour, 1999; Nock & Nielsen, 2009). Matsushita loss, which is introduced in the non-private setting in Kearns and Mansour's algorithm, is an entropy function that is known to be optimal from the boosting standpoint (Kearns & Mansour, 1999), and has not previously received treatment in differential privacy. Chapter 3 proposes this splitting function and calculates its global sensitivity. This splitting criterion is used in the experiments in both Chapters 3 and 4 of this thesis.

Four splitting criteria are used in the experiments in Chapters 3 and 4 of this thesis, Information Gain (IG, also called binary entropy) in ID3 and C4.5 (Quinlan, 1993), Gini index in CART (Breiman et al., 1984), Misclassification Error score (Friedman & Schuster, 2010) and Matsushita loss. Applying differential privacy on these splitting criteria generalizes the original greedy differentially private trees and forests to all loss functions, and gives a better big picture about the splitting functions, precisely because the convergence rates of these splitting function affects both generalization and the consumption of the privacy budget. When the rate of convergence of a splitting criterion is higher then typically fewer iterations are needed to yield a useful

approximation of the splitting function behavior and results. A generalized observation is learned from the experiments in this thesis: the splitting functions that have a better rate of convergence are more expensive in differential privacy and need more privacy budget to obtain accurate results.

5.2 Limitations and possible future research directions

There are a couple of limitations and future research directions that can be pursued from this thesis and are detailed in this section.

5.2.1 Tuning more parameters

Apart from the parameters that are tuned throughout this thesis, tuning some other heuristic parameters in the two proposed algorithms has the potential to improve the quality of the work. All of these parameters have been carefully set based on the experiment results in the thesis, and the introduction of a theoretical approach to choose them is suggested in this section.

- The first heuristic parameter is the initial cut of the privacy budget that is assigned to the tree in ADiffP-ϕ and is tuned in Section 3.5.3. This cut is the minimum amount of privacy budget that is used for each iteration of the tree and is calculated based on Equation (3.13). This portion of the budget is chosen heuristically and based on the standard uniform convergence bounds to reduce the absolute difference between the empirical risk and true error. Although this privacy budget cut adaptively changes for different operations of

the tree according to the characteristics of the data sample, the initial budget adjustment can be optimized in the algorithm by a robust theory.

- Another parameter to be discussed in the ADiffP-ϕ algorithm is the assignment of the privacy budget portion in Line 7 of Algorithm 1. When the algorithm needs a large portion of the privacy budget to be able to return a highly accurate split, the existing budget for the iteration is multiplied by five in the algorithm. The value of five is assigned based on the results of the experiments on the algorithm, but a possible optimization of this value is left for further studies.

- The third parameter that can be theoretically adjusted in ADiffP-ϕ is the threshold value in Section 3.5.4. This value is defined based on the experiments in Figure 3.2. However, the preciseness of this threshold value may be seen as debatable due to the lack of a proof.

- The number of trees in the forest is an important parameter in differentially private forest algorithms. In RFDiffP-ϕ, this number is assigned heuristically based on the results of the experiments in Figure 4.2 in Chapter 4. After applying a different number of trees on the forest, the choice of ten trees seems reasonable for the experiments. Assigning the number of trees heuristically based on the experiments is a common practice in the literature. This number is not fixed in the previous greedy or random style differentially private forests in the literature. Greedy style forests applied fewer trees in the ensemble: for example Patil and Singh (2014) used twenty trees and Fletcher and Islam (2015a) used four trees. However, the forests that are built with random style trees used a wider range of tree numbers in their experiments, such as ten trees in Jagannathan

et al. (2012), and a hundred trees in Fletcher and Islam (2017). Number of trees is also likely to be dependent on the particular choice of datasets in the experiments. None of these previous studies suggests a theoretical method to assign the tree numbers to the differentially private forest and the choice of the datasets, and a solution for this problem still remains an open question.

5.2.2 Adapting the splitting criterion to the dataset

Characterizing the behaviour of the algorithm for different types of datasets and choosing different attribute selection methods is an interesting open problem. Although this thesis shows that these algorithms can generate more accurate results than state-of-the-art algorithms in differentially private decision trees and forests, the choice of splitting criteria on the greedy style trees requires some consideration. This choice has an enormous effect on the quality of the resulting classifiers. Some datasets perform much better with particular criteria and others preform worse. In differential privacy, in addition to how the splitting function reacts, their sensitivity also has an effect on the results. Different accuracy levels in the results of the four choices of splitting criteria on the same synthetic dataset in Figure 4.1 demonstrate this point. Therefore, one open problem in generating differentially private decision trees and forests relies on trying to adapt not just the local privacy budget to the data at hand, but also the splitting criterion – and therefore the loss function – to the data sample. Consequently finding a possible relationship between the characteristics of the data sample and the choice of the splitting criterion in differential privacy is an interesting topic to explore.

5.2.3 New splitting criteria in differentially private settings

This thesis started a new path of research by introducing a new splitting function for differentially private trees. There are not too many splitting functions proposed in the literature. The reason for this may be that it is challenging to calculate the sensitivity of some non-private splitting functions or they may be not easily adopted to the differential privacy framework. For example, Friedman and Schuster (2010) showed that it is not possible to calculate the sensitivity of the Gain ratio in a differentially private setting. This thesis demonstrates that there could be more functions to split the data samples in differential privacy which would be an interesting line of research.

5.2.4 Comparison of RFDiffP-ϕ with other algorithms

Another path to continue this research is to compare the RFDiffP-ϕ algorithm with some other differentially private random forests in the literature. In Chapter 4, this algorithm is compared with the PRDT algorithm and shows promising results. PRDT has been often used to evaluate most of the previous differentially private decision tree and random forest algorithms and sets a high benchmark for evaluation criteria. The fact that applying random style trees, instead of greedy style trees, in a differentially private forest, shows promising results in the experiments and makes PRDT an excellent candidate for comparing with RFDiffP-ϕ. However, there are several recent studies on differentially private random forests that might be good candidates for comparison as well, including the greedy style trees in Fletcher and Islam (2015a) or the forests that are built with randomly assigned trees including Fletcher and Islam (2015b) and (2017). Comparing the results of RFDiffP-ϕ to these models has remained future work to this thesis. Rana et al.'s algorithm (2015) presents a

weakened version of privacy settings. However, for highly sensitive data, which is the primary target of this thesis, the weakened version of differential privacy is not recommended.

5.2.5 Applying the method on other data mining techniques

ADiffP-ϕ and RFDiffP-ϕ can be considered as a prototype for budget allocation optimization in differential privacy. Applying the budget allocation schema on other data mining algorithms, apart from decision trees and random forest, is another possible path for research. The mechanism works based on the assessment of the data sample and determines which queries are more or less sensitive to the adjustment of the privacy budget expenditure. This concept can be applied more broadly, to adaptively adjust the privacy budget on other data mining algorithms.

5.2.6 Increasing number of trees in the Differentially Private Forest with new methods

Experiments in Chapter 4 used ten trees in RFDiffP-ϕ algorithm to shape differentially private forests. As we mentioned in the last paragraph of section 5.2.1, there is no theoretical method in the literature to assign a tree numbers to the differentially private forest. However, randomness in the trees work better by increasing the number of trees in the forests. Some of the new differentially private forests implements higher number of trees in their forests, including 500+ trees in Rana et al. (2015) and 100 trees in Fletcher and Islam (2017). Increasing the number of trees in RFDiffP-ϕ needs to be applied by a change in the design of the algorithm. In the current design the budget divides between the individual trees in the forest and the algorithm

calculates the budget for each bootstrap sample based on the overall privacy budget assigned to the forest and the number of bootstrap samples. When the bootstrap sampling is used, a record may be or may not be in the sample size. So the privacy is partially preserved. Therefore, either there is no need to add Laplacian noise, or the noise can be decreased dramatically. This method can preserve overall privacy budget in the tree and save it to add more trees in the forest.

5.2.7 Applying the algorithms on more datasets from different areas

The datasets for this thesis are selected from publicly available sources and no consideration for subject sensitive datasets is applied in this thesis. Some areas, including health data, have specific needs and treatments. Dealing with the consequences of working with these kinds of datasets in differential privacy is outside the scope of this thesis and left as future work.

5.3 Closing Note

To conclude, privacy budget allocation optimization on data mining techniques is a significant area of research to pursue. This thesis develops two novel algorithms to address the efficient allocation of the privacy budget on differentially private decision trees and random forests. Matsushita loss, a new splitting criterion in differentially private setting, is also introduced for differentially private trees and forests in this thesis. Experiment evaluations in this thesis confirm that by adding some extra computational costs to how to spend budget on both single trees and forests, the

adaptive differentially private budget allocation algorithms manage the privacy costs of the model more efficiently and achieve a better trade-off between accuracy and privacy in the outputs. These algorithms result in more accurate outputs in comparison to the previous studies in this area, especially on sensitive private datasets, where the smaller setting of ϵ provides a better guarantee for privacy in the output.

Chapter 6

Appendix

We acknowledge the contribution of Richard Nock from Data61/CSIRO in providing the proof of theorem and lemmas in this Appendix.

6.1 Sketch of proof of Theorem 4

We first show that $\Delta_\phi \leq \check{\phi}(1, m+1)$. We then show that the bound is realized. We do not assume that ϕ is differentiable.

The proof is split in three parts. The two first being the following two Lemmata.

Lemma 9. *$\check{\phi}(1, x)$ is non-decreasing over $\mathbb{R}_{+*}$.*

Proof. We know that $-\phi$ is convex and therefore $D_{-\phi}(a\|b)$ is non negative, $D_{-\phi}$ being the Bregman divergence with generator $-\phi$ (Nock & Nielsen, 2009). We obtain, with $a = 0, b = 1/x$,

$$D_{-\phi}(a\|b) = \phi\left(\frac{1}{x}\right) - \phi(0) - \left(0 - \frac{1}{x}\right) \cdot (-\phi)' \quad \geq \quad 0 \ ,$$

where $\phi' \in \partial\phi(1/x)$, ∂ denoting the subdifferential. Simplifying yields

$$\phi\left(\frac{1}{x}\right) - \frac{1}{x} \cdot \phi' \quad \geq \quad 0 \ . \tag{6.1}$$

We then remark that

$$\partial\check{\phi}(1,x) \;=\; \left\{\phi\left(\frac{1}{x}\right) - \frac{1}{x}\cdot\phi' \;, \phi' \in \partial\phi\left(\frac{1}{x}\right)\right\} \;. \tag{6.2}$$

Therefore, $\check{\phi}(1,x)$ is non-decreasing when $x \neq 0$, and we obtain the statement of Lemma 9. □

Lemma 10. *The following holds true:*

(A) ϕ is not decreasing (resp. not increasing) over $[0,1/2]$ (resp. $[1/2,1]$);

(B) for any $0 \leq p \leq q \leq 1/2$, or any $1/2 \leq q \leq p \leq 1$, we have

$$0 \leq \phi(q) - \phi(p) \leq \phi(|q-p|) \;. \tag{6.3}$$

Proof. A fact that we will use repeatedly hereafter is the fact that a concave function sits above all its chords. We first prove (A): if ϕ were decreasing somewhere on $[0,1/2]$, there would be some $a \in [0,1/2]$ such that $\phi(a) > 1 = \phi(1/2)$, and therefore because ϕ is symmetric, $\phi(1-a) = \phi(a) > 1 = \phi(1/2)$, so point $(1/2, \phi(1/2))$ would sit under the chord $[(a,\phi(a)),(1-a,\phi(1-a))]$ and ϕ cannot be concave, a contradiction. Notice that $\phi(a) > 1 = \phi(1/2)$ can in fact not be $\phi(a) > b$ for some $b < 1$ as otherwise ϕ would not be concave in $[0,1/2]$. The same reasoning holds for the interval $[1/2,1]$ because of the symmetry of ϕ.

We now prove (B). We prove it for the case $0 \leq p \leq q \leq 1/2$, the other following from the symmetry of ϕ. Non-negativity follows from (A) and the fact that $\phi(0) = 0$. The right inequality follows from the concavity of ϕ: indeed, since $\phi(0) = 0$, this inequality is equivalent to proving

$$\frac{\phi(q) - \phi(p)}{q-p} \leq \frac{\phi(q-p) - \phi(0)}{q-p-0} \;, \tag{6.4}$$

which since $p \geq 0$, is just stating that slopes of chords that intersect ϕ at points of constant difference between abscissae do not increase, *i.e.* ϕ is concave. We obtain the statement of Lemma 10. □

We now distinguish two high level cases.

⋆ Suppose $m \geq 2$. Let m_ℓ and m_ℓ^+ (resp. m'_ℓ and $m_\ell'^+$) be the local cardinals for $\mathcal{S}$ (resp. $\mathcal{S}'$). Let us fix for short

$$\Delta \doteq \Delta_\phi = |f_\phi(h, \ell, \mathcal{S}') - f_\phi(h, \ell, \mathcal{S})| \ .$$

Let us define the following events:

$$\begin{aligned} A_1 \doteq "m'_\ell = m_\ell" \quad ; \quad A_2 \doteq "m'_\ell = m_\ell + 1" \\ B_1 \doteq "m_\ell'^+ = m_\ell^+ + 1" \quad ; \quad B_2 \doteq "m_\ell'^+ = m_\ell^+" \\ C_1 \doteq "m_\ell^+ + 1 \leq m_\ell/2" \quad ; \quad C_2 \doteq "m_\ell^+ \geq m_\ell/2" \\ C_3 \doteq "m_\ell^+ \geq (m_\ell + 1)/2" \qquad . \end{aligned}$$

The proof consists in analyzing combinations (logical-and) of events (at least one for each $A_\cdot, B_\cdot, C_\cdot$). Because the events combinatorics is lengthy, we give just one case as an example.

↪ Case $A_1 \wedge B_1 \wedge \neg C_1 \wedge \neg C_2$. In this case, we have $m_\ell^+ < m_\ell/2$ and $m_\ell^+ + 1 > m_\ell/2$, implying m_ℓ is odd and

$$m_\ell^+ = \frac{m_\ell - 1}{2} \ . \tag{6.5}$$

Assuming without loss of generality that $\phi(m_\ell^+/m_\ell) \leq \phi((m_\ell^+ + 1)/m_\ell)$, we obtain

this time

$$\begin{aligned}
\Delta &= m_\ell \cdot \left(\phi\left(\frac{m_\ell^+ + 1}{m_\ell} \right) - \phi\left(\frac{m_\ell^+}{m_\ell} \right) \right) \\
&\leq m_\ell \cdot \left(\phi\left(\frac{1}{2} \right) - \phi\left(\frac{m_\ell^+}{m_\ell} \right) \right) \\
&= m_\ell \cdot \left(\phi\left(\frac{1}{2} \right) - \phi\left(\frac{1}{2} - \frac{1}{2m_\ell} \right) \right) \\
&\leq m_\ell \cdot \left(\frac{2}{m_\ell} \right) \qquad (6.6) \\
&= 2 = \check{\phi}(1,2) \\
&\leq \check{\phi}(1,m) \ . \qquad (6.7)
\end{aligned}$$

Ineq. (6.6) holds because

$$\phi(x) \ \geq \ 2x \ , \forall x \in [0, 1/2] \ , \qquad (6.8)$$

since $y = 2x$ is a chord for ϕ over $[0, 1/2]$ and ϕ is concave (it therefore sits over its chords). We get for any $x \geq 2$,

$$\begin{aligned}
\phi\left(\frac{1}{2} \right) - \phi\left(\frac{1}{2} - \frac{1}{x} \right) &= 1 - \phi\left(\frac{1}{2} - \frac{1}{x} \right) \\
&\leq 1 - 2 \cdot \left(\frac{1}{2} - \frac{1}{x} \right) \\
&= \frac{2}{x} \ , \qquad (6.9)
\end{aligned}$$

from which we obtain ineq. (6.6) with $x = 2m_\ell$ (since $m_\ell \geq 1$). Ineq. (6.7) holds because $m \geq 2$ and Lemma 9.

$\star$ To finish up, suppose $m = 1$. In this case, since $0 \leq m_\ell^+ \leq m_\ell \leq m$, $\phi(m_\ell^+/m_\ell) = 0$, so we get $\Delta = \phi(m_\ell'^+/m_\ell') \in \{\phi(0), \phi(1/2), \phi(1)\} = \{0, 1\}$, so $\Delta \leq 2 = \check{\phi}(1,2) = \check{\phi}(1, m+1)$. This finishes the proof that $\Delta_\phi \leq \check{\phi}(1, m+1)$ in Theorem 4.

to prove that the bound is realized, consider $m = 1$, $m_\ell = 0$, $m' = 2$, $m'_\ell = 1$. In this case,

$$\Delta = 2 \cdot \phi\left(\frac{1}{2}\right) - 1 \cdot \phi\left(\frac{0}{1}\right) = 2 \cdot \phi\left(\frac{1}{2}\right) = \check{\phi}(1, m+1) \ , \tag{6.10}$$

as claimed.

6.2 Proof of Lemma 5

We have

$$\check{\phi}_{\mathrm{M}}(1, m+1) = (m+1) \cdot 2\sqrt{\frac{1}{m+1} \cdot \frac{m}{m+1}} = 2\sqrt{m} \ , \tag{6.11}$$

as claimed.

We have

$$\begin{aligned} &\check{\phi}_{\mathrm{IG}}(1, m+1) \\ &= (m+1) \cdot \left(-\frac{1}{m+1} \log \frac{1}{m+1} - \frac{m}{m+1} \log \frac{m}{m+1}\right) \\ &= \log(m+1) + m \log \frac{m+1}{m} \\ &\leq \log(m+1) + \frac{1}{\ln 2} \ . \end{aligned} \tag{6.12}$$

The last inequality follows from (Friedman & Schuster, 2010, Claim 1).

We have

$$\check{\phi}_{\mathrm{G}}(1, m+1) = (m+1) \cdot \frac{4}{m+1} \cdot \frac{m}{m+1} = \frac{4m}{m+1} \ , \tag{6.13}$$

as claimed.

Finally, we have

$$\check{\phi}_{\mathrm{Err}}(1, m+1) = (m+1) \cdot 2 \min\left\{\frac{1}{m+1}, \frac{m}{m+1}\right\} = 2 \ , \tag{6.14}$$

as claimed.

6.3 Proof of Lemma 6

We perform a Taylor expansion of ϕ up to second order and obtain:

$$\phi(0) = \underbrace{\phi\left(\frac{1}{x}\right) + \left(0 - \frac{1}{x}\right) \cdot \phi'\left(\frac{1}{x}\right)}_{\doteq J} + \left(0 - \frac{1}{x}\right)^2 \cdot \phi''(a) \ ,$$

for some $a \in [0, x]$. There remains to see that $J = \check{\phi}'(1, x)$ (eq. (6.2)), fix $x = m + 1$ and reorder given $\phi(0) = 0$ since ϕ is permissible.

Bibliography

Abernethy, J. D., & Frongillo, R. M. (2012). A characterization of scoring rules for linear properties. In *Conference on Learning Theory* (pp. 27–1).

Aggarwal, C. C., & Philip, S. Y. (2008). *Privacy-preserving data mining: models and algorithms*. Springer Science & Business Media.

Agrawal, R., & Srikant, R. (2000). Privacy-preserving data mining. In *ACM Sigmod Record* (Vol. 29, pp. 439–450).

Anupam Gupta, F. M. A. R., Katrina Ligett, & Talwar, K. (2010). Differentially private combinatorial optimization. *In Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics*, 1106–1125.

Bache, K., & Lichman, M. (2013). *UCI Machine Learning Repository*.

Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., & Talwar, K. (2007). Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proceedings of the 26th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (pp. 273–282).

Barros, R. C., de Carvalho, A. C., & Freitas, A. A. (2015). *Automatic design of decision-tree induction algorithms*. Springer.

Blum, A., Dwork, C., McSherry, F., & Nissim, K. (2005). Practical Privacy: The

SuLQ Framework. In *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (pp. 128–138).

Blum, A., Ligett, K., & Roth, A. (2013). A learning theory approach to non-interactive database privacy. *Journal of the ACM (JACM)*, *60*(2), 12.

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

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123–140.

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

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). Classification and regression trees. *Wadsworth Int Group*, *37*(15), 237–251.

Casey, R., & Nagy, G. (1984). Decision tree design using a probabilistic model (corresp.). *IEEE Transactions on Information Theory*, *30*(1), 93–99.

Chaudhuri, K., Monteleoni, C., & Sarwate, A. (2011). Differentially private empirical risk minimization. *Journal of Machine Learning Research*, *12*(Mar), 1069–1109.

Chaudhuri, K., Sarwate, A., & Sinha, K. (2013). Near-optimal algorithms for differentially-private principal components. *The Journal of Machine Learning Research*, *14*(1), 2905–2943.

Chen, R., Mohammed, N., Fung, B. C., Desai, B. C., & Xiong, L. (2011). Publishing set-valued data via differential privacy. *Proceedings of the VLDB Endowment*, *4*(11), 1087–1098.

Clifton, C., & Tassa, T. (2013). On syntactic anonymity and differential privacy. In *IEEE 29th International Conference on Data Engineering Workshops (ICDEW)*

(pp. 88–93).

Dankar, F. K., & El Emam, K. (2012). The application of differential privacy to health data. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops* (pp. 158–166).

Darakhshan Mir, A. N., S Muthukrishnan, & Wright, R. N. (2011). Pan-private algorithms via statistics on sketches. In *Proceedings of the 13th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (pp. 37–48).

Devroye, L., Györfi, L., & Lugosi, G. (2013). *A Probabilistic Theory of Pattern Recognition* (Vol. 31). Springer Science & Business Media.

Dietterich, T. G., Kearns, M. J., & Mansour, Y. (1996). Applying the weak learning framework to understand and improve C4.5. In *ICML* (pp. 96–104).

Dinur, I., & Nissim, K. (2003). Revealing information while preserving privacy. In *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (pp. 202–210). ACM.

Domingos, P., & Hulten, G. (2000). Mining high-speed data streams. In *KDD* (Vol. 2, p. 4).

Dwork, C. (2007). Ask a better question, get a better answer a new approach to private data analysis. In *International Conference on Database Theory* (pp. 18–27).

Dwork, C. (2008). Differential privacy: A survey of results. *International Conference on Theory and Applications of Models of Computation*, 1–19.

Dwork, C. (2010). Differential privacy in new settings. *Proceedings of the 21st annual ACM-SIAM symposium on Discrete Algorithms*, 174–183.

Dwork, C. (2011a). Differential privacy. *Encyclopedia of Cryptography and Security*,

338–340.

Dwork, C. (2011b). A firm foundation for private data analysis. *Communications of the ACM*, *54*(1), 86–95.

Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference* (pp. 265–284). Springer, Berlin, Heidelberg.

Dwork, C., Naor, M., Pitassi, T., Rothblum, G., & Yekhanin, S. (2010). Pan-private streaming algorithms. *ICS*, 66–80.

Dwork, C., Naor, M., Pitassi, T., & Rothblum, G. N. (2010). Differential privacy under continual observation. *Proceedings of the 42nd ACM symposium on Theory of Computing*, 715–724.

Dwork, C., Nikolov, A., & Talwar, K. (2014). Using convex relaxations for efficiently and privately releasing marginals. *In Annual Symposium on Computational Geometry, ACM*, 261.

Dwork, C., & Nissim, K. (2004). Privacy-preserving datamining on vertically partitioned databases. In *Annual International Cryptology Conference* (pp. 528–544). Springer.

Dwork, C., & Pottenger, R. (2013). Toward practicing privacy. *Journal of the American Medical Informatics Association*, *20*(1), 102–108.

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

Dwork, C., & Smith., A. (2009). Differential privacy for statistics: What we know and what we want to learn. *Journal of Privacy and Confidentiality*, *1*(2), 135–154.

Dwork, G. N. R., Cynthia, & Vadhan, S. (2010). Boosting and differential privacy.

In *Proceedings of 51st annual symposium on foundations of computer science* (pp. 51–60). IEEE.

Erlingsson, Ú., Pihur, V., & Korolova, A. (2014). Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security* (pp. 1054–1067).

Fan, Wang, H., Yu, P., & Ma, S. (2003). Is random model better? on its accuracy and efficiency. In *Third IEEE International Conference on Data Mining, ICDM* (pp. 51–58).

Fan, L., & Xiong, L. (2012). Real-time aggregate monitoring with differential privacy. In *Proceedings of the 21st ACM International conference on Information and knowledge Management.*

Fienberg, S. E., Rinaldo, A., & Yang, X. (2010). Differential privacy and the risk-utility tradeoff for multi-dimensional contingency tables. In *International conference on privacy in statistical databases* (pp. 187–199).

Fletcher, S., & Islam, M. Z. (2015a). A differentially private decision forest. In *Proceedings of the 13th Australasian Data Mining Conference, Sydney, Australia* (pp. 1–10).

Fletcher, S., & Islam, M. Z. (2015b). A differentially private random decision forest using reliable signal-to-noise ratios. In *Australasian joint conference on artificial intelligence* (pp. 192–203). Springer, Cham.

Fletcher, S., & Islam, M. Z. (2016). Decision tree classification with differential privacy: A survey. *arXiv preprint arXiv:1611.01919*.

Fletcher, S., & Islam, M. Z. (2017). Differentially private random decision forests using smooth sensitivity. In *Expert Systems with Applications* (Vol. 78, pp.

16–31). Elsevier.

Friedman, A., & Schuster, A. (2010). Data mining with differential privacy. In *Proceedings of the 16th ACM SIGKDD International conference on Knowledge Discovery and Data mining* (pp. 493–502). ACM.

Gelfand, S. B., Ravishankar, C., & Delp, E. J. (1989). An iterative growing and pruning algorithm for classification tree design. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics* (pp. 818–823).

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

Ghosh, A., & Roth, A. (2015). Selling privacy at auction. *Games and Economic Behavior*, *91*, 334-346.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Hardt, M., Ligett, K., & McSherry, F. (2012). A Simple and Practical Algorithm for Differentially Private Data Release. In *Advances in Neural Information Processing Systems* (pp. 2348–2356).

Hardt, M., & Roth, A. (2013). Beyond worst-case analysis in private singular vector computation. In *Proceedings of the 45th annual ACM symposium on Theory of Computing* (pp. 331–340).

Hardt, M., & Talwar, K. (2010). On the geometry of differential privacy. In *Proceedings of the 42nd ACM symposium on Theory of computing* (pp. 705–714). ACM.

Hartmann, C., Varshney, P., Mehrotra, K., & Gerberich, C. (1982). Application of

information theory to the construction of efficient decision trees. *IEEE Transactions on Information Theory*, *28*(4), 565–577.

Helmbold, D. P., & Schapire, R. E. (1997). Predicting nearly as well as the best pruning of a decision tree. *Machine Learning*, *27*(1), 51–68.

Hsu, J., Gaboardi, M., Haeberlen, A., Khanna, S., Narayan, A., Pierce, B. C., & Roth, A. (2014). Differential privacy: An economic method for choosing epsilon. In *IEEE 27th Computer Security Foundations Symposium* (pp. 398–410).

Hunt, E. B., Marin, J., & Stone, P. J. (1966). *Experiments in induction*. Academic press.

Huysmans, J., Dejaeger, K., Mues, C., Vanthienen, J., & Baesens, B. (2011). An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, *51*(1), 141–154.

Islam, Z., & Giggins, H. (2011). Knowledge discovery through SysFor: a systematically developed forest of multiple decision trees. In *Proceedings of the Ninth Australasian Data Mining Conference* (Vol. 121, pp. 195–204).

Jagannathan, G., Monteleoni, C., & Pillaipakkamnatt, K. (2013). A semi-supervised learning approach to differential privacy. In *IEEE 13th International Conference on Data Mining Workshops (ICDMW)* (pp. 841–848).

Jagannathan, G., Pillaipakkamnatt, K., & Wright, R.-N. (2012). A Practical Differentially Private Random Decision Tree Classifier. *Transactions on Data Privacy*, *5*(1), 273–295.

Jain, P., & Thakurta., A. (2013). Differentially private learning with kernels. *In Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 118–126.

Jeremiah Blocki, A. D., Avrim Blum, & Sheet, O. (2013). Differentially private data analysis of social networks via restricted sensitivity. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science, ACM* (pp. 87–96).

Ji, Z., Lipton, Z.-C., & Elkan, C. (2014). Differential privacy and machine learning: a survey and review. *CoRR*, *abs/1412.7584*.

Justin Thaler, J. U., & Vadhan, S. (2012). Faster algorithms for privately releasing marginals. *In Automata, Languages, and Programming, Springer*, 810–821.

Kapralov, M., & Talwar., K. (2013). On differentially private low rank approximation. In *Proceedings of the 24th annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics* (Vol. 5, p. 1).

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., & Smith, A. (2011). What can we learn privately? *SIAM Journal on Computing*, *40*(3), 793–826.

Kearns, M., & Mansour, Y. (1998). A Fast, Bottom-up Decision Tree Pruning algorithm with Near-Optimal generalization. In *International conference on machine learning (icml)* (Vol. 98, pp. 269–277).

Kearns, M., & Mansour, Y. (1999). On the boosting ability of top-down decision tree learning algorithms. *Journal of Computer and System Sciences*, *58*(1), 109–128.

Kobbi Nissim, R. S., & Tennenholtz, M. (2012). Approximately optimal mechanism design via differential privacy. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ACM* (pp. 203–213).

Le, N. J., & Pappas, G. (2012). Differentially private filtering. *IEEE Transactions*

on Automatic Control (CDC), *59*(2), 341–354.

Letham, B., Rudin, C., McCormick, T. H., Madigan, D., et al. (2015). Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, *9*(3), 1350–1371.

Li, N., Li, T., & Venkatasubramanian, S. (2007). t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE 23rd International Conference on Data Engineering* (pp. 106–115).

Li, N., Qardaji, W. H., & Su, D. (2011). Provably private data anonymization: Or, k-anonymity meets differential privacy. *Arxiv preprint, CoRR, abs/1101.2604*, *49*, 55.

Lior, R. (2014). *Data mining with decision trees: theory and applications* (Vol. 81). World scientific.

Loukides, G., Liagouris, J., Gkoulalas-Divanis, A., & Terrovitis, M. (2014). Disassociation for electronic health record privacy. *Journal of biomedical informatics*, *50*, 46–61.

Machanavajjhala, A., Kifer, D., Abowd, J., Gehrke, J., & Vilhuber, L. (2008). Privacy: Theory meets practice on the map. In *Proceedings of the 24th IEEE International Conference on Data Engineering* (pp. 277–286).

Machanavajjhala, A., Kifer, D., Gehrke, J., & Venkitasubramaniam, M. (2007). l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *1*(1), 3.

Mansour, Y. (1997). Pessimistic decision tree pruning based on tree size. In *Machine learning-international workshop then conferences* (pp. 195–201). Morgan Kaufmann Publishers, INC.

Maréchal, P. (2005a). On a functional operation generating convex functions, part I: duality. *Journal of Optimization Theory and Applications*, *126*, 175–189.

Maréchal, P. (2005b). On a functional operation generating convex functions, part II: algebraic properties. *Journal of Optimization Theory and Applications*, *126*, 357–366.

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

McSherry, F., & Talwar, K. (2007). Mechanism design via differential privacy. In *FOCS* (Vol. 7, pp. 94–103).

Mohammed, N., Chen, R., Fung, B.-C.-M., & Yu, P.-S. (2011). Differentially private data release for data mining. In *Proceedings of the 17th ACM SIGKDD International conference on Knowledge discovery and data mining* (pp. 493–501).

Mohammed, N., Jiang, X., Chen, R., Fung, B. C., & Ohno-Machado, L. (2012). Privacy-preserving heterogeneous health data sharing. *Journal of the American Medical Informatics Association*, *20*(3), 462–469.

Muralidhar, K., & Sarathy, R. (2010). Does differential privacy protect terry gross privacy? In *Privacy in statistical databases* (pp. 200–209).

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Narayanan, A., & Shmatikov, V. (2008). Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy* (pp. 111–125).

Nergiz, M. E., Atzori, M., & Clifton, C. (2007). Hiding the presence of individuals from shared databases. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data* (pp. 665–676).

Nissim, K., Raskhodnikova, S., & Smith, A. (2007). Smooth sensitivity and sampling in private data analysis. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing* (pp. 75–84). ACM.

Niyogi, S.-S. . W. S., P. (2008). Finding the homology of submanifolds with high confidence from random samples. *Discrete and Computational Geometry*, *39*(1-3), 419–441.

Nock, R., & Nielsen, F. (2004). On Domain-Partitioning Induction Criteria: Worst-case Bounds for the Worst-case Based. *Theoretical Computer Science*, *321*(2-3), 371–382.

Nock, R., & Nielsen, F. (2008). On the efficient minimization of classification-calibrated surrogates. In *Advances in neural information processing systems* (pp. 1201–1208).

Nock, R., & Nielsen, F. (2009). Bregman divergences and surrogates for learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*, 2048–2059.

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

Poulis, G., Loukides, G., Skiadopoulos, S., & Gkoulalas-Divanis, A. (2017). Anonymizing datasets with demographics and diagnosis codes in the presence of utility constraints. *Journal of biomedical informatics*, *65*, 76–96.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, *1*(1), 81–106.

Quinlan, J. R. (1993). *C4.5 : programs for machine learning*. Morgan Kaufmann.

Quinlan, J. R. (1996). Bagging, boosting and c4.5. In *AAAI/IAAI* (Vol. 1, pp.

725–730).

Quinlan, J. R. (2014). *C4.5: programs for machine learning*. Elsevier.

Rana, S., Gupta, S. K., & Venkatesh, S. (2015). Differentially private random forest with high utility. *In IEEE International Conference on Data Mining*, 955-960.

Rastogi, V., Hay, M., Miklau, G., & Suciu, D. (2009). Relationship privacy: Output perturbation for queries with joins. In *Proceedings of the 28th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS)* (pp. 107–116).

Rastogi, V., Suciu, D., & Hong, S. (2007). The boundary between privacy and utility in data publishing. In *Proceedings of the 33rd International conference on Very Large Data Bases* (pp. 531–542). VLDB Endowment.

Reid, M.-D., & Williamson, R.-C. (2010). Composite binary losses. *Journal of Machine Learning Research*, *11*(Sep), 2387–2422.

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press.

Rokach, L., & Maimon, O. (2005). Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *35*(4), 476–487.

Rubinstein, B., Bartlett, P., Huang, L., & Taft, N. (2012). Learning in a large function space: Privacy-preserving mechanisms for svm learning. *Journal of Privacy and Confidentiality*, *4*(1), 4.

Sankar, L., Rajagopalan, S., & Poor, H. (2013). Utility-privacy tradeoff in databases: An information-theoretic approach. *IEEE Transactions on Information Forensics and Security*, *8*(6), 838–852.

Sarathy, R., & Muralidhar, K. (2011). Evaluating Laplace Noise Addition to Satisfy

Differential Privacy for Numeric Data. *Transactions on Data Privacy*, *4*(1), 1–17.

Sarwate, A. D., & Chaudhuri, K. (2013). Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE signal processing magazine*, *30*(5), 86–94.

Schapire, R. E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning Journal*, *37*(3), 297–336.

Shannon, C. E., Weaver, W., & Burks, A. W. (1951). *The mathematical theory of communication*.

Shiva Prasad Kasiviswanathan, S. R., Kobbi Nissim, & Smith, A. (2013). Analyzing graphs with node differential privacy. *In Theory of Cryptography, Springer*, *39*, 457-476.

Simonite, T. (2016). Apple's new privacy technology may pressure competitors to better protect our data. *MIT Technology Review (August 3)*.

Smith, A. (2011). Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the 43rd annual ACM symposium on Theory of computing, ACM* (pp. 813–822).

Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *10*(05), 557–570.

Vaidya, C. W. C., Jaideep, & Zhu, Y. M. (2006). Privacy preserving data mining. *Springer Science and Business Media*, *19*.

Van Drongelen, W. (2011). *Signal processing for neuroscientists: an introduction to the analysis of physiological signals*. Oxford: Elsevier.

Vu, D., & Slavkovic, A. (2009). Differential privacy for clinical trial data: Preliminary evaluations. In *IEEE International Conference on Data Mining Workshops* (pp. 138–143).

Wang, K., Chen, R., Fung, B., & Yu, P. (2010). Privacy-preserving data publishing: A survey on recent developments. *ACM Computing Surveys*, *42*(4), 1–53.

Wei, W., Du, J., Yu, T., & Gu, X. (2009). Securemr: A service integrity assurance framework for mapreduce. In *Annual Computer Security Applications Conference* (pp. 73–82).

Williams, O., & Frank, M. (2010). Probabilistic inference and differential privacy. *Advances in Neural Information Processing Systems*, *31*, 2451–2459.

Witten, I., & Frank, E. (1999). *Data Mining: Practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann.

Wong, R. C.-W., Li, J., Fu, A. W.-C., & Wang, K. (2006). (α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data mining* (pp. 754–759).

Xiao, X., Bender, G., Hay, M., & Gehrke, J. (2011). iReduct: differential privacy with reduced relative errors. In *Proceedings of the ACM SIGMOD International Conference on Management of data* (pp. 229–240). ACM.

Xiao, X., & Tao, Y. (2006). Personalized privacy preservation. In *Proceedings of the ACM SIGMOD International Conference on Management of data* (pp. 229–240).

Zhang, J., Zhang, Z., Xiao, X., Yang, Y., & Winslett, M. (2012). Functional mechanism: Regression analysis under differential privacy. *Proceedings of the VLDB*

Endowment, *5*(11), 1364–1375.

Zhu, T., Xiong, P., Xiang, Y., & Zhou, W. (2013). An effective differentially private data releasing algorithm for decision tree. In *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* (pp. 388–395). IEEE.