Data Classification for L-Diversity

Koray Mancuhan

Purdue University

Follow this and additional works at: https://docs.lib.purdue.edu/open_access_dissertations

Recommended Citation
Mancuhan, Koray, "Data Classification for L-Diversity" (2017). Open Access Dissertations. 1598.
https://docs.lib.purdue.edu/open_access_dissertations/1598

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.
DATA CLASSIFICATION FOR $\ell$-DIVERSITY

A Dissertation
Submitted to the Faculty
of
Purdue University
by
Koray Mancuhan

In Partial Fulfillment of the
Requirements for the Degree
of
Doctor of Philosophy

December 2017
Purdue University
West Lafayette, Indiana
THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Chris Clifton, Chair
   Department of Computer Science
Dr. Dan Goldwasser
   Department of Computer Science
Dr. Jean Honorio
   Department of Computer Science
Dr. Jennifer Neville
   Department of Computer Science

Approved by:

Dr. Voicu Popescu by Dr. William J. Gorman
   Chair, Graduate Study Committee
In dedication to my grand parents who passed away

In dedication to my parents for their continuous support
ACKNOWLEDGMENTS

This work was made possible partially by NPRP grant 09-256-1-046 from the Qatar National Research Fund, by Research Grant 2014-2015 from the Purdue Research Foundation (PRF) and by generous support of the Northrop Grumman Cybersecurity Consortium. The statements made herein are solely the responsibility of the author.

I wish to thank to Dr. Chris Clifton who has mentored patiently during the completion of this dissertation. I wish many thanks to my parents for their continuous support during this work. I am also grateful to Dr. Dan Goldwasser, Dr. Jean Honorio and Dr. Jennifer Neville for helpful suggestions and pointing out the ambiguities through the preparation of the material. I wish to thank to Dr. Ryan Riley for his helpful comments throughout the preparation of the parts of this work. I also thank to Dahjong Chung who has been very supportive. I would like to thank to Hogun Park, Jaewoo Lee and many others as well for the constructive discussion that we had throughout the realization. At last, I wish to thank to Department of Computer Science at Purdue University and to Department of Computer Science and Engineering at Qatar University for providing physical facilities required in the experimentation.
PREFACE

This work selects a basic privacy standard such as ℓ-diversity under the high utility data publishing scheme anatomization, and shows theoretically that every additional information matters to fit a classifier on the private data. The main part of this work is one of the few theoretical takes on the data classification in the privacy context, the first take on the non-differential privacy based alternatives. However, this work is not the first attempt to study the data classification in the context of data privacy. It doesn’t attempt to compete with other popular privacy standards such as differential privacy either.

Some parts of the work are quite empirical whereas other parts of the work are quite theoretical. Chapter 4 requires some information security and data mining background. Basic knowledge of privacy, decision trees and data classification would be sufficient to capture the core idea of the proposed algorithm. Chapters 5 and 6 on the other hand require solid theoretical background on the statistical machine learning and pattern recognition. Chapter 5 in particular is quite dry in most parts of the proofs. I tried to give some examples to reveal the core idea of the algorithms and the motivation of the theoretical outcome in these chapters.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>ix</td>
</tr>
<tr>
<td>SYMBOLS</td>
<td>xi</td>
</tr>
<tr>
<td>ABBREVIATIONS</td>
<td>xiv</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>xv</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2 RELATED WORK AND PROBLEM STATEMENT</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Classification Related Work and Problem Statement</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Other Related Work</td>
<td>8</td>
</tr>
<tr>
<td>3 DEFINITIONS</td>
<td>10</td>
</tr>
<tr>
<td>4 DISTRIBUTED DECISION TREE CLASSIFICATION FOR ( \epsilon )-DIVERSITY</td>
<td>16</td>
</tr>
<tr>
<td>4.1 Definitions</td>
<td>16</td>
</tr>
<tr>
<td>4.2 Distributed Decision Tree Classifier</td>
<td>18</td>
</tr>
<tr>
<td>4.2.1 Algorithm</td>
<td>19</td>
</tr>
<tr>
<td>4.2.2 Privacy Preservation</td>
<td>22</td>
</tr>
<tr>
<td>4.2.3 Costs for Client</td>
<td>23</td>
</tr>
<tr>
<td>4.3 Experiments and Results</td>
<td>24</td>
</tr>
<tr>
<td>4.3.1 Prerequisites</td>
<td>24</td>
</tr>
<tr>
<td>4.3.2 Analysis of Results</td>
<td>27</td>
</tr>
<tr>
<td>5 NON-PARAMETRIC CLASSIFICATION FOR ( \epsilon )-DIVERSITY</td>
<td>30</td>
</tr>
<tr>
<td>5.1 Notations</td>
<td>30</td>
</tr>
<tr>
<td>5.2 Anatomized k-NN Classifier</td>
<td>33</td>
</tr>
<tr>
<td>5.2.1 Illustration</td>
<td>33</td>
</tr>
<tr>
<td>5.2.2 Implementation</td>
<td>35</td>
</tr>
<tr>
<td>5.2.3 Privacy Preservation</td>
<td>35</td>
</tr>
<tr>
<td>5.2.4 Theoretical Analysis</td>
<td>35</td>
</tr>
<tr>
<td>5.3 Experiments and Results</td>
<td>50</td>
</tr>
<tr>
<td>5.3.1 Prerequisites</td>
<td>50</td>
</tr>
<tr>
<td>5.3.2 Analysis of Results</td>
<td>53</td>
</tr>
<tr>
<td>5.3.3 Student ( t )-test for Anatomized k-NN versus Original k-NN</td>
<td>66</td>
</tr>
</tbody>
</table>
# SUPPORT VECTOR CLASSIFICATION FOR t-DIVERSITY

6.1 Notations ................................................. 69
6.2 Pruning Mechanism for Anatomization .................. 71
   6.2.1 Algorithm ........................................ 71
   6.2.2 Pseudo code and Complexity ........................ 74
   6.2.3 Privacy Preservation ................................. 76
   6.2.4 Generalization Error of Pruned SVC............... 76
6.3 Experiments and Results ................................ 79
   6.3.1 Prerequisites ...................................... 79
   6.3.2 Analysis of Results ................................ 81
   6.3.3 Student t-test for Pruned SVC/SVM versus Original SVC/SVM 88

7 SUMMARY .................................................... 90

REFERENCES .................................................. 93

VITA .......................................................... 100
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Anatomized k-NN vs Original k-NN on Adult</td>
</tr>
<tr>
<td>5.2</td>
<td>Anatomized k-NN vs Original k-NN on IPUMS</td>
</tr>
<tr>
<td>5.3</td>
<td>Anatomized k-NN vs Original k-NN on Marketing</td>
</tr>
<tr>
<td>5.4</td>
<td>Anatomized k-NN vs Original k-NN on Fatality</td>
</tr>
<tr>
<td>6.1</td>
<td>Pruned SVC vs Original SVC</td>
</tr>
<tr>
<td>6.2</td>
<td>Pruned SVM vs Original SVM</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Distributed DT Classifier <strong>Main Function</strong> (Called by Server)</td>
</tr>
<tr>
<td>4.2</td>
<td>Distributed DT Classifier <strong>FindBDTLeaf Function</strong> (Called by Server)</td>
</tr>
<tr>
<td>4.3</td>
<td>Distributed DT Classifier <strong>Inference Function</strong> (Called by Client)</td>
</tr>
<tr>
<td>4.4</td>
<td>Execution Time Savings as a Fraction of Original DT Time</td>
</tr>
<tr>
<td>4.5</td>
<td>Memory Savings as a Fraction of Original DT Memory Requirement</td>
</tr>
<tr>
<td>4.6</td>
<td>Accuracy Graph</td>
</tr>
<tr>
<td>5.1</td>
<td>Training Data Example</td>
</tr>
<tr>
<td>5.2</td>
<td>Anatomized 3-NN Classifier Example</td>
</tr>
<tr>
<td>5.3</td>
<td>10 Cross Validation Errors Rates for NN Classifiers</td>
</tr>
<tr>
<td>5.4</td>
<td>NN Error Rates on Adult (k for k and L for l)</td>
</tr>
<tr>
<td>5.5</td>
<td>10 Cross Validation Errors Rates of Adult Data</td>
</tr>
<tr>
<td>5.6</td>
<td>10 Cross Validation Errors Rates of Ipums Data</td>
</tr>
<tr>
<td>5.7</td>
<td>10 Cross Validation Errors Rates of Marketing Data</td>
</tr>
<tr>
<td>5.8</td>
<td>10 Cross Validation Errors Rates of Fatality Data</td>
</tr>
<tr>
<td>5.9</td>
<td>Convergence of NN Using Multiple Size Training Sets</td>
</tr>
<tr>
<td>6.1</td>
<td>Training Data Example</td>
</tr>
<tr>
<td>6.2</td>
<td>Linear Classifier from Original Training Data</td>
</tr>
<tr>
<td>6.3</td>
<td>Possible Circles of Shatterable Points</td>
</tr>
<tr>
<td>6.4</td>
<td>Linear Classifier from Pruned Training Data</td>
</tr>
<tr>
<td>6.5</td>
<td>Prerequisite Step Pseudocode</td>
</tr>
<tr>
<td>6.6</td>
<td>Pruning Step Pseudocode</td>
</tr>
<tr>
<td>6.7</td>
<td>SVC on Adult</td>
</tr>
<tr>
<td>6.8</td>
<td>SVM on Adult</td>
</tr>
<tr>
<td>6.9</td>
<td>SVC on IPUMS</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>----------------</td>
<td>------</td>
</tr>
<tr>
<td>6.10 SVM on IPUMS</td>
<td>85</td>
</tr>
<tr>
<td>6.11 SVC on Marketing</td>
<td>85</td>
</tr>
<tr>
<td>6.12 SVM on Marketing</td>
<td>86</td>
</tr>
<tr>
<td>6.13 SVC on Fatality</td>
<td>87</td>
</tr>
<tr>
<td>6.14 SVM on Fatality</td>
<td>87</td>
</tr>
</tbody>
</table>
SYMBOLS

\( D \)  
Original Training Data

\( N \)  
Number of instances

\( D_A \)  
Anatomized Training Data

\( D_P \)  
Pruned Training Data

\( D_{id} \)  
Identifying Training Data

\( T \)  
Test set

\( \ell \)  
Parameter of \( \ell \)-diversity standard

\( \gamma \)  
A base decision tree leaf

\( \Gamma \)  
A set of base decision tree leaves

\( d \)  
Number of identifying attributes

\( A_{id} \)  
Identifying attributes

\( A_i \)  
Identifying attribute \( i \)

\( A_s \)  
Sensitive attribute

\( a_i \)  
Identifying attribute value

\( a_s \)  
Sensitive attribute value

\( G_i \)  
Anatomy group with id \( i \)

\( s \)  
Sequence number value

\( E_k(salt, s) \)  
Encrypted sequence number with key \( k \) and \( salt \)

\( X'_N(k) \)  
k number of nearest neighbors of a test instance for original k-NN classifier

\( X'_{N\ell}(k) \)  
k number of nearest neighbors of a test instance for anatomized k-NN classifier

\( X \)  
Random variable

\( x \)  
Test instance
$x_i, x_j$  
Training instance

$y, c$  
Binary class label

$P(X)$  
Probability distribution where $x$ and $x_i$ are drawn
(case of $D$)

$P(X, y)$  
Joint density of training instances coming from $X$
with class $y$

$q_i(x)$  
Posterior probability of class $i$ for test instance $x$
(case of $D$)

$P_i(x)$  
Likelihood probability of class $i$ for test instance $x$
(case of $D$)

$P_i$  
Prior probability of class $i$ (case of $D$)

$q_{A_i}(x)$  
Posterior probability of class $i$ for test instance $x$
(case of $D_A$)

$P_{A_i}(x)$  
Likelihood probability of class $i$ for test instance $x$
(case of $D_A$)

$P_{A_i}$  
Prior probability of class $i$ (case of $D_A$)

$R^k(x), R^k(X'_N(k), x)$  
Error rate of $k$ nearest neighbors (case with $D$)

$R^k_A(x), R^k(X'_N(k), x)$  
Error rate of $k$ nearest neighbors (case with $D_A$)

$R^*(x)$  
Bayesian error of $k$ nearest neighbor (case with $D$)

$R^*_A(x)$  
Bayesian error of $k$ nearest neighbor (case with $D_A$)

$\mathbb{R}^{d+1}$  
Metric or Euclidean space

$\mathcal{F}$  
Functional space

$f, f(X)$  
Linear classifier

$\hat{f}_N$  
Empirical risk minimizer

$R$  
Radius of a sphere in euclidean space

$w$  
Set of weights in a linear classifier

$b$  
Bias term of a linear classifier

$||*||$  
Euclidean norm of a vector *

$I(*)$  
Indicator function
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Pi(*)$</td>
<td>Database projection operator</td>
</tr>
<tr>
<td>$\bowtie$</td>
<td>Database join operator</td>
</tr>
<tr>
<td>$R(f)$</td>
<td>Risk of a linear classifier</td>
</tr>
<tr>
<td>$\tilde{R}_N(f)$</td>
<td>Empirical risk of a linear classifier</td>
</tr>
<tr>
<td>$R(\hat{f}_N)$</td>
<td>Risk of an empirical risk minimizer</td>
</tr>
<tr>
<td>$\sigma_G$</td>
<td>Pruning mechanism</td>
</tr>
</tbody>
</table>
# ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>Decision Tree</td>
</tr>
<tr>
<td>BDT</td>
<td>Base Decision Tree</td>
</tr>
<tr>
<td>k-NN</td>
<td>k-nearest neighbor classifier</td>
</tr>
<tr>
<td>NN</td>
<td>nearest neighbor</td>
</tr>
<tr>
<td>SVC</td>
<td>Support vector classifier</td>
</tr>
<tr>
<td>SVM</td>
<td>Support vector machine</td>
</tr>
<tr>
<td>IT</td>
<td>Identifying table</td>
</tr>
<tr>
<td>ST</td>
<td>Sensitive table</td>
</tr>
<tr>
<td>GID</td>
<td>Group id</td>
</tr>
<tr>
<td>HIPAA</td>
<td>Healthcare Insurance Portability and Accountability Act</td>
</tr>
<tr>
<td>TDS</td>
<td>Top Down Specialization</td>
</tr>
<tr>
<td>SSN</td>
<td>Social security number</td>
</tr>
<tr>
<td>SEQ</td>
<td>Sequence number</td>
</tr>
<tr>
<td>ESEQ</td>
<td>Encrypted sequence number</td>
</tr>
</tbody>
</table>
Corporations are retaining ever-larger corpuses of personal data; the frequency of breaches and corresponding privacy impact has been rising accordingly. One way to mitigate this risk is through use of anonymized data, limiting the exposure of individual data to only where it is absolutely needed. This would seem particularly appropriate for data mining, where the goal is generalizable knowledge rather than data on specific individuals. In practice, corporate data miners often insist on original data, for fear that they might miss something with anonymized or differentially private approaches. This dissertation provides both empirical and theoretical justifications for the use of anonymized data, in particular for a specific scheme of anonymization called anatomization (or anatomized data). Anatomized data preserves all attribute values, but introduces uncertainty in the mapping between identifying and sensitive values, thus satisfying $\ell$-diversity. We first propose a promising decision tree learning algorithm. Empirical results show that this algorithm produces decision trees approaching the accuracy of non-private decision trees. We then show that a $k$-nearest neighbor classifier and a support vector classifier trained on anatomized data are theoretically expected to do as well as on the original data under certain conditions. The theoretical effectiveness of the latter approaches are validated using several publicly available datasets, showing that we outperform the state of the art for nearest neighbor and support vector classification using training data protected by $k$-anonymity, and are comparable to learning on the original data.
1 INTRODUCTION

Many privacy definitions have been proposed based on generalizing/suppressing data ($\ell$-diversity \cite{1}, $k$-anonymity \cite{2,3}, $t$-closeness \cite{4}, $\delta$-presence \cite{5}, $(\alpha,k)$-anonymity \cite{6}). Other alternatives include value swapping \cite{7}, distortion \cite{8}, randomization \cite{9}, and noise addition (e.g., differential privacy \cite{10}). Generalization consists of replacing identifying attribute values with a less specific version \cite{3}. Suppression can be viewed as the ultimate generalization, replacing the identifying value with an “any” value \cite{3}. Generalization has the advantage of preserving truth, but a less specific truth that reduces utility of the published data.

Xiao and Tao proposed anatomization as a method to enforce $\ell$-diversity while preserving specific data values \cite{11}. Anatomization splits instances across two tables, one containing identifying information and the other containing private information. The more general approach of fragmentation \cite{12} divides a given dataset’s attributes into two sets of attributes (2 partitions) such that an encryption mechanism avoids associations between two different small partitions. Vimercati et al. extend fragmentation to multiple partitions \cite{13}, and Tamas et al. propose an extension that deals with multiple sensitive attributes \cite{14}. The main advantage of anatomization/fragmentation is that it preserves the original values of data; the uncertainty is only in the mapping between individuals and sensitive values.

This dissertation proposes that this additional information has real value. First, we propose a simple decision tree learning algorithm for distributed data classification and justify the algorithm through the experiments on real data. We then demonstrate that in theory, learning from anatomized data can be as good as learning from the raw data for k-nearest neighbor and support vector classification. The empirical results of these classification algorithms show that learning from anatomized data beats learning from generalization-based anonymization.
In the theoretical analysis, this dissertation looks only at k-nearest neighbor classifier (k-NN) and support vector classifier (SVC). This focus was chosen because there are solid theoretical results on the limits of learning using k-NN from statistical pattern recognition. There is also solid theory about the generalization ability of the SVC from the statistical learning theory. The existing theory allows us to compare theoretical bounds on learning from anatomized data with learning from the underlying unprotected data. In k-NN, we demonstrate this for a simple approach of anatomizing the data; we simply consider all possible mappings of individuals to sensitive values as equally likely. In SVC, we pick one of the equally likely sensitive values for each individual so that the resulting SVC would have the best expected generalization.

There is concern that anatomization is vulnerable to several prior knowledge attacks [15, 16]. In addition, the membership attacks is also an existing issue of anatomization despite the lack of sufficient prior knowledge [17]. While this can be an issue, any method that provides any meaningful utility fails to provide perfect privacy against a sufficiently strong adversary [10,18]. Legally recognized standards such as the so-called “Safe Harbor” rules of the U.S. Healthcare Insurance Portability and Accountability Act (HIPAA) [19] allow release of data that bears some risk of re-identification; the belief is that the greater good of beneficial use of the data outweighs the risk to privacy. Besides, anatomization can be robust against some recent attacks such as minimality [20,21].

This dissertation does not attempt to resolve that issue (although we use a randomized rather than optimized grouping that provides resistance to the prior knowledge attacks above). Instead, it investigates the feasible data classification models for anatomization. Our theoretical analysis holds for any assignment of items to anatomy groups, including a random assignment, which provides a high degree of robustness against minimality and correlation-based attacks. While this does not eliminate privacy risk, if the alternative is to use the original data, we show that anatomy provides comparable utility while reducing the privacy risk.
The rest of this dissertation is organized as follow. Chapter 2 summarizes related work, defines the problem statement and underlines the key contributions. Chapter 3 gives the set of definitions that are required to elaborate further discussion. Chapter 4 discusses the distributed decision tree classification for the anatomized data while Chapters 5 and 6 discuss the k-NN and SVC. Chapter 7 concludes the dissertation with an overall summary of the work.
2 RELATED WORK AND PROBLEM STATEMENT

2.1 Classification Related Work and Problem Statement

There have been studies on how to train classifiers on anonymized data. The existing work can be grouped into three categories: generalization based classification, distribution reconstruction based classification, and differential privacy based classification.

In generalization based classification, nearest neighbor classification using generalized data was investigated by Martin [22]. Nested generalization and non-nested hyperrectangles were used to generalize the data from which the nearest neighbor classifiers were trained. Iyengar suggested using a classification metric so as to find the optimum generalization. Then, a C4.5 decision tree classifier was trained from the optimally generalized training data [23]. Zhang et al. studied Naive Bayes using partially specified training data [24], proposing a conditional likelihoods computation algorithm exploring the instance space of attribute-value generalization taxonomies. Inan et al. proposed nearest neighbor and support vector machine classifiers using anonymized training data that satisfy k-anonymity. Taylor approximation was used to estimate the dot product, Euclidean distance and kernel function from generalized data [25].

In distribution reconstruction based classification, Agrawal et al. proposed an iterative distribution reconstruction algorithm for distorted training data from which a C4.5 decision tree classifier was trained [8,26]. Fung et al. gave a top-down specialization method (TDS) for anonymization so that the anonymized data allows accurate decision trees. A new scoring function was proposed for the calculation of decision tree splits from the compressed training data [27]. Dowd et al. studied C4.5 decision tree learning from training data perturbed by random substitutions. A matrix
based distribution reconstruction algorithm was applied on the perturbed training data from which an accurate C4.5 decision tree classifier was learned [28].

In differential privacy based classification, Rubinstein et al. studied the kernels of support vector machine under differential privacy and showed the trade-off between privacy level and data utility. They analyzed finite and infinite dimensional kernels as a function of the approximation error under differential privacy [29]. Lin at al. studied training support vector classification for outsourced data. Random transformation was applied on the training set so that the cloud server could compute an accurate model without knowing what the actual values were [30]. Jain et al. studied support vector machine kernels in the differential privacy setting. They proposed differentially private mechanisms to train support vector machines for interactive, semi-interactive and non-interactive learning scenarios, providing theoretical analysis of the proposed approaches [31]. Freidman et al. investigate learning C4.5 decision trees from datasets that satisfy differential privacy [32]. Jagannathan et al. propose a tree classifier based on random forests that is built from differentially private data [33]. Vaidya et al. give a Naive Bayes Classification scheme that publishes the model parameters with differential privacy [34]. Duchi et al. study the statistical inference under a privacy description called local privacy that is relevant to differential privacy [35]. They derive the convergence rate of a statistical estimator where the data is kept confidential even from the learner. Duchi et al. later extend this work to the concept of differentially local privacy which captures both local and differential privacy concepts in the context of convex risk minimization [36].

Other work related to the classifiers on anonymized data are the ones for vertically/horizontally partitioned data and outsourced data to a semi-honest third party. Vaidya et al. propose decision trees for vertically partitioned data [37]. Giannella et al. study a similar decision tree learning problem reducing the communication overhead [38]. Vaidya et al. later proposes a new random decision tree learning framework to learn models from vertically or horizontally partitioned data [39]. They extend some existing techniques in [37] such that the learning is more scalable for par-
allel processing. Yu et al. investigate support vector machine learning for vertically partitioned data [40]. Hu et al. propose an alternative support vector machine learning with less computation overhead of vertically partitioned data [41]. Mangasarian et al. investigate the support vector machine learning in case of horizontally partitioned data [42]. Vertically/horizontally partitioned techniques are generally based on secure multiparty computation. These techniques make strong assumptions about data partitions and what data can be shared, and typically assume that each party holding data has significant computational power. For example, some decision tree techniques assume that two tuples of two partitions can be linked directly to each other, that the tuples are ordered in the same way; and that the class labels are known for both partitions.

None of the earlier work has provided a classifier directly applicable to anatomized training data. Such a classifier requires specific theoretical and experimental analysis, because anatomized training data provides additional detail that has the potential to improve learning; but also additional uncertainty that must be dealt with. Furthermore, most of the previous work didn’t justify theoretically why the proposed heuristics let classifiers generalize well. Therefore, this dissertation states problem 2.1.1.

**Problem 2.1.1** Define heuristics to train classifiers on anatomized data without violating $\ell$-diversity while using the sensitive information, with a theoretical guarantee of good generalization under reasonable assumptions.

Our solution to the problem 2.1.1 will have the following key contributions to the privacy preserving data mining community in the upcoming chapters:

1. We define a classification task on anatomized data without violating the random worlds assumption. A violating classification task would be the prediction of sensitive attribute, a task that was found to be $\mathbb{#P}$-complete by Kifer [15]. Besides, it would also be out of the scope of problem 2.1.1.
2. This dissertation proposes a distributed decision tree algorithm which solves partially the problem \[2.1.1\] The promising results of this method motivate the theoretically solid methods.

3. To our best knowledge, this is the first dissertation in the privacy community that studies the theoretical effect of training the k-nearest neighbor (k-NN) and support vector classification (SVC) on anatomized data. We show the generalization error for a non-parametric classifier and support vector classifier using the anatomized training data. In case of k-NN, we also show the anatomization effect for the error rate bounds and the convergence rate.

4. In the experiments, we compare the distributed decision tree, \(k\)-NN and SVC classifiers trained on the anatomized data with the respective counterparts trained on the unprotected data. In cases of nearest neighbor (NN), SVC and support vector machines (SVMs), we also make an additional comparison to generalization based learning scheme \[25\].

Note that the test data is assumed to be neither anonymized nor anatomized. Inan et al. give a practical application of such a learning scenario \[25\]:

Several hospitals may collaborate to create one large anonymized data set which is then shared among all collaborating hospitals. For a researcher in a participating hospital who is interested in using data mining to classify the patients of the hospital, there will be at least two options. Either, the researcher can build a classifier using the local data set and use it for new patients or he/she can build a classifier using the large anonymized data set that involves many more samples and use it for new patients. To see which of these two options is better, we need to be able to classify original data (e.g., the medical records of the new patient) using data mining models built on anonymized data.

Another case would be outsourcing where the owner of the personal data releases the anonymized or anatomized version to a third party such as cloud server for both
storage and machine learning purposes [43]. The data should be protected under some privacy standard due to the sensitivity of the personal information. The owner of data can still use the processing power of the cloud server to train computationally expensive machine learning models in addition to the storage ability.

2.2 Other Related Work

There have been studies about handling private data in data mining tasks such as clustering and pattern mining.

In the clustering task, Oliveira et al. studies privacy preserving clustering techniques based on data transformation [14]. Ngai et al. propose a clustering algorithm that handles the anonymized data as uncertain information [15]. Kriegel et al. propose a hierarchical density based clustering method using fuzzy objects [16]. Xiao et al. discusses the problem of distance calculation for uncertain objects [17].

In pattern mining task, Evfimievski et al. studies the frequent itemset mining from randomized data [9, 48]. Chui et al. [19] and Leung et al. [50] address the frequent itemset mining problem as well. Atzori et al. focus on the privacy preservation of pattern mining results [51]. Atzori et al. later propose a new frequent itemset mining algorithm for $k$-anonymized data [52]. Zeng et al. proposes a new apriori algorithm for frequent itemset mining of differentially private data [53]. Lee et al. focus on extracting top frequent itemsets using FP-trees that are differentially private [54].

The last group of approaches related to mining the anonymized data are the employment of data mining to improve the data utility and the privacy guarantee of anonymization. Li et al. formulate the background knowledge using frequent itemset mining in order to hack and to improve the anonymization [55]. Al Bouna et al. improve the anonymization in case of transactional databases so that the anonymized data supports effective frequent itemset mining [56]. Mirakabad et al. devise an apriori algorithm to publish data respecting $\ell$-diversity, an alternative to the bucke-
tization of Xiao et al. [57]. Mirakabad et al. also propose an extended anatomization using clustering so as to minimize information loss [58].

We have seen from the previous and the current section that the classification task has been studied more thoroughly than the other data mining tasks in the privacy preserving data mining community. The reason of this focus is that the classification algorithms have deeper theoretical understanding about the generalization properties. It is also easier to evaluate the classification tasks as it is part of the supervised learning which includes the ground truth (class label). Thus, the proposed algorithms depend less on the empirical results of the existing data. This dissertation will also focus on the classification tasks for similar reasons. Pattern mining has recently been studied by Hamzaoui et al. for anatomized data satisfying $\ell$-diversity [59].
3 DEFINITIONS

The first four definitions restate standard definitions of unprotected data and attribute types.

**Definition 3.0.1** A dataset $D$ is called a person specific dataset for population $P$ if each instance $x_i \in D$ belongs to a unique individual $p \in P$. The person specific dataset has the schema in \((3.1)\)

$$\begin{align*}
(C, A_1, ..., A_d, A_s)
\end{align*}$$

where $C$ is the class attribute, $A_1, ..., A_d$ are quasi-identifying attributes, and $A_s$ is the sensitive attribute. Quasi-identifying and sensitive attributes are defined below.

The person specific dataset will be called the original training data in this paper. Next, we define the types of attributes.

**Definition 3.0.2** A set of attributes are called direct identifying attributes if they let an adversary associate an instance $x_i \in D$ to a unique individual $p \in P$ without any background knowledge.

**Definition 3.0.3** A set of attributes are called quasi-identifying attributes if there is background knowledge available to the adversary that associates the quasi-identifying attributes with a unique individual $p \in P$.

We include both direct and quasi-identifying attributes under the name identifying attributes. First name, last name and social security number (SSN) are common examples of direct identifying attributes. Direct identifying attributes are generally not useful in machine learning as unique identifiers would not generalize. Hence, the schema in \((3.1)\) typically does not include any direct identifying attributes. They
can be suppressed but other information can be both useful and assist in identifying individuals. Some common examples of quasi-identifying attributes are age, postal code, and occupation. These are assumed to be public knowledge and thus need not be protected. Thus, the quasi-identifying attributes are included in schema \(3.1\) as \(A_1, \ldots, A_d\). Next, we will give the second type of attribute.

**Definition 3.0.4** An attribute \(A_s\) of \(D\) is called a sensitive attribute if we should protect against adversaries correctly inferring the value for any individual \(x_i \in D\).

Patient disease and individual income are common examples of sensitive attributes. Unique individuals \(p \in P\) typically don’t want these sensitive information to be revealed to individuals without a direct need to know that information.

Given the former definitions, we will next define the anonymized training data following the definition of \(k\)-anonymity [3].

**Definition 3.0.5** A training dataset \(D\) that satisfies the following conditions is said to be \(k\)-anonymized training data \(D_k\) [3]:

1. The training data \(D_k\) does not contain any unique identifying attributes.

2. Every instance \(x_i \in D_k\) is indistinguishable from at least \((k - 1)\) other instances in \(D_k\) with respect to its quasi-identifying attributes.

Anatomy satisfies a slightly weaker definition; the indistinguishability applies only to sensitive data. This will be captured in Definitions 3.0.8-3.0.10.

The rest of the dissertation assumes that the anonymized training data \(D_k\) is created according to a generalization based data publishing method. We next define the comparison classifiers.

**Definition 3.0.6** A classifier that is trained on the original training data \(D\) is called the original classifier.

**Definition 3.0.7** A classifier trained on the anonymized training data \(D_k\) is called the anonymized classifier.
We go further from Definition 3.0.5, requiring that there must be multiple possible sensitive values that could be linked to an individual. **The proposed algorithms will be centered around the following definitions.** This new requirement uses the definition of groups \[11\].

**Definition 3.0.8** A group \(G_j\) is a subset of instances in \(D\) such that \(D = \bigcup_{j=1}^{m} G_j\), and for any pair \((G_{j_1}, G_{j_2})\) where \(1 \leq j_1 \neq j_2 \leq m\), \(G_{j_1} \cap G_{j_2} = \emptyset\).

Next, we define the concept of \(\ell\)-diversity or \(\ell\)-diverse (multiple possible sensitive values) for all the groups in the original training data \(D\) regarding to Xiao et al. \[11\].

**Definition 3.0.9** A set of groups is \(\ell\)-diverse if and only if \(\forall G_j, v \in \Pi_{A_s}(G_j); \frac{\text{freq}(v,G_j)}{|G_j|} \leq \frac{1}{\ell}\) where \(A_s\) is the sensitive attribute in \(D\), \(\Pi_{A_s}(\ast)\) is the database \(A_s\) projection operation on original training data \(\ast\) (or on data table in the database community), \(\text{freq}(v,G_j)\) is the frequency of \(v\) in \(G_j\) and \(|G_j|\) is the number of instances in \(G_j\).

We extend the data publishing method **anatomization** that is originally based on \(\ell\)-diverse groups by Xiao et al. \[11\]. We should note that \(\ell\)-diversity comes from Machanavajjhala et al. Xiao et al. extend this to anatomization rather than generalization \[1,11\]. As we are following Xiao et al., we will be using groups rather than the blocks of Machanavajjhala et al.

**Definition 3.0.10** Given an original training data \(D\) partitioned in \(m\) \(\ell\)-diverse groups according to Definition 3.0.9, anatomization produces an identifying table \(IT\) and a sensitive table \(ST\) as follows. \(IT\) has schema

\[(C, A_1, ..., A_d, GID)\]

including the class attribute, the quasi-identifying attributes \(A_i \in IT\) for \(1 \leq i \leq d\), and the group id \(GID\) of the group \(G_j\). For each group \(G_j \in D\) and each instance \(x_i \in G_j\), \(IT\) has an instance \(x_i\) of the form:

\[(x_i.c, x_i.a_1, ..., x_i.a_d, j)\]
ST has schema

\[(GID, A_s)\]

where \(A_s\) is the sensitive attribute in \(D\) and \(GID\) is the group id of the group \(G_j\).

For each group \(G_j \in D\) and each instance \(x_i \in G_j\), \(ST\) has an instance of the form:

\[(j, x_i, a_s)\]

Unlike Machanavajjhala et al., the data publishing scheme here doesn’t require the instances in a group to generalize to the same value. This comes from Xiao et al.’s definition of anatomization. Unlike Xiao et al., our definition of anatomization doesn’t include the count of sensitive attributes within a group. We intend to assume a 1-to-1 mapping between the sensitive attribute and the identifying attributes.

The \(IT\) table includes only the quasi-identifying and class attributes. Provided an instance \(x_i \in D\), the class label is denoted by \(x_i.c\). We don’t consider the case where \(c\) is sensitive, as this would make the purpose of classification to violate privacy. \(c\) is neither sensitive nor identifying in this dissertation, although our analysis holds for \(c\) being an identifying attribute. We assume that direct identifying attributes are removed before creating the \(IT\) and \(ST\) tables. We have the following observation from Definition 3.0.10 to train a classifier: every instance \(x_i \in IT\) can be matched to the instances \(x_j \in ST\) within the same group using the common attribute \(GID\) in both data tables. This observation yields the anatomized training data.

**Definition 3.0.11** Given two data tables \(IT\) and \(ST\) resulting from the anatomization on original training data \(D\), the anatomized training data \(D_A\) is

\[D_A = \Pi_{IT.C, IT.A_1, \ldots, IT.A_d, ST.A_s}(IT \Join ST)\]

where \(\Join\) is the database inner join operation with respect to the condition \(IT.GID = ST.GID\) and \(\Pi(*)\) is the database projection operation on training data *.

Anatomized training data shows one of two naïve data preprocessing approaches. Another one is ignoring the sensitive attribute in \(ST\) table:
Definition 3.0.12 Given two data tables $IT$ and $ST$ resulting from the anatomization on original training data $D$, the identifying training data $D_{id}$ is

$$D_{id} = \Pi_{IT.C, IT.A_1, \ldots, IT.A_d}(IT)$$

where $\Pi(\ast)$ is the database projection operation on training data $\ast$.

Identifying training data doesn’t use all the information available in the published data (and would likely lead to users insisting on having the original data.) The anatomized training data (cf. Definition 3.0.11), on the other hand, is both computationally costly (a factor of $\ell$ increase in size) and noisy: for every true instance, there are $\ell - 1$ incorrect instances that might cause incorrect classification. However, if the test instance comes from the original data’s distribution and the original training data is sufficiently large; then training a non-parametric classifier from the anatomized training data is expected to work well. In this dissertation, we propose the following classifier based on this naïve preprocessing method.

Definition 3.0.13 A $k$-nearest neighbor (k-NN) classifier that is trained on the anatomized training data $DA$ is called the anatomized k-NN classifier.

We will later show that such a naïve classifier will have very interesting theoretical properties. Having $\ell - 1$ incorrect instances for every true instance does not always result in a bad classifier if the test instance to classify does not have uncertainty between the identifying and the sensitive information. In fact, it is theoretically possible that anatomized k-NN classifier outperforms the original k-NN despite the uniform noise of $\ell$-diversity!

However, the naïve training method of Definition 3.0.11 might be too noisy for linear classification. For every true instance, there are $\ell - 1$ incorrect instances that may not be linearly separable or that could hinder the shattering properties of the original training data. We will propose a preprocessing algorithm to eliminate $\ell - 1$ instances within each group to give (hopefully) a linearly separable training data (with or without soft margin.) This gives the definition of our preprocessing proposition for SVC and SVM: pruned training data.
Definition 3.0.14 Given two data tables IT and ST resulting from anatomization of the original training data D, the pruned training data \( D_P \) is

\[
D_P = \Pi_{IT.C, IT.A_1, \ldots, IT.A_d, ST.A_s}(\sigma_G(IT, ST))
\]

where \( \sigma_G(IT, ST) \) is a pruning mechanism eliminating \( \ell - 1 \) instances for all groups \( G \) in the IT/ST pair that are unlikely to be separable (cf. Chapter 6), and \( \Pi(*) \) is the database projection operation on training data *.

We should note that the \( \sigma_G \) does not know the correct or incorrect mappings. We will next define the proposed linear classification method in this dissertation.

Definition 3.0.15 A linear support vector classifier (SVC) that is trained on the pruned training data \( D_P \) is called the pruned SVC. Similarly, a support vector machine (SVM) that is trained on the pruned training data \( D_P \) is called the pruned SVM.

Next will be the final definition in this chapter.

Definition 3.0.16 A classifier that is trained on the identifying training data \( D_{id} \) is called the identifying classifier.

We will give the specific definitions for the distributed DT classifier in the next chapter as the distributed learning setting is different from the learning setting of anatomized k-NN and the pruned SVC/SVM classifiers.
4 DISTRIBUTED DECISION TREE CLASSIFICATION FOR \( \ell \)-DIVERSITY

In this chapter, we will consider an empirical approach for data classification in a distributed learning scheme. In this scheme, we assume that the data is released to a third party for outsourcing purpose. The publisher of the data will be called client whereas the third party will be called cloud server or server interchangeably. As mentioned in Problem [2.1.1] the released data will be assumed to respect \( \ell \)-diversity.

We will consider anatomized training data for training, the test data will be neither anatomized nor k-anonymized. In the distributed scheme, the server is not allowed to reconstruct the original training data whereas client is allowed to do reconstruction when needed. This will be the motivation of the proposed algorithm in this chapter.

The chapter will be organized in the following way. Section 4.1 will present definitions specific to the distributed learning scheme in addition to the notations. Next, the proposed distributed decision tree (DT) classifier will be presented in Section 4.2.1. Section 4.3 will conclude this chapter by presenting the experimental results of the proposed algorithm. The discussion will follow mostly Mancuhan et al. [43].

4.1 Definitions

Using the definitions [3.0.8] and [3.0.9] we will define a different version of anatomization.

**Definition 4.1.1** Given an original training data \( D \) partitioned in \( m \) \( \ell \)-diverse groups according to Definition [3.0.9], anatomization produces an identifying table \( IT \) and a sensitive table \( ST \) as follows. \( IT \) has schema

\[
(C, A_1, ..., A_d, GID, ESEQ)
\]
including the class attribute, the quasi-identifying attributes $A_i \in IT$ for $1 \leq i \leq d$, the group id $GID$ of the group $G_j$ and an encrypted unique sequence number $ESEQ$. For each group $G_j \in D$ and each instance $x_i \in G_j$, $IT$ has an instance $x_i$ of the form:

$$(x_i.c, x_i.a_1, ..., x_i.a_d, j, E_k(salt, s))$$

$ST$ has schema

$$(SEQ, GID, A_s)$$

where $A_s$ is the sensitive attribute in $D$, $GID$ is the group id of the group $G_j$ and $SEQ$ is the unique sequence number. For each group $G_j \in D$ and each instance $x_i \in G_j$, $ST$ has an instance of the form:

$$(s, j, x_i.a_s)$$

Definition 4.1.1 was originally extended by Nergiz et al. [60, 61] from [11]. The difference from Definition 3.0.10 is that it includes $ESEQ$ and $SEQ$ in the schema of $IT$ and $ST$ respectively. Given an instance $x_i \in IT$, the publisher of the data can decrypt $E_k(salt, s)$ to obtain the specific $s$ value and match it with the $s$ value in the group $G_j$ of $ST$ table. Here, the same $s$ value in $ST$ table would indicate the correct match of the sensitive attribute among $\ell$ potential values. The publisher of the data can reconstruct the original data form the tables $IT$ and $ST$.

The distributed decision tree algorithm assumes that a client outsources its training data to a cloud server according to the Definition 4.1.1. The client can therefore reconstruct the original training data from the anatomized training data or the subsamples of the original training data from the subsamples of the anatomized training data. Next, we will describe the definitions specific to the distributed DT classifier for $\ell$-diversity.

**Definition 4.1.2** Given two partitions $IT$ and $ST$ of anatomized training data $D_A$, the base decision tree (BDT) classifier is a decision tree classifier that is built from the attributes $A_1, \cdots, A_d \in IT$. Given a BDT classifier that has leaves $\Gamma$; every leaf $\gamma \in \Gamma$ has instances in the following format:

$$(x_i.c, x_i.a_1, ..., x_i.a_d, j, E_k(salt, s))$$
Definition 4.1.3 *Given a base decision tree BDT and the leaves \( \Gamma \in BDT \); a leaf \( \gamma \in \Gamma \) is called refined leaf if and only if it points to a sub-tree rooted at \( \gamma \) that is encrypted (using symmetric key encryption).*

We will elaborate in the privacy discussion of the next section how and why a base decision tree has encrypted sub-trees.

Definition 4.1.4 *Given a base decision tree BDT, the leaves \( \Gamma \in BDT \); a leaf \( \gamma \in \Gamma \) is called unrefined leaf if and only if it doesn’t point to a sub-tree rooted at \( \gamma \).*

The basic difference between an identifying DT classifier and the BDT is that BDT has pointers to the subtrees within each leaf that might eventually point to a specific subtree. Rigorously speaking, a BDT which has all unrefined leaves is identical to an identifying DT classifier in Definition 3.0.16.

4.2 Distributed Decision Tree Classifier

As we have mentioned in the beginning of this chapter, we have a client server scenario which is one of the basic cases of distributed architecture. The data is kept on the server side respecting \( \ell \)-diversity through anatomization whereas the client can retrieve the data and reconstruct the original data. Thus, we propose a distributed decision tree algorithm that involves both parties: client and server.

The idea is very simple. In a nutshell, we let the server learn a BDT classifier with all unrefined leaves and let the client create a refined leaf from the unrefined leaf while it is making a prediction on the test data. In some sense, this is an “on-the-fly” learning algorithm which builds parts of the model while the prediction process is going on. Our objective is to reduce the runtime cost of training for the client while keeping the predictive power high.

Next, we will explain the algorithm in detail. Section 4.2.2 will discuss the preservation of \( \ell \)-diversity and Section 4.2.3 will conclude the discussion with costs for the client.
4.2.1 Algorithm

Given an anatomized training data $D_A$ satisfying Definition 4.1.1 that is stored on the cloud server, the cloud server builds a BDT classifier with respect to Definition 4.1.2. Then, the BDT leaves $\Gamma$ are improved by the client. For every leaf $\gamma \in \Gamma$, the improvement is a sub-tree learned by the client. The client uses the instances in leaf $\gamma$ to learn an improved sub-tree and eventually $\gamma$ becomes a refined leaf of the BDT (cf. Definition 4.1.3). Figures 4.1, 4.2 and 4.3 give pseudo code of distributed DT classifier algorithm involving both parties server and client. Throughout the pseudo codes, we assume that both client and server know each other’s ip addresses.

```
Main():
begin:
    BDT:= LearnDT(IT)
    while(TRUE):
    begin:
        x:=getInstance(client_ip)
        FindBDTLeaf(BDT,x)
    end
end
```

Figure 4.1.: Distributed DT Classifier Main Function (Called by Server)

Figure 4.1 shows the Main() function that is called by server to initiate distributed DT learning. The LearnDT(IT) function call builds the BDT from the IT partition. The client improvements on the BDT are made on the fly when doing predictions. To make a prediction for an instance $x$, the client sends $(x.a_1, \ldots, x.a_d)$ to the server. getInstance() function call receives the instance $x$. FindBDTLeaf(BDT,x) function is called by server for every $x$ once $x$ is received.

Figure 4.2 shows the FindBDTLeaf() function that is called by the server within the function Main(). This function is essentially the interaction point between client and server for the potential improvements to make a leaf refined. This function finds
FindBDTLeaf(BDT, x):
Input:
  BDT: base decision tree on server
  x: predicted instance
begin:
  γ := findLeaf(BDT, x)
if (γ.hasEncryptedSubtree() == TRUE):
begin:
  encrSubtree := γ.getEncryptedSubtree()
  sendMessage(encrSubtree)
end
else:
begin:
  leafIns := γ.getInstances()
  leafIns := joinTables(leafIns, ST, GID)
  sendMessage(leafIns)
end
end

Figure 4.2.: Distributed DT Classifier FindBDTLeaf Function (Called by Server)

the appropriate BDT leaf $\gamma \in \Gamma$ for the client to improve (if not already done) and sends leaf $\gamma$ to the client. Given test instance $x$ and the BDT, the findLeaf(BDT, x) function call finds the respective leaf $\gamma \in \Gamma$ using a usual decision tree inference with attribute values $(x.a_1, \cdots, x.a_d)$. Then, FindBDTLeaf verifies if $\gamma$ points to an encrypted sub-tree that was previously learned by a client (if statement in pseudocode). (The subtree is encrypted to ensure privacy constraints are satisfied, this will be discussed in more detail later) If $\gamma$ points to an encrypted sub-tree, it sends to the client the encrypted sub-tree as a response message (sendMessage(encrSubTree) function call). Otherwise, server sends the tuples belonging to $\gamma$ (sendMessage(leafIns) function call). Note that there is a function call joinTables(leafIns, ST, GID) before sending the instances of the leaf. This function matches every instance $x \in \gamma$ with all $\ell$ potential sensitive attributes using the $GID$ field as the join key (a group-level join, giving all possible instances provided the $\ell$-diverse dataset, not just the true matching values which the server doesn’t know). The client will be responsible
for finding the correct matching sensitive attribute for every instance \( x \in \gamma \) and use this to make further splits from the leaf \( \gamma \).

```
Inference(server_ip, x, key):
Input:
  server_ip: ip address of server
  x: predicted instance
  key: encryption key
Output:
  clsLbl: class label predicted
begin:
  sendInstance(server_ip, x)
  message:=receiveMessage(server_ip)
  if (message==encrTree):
    begin:
      subTree:=decipherTree(encrTree, key)
      \( \gamma' := \text{findLeaf}(\text{subTree}, x) \)
      leafIns:=\( \gamma' \).getInstances()
      clsLbl:=majorityLabel(leafIns)
    end
  if (message==leafIns):
    begin:
      instances:=trueInstances(leafIns, ST)
      subTree:=learnDT(instances)
      encrSubtree:=encryptTree(subTree, key)
      updateBDT(server_ip, encrSubtree)
      \( \gamma' := \text{findLeaf}(\text{subTree}, x) \)
      leafIns:=\( \gamma' \).getInstances()
      clsLbl:=majorityLabel(leafIns)
    end
  return clsLbl
end
```

Figure 4.3: Distributed DT Classifier Inference Function (Called by Client)

Figure 4.3 shows the Inference() function that is called by the client. Client makes predictions on the fly improvements using the Inference() function. It sends a test instance \( x \) (only with attributes \((x.a_1, \ldots, x.a_d)\)) to the cloud server using sendInstance() function. Then, client calls receiveMessage() to receive message sent by function FindBDTLeaf(). As mentioned in the FindBDTLeaf() discussion (cf. Figure 4.2), the message can be either an encrypted sub-tree (encrTree) or instances of leaf \( \gamma \) (leafIns). leafIns is in fact all instances of \( \gamma \) such that every instance is matched with all \( \ell \) potential sensitive attribute values. If the message is
an encrypted sub-tree, \texttt{Inference()} just decrypts the encrypted sub-tree using the \texttt{decipherTree()} function, finds the appropriate leaf \( \gamma' \) as in regular tree inference using the \texttt{findLeaf()} function, and predicts the class label of test instance \( x \) by taking majority of the class labels in \( \gamma' \) using the \texttt{majorityLabel()} function. If \texttt{message} is \texttt{leafIns}, the \texttt{Inference()} function reconstructs the original instances (\texttt{instances}) from \texttt{leafIns} using \texttt{ESEQ} and \texttt{SEQ} in Definition 4.1.1. This operation is done in function call \texttt{trueInstances()}. Nergiz et al. explains in detail how this reconstruction works in the context of a data outsourcing scenario in the client server model [60, 61]. To summarize, the client decrypts the encrypted sequence numbers \texttt{ESEQ} in identifying attributes of \texttt{leafIns} to find true sequence numbers and eliminates the instances of \texttt{leafIns} that don’t have sensitive table sequence numbers \texttt{SEQ} the same as the true sequence numbers. After this reconstruction, \texttt{Inference()} learns a sub decision tree (\texttt{subTree}) from \texttt{instances} using the \texttt{learnDT()} function, encrypts \texttt{subTree} (\texttt{encryptTree()} function call) and sends \texttt{subTree} back to the server. Server stores client’s encrypted subtree for future predictions. Finally, \texttt{Inference()} finds the appropriate leaf \( \gamma' \) as in regular tree inference using the \texttt{findLeaf()} function and predicts the class label of instance \( x \) by taking majority of the class labels in \( \gamma' \) using \texttt{majorityLabel()} function.

4.2.2 Privacy Preservation

We now discuss the preservation of \( \ell \)-diversity at the cloud server after learning the distributed DT classifier. The learning algorithm in Section 4.2.1 has two inference channels through the learning process:

1. The cloud server learns a BDT.

2. The client, owner of the data, creates refined leaves from the unrefined leaves of BDT.

The preservation of \( \ell \)-diversity here involves whether any true match between \( IT \) and \( ST \) occurs after the two steps above. For step 1, there is no revealed match as
the BDT is composed of splits that are for the attributes \( A_1, \ldots, A_d \) in the schema of \( IT \). The process doesn’t involve any matching. For step 2, the reconstruction of the original instances within the leaf \( \gamma \) in BDT is done at the client. In addition, the client encrypts the learned subtree that could include splits with the sensitive attribute and then sends it back to the server. Unless the server obtains the encryption key of the client, it doesn’t learn any information from the encrypted subtree that it stores in the refined leaves. Hence, the \( \ell \)-diversity of the anatomized training data is preserved.

4.2.3 Costs for Client

The cost discussion for both training and inference of the distributed DT classifier includes multiple factors such as network communication latency cost, the encryption/decryption overheads, the computation complexity of creating splits and the generalization ability. Here, we will focus on the computation complexity of creating splits. As the decision trees are empirical models, we don’t discuss here the generalization ability of the distributed DT classifier, unlike the other classifiers in the next chapters. We will also ignore any cost incurring the server as the cloud servers are usually assumed to have infinite processing power [60,61].

It is not possible to come up with a theoretical upper bound on the client’s training cost in distributed DT classifier that is tighter than the upper bound of client’s training cost in the original DT classifier. As the leaves \( \Gamma \) of BDT must have fewer instances than the entire original training set and some good predictors in \( IT \) must have been used in building the BDT, the client’s training cost of distributed DT classifier should be less than the training cost of original DT classifier. On the other hand, if all the identifying attributes in \( IT \) are bad predictors of the class label, then BDT wouldn’t have many splits, maybe not even one split. In this case, it would be more meaningful for the client to just learn an original DT classifier itself.

The inference cost has similar issues to the training cost in terms of theoretical upper bounds. If the identifying attributes in the \( IT \) table are good predictors, the
BDT would already have some significant generalization ability. In the end, the refined leaves created by clients wouldn’t have many splits other than the sensitive attribute itself (maybe not even that). Therefore, the length of inference path in distributed DT classifier would be mostly located at server. The length of inference path in the distributed DT classifier would be less than the original DT classifier one’s length for the client. If the identifying attributes in IT are bad predictors, the client would have to do the splits using the sensitive attribute and the length of the inference path in distributed DT classifier wouldn’t be much less than the length of inference path in the original DT classifier.

4.3 Experiments and Results

In the experiments, we will be comparing the distributed DT classifier with the original DT classifier and identifying DT classifier. We will start discussion by prerequisites. We will then discuss about the results.

4.3.1 Prerequisites

Datasets

We are testing the distributed DT classifier on four datasets of the UCI repository [62].

1. The Adult dataset is composed of US census data. An individual’s income is the class attribute (more than 50K vs less than 50K). It has 48842 tuples where each tuple has 15 attributes. Here, the sensitive attribute is considered to be the attribute age and relationship in different experiments. We also considered individuals in non-private work class, so the experiments are made on a subsample of 14936 instances.

2. The Vote dataset contains 485 tuples where each tuple has 16 binary attributes. An attribute is the vote of a senator in a session. A senator’s party affiliation is
the class attribute (Democrat vs. Republican). Here, the attribute for the vote session physician freeze is considered a sensitive attribute.

3. The Autos datasets contains 205 tuples where each tuple has 26 attributes. The class label is either symboling (risk rating) or the price of a car. The autos datasets continuous attributes are discretized in our experiments and a binary class label is created from price of a car (low price vs. high price). The attribute symboling (risk rating) is set as the sensitive attribute.

4. The Australian credit dataset has 690 tuples where each tuple has 16 attributes. The class attribute is whether a credit application is approved or not (+ vs. −). The sensitive attribute in our experiments is set to A9.

While privacy may not be a real issue for some of these datasets, we chose these four datasets because they are large enough to demonstrate performance differences, and reasonably challenging for decision tree learning. True privacy-sensitive data is hard to obtain and make public experiments on, precisely because it is privacy-sensitive.

Privacy Setup

The anatomization was done according to Xiao et al.’s bucketization algorithm [11]. When the $\ell$-diversity condition is not satisfied, the instances were divided into groups of size $\ell$ according to the original bucketization algorithm. Leftover instances were suppressed (not used in training models). The $\ell$ value was set to 2 in the experiments.

The sensitive attributes were chosen such that these attributes have the most significant predictor power for the class attribute. In the experiments, we wanted to simulate the worst case for the models in terms of the sensitive attributes’ predictive power. The only exception here is the attribute relationship in adult dataset which is
a moderate predictor of the class attribute. The objective here was to see the effect of different sensitive attributes with different predictive power on a fixed dataset.

Model Evaluation Setup

10-fold cross validation is done on all the datasets and the accuracy was measured. Weka J48 is used for decision tree learning with reduced error pruning. 20% of the training sets are used for reduced error pruning. Given a training/test pair of 10-fold cross validation, the prune set is chosen randomly once so that the original, distributed and identifying DT classifier are compared within the same model space (same pruning set and training set for different types of DT classifiers). The experiments on vote and australian credit datasets learn binary decision trees whereas the experiments on auto and adult datasets learn non-binary decision trees. Experiments are done using a physically remote server; and a laptop with Intel i5 processor and 4 GB RAM. Internet connection speed was 100 Mbps. AES-128 is chosen for symmetric key encryption.

In the distributed DT classifier, we also measure memory savings and execution time savings for a client. We will define below these evaluation metrics.

\[
ets = \frac{\text{distributed DT classifier execution time}}{\text{original DT classifier execution time}}
\]

\[
ms = \frac{\text{distributed DT classifier memory requirement}}{\text{original DT classifier memory requirement}}
\]

Equations 4.1 and 4.2 calculate execution time savings (ets) and memory savings (ms) respectively in the experiments. Savings are maximized as both formulas approach to zero. In contrary, savings are minimized as both formulas approach to 1.

Execution times (Eq. 4.1) include client’s encryption/decryption, network and learning/inference costs on a training set and test set pair (ith iteration of cross validation). In equation 4.2 the number of tuples in training set is the original DT classifier’s memory requirement whereas the number of tuples in the BDT’s biggest
leaf is the distributed DT classifier’s memory requirement. Since the client learns sub-trees on the fly from the unrefined leaves, the biggest leaf is the memory upper bound.

4.3.2 Analysis of Results

![Figure 4.4: Execution Time Savings as a Fraction of Original DT Time](image.png)

Figures 4.4 and 4.5 provide box plots showing elapsed time and memory savings measurements on each of the 10 folds. Identifying DT classifier is not shown, as the client cost is 0. The blue dots on the boxplots show the mean time and memory savings. Savings graphs exhibit visually the tradeoff between ms and ets. Given a dataset, if the memory savings are high, the time requirements are low (as expected). This is expected since the high ms indicates a complex BDT in distributed DT classifier. The client needs to do more improvement in distributed DT classifier since the
Figure 4.5.: Memory Savings as a Fraction of Original DT Memory Requirement

leaf provided is already a bad predictor. In addition, high ms and low ets can lead to complex decision trees that can produce overfitting. On the other hand, low ms indicate a simple BDT in distributed DT classifier. Little remains to be done at the client since the leaf provided is already a good predictor.

Figure 4.6 provides measured prediction accuracies in the experiments. On average, the original DT classifiers have the best accuracy, as expected. Identifying DT classifiers have the worst accuracy and distributed DT classifiers are generally somewhere between. This accuracy trend is expected since identifying DT classifiers are identical to the BDT in distributed DT classifier. So, the distributed DT classifier’s accuracy values show the effect of client improvements which are done to the identifying DT classifier. The exception is in the experiment with adult dataset where the sensitive attribute is relationship. The sensitive attribute is a moderate
predictor. This exceptional case shows that the distributed DT classifier becomes too complex after client improvement (overfitting). Execution time savings and memory savings graphs justify this observation (Figures 4.4 and 4.5). A possible solution to avoid distributed DT classifier overfitting is a memory savings threshold (threshold for BDT leaf size on server). However, it is hard to define an exact threshold value. Given a fixed training set on server, distributed DT classifier can be learned with various BDT thresholds. The threshold having the best accuracy on the pruning set can be chosen and client improvements can be applied on this model.
5 NON-PARAMETRIC CLASSIFICATION FOR $\ell$-DIVERSITY

In this chapter, we will consider a theoretical approach for data classification in a non-distributed learning scheme. In this scheme, we assume again that the data is released to a third party. However, anybody who has access to this data could learn models. We will still aim to solve the Problem 2.1.1. However, the owner of the released data is assumed to be not involved with training models. We will present an anatomized k-nearest neighbor (k-NN) classifier that is trained on the anatomized training data according to Definitions 3.0.10 and 3.0.13. The test data will be in original form as before (no uncertainty of $\ell$-diversity).

The chapter is organized as follows. First, we will outline the notations and remind the results of the theoretical analysis of original k-NN classifier. In Section 5.2, the anatomized k-NN classifier will be presented in detail in addition to the theoretical properties. Section 5.3 will conclude the chapter with the experiment results on the publicly available datasets. The discussion expands that of Mancuhan et al. [64].

5.1 Notations

We now give the notations that will be used in the rest of this chapter. $D$ will denote the original training data, whereas $D_A$ will denote the anatomized training data. $D$ has $N$ instances and $D_A$ has $N\ell$ instances from definition 3.0.11. All instances are i.i.d whether they are in training or test data. The total number of attributes are assumed to be $d+1$ ($d$ identifying attributes and 1 sensitive attribute.) For the sake of simplicity, $A_{id}$ will denote the identifying attributes $A_1 \cdots A_d \in IT$. $T$ stands for test data which are not processed by any anatomization and generalization method. $x$ will be an instance of the test data $T$. $d(U,V)$ is the quadratic distance metric for a pair of instances $U$ and $V$ in metric space $M \subset \mathbb{R}^{d+1}$. $x_N'(k)$ denotes the set of $k$
nearest neighbors of \( x \) in \( D \) that the original k-NN classifier uses while \( x'_N(k) \) denotes the set of \( k \) nearest neighbors of \( x \) in \( D_A \) that the anatomized k-NN classifier uses. \( x_i \) will interchangeably be an instance of \( D \) or \( D_A \) and \( x_j \) will interchangeably be an instance of \( x'_N(k) \) or \( x'_{N_A}(k) \). In case of \( k = 1 \), we will use \( x'_N \) and \( x'_{N_A} \) for the nearest neighbors in \( D \) and \( D_A \). \( X \) is the random variable with probability distribution \( P(X) \) from which \( x \) and \( x_i \) are drawn. Training and test instances will be column vectors in format of \( (A_1, ..., A_d, A_s)^T \). \( C \) is the class attribute in \( D \) and \( D_A \) with binary labels 1 and 2.

We also recall the notations from the statistical pattern recognition that will be used in the theoretical analysis \[65]. In the probability notations and proofs, we will abuse \( x \) to represent \( X = x \). Given the training data \( D \) and the class label \( i \); \( q_i(x) \), \( P_i(x) \) and \( P_i \) stand for the posterior probability, the likelihood probability and the prior probability respectively. If the anatomized training data \( D_A \) is used, \( q_A_i(x) \), \( P_{A_i}(x) \) and \( P_{A_i} \) are the symmetric definitions for the class label \( i \). When \( \forall x \in T \) hold, the probability of classifying \( x \) incorrectly will be referred to an error rate. \( R(x'_N(k), x) \) is the error rate when \( x \in T \) is classified using \( x'_N(k) \). If \( x'_{N_A}(k) \) is used to classify \( x \), \( R_A(x'_{N_A}(k), x) \) will be the error rate. When \( \forall x_j \in x'_N(k); x_j \equiv x \) hold, we denote the error rate by \( R^k(x) \) in \[5.1\].

\[
R^k(x) = \sum_{i=1}^{k+1/2} \frac{1}{i} \binom{2i - 2}{i - 1} [q_1(x)q_2(x)]^i \\
+ \frac{1}{2} \binom{k + 1}{k + 1/2} [q_1(x)q_2(x)]^{k+1/2}
\] (5.1)

\( R^k_A(x) \) is the error rate when \( \forall x_j \in x'_{N_A}(k); x_j \equiv x \) holds. \( R^k_A(x) \) can trivially be derived from \[5.1\] by substituting \( q_i(x) \) with \( q_{A_i}(x) \). For all \( x \in T \), the smallest error rate that could be obtained from the best possible classifier will be referred by Bayesian error \[65\]. The Bayesian errors given \( x \) are denoted by \( R^*(x) \) and \( R^*_A(x) \)
when \( \forall \ x_j \in x'_N(k); x_j \equiv x \) and \( \forall \ x_j \in x'_{N\ell}(k); x_j \equiv x \) hold respectively. \[ \text{(5.2)} \]

computes \( R^*(x) \).

\[
R^*(x) = \min\{q_1(x), q_2(x)\} \\
\cong \sum_{i=1}^{\infty} \frac{1}{i} \left(\frac{2i - 2}{i - 1}\right) [q_1(x)q_2(x)]^i \tag{5.2}
\]

\( R^*_A(x) \) can trivially be derived again from \[ \text{(5.2)} \] by substituting \( q_i(x) \) with \( q_{A_i}(x) \). \( R^k \) and \( R^k_A \), which are the respective expectations of \( R^k(X) \) \((E\{R^k(X)\})\) and \( R^k_A(X) \) \((E\{R^k_A(X)\})\) with respect to \( X \), will stand for the error rate of original k-NN and anatomized k-NN classifiers respectively. \( R^* \) and \( R^*_A \), which are the respective expectations of \( R^*(X) \) \((E\{R^*(X)\})\) and \( R^*_A(X) \) \((E\{R^*_A(X)\})\) with respect to \( X \), will stand for the Bayesian errors of the original training data and anatomized training data respectively. We will denote \( R^1(x) \) and \( R^1_A(x) \) by \( R(x) \) and \( R_A(x) \) for convenience. Similarly, \( R \) and \( R_A \) will denote \( R^1 \) and \( R^1_A \).

We finish this section with the assumptions in the theoretical analysis. We assume that all the training data has a smooth probability distribution. Although anatomization requires a discrete probability distribution for the sensitive attribute \( A_s \), such smoothness violation is negligible since the original k-NN classifier is known to fit well on discrete training data [66]. The sensitive attribute \( A_s \) is assumed to be non-binary. The binary case would be meaningless in reality for \( \ell \)-diversity since \( \ell \) can only be 2, implying a coin flip privacy guarantee on the sensitive attribute. The anatomized k-NN is assumed to have odd \( k \) values, because even \( k \) values encompass the tie cases among \( x'_N(k) \) and \( x'_{N\ell}(k) \) that make the bounds ambiguous and complicated [65]. All instances are assumed to be in a separable metric space \( M \subset \mathbb{R}^{d+1} \) following Cover et al., Devroye et al. and Fukunaga et al. [65,67,68]. Last, the theoretical analysis will assume that the original training data \( D \) satisfies \( \ell \)-diversity condition and that every group has \( \ell \) instances in the creation of anatomized training data \( D_A \) from the original training data \( D \). If this assumption is violated in practice, we will be using suppression.
5.2 Anatomized k-NN Classifier

5.2.1 Illustration

We will illustrate the anatomized k-NN classifier through the example in Figure 5.1. Although the example is for an anatomized 3-NN classifier, the procedure is general for any “k” value (cf. Definition 3.0.13).

Figure 5.1(a) shows the original training data with six instances: two instances of a blue class (on the left side) and four of a red class (on the right side), with two attributes \( A_1 \) and \( A_2 \). \( A_1 \) is identifying attribute and \( A_2 \) is sensitive attribute. Here, every instance has a different shape color since it belongs to a unique individual with unique \( A_1 \) value. Figure 5.1(b) shows the anatomized training data created from \( IT(A_1, GID) \) and \( ST(GID, A_2) \) when \( \ell = 2 \) (cf. Definitions 3.0.10 and 3.0.11). Note that each point is duplicated with two possible values for \( A_2 \). It has 12 instances in total.

Figure 5.2 shows the anatomized 3-NN classifier in the toy example. To classify an unlabeled test instance (black + in Figure 5.2(a)), the anatomized 3-NN classifier finds 3 instances from the anatomized training data that are closest to the test instance.
In Figure 5.2(b), the 3 points that are located in the green circle are the 3 closest points to the test instance. Since the test instance’s true class label is red, we obtain the true classification despite using 2 noisy instances (incorrect matches) and 1 true instance (correct match.) Since the training data is separable in the toy example and the test instance comes from the subspace where the red instances are clustered, the correct classification is obtained despite the distortion of $\ell$-diversity.

Note that the $\ell$-diversity’s distortion might be much more critical in practice than our toy example’s. For example, such distortion could transform the linearly separable original training data to non-linearly separable anatomized training data. This could hence result in incorrect classifications as the distortion changes the likelihood probabilities that define the generalization error of any non-parametric classifier. As long as the $\ell$-diversity’s distortion on the likelihood probabilities is minimized, the generalization ability of anatomized $k$-NN is likely to be same as the original $k$-NN’s. Before elaborating on the theoretical aspects of $\ell$-diversity’s perturbation, we will discuss the implementation details and complexity, and the preservation of $\ell$-diversity.
5.2.2 Implementation

The anatomized k-NN classifier has two phases of implementation. First is the creation of the anatomized training data whereas the second phase is the implementation of the original k-NN classifier to train on the anatomized training data. The first phase is the inner join operation between the IT and ST tables and the projection operation on the former inner join result (cf. Definition 3.0.11). The inner join operation can be implemented in $O(N \log N)$ time whereas the projection operation can be implemented in $O(N)$. Hence, the anatomized training data creation can be implemented in $O(N \log N)$ time [69]. The second phase is the k-NN classifier implementation. The naïve approach of finding the exact k neighbors requires going through the entire anatomized training data which takes $O(N \ell \left(d + 1\right))$ time. LSH based algorithms, on the other hand, can compute the $c$-approximate k nearest neighbors in $k \frac{1}{c^2} + O(k \log \left(\frac{\log N \ell}{\log^2 N \ell} \right))$ time [70].

5.2.3 Privacy Preservation

Correct or incorrect classification from the anatomized k-NN classifier doesn’t help an adversary learn additional information about the sensitive attribute’s distribution within a group. It is possible to have a correct classification from the k closest training instances even if these are all incorrect matches. In addition, the original training data might be linearly separable whereas the anatomized training data could be linearly inseparable. Anatomized k-NN classifier just uses k “plausible” instances in the anatomized training data which is more biased according to $\ell$-diversity’s distortion (cf. Section 5.2.4). Thus, the anatomized k-NN classifier preserves $\ell$-diversity.

5.2.4 Theoretical Analysis

We will analyze the anatomized k-NN classifier in terms of the asymptotical error rate bounds (infinite size training data), error rate convergence on the finite size
training data, and the generalization error. The analysis is based on that of Cover et al. and Fukunaga [65, 67], extended to address \( \ell \)-diversity.

The convergence analysis will be limited to the anatomized NN classifier since the original k-NN classifier is analyzed only for 1 and 2 neighbors. The analysis of 2 neighbors is omitted here since it directly follows the 1 neighbor case.

The generalization error will follow the non-parametric density estimation classifier to keep the discussion easy to follow. The analysis result is general for k-NN classifiers and any density based non-parametric classifier [65].

Asymptotic Error Rate Bounds

We will first show the error bounds for the anatomized NN classifier. We will then discuss the extension to the anatomized k-NN classifier for all odd \( k > 1 \). Corollary 5.2.1 expresses formally the convergence of the nearest neighbor which is critical for the error bounds of the anatomized NN classifier.

**Corollary 5.2.1** Let \( x \in T \) and \( x_1, \ldots, x_N \in D \) be i.i.d instances taking values separable in any metric space \( M \subset \mathbb{R}^{d+1} \). Let \( x'_{N} \) be the nearest neighbor of \( x \) in \( D \). Then, \( \lim_{N \to \infty} x'_{N} = x \) with probability one [67].

**Proof** Let \( S_x(r) = \{ \bar{x} \in M : d(x, \bar{x}) \leq r \} \) be the sphere with radius \( r > 0 \) centered at \( x \). Let’s consider that \( x \) has a sphere \( S_x(r) \) with non-zero probability. Therefore, for any radius \( \delta > 0 \);

\[
\lim_{N \to \infty} P\left\{ \min_{i=1, \ldots, N} d(x_i, x) \geq \delta \right\} = \lim_{N \to \infty} [1 - P(S_x(\delta))]^N = 0 \quad (5.3)
\]

Basically, Corollary 5.2.1 says that if the original training data has an infinite number of instances, it is guaranteed to find the nearest neighbor of a test instance that is drawn from the same probability distribution.

Obviously, the nearest neighbor could have either a correct or an incorrect sensitive attribute if we use the anatomized training data instead of the original training
data. Although the sensitive attribute value could change the specific instance which becomes the nearest neighbor in terms of the distance, there would still be a nearest neighbor if the original has an infinite number of instances. Finding the nearest neighbor from the anatomized training data is equivalent to finding the nearest neighbor from $D$ where one attribute value is swapped in the multivariate distribution. Next, Theorem 5.2.1 shows the error bounds of the anatomized NN classifier under this assumption from Corollary 5.2.1.

**Theorem 5.2.1** Let $M \subset \mathbb{R}^{d+1}$ be a metric space. Let $P_{A_1}(x)$ and $P_{A_2}(x)$ be the likelihood probabilities of $x$ such that $P_A(x) = P_{A_1}P_{A_1}(x) + P_{A_2}P_{A_2}(x)$ with class priors $P_{A_1}$ and $P_{A_2}$. Finally, we assume that $x$ is either a point of non-zero probability measure or a continuity point of $P_{A_1}(X)$ or $P_{A_2}(X)$ with probability 1 [67]. Then, the nearest neighbor has the probability of error $R_A$ with the bounds

$$R_A^* \leq R_A \leq 2R_A^*$$  \hspace{1cm} (5.4)

where $R_A^*$ denotes the Bayesian error when the anatomized training data $D_A$ is used.

We now give a sketch of proof for Theorem 5.2.1. Let $R_A(x_{N\ell}, x)$ denote the probability of error for a pair of instances $x \in T$ and $x_{N\ell} \in D_A$. From Corollary 5.2.1, we can assume that $\lim_{N \to \infty} x_{N\ell} = x$ always hold. Using the assumptions of the theorem and (5.1), with $k = 1$ and $q_{A_i}(x)$ instead of $q_i(x)$, (5.5) holds with probability 1.

$$\lim_{N \to \infty} R_A(x_{N\ell}, x) \cong R_A(x) = 2q_{A_1}(x)q_{A_2}(x)$$  \hspace{1cm} (5.5)

The rest of the derivation follows Cover et al. using (5.1), (5.2) [67].

Extending (5.4) from the anatomized NN classifier to the anatomized k-NN classifier for all odd $k > 1$ follows the steps in Corollary 5.2.1 and Theorem 5.2.1. The key is to show that $\lim_{N \to \infty} x_j = x$ holds for all $x_j \in x_{N\ell}(k)$. The rest is to derive an expression of $R_A^k(x)$ as in (5.5) for all odd $k > 1$ and show that $R_A^k(x)$ is always less than $2R_A^*$ and $R_A^{k-2}(x)$. This can be derived following the analysis of original k-NN classifier in Fukunaga [65]. The anatomized k-NN classifier has the bound (5.6).

$$\forall \text{ odd } k > 1; R_A^* \leq \cdots \leq R_A^5 \leq R_A^3 \leq R_A \leq 2R_A^*$$  \hspace{1cm} (5.6)
Note that the Bayesian errors $R_A^*$ and $R^*$ are not always same due to the $\ell$-diverse groups of the anatomization. The $\ell$-diverse groups cause new likelihood $P_{A_i}(x)$ and eventually posterior probabilities $q_{A_i}(x)$. $R_A^*$ thus differs from (5.2), because (5.2) uses $q_i(x)$ instead of $q_{A_i}(x)$. We will return back to this in the generalization error discussion.

Error Rate Convergence on Finite Size Training Data

We now discuss the error rate of the anatomized NN classifier when the size of anatomized training data is finite. This error rate will let us derive the convergence rate to the Bayesian error for the anatomized NN classifier. The discussion here won’t be generalized to the anatomized k-NN classifier since the finite size training data performance of k-NN classifiers are not generalized to 3 or more neighbors in the pattern recognition literature [65,68]. Theorem 5.2.2 extends the analysis of Fukunaga and Fukunaga et al. [65,71].

Theorem 5.2.2 Let $M \subset \mathbb{R}^{d+1}$ be a metric space. Let’s assume that the original training data $D$ satisfies $\ell$-diversity condition and that every group has $\ell$ instances in the creation of anatomized training data $D_A$ from the original training data $D$. Let $P_A(X)$ and $P(X)$ be the smooth density functions of $x$. Let $P_{A_1}(X)$ and $P_{A_2}(X)$ be the class likelihood density functions of $x$. Let $P_{A_1}$ and $P_{A_2}$ be the class priors such that $P_{A}(x) = P_{A_1}P_{A_1}(x) + P_{A_2}P_{A_2}(x)$. Let $q_{A_1}(x)$ and $q_{A_2}(x)$ be the smooth posterior probability densities such that $q_{A_1}(x) + q_{A_2}(x) = 1$ and $N\ell \rightarrow \infty$. Let $q_{A_1}(x'_{N\ell})$ and $q_{A_2}(x'_{N\ell})$ be the smooth posterior probability densities such that $q_{A_1}(x'_{N\ell}) + q_{A_2}(x'_{N\ell}) = 1$ and $N\ell \rightarrow \infty$. Let $\delta > 0$ be the difference between $q_{A_1}(x)$ and $q_{A_1}(x'_{N\ell})$ for class labels $i = \{1, 2\}$. Let $d(x'_{N\ell}, x)$ be the quadratic distance with matrix $A$ and $\rho$ be the calculated value of $d(x'_{N\ell}, x)$. Let $R_A$ be the error rate of the anatomized NN classifier when $N\ell \rightarrow \infty$. Last, let $R_{AN}$ be the error rate of the anatomized NN classifier when $N\ell \rightarrow \infty$. Then,

$$R_{AN} \cong R_A + \beta \frac{1}{(N\ell)^{2\pi + 1}} E_X \{ |A|^{-\frac{1}{\pi + 1}} \text{tr}\{AB(x)\} \}$$

(5.7)
where \( \beta \) is
\[
\beta = \frac{\Gamma(\frac{\nu+2}{2})\Gamma\left(\frac{\nu}{2d+2}+1\right)}{\pi(d+1)} \tag{5.8}
\]
and \( B(x) \) is
\[
B(x) = P_A^{-\frac{2}{\nu+1}}(x)[q_{A_2}(x) - q_{A_1}(x)]
\times \left[ \frac{1}{2} \nabla^2 q_{A_1}(x) + P_A^{-1}(x)\nabla P_A(x)\nabla^T q_{A_1}(x) \right] \tag{5.9}
\]

**Proof** We first define \( q_{A_1}(x'_{Nt}) \) in function of \( q_{A_1}(x) \) and \( \delta \).

\[
q_{A_1}(x'_{Nt}) = q_{A_1}(x) + \delta \tag{5.10}
\]
\[
q_{A_2}(x'_{Nt}) = q_{A_2}(x) - \delta \tag{5.11}
\]

\( R_{Nt} \) is written in function of \( R_A \) and \( \delta \) using (5.10) and (5.11) in (5.12)
\[
R_{Nt} = E\{q_{A_1}(x)(q_{A_2}(x) - \delta) + q_{A_2}(x)(q_{A_1}(x) + \delta)\}
= R_A + E[[q_{A_2}(x) - q_{A_1}(x)]\delta]
\tag{5.12}
\]
where \( E[[q_{A_2}(x) - q_{A_1}(x)]\delta] \) is (5.13)
\[
E\{(q_{A_2}(x) - q_{A_1}(x))\delta)\} = E_X\{E_{x_{Nt}}\{[q_{A_2}(x) - q_{A_1}(x)]\delta|\rho, x]\}
= E_X\{[q_{A_2}(x) - q_{A_1}(x)]E_{x_{Nt}}\{\delta|\rho, x]\}
\tag{5.13}
\]

Following Fukunaga, the last line of (5.13) requires a 3-step expectation calculation. Step 1 gives
\[
E_{x_{Nt}}\{\delta|\rho, x\} \cong \frac{\rho^2}{d+1} \times tr\{A\left[ \frac{1}{2} \nabla^2 q_{A_1}(x) + P_A^{-1}(x)\nabla P_A(x)\nabla^T q_{A_1}(x) \right]\}
\tag{5.14}
\]

Step 2 uses (5.14) to calculate \( E_{\rho}\{E_{x_{Nt}}\{\delta|\rho, x]\}|x\}. \) This eventually requires the computation of \( E\{\rho^2\}. \) Although the probability distribution of \( \rho \) is unknown, the probability distribution of the local region \( u \) around test instance \( x \in T \) including the nearest neighbor \( x'_{Nt} \) is known. We therefore need to formulate \( \rho^2 \) in function of \( u^2 \). The approximation of \( u \) as a function of \( \rho \) is given in (5.15).
\[
u \cong p(x)\frac{\pi^{\frac{d+1}{2}}}{\Gamma(\frac{d+2}{2})} \rho^{d+1} |A|^{\frac{1}{2}} 
\tag{5.15}
\]
Rewriting (5.15) and taking the expectation of both sides result in (5.16).

\[ E\{\rho^2\} = \frac{\Gamma(\frac{d+3}{2})}{p^{d+1}(x)\pi|A|^{\frac{d}{d+1}}} E\{u^{\frac{2}{d+1}}\} \]  \hspace{1cm} (5.16)

(5.17) computes \( E\{u^{\frac{2}{d+1}}\} \):

\[ \int_0^1 u^{\frac{2}{d+1}} P_u(u)du = N\ell \int_0^1 u^{\frac{2}{d+1}} (1-u)^{N\ell-1}du = \frac{\Gamma(\frac{2}{d+1} + 1)\Gamma(N\ell + 1)}{\Gamma(N\ell + \frac{2}{d+1} + 1)} \]  \hspace{1cm} (5.17)

Replacing the result of (5.17) in (5.16) gives (5.18).

\[ E\{\rho^2\} = \frac{\Gamma(\frac{d+3}{2})\Gamma(\frac{2}{d+1} + 1)}{p^{d+1}(x)\pi|A|^{\frac{d}{d+1}}} \frac{\Gamma(N\ell + 1)}{\Gamma(N\ell + \frac{2}{d+1} + 1)} \]  \hspace{1cm} (5.18)

Assuming \( N \) and \( d \) have values large enough, \( \frac{\Gamma(N\ell + 1)}{\Gamma(N\ell + \frac{2}{d+1} + 1)} \) is approximated in (5.19).

\[ \frac{\Gamma(N\ell + 1)}{\Gamma(N\ell + \frac{2}{d+1} + 1)} \approx \frac{N\ell}{N\ell + \frac{2}{d+1}} \times \frac{\Gamma(N\ell)}{\Gamma(N\ell + \frac{2}{d+1})} \approx \frac{1}{(N\ell)^{\frac{d}{d+1}}} \]  \hspace{1cm} (5.19)

Replacing the result of (5.19) in (5.18) results in (5.20).

\[ E\{\rho^2\} \approx \frac{\Gamma(\frac{d+3}{2})\Gamma(\frac{2}{d+1} + 1)}{p^{d+1}(x)\pi|A|^{\frac{d}{d+1}}} \frac{1}{(N\ell)^{\frac{d}{d+1}}} \]  \hspace{1cm} (5.20)

Using (5.20) in \( E_\rho\{E_{x'_{N\ell}}\{\delta|\rho,x\}|x\} \) results in

\[ E_\rho\{E_{x'_{N\ell}}\{\delta|\rho,x\}|x\} \approx \beta \frac{1}{(N\ell)^{\frac{d}{d+1}}} |A|^{-\frac{d}{d+1}} \times tr\{A P_A^{-\frac{2}{d+1}}(x) \}
\begin{align*}
&\times \left[\frac{1}{2} \nabla^2 q_{A_i}(x) + P_A^{-1}(x) \nabla P_A(x) \nabla^T q_{A_i}(x)\right]\}
\end{align*} \hspace{1cm} (5.21)

where \( \beta \) is (5.22).

\[ \beta = \frac{\Gamma(\frac{d+3}{2})\Gamma(\frac{2}{d+1} + 1)}{\pi(d+1)} \]  \hspace{1cm} (5.22)

Step 3 uses (5.21) to calculate the last line of (5.13). Rewriting results in (5.23)

\[ E_X\{[q_{A_2}(x) - q_{A_1}(x)] E_\rho\{E_{x'_{N\ell}}\{\delta|\rho,x\}|x\} \} \approx \beta \frac{1}{(N\ell)^{\frac{d}{d+1}}} E_X\{|A|^{-\frac{1}{d+1}} tr\{AB(x)\} \}
\]  \hspace{1cm} (5.23)

where \( B(x) \) is (5.24)

\[ B(x) = P_A^{-\frac{2}{d+1}}(x)[q_{A_2}(x) - q_{A_1}(x)] \times \left[\frac{1}{2} \nabla^2 q_{A_i}(x) + P_A^{-1}(x) \nabla P_A(x) \nabla^T q_{A_i}(x)\right]. \]  \hspace{1cm} (5.24)

Replacing (5.23) in (5.12) and rewriting (5.12) yields (5.7).
From Theorem 5.2.3, we see that the anatomized NN classifier has a faster convergence rate than the original NN classifier’s ($O(\frac{1}{N})$ vs $O(\frac{1}{N^2})$.) This is a surprising result despite the $\ell$-diversity condition. However, we still don’t know what kind of error rate ($R_A$) the anatomized NN classifier is converging to. Formally, we need to compare the bounds of error rate $R_A$ to the bounds of error rate $R$. The generalization error analysis will elaborate this comparison through non-parametric density estimation classifier.

Generalization Error Analysis

In pattern recognition literature, the generalization ability of any classifier is defined through the classifier’s Bayesian error estimation ability [65, 68, 72]. This is reasonable for k-NN classifiers as well since the error rate of original or anatomized k-NN classifier is bounded by the Bayesian errors (See (5.6) for anatomized k-NN.)

In this section, the Bayesian error will be estimated for binary classification using non-parametric density estimation classifier. Parzen density estimation will be used with mixed kernel function [65]. This approach is chosen because its derivation is easier and more readable than the k-NN density estimation’s one. The analysis, which follows Fukunaga [65] and Fukunaga et al. [73], is general enough for any non-parametric density estimation classifier including k-NN [65]. The multi-label classification is ignored since its theoretical work is limited for the original training data [72]. We first give three Axioms and a Lemma.

**Axiom 5.2.1** Given the anatomized training data $D_A$ and the training data $D$; let $P_{A_i}$ and $P_i$ be the class priors for class labels $i = \{1, 2\}$. Assume that $D$ satisfies the $\ell$-diversity condition and that every group has $\ell$ instances in the creation of anatomized training data $D_A$ from the original training data $D$. Then, $P_i = P_{A_i}$.

**Axiom 5.2.2** Given the anatomized training data $D_A$ and the training data $D$; let $P_A(X.A_{id})$ and $P(X.A_{id})$ be the smooth joint densities of identifying attributes $A_{id}$. Assume that $D$ satisfies $\ell$-diversity condition and that every group has $\ell$ instances in
the creation of anatomized training data $D_A$ from the original training data $D$. Then, $P(X.A_{id}) = P_A(X.A_{id})$.

**Axiom 5.2.3** Given the anatomized training data $D_A$ and the training data $D$; let $P_A(X.A_s)$ and $P(X.A_s)$ be the smooth densities of sensitive attribute $A_s$. Assume that $D$ satisfies $\ell$-diversity condition and that every group has $\ell$ instances in the creation of anatomized training data $D_A$ from the original training data $D$. Then, $P(X.A_s) = P_A(X.A_s)$.

Axioms 5.2.1 [5.2.2] and 5.2.3 are obvious due to the following: provided a sample of size $N$ drawn from a probability distribution $P$, repeating every instance for fixed $\ell > 0$ times and obtaining a sample of size $N\ell$ does not change the probability distribution $P$. The estimated parameters $\hat{\mu}$ and $\hat{\sigma}^2$ of distribution $P$ remain same as long as there is no suppression.

**Lemma 5.2.1** Given the anatomized training data $D_A$ and the training data $D$, let identifying attributes $A_{id}$ and the sensitive attribute $A_s$ be independent. Let’s assume that $D$ satisfies $\ell$-diversity condition and that every group has $\ell$ instances in the creation of anatomized training data $D_A$ from the original training data $D$. Then, $P_A(X) = P(X)$ is always true under the axioms 5.2.2 and 5.2.3.

Using Axioms 5.2.2 and 5.2.3 the proof of Lemma 5.2.1 is straightforward. Lemma 5.2.1 and Axioms 1-to-3 yield the Theorem 5.2.3. Using Lemma 5.2.1 we will assume that $R^*_A = R^*$ holds asymptotically for Bayesian errors.

**Theorem 5.2.3** Let $M \subset \mathbb{R}^{d+1}$ be a metric space. Let $D$ be an original training data of $N$ instances satisfying the $\ell$-diversity condition. Let $D_A$ be an anatomized training data of $N\ell$ instances created from the instances in $D$ with respect to the bucketization algorithm without any leftover instances. Let $x_{i_{id}} \in D$ denote the identifying attribute values of a training instance $x_i \in D$ such that, with probability 1, every $x_{i_{id}} \in D_A$ has fixed choice of $\ell$ sensitive attribute values among the domain of sensitive attribute $A_s$. Provided the former anatomized training data $D_A$, let $P_{A_1}(X)$ and $P_{A_2}(X)$ be the class
likelihood probability density functions of \( x \). Let \( P_{A_1} \) and \( P_{A_2} \) be the class priors. Let \( P_A(X) \) be the smooth density function of \( x \) such that \( P_A(x) = P_{A_1}P_{A_1}(x) + P_{A_2}P_{A_2}(x) \).

Provided original training data \( D \), let \( P_1(X) \) and \( P_2(X) \) be the class likelihood probability density functions of \( x \). Let \( P_1 \) and \( P_2 \) be the class priors. Let \( P(X) \) be the smooth density function of \( x \) such that \( P(x) = P_1P_1(x) + P_2P_2(x) \). Let \( h_A(x) = -\ln\frac{P_{A_1}(x)}{P_{A_2}(x)} \) and \( h(x) = -\ln\frac{P_1(x)}{P_2(x)} \) be the density classifiers with biases \( \Delta h_A(x) \) and \( \Delta h(x) \) respectively. Let \( t = \ln\frac{P_{A_1}}{P_{A_2}} = \ln\frac{P_1}{P_2} \) be the decision threshold with threshold bias \( \Delta t \). Let \( \epsilon_A > 0 \) be the small changes on \( P_1(x) \) and \( P_2(x) \) resulting in \( P_{A_1}(x) \) and \( P_{A_2}(x) \); and \( \Delta R_A^*, \Delta R^* \) be the Bayesian error estimations with respective biases \( \Delta R_A^* \), \( \Delta R^* \). Let \( \hat{P}_{A_1}(x) \) and \( \hat{P}_1(x) \) be the Parzen density estimations of likelihood densities; and \( K(\cdot) \) be the kernel function for original training data \( D \) with shape matrix \( A \) and size/volume parameter \( r \) \([65]\). Last, let’s assume the following:

1. \( A_{id} \) and \( A_s \) are independent in the original training data \( D \) and the anatomized training data \( D_A \) (independence in joint distributions, not for the distributions with respect to class labels \( \{1,2\} \)).

2. \( R_A^* = R^* \) hold.

3. \( \Delta t < 1 \).

4. The original training data \( D \) satisfies \( \ell \)-diversity condition and every group has \( \ell \) instances in the creation of anatomized training data \( D_A \) from the original training data \( D \).

The generalization error therefore has the approximation \([5.25]\) with probability 1

\[
\hat{R}_A^* \approx R^* + a_1r^2 + a_2r^4 + a_3r^{-(d+1)} N + \epsilon_Aa_4r^2 + \epsilon_Aa_5r^4 - \epsilon_Aa_6r^{-(d+1)} N \tag{5.25}
\]

where \( a_i \) is an integration term. \( a_1r^2, a_2r^4, \epsilon_Aa_4r^2 \) and \( \epsilon_Aa_5r^4 \) are the bias terms while \( a_3r^{-(d+1)} N \) and \( \epsilon_Aa_6r^{-(d+1)} N \) are the variance terms.
In the proof, we will use Taylor approximations up to the second order and the negligible terms will be ignored for convenience throughout the derivation. We will abuse $x$ to denote random variable $x$. Throughout the proof, we will derive in detail the approximations for the class 1 and omit the details for class 2 since both derivations are symmetric.

**Proof** Under the bucketization and the no leftover instances assumption, $P_{A_1}(x)$ and $P_{A_2}(x)$ are $P_1(x) + \epsilon_1$ and $P_2(x) - \epsilon_2$ respectively with probability 1 where $\epsilon_i$ is a small change for any class label $i = \{1, 2\}$ and $\epsilon_1 \neq \epsilon_2$. We first write $P_A(x)$.

$$P_A(x) = P_1P_{A_1}(x) + P_2P_{A_2}(x) = P_1[P_1(x) + \epsilon_1] + P_2[P_2(x) - \epsilon_2] = P_1P_1(x) + P_2P_2(x) + P_1\epsilon_1 - P_2\epsilon_2 = P(x) + P_1\epsilon_1 - P_2\epsilon_2$$

(5.26)

In the last line of (5.26), lemma 5.2.1 tells that $P_A(x) = P(x)$ always holds under the theorem’s independence assumptions. Thus, (5.27) is valid

$$\epsilon_2 = \frac{P_1}{P_2}\epsilon_1 = e^t\epsilon_1$$

(5.27)

when $t = \ln(P_1/P_2)$ [65]. Let $\epsilon_A$ stand for $\epsilon_1$ in the remainder of the text. Then, the class likelihoods are $P_{A_1}(x) = P_1(x) + \epsilon_A$ and $P_{A_2}(x) = P_2(x) - \epsilon^t\epsilon_A$. Next is the approximation of $E\{\tilde{P}_{A_1}(x)\}$ in function of $P_i(x)$ and $\epsilon_A$. (5.28) approximates $E\{\tilde{P}_{A_1}(x)\}$ using convolution and $\int K(x)dx = 1$.

$$E\{\tilde{P}_{A_1}(x)\} = \int P_1(Y)K_1(x - Y)dY = \int [P_1(Y) + \epsilon_A]K_1(x - Y)dY = \int P_1(Y)K_1(x - Y)dY + \epsilon_A\int K_1(x - Y)dY = P_1(x) * K_1(x) + \epsilon_A$$

$$\simeq P_1(x) + P_1(x)\frac{1}{2}\alpha_1(x)r^2 + \epsilon_A$$

(5.28)

In (5.28), $\alpha_i(x)$ is $tr\left\{\frac{\Sigma^2 P_i(x)}{P_i(x)}A\right\}$. Through a similar approach, we have

$$E\{\tilde{P}_{A_2}(x)\} \simeq P_2(x) + P_2(x)\frac{1}{2}\alpha_2(x)r^2 - \epsilon^t\epsilon_A$$

(5.29)
According to Fukunaga, variance is

$$\text{Var}\{\hat{P}_{A_1}(x)\} = \frac{1}{N} [P_{A_1}(x) \ast K_1^2(x) - E^2\{\hat{P}_{A_1}(x)\}].$$ \hspace{1cm} (5.30)

We will only use the first order terms to keep the calculation tractable. \(E^2\{\hat{P}_{A_1}(x)\}\) is (5.31)

$$[E\{\hat{P}_1(x) + \epsilon_A\}^2 \approx [P_1(x) + \epsilon_A]^2 \approx P_1(x)[P_1(x) + 2\epsilon_A]$$ \hspace{1cm} (5.31)

and \(P_{A_1}(x) \ast K_1^2(x)\) is (5.32) using \(w_1 = \int K^2(x)dx\).

$$\int P_{A_1}(Y)K_1^2(x - Y)dY = \int [P_1(Y) + \epsilon_A]K_1^2(x - Y)dY$$

$$= \int P_1(Y)K_1^2(x - Y)dY + \epsilon_A \int K_1^2(x - Y)dY$$

$$= P_1(x) \ast K_1^2(x) + \epsilon_1 w_1$$

$$\approx w_1 P_1(x) + \epsilon_A w_1$$

Replacing (5.31) and the last line of (5.32) in (5.30) results in (5.33).

$$\text{Var}\{\hat{P}_{A_1}(x)\} \approx \frac{1}{N} [[w_1 P_1(x) + \epsilon_A w_1] - [P_1(x)[P_1(x) + 2\epsilon_A]]]$$

$$= \frac{1}{N} [w_1 P_1(x) - P_1^2(x) + \epsilon_A[w_1 - 2P_1(x)]]$$ \hspace{1cm} (5.33)

$$= \frac{1}{N} [w_1 P_1(x) - P_1^2(x)] + \frac{\epsilon_A}{N}[w_1 - 2P_1(x)]$$

The approximation of \(P_{A_2}(x) \ast K_2^2(x)\) and \(E^2\{\hat{P}_{A_2}(x)\}\) yields (5.34).

$$\text{Var}\{\hat{P}_{A_2}(x)\} \approx \frac{1}{N} [w_2 P_2(x) - P_2^2(x)] - \frac{\epsilon_A w_2}{N}[w_2 - 2P_2(x)]$$ \hspace{1cm} (5.34)

According to Fukunaga, the bias \(\Delta R^*_A\) of the Bayesian error estimation is (5.35)

$$E[\Delta R^*_A] \approx \frac{1}{2\pi} \int \int E[\Delta h_A(x) + \frac{(j\omega)}{2}\Delta h_A^2(x)]e^{j\omega h_A(x)}$$

$$\times [P_{A_1}P_{A_1}(x) - P_{A_2}P_{A_2}(x)]d\omega dx.$$ \hspace{1cm} (5.35)

(5.35) requires the approximations of the expected decision function biases \(E\{\Delta h_A(x)\}\) and \(E\{\Delta h_A^2(x)\}\). \(E\{\frac{\Delta P_{A_1}(x)}{P_{A_1}(x)}\}\) and \(E\{(\frac{\Delta P_{A_1}(x)}{P_{A_1}(x)})^2\}\) will be approximated next to approx-
imate the former terms. $E\{\frac{\Delta P_{A_1}(x)}{P_{A_1}(x)}\}$ is approximately the last line of (5.36) using $\frac{\epsilon_A}{P_{1}(x)} < 1$ and the first order Taylor series.

$$E\{\Delta P_{A_1}(x)\} = \frac{1}{P_{A_1}(x)}E\{\hat{P}_{A_1}\} - 1$$

$$\approx \frac{1}{P_{1}(x) + \epsilon_A} [P_{1}(x) + P_{1}(x) \frac{1}{2} \alpha_1(x) r^2 + \epsilon_A] - 1$$

$$= \frac{1}{1 + \frac{\epsilon_A}{P_{1}(x)}} + \frac{1}{1 + \frac{\epsilon_A}{P_{1}(x)}} \frac{1}{2} \alpha_1(x) r^2 + \frac{1}{1 + \frac{P_{1}(x)}{\epsilon_A}} - 1$$

$$\approx \frac{1}{2} \alpha_1(x) r^2 - \frac{\epsilon_A}{P_{1}(x)} \frac{1}{2} \alpha_1(x) r^2 + 1 - \frac{1}{1 + \frac{\epsilon_A}{P_{1}(x)}}$$

$$\approx \frac{1}{2} \alpha_1(x) r^2 - \frac{\epsilon_A}{P_{1}(x)} \frac{1}{2} \alpha_1(x) r^2 + 1 - (1 - \frac{\epsilon_A}{P_{1}(x)})$$

$$= \frac{1}{2} \alpha_1(x) r^2 - \frac{\epsilon_A}{2} \frac{1}{P_{1}(x)} r^2$$

Similarly $E\{\frac{\Delta P_{A_2}(x)}{P_{A_2}(x)}\}$ is (5.37).

$$E\{\frac{\Delta P_{A_2}(x)}{P_{A_2}(x)}\} \approx \frac{1}{2} \alpha_2(x) r^2 + \epsilon_A \frac{1}{2} \alpha_2(x) r^2$$

$$E\{\left(\frac{\Delta P_{A_1}(x)}{P_{A_1}(x)}\right)^2\} \text{ is (5.38)}$$

$$E\{\left(\frac{\Delta P_{A_1}(x)}{P_{A_1}(x)}\right)^2\} = \frac{1}{P_{A_1}(x)} Var\{\hat{P}_{A_1}(x)\} + \frac{1}{P_{A_1}(x)} E^2\{\Delta P_{A_1}(x)\}$$

That requires the approximation of $\frac{1}{P_{A_1}(x)} Var\{\hat{P}_{A_1}(x)\}$ and $\frac{1}{P_{A_1}(x)} E^2\{\Delta P_{A_1}(x)\}$.

$\frac{1}{P_{A_1}(x)} Var\{\hat{P}_{A_1}(x)\}$ is approximated in (5.39) using $(\frac{P_{1}(x)}{P_{1}(x) + \epsilon_A})^2 \approx 1 - \frac{2\epsilon_A}{P_{1}(x)}$ and $w_1 = s_1 r^{d+1}$

$$\frac{1}{P_{A_1}^2(x)} Var\{\hat{P}_{A_1}(x)\} \approx \frac{1}{N(P_{1}(x) + \epsilon_A)^2 [w_1 P_{1}(x) - P_{1}^2(x)] + \epsilon_A N(P_{1}(x) + \epsilon_A)^2 [w_1 - 2P_{1}(x)]}$$

$$= \frac{P_{1}^2(x)}{(P_{1}(x) + \epsilon_A)^2 [NP_{1}(x) + \epsilon_A w_1 + \epsilon_A w_1 - \epsilon_A + \epsilon_A]}$$

$$= \frac{1}{NP_{1}(x) + \epsilon_A} + \frac{\epsilon_A}{NP_{1}(x) + \epsilon_A} - 1 - \frac{2\epsilon_A}{NP_{1}(x) + \epsilon_A}$$

$$\approx [1 - \frac{2\epsilon_A}{P_{1}(x)}] [NP_{1}(x) + \epsilon_A - NP_{1}(x) - \epsilon_A]$$

$$= \frac{s_1}{NP_{1}(x) r^{d+1}} - \frac{\epsilon_A}{NP_{1}(x) r^{d+1}}$$

$$\approx \frac{s_1}{NP_{1}(x) r^{d+1}} - \frac{\epsilon_A}{NP_{1}(x) r^{d+1}}$$
whereas \( \frac{1}{P_{A_1}^2(x)} E^2\{\Delta P_{A_1}(x)\} \) is approximated in (5.40).

\[
\frac{1}{P_{A_1}^2(x)} E^2\{\Delta P_{A_1}(x)\} \approx \left( \frac{1}{2} \alpha_1(x) r^2 - \epsilon_A \frac{1}{2} \alpha_1(x) r^2 \right)^2 \\
\approx \frac{1}{4} \alpha_1^2(x) r^4 - \alpha_1(x) r^2 \epsilon_A \frac{1}{2} \alpha_1(x) r^2 \\
= \frac{1}{4} \alpha_1^2(x) r^4 - \epsilon_A \frac{1}{2} \alpha_1(x) r^4
\]

(5.40)

Replacing the last lines of (5.39) and (5.40) in (5.38) results in (5.41).

\[
E\left\{ \left( \frac{\Delta P_{A_1}(x)}{P_{A_1}(x)} \right)^2 \right\} \approx \frac{s_1}{NP_1(x)r^{d+1}} - \epsilon_A \frac{s_1}{NP_1^2(x)r^{d+1}} \\
+ \frac{1}{4} \alpha_1^2(x) r^4 - \epsilon_A \frac{1}{2} \alpha_1^2(x) r^4
\]

(5.41)

The same derivation for \( E\left\{ \left( \frac{\Delta P_{A_2}(x)}{P_{A_2}(x)} \right)^2 \right\} \) results in (5.42).

\[
E\left\{ \left( \frac{\Delta P_{A_2}(x)}{P_{A_2}(x)} \right)^2 \right\} \approx \frac{s_2}{NP_2(x)r^{d+1}} + \epsilon_A \frac{s_2}{NP_2^2(x)r^{d+1}} \\
+ \frac{1}{4} \alpha_2^2(x) r^4 + \epsilon_A \frac{1}{2} \alpha_2^2(x) r^4
\]

(5.42)

According to Fukunaga, \( E\{\Delta h_A(x)\} \) is approximately (5.43)

\[
E\left\{ \frac{\Delta P_{A_2}(x)}{P_{A_2}(x)} \right\} - \frac{1}{2} E\{\Delta P_{A_2}(x)^2\} = E\left\{ \frac{\Delta P_{A_1}(x)}{P_{A_1}(x)} \right\} + \frac{1}{2} E\{\Delta P_{A_1}(x)^2\} - \Delta t
\]

(5.43)

and \( E\{\Delta h_A^2(x)\} \) is approximately (5.44) using the second order Taylor approximation.

\[
E\{\left( \frac{\Delta P_{A_2}(x)}{P_{A_2}(x)} \right)^2\} + E\{\left( \frac{\Delta P_{A_1}(x)}{P_{A_1}(x)} \right)^2\} - 2 E\{\left( \frac{\Delta P_{A_2}(x)}{P_{A_2}(x)} \right)\left( \frac{\Delta P_{A_1}(x)}{P_{A_1}(x)} \right)\} + \Delta t^2 \\
- 2 \Delta t E\{\frac{\Delta P_{A_2}(x)}{P_{A_2}(x)}\} - \frac{1}{2} E\{\left( \frac{\Delta P_{A_2}(x)}{P_{A_2}(x)} \right)^2\} \\
- E\{\frac{\Delta P_{A_2}(x)}{P_{A_2}(x)}\} + \frac{1}{2} E\{\left( \frac{\Delta P_{A_1}(x)}{P_{A_1}(x)} \right)^2\}
\]

(5.44)
Replacing the result of (5.36), (5.37), (5.41) and (5.42) in (5.43) and rewriting yield (5.45).

\[
E\{\Delta h_A(x)\} \cong \frac{r^2}{2} [\alpha_2(x) - \alpha_1(x)] + \frac{r^4}{8} (\alpha_1^2(x) - \alpha_2^2(x)) \\
+ \frac{r^{-(d+1)}}{2N} \left[ \frac{s_1}{P_1(x)} - \frac{s_2}{P_2(x)} \right] + \epsilon_A \frac{r^2}{2} \left[ \frac{\alpha_1(x)}{P_1(x)} + e^t \frac{\alpha_2(x)}{P_2(x)} \right] \\
- \epsilon_A \frac{r^4}{4} \left[ \frac{\alpha_1^2(x)}{P_1(x)} + e^t \frac{\alpha_2^2(x)}{P_2(x)} \right] - \epsilon_A \frac{r^{-(d+1)}}{2N} \left[ \frac{s_1}{P_1^2(x)} + e^t \frac{s_2}{P_2^2(x)} \right] \\
= E\{\Delta h(x)\} + \epsilon_A \frac{r^2}{2} \left[ \frac{\alpha_1(x)}{P_1(x)} + e^t \frac{\alpha_2(x)}{P_2(x)} \right] \\
- \epsilon_A \frac{r^4}{4} \left[ \frac{\alpha_1^2(x)}{P_1(x)} + e^t \frac{\alpha_2^2(x)}{P_2(x)} \right] - \epsilon_A \frac{r^{-(d+1)}}{2N} \left[ \frac{s_1}{P_1^2(x)} + e^t \frac{s_2}{P_2^2(x)} \right] \\
(5.45)
\]

Note that the first three terms of approximation are \( E\{\Delta h(x)\} \) according to Fukunaga \[65\] and the remaining terms with \( \epsilon_A \) are the effect of \( \ell \)-diversity. Replacing the result of (5.36), (5.37), (5.41) and (5.42) in (5.44) and rewriting yield (5.46).

\[
E\{\Delta h_A^2(x)\} \cong \left[ \frac{1}{2} r^2 (\alpha_2(x) - \alpha_1(x)) - \Delta t \right]^2 - \frac{\Delta t}{4} r^4 (\alpha_1^2(x) - \alpha_2^2(x)) \\
+ \frac{r^{-(d+1)}}{N} \left[ \frac{s_1}{P_1(x)} - \frac{s_2}{P_2(x)} \right] - \epsilon_A \Delta t \left[ \frac{\alpha_1(x)}{P_1(x)} + e^t \frac{\alpha_2(x)}{P_2(x)} \right] \\
+ \epsilon_A \frac{r^2}{2} \left[ \frac{\alpha_1(x)\alpha_2(x)}{P_1(x)} \right] - e^t \left[ \frac{\alpha_1(x)\alpha_2(x)}{P_2(x)} \right] \\
- \epsilon_A \frac{r^4}{2} \left[ \frac{\alpha_2^2(x)(1 - \Delta t)}{P_1(x)} \right] - e^t \left[ \frac{\alpha_2^2(x)(1 + \Delta t)}{P_2(x)} \right] \\
- \epsilon_A \frac{r^{-(d+1)}}{N} \left[ \frac{1 - \Delta t}{P_1^2(x)} + e^t \frac{1 + \Delta t}{P_2^2(x)} \right] \\
= E\{\Delta h^2(x)\} \\
- \epsilon_A \Delta t \left[ \frac{\alpha_1(x)}{P_1(x)} + e^t \frac{\alpha_2(x)}{P_2(x)} \right] + \epsilon_A \frac{r^4}{2} \left[ \frac{\alpha_1(x)\alpha_2(x)}{P_1(x)} - e^t \frac{\alpha_1(x)\alpha_2(x)}{P_2(x)} \right] \\
- \epsilon_A \frac{r^2}{2} \left[ \frac{\alpha_2^2(x)(1 - \Delta t)}{P_1(x)} - e^t \frac{\alpha_2^2(x)(1 + \Delta t)}{P_2(x)} \right] \\
- \epsilon_A \frac{r^{-(d+1)}}{N} \left[ \frac{1 - \Delta t}{P_1^2(x)} + e^t \frac{1 + \Delta t}{P_2^2(x)} \right] \\
(5.46)
\]

The first three terms of the approximation are \( E\{\Delta h^2(x)\} \) according to Fukunaga \[65\] and the remaining terms with \( \epsilon_A \) are again the effect of \( \ell \)-diversity. Plugging the results
of (5.45) and (5.46) in (5.35) and rewriting (5.35) give (5.25) where each $a_i$ stands for an integration term.

(5.25) relies on the following intuition. Given a fixed original training data that satisfies the $\ell$-diversity condition without leftover instances and the bucketization algorithm, Definition 3.0.11 always results in the same instances in the anatomized training data. Although the groups that a given instance is assigned could be different in independent runs of the bucketization algorithm, the anatomized training data always has the same set of instances without specific group information. This makes (5.25) general for a fixed original training data.

(5.25) shows that the anatomized training data $D_A$ reduces the variance term of the non-parametric density classifier that estimates the Bayesian error. This explains the faster convergence of the anatomized k-NN classifier that was derived in the previous section. Given the original training data of finite size $N$, using the anatomized training data of finite size $N\ell$ reduces the search space of possible models. This means that the anatomized k-NN classifier considers fewer options for probabilistic models than the original k-NN classifier.

However, the bias term is increased which makes the non-parametric density classifier more susceptible to underfitting. In overall, the non-parametric density classifier on the anatomized training data has a shifted bias-variance trade-off relative to the non-parametric models on the original training data. The following conclusions can be drawn for the generalization ability:

1. If the original k-NN classifier overfits, the anatomized k-NN classifier suffers less from overfitting provided that both classifiers have the same k hyper-parameter. In this case, the anatomized k-NN always generalizes better than the original k-NN.

2. If the original k-NN classifier fits well (optimum bias-variance tradeoff), the anatomized k-NN classifier always underfits provided that both classifiers have
the same k hyper-parameter. In this case, the original k-NN generalizes better than the anatomized k-NN.

3. If the original k-NN classifier underfits, the anatomized k-NN classifier suffers more from underfitting provided that both classifiers have the same k hyper-parameter. In this case, the original k-NN generalizes better than the anatomized k-NN as well.

The key result to keep is: \( \ell \)-diversity regularizes the original k-NN classifier while it provides privacy.

5.3 Experiments and Results

5.3.1 Prerequisites

Datasets

We tested our algorithm on the adult, IPUMS and marketing datasets of the UCI data repository \([62]\) and the fatality dataset of the Keel data repository \([74]\):

1. **Adult:** The adult dataset is drawn from 1994 census data of the United States \([62]\). It is composed of 45222 instances after the removal of instances with missing values. The binary classification task is to predict whether a person’s adjusted gross income is \( \leq 50K \) or \( > 50K \). The attribute *final weight* is ignored. *Education* is treated as the sensitive attribute in the experiments. The quasi-identifying attributes are *age, workclass, maritalstatus, occupation, race, sex, capitalgain, capitalloss, hoursperweek* and *nativecountry*. The class attribute is *income*.

2. **IPUMS:** This data is drawn from the 1970, 1980 and 1990 census data of the Los Angeles and Long Beach areas \([62]\). It has 233584 instances in total. We pick the 10 attributes that are included in the adult data. The binary classification task is to predict whether a person’s total income is \( \leq 50K \) or
> 50K. The classifiers are expected to show a different behavior from the former adult data since the population (and to some extent, classification task, as it is total income rather than adjusted gross income) are different. Educrec is treated as the sensitive attribute in the experiments. The quasi-identifying attributes are age, sex, raceg, marst, occ1950, classwkg, hrswork2, migplac5 and vetstat. The class attribute is bintotinc, a binary attribute that we created from the totinc (total income) of the original dataset based on the former binary classification task.

3. **Marketing Data:** This data is drawn from a phone based marketing campaign of a Portuguese banking institution for long term deposits [62]. We created the following binary classification task which is linearly separable: “among all the people who didn’t submit a long term deposit, predict whether a person has a housing loan or not”. We performed the following preprocessing using Weka filters [63]: 1) pick 39922 instances who didn’t make a long term deposit 2) choose four attributes job, day, month and age using the correlation with the class attribute housing. Discretized age is treated as the sensitive attribute whereas the quasi-identifying attributes are job, day and month.

4. **Fatality Data:** This data is a U.S. National Center for Statistics and Analysis compilation of 2001 car accidents. The original class attribute injury.severity has eight labels indicating the level of injury suffered [74]. We create the binary attribute is_injured with values “Injured” and “No_Injury” in the following way: 1) remove the instances with labels “Injured_Severity_Unknown”, “Died_Prior_to_Accident”, “Unknown” and “Possible_Injury” from the original data. This results in 91085 instances 2) label “Injured” the instances with labels “Nonincapaciting_Evident_Injury”, “Incapaciting_Injury” and “Fatal_Injury”. No feature selection is applied on this dataset. The sensitive attribute is police_reported_alcohol_involvement. The remaining attributes in the data catalog are the quasi-identifying attributes [74].
Privacy Setup

The anatomization was done according to Xiao et al.’s bucketization algorithm \[11\]. When $\ell$-diversity condition is not satisfied, the instances were divided into groups of size $\ell$ according to the original bucketization algorithm. Leftover instances were suppressed (not used in training models.) Although this is a violation of the assumption in the theoretical analysis, we believe such experiments will still be useful to show whether the theoretical analysis is representation of data that violates the assumptions.

$k$-anonymized training data was created for the adult dataset. The $k$-anonymized k-NN is not included for other datasets since Inan et al. provided generalization hierarchies only in the adult dataset \[25\]. Hence, we used Inan et al.’s value generalization hierarchies in the experiments. The privacy parameters were $k = \ell$ for $k$-anonymity and $\ell$-diversity to compare the classifiers using same group sizes in training data.

$k$-anonymized and anatomized training data had the same identifying and sensitive attributes. The sensitive attributes were chosen such that the $\ell$-diversity is satisfied for at least $\ell = 2$.

Model Evaluation Setup

Weka’s IBk class was used to train k-NN classifiers on the original, identifying and anatomized training data \[63\]. The anatomized training data is created from the $IT$ and $ST$ tables using the merging and dropping functions of Pandas \[75\].

10-fold cross validation was used for evaluation of the error rate bounds and generalization ability, and the error rate was used as the evaluation metric. The comparison includes anatomized k-NN, original k-NN and identifying k-NN. The comparison on adult dataset also includes $k$-anonymized k-NN due to the privacy setup in the previous section. The error rates of anatomized and original k-NN are compared using the Student $t$-test. Other models are not included in Student $t$-test, because Theorem \[5.2.3\] covers only anatomized and original k-NN.
10-fold cross validation was also used for evaluation of the NN classifier’s convergence rate. The error rate was again used for the evaluation metric. However, we trained the anatomized and original NN classifiers incrementally at each iteration of the cross validation. In a given iteration, the training set was divided into 9 partitions and the models are trained 9 times. The training started from the first partition and continued further by adding a partition at a time. The average error rates are computed over 10 different error rate values for a given training set size in the analysis (cf. Section 5.3.2)

5.3.2 Analysis of Results

Figures 5.3 through 5.8 show the boxplots of error rates for k-NN classifiers. Figure 5.9 shows the lines of convergence rate for the error rates of original and anatomized NN classifiers. In Figures 5.3 to 5.8, “Org.” and “Id.” labels will stand for original k-NN and identifying k-NN respectively. The anatomized and k-anonymized k-NN will be represented by their respective privacy parameters (L for \( \ell \) and k for \( k \).) Our analysis have four observation aspects:

1. **Comparison between the anatomized and the original k-NN:** From Theorem 5.2.3 there are two possibilities for the error rates. In the first possibility, the anatomized k-NN classifier is effected from overfitting less than the original k-NN classifier. In this case, we expect that the anatomized k-NN has smaller error rate than the original k-NN on average. In the second possibility, the anatomized k-NN suffers from underfitting while the original k-NN classifier either fits well or suffers less from underfitting. In this case, the anatomized k-NN’s error rate is expected to be greater than the original k-NN’s. Increasing the \( \ell \) parameter would result in the increase of distortion (\( \epsilon_A \) in (5.25)). From Theorem 5.2.3, the error rate expectation in the former two possibilities would still be valid in function of the increase in \( \ell \) unless there is suppression. If some instances are suppressed to create the anatomized training data, then the theo-
retical analysis of the anatomized k-NN classifier would be invalid because the assumption of Theorem 5.2.3 is violated.

2. **Comparison between anatomized and identifying k-NN:** In the first case, the identifying k-NN is likely to outperform the anatomized k-NN if the sensitive attribute is a bad predictor of the class attribute in the original training data. The sensitive attribute changes the likelihood probability in the density based decision function and the model either overfits by the increase in variance or underfits by the increase in the bias. The anatomized k-NN therefore estimates a model of the original training data that is not likely to generalize well. In the second case, the anatomized k-NN are likely to outperform the identifying k-NN if the sensitive attribute is a good predictor of the class attribute in the original training data. The sensitive attribute changes the likelihood probability in the density based decision function and the anatomized k-NN classifier catches a better tradeoff between the bias and the variance terms. The anatomized k-NN classifier would avoid the potential underfitting that the identifying k-NN could have.

3. **Comparison between the anatomized and the k-anonymized k-NN:** The anatomized k-NN are expected to outperform the k-anonymized k-NN because anatomization preserves the original values for all the attributes. The generalization based k-anonymity, on the other hand, distorts most of the original attribute values $25$.

4. **Comparison of the anatomized and original NN in the convergence rates:** From Theorem 5.2.2 we expect that the anatomized NN will converge faster than the original NN classifier to the lowest possible error rate if there is no suppression in the creation of the anatomized training data. Suppression would again violate the assumption of Theorem 5.2.2.

We should note that we are comparing an anatomy method satisfying $\ell$-diversity against a weaker $k$-anonymity requirement for the generalization-based approach.
While we don’t specifically use an $\ell$-diversity based generalization [1], such a method would be expected to generalize more and give worse results. The generalization-based $k$-anonymization we use already produces 2-diverse datasets. More than 44000

![Figures 5.3: 10 Cross Validation Errors Rates for NN Classifiers](image-url)
instances in the adult dataset are 11-diverse for $k$-values 2 to 5. As 97% of the generalization based $k$-anonymized adult dataset satisfy $\ell$-diversity, the results should be similar with generalization based $\ell$-diversity.

We are now presenting the analysis of error rates in first three aspects for all the datasets. We will then present the analysis of convergence rate for anatomized NN in the fourth aspect.

Analysis of Error Rates for NN

Figure 5.3 shows the error rates for the original, anatomized and identifying NN classifiers that are tested on the four datasets.

In Figure 5.3(a), the original NN classifier outperforms the anatomized NN classifier when $\ell$ is 2 and 3. The anatomized NN classifier hence underfits more than the original NN due to the distortion of $\ell$-diversity. In the second aspect, the sensitive attribute is a bad predictor of the class attribute because the average error rate of the identifying NN classifier is less than the original NN’s. The anatomized NN classifier’s error rates are greater than both the original and identifying NN classifier’s because the anatomized NN classifers are estimating an original NN classifier that doesn’t fit well. The former claims hold when $\ell$ is 4 and 5 despite suppression.

In Figure 5.3(b) the original NN outperforms the anatomized NN when $\ell$ is 2-to-4. Theorem 5.2.3 concludes that the anatomized NN classifier underfits more than the original NN classifier. In the second aspect, the sensitive attribute is a bad predictor of the class attribute because the average error rate of the identifying and original NN classifiers are almost same. The anatomized NN classifier’s error rates thus are greater than both the original and identifying NN classifier’s because the anatomized NN classifers are estimating an original NN classifier that doesn’t fit well. When $\ell$ is 5, the former claims hold although the assumptions of our theoretical analysis is violated due to suppression.
In Figure 5.3(c) the original NN classifier has higher error rates than the anatomized NN classifiers when $\ell$ is 2-to-5. From Theorem 5.2.3 the original NN classifier suffers from overfitting whereas the anatomized NN classifier catches a better bias variance tradeoff. Note that when $\ell$ is increased from 4 to 5, the error rates start increasing. Increase in $\ell$ causes here the increase in the bias such that the anatomized NN classifier starts underfitting. In the second aspect, the identifying NN classifier has a much lower error rate than the original NN classifier has. This means that the sensitive attribute is a bad predictor of the class attribute. Since the sensitive attribute is a bad predictor, the anatomized NN classifier approximates a bad classifier of the original training data which increases its generalization error relative to the identifying NN classifier.

In Figure 5.3(d) the original NN classifier has higher error rates than the anatomized NN classifiers when $\ell$ is 2. From Theorem 5.2.3 the original NN classifier suffers from overfitting whereas the anatomized NN classifier captures a better bias variance tradeoff. When $\ell$ is 3 or 4, the assumptions of the theoretical analysis is violated due to suppression. The general conclusion here would be the significant distortion of the likelihood probabilities which increases the bias term (underfitting). In the second aspect, the identifying NN classifier’s error rates are less than the original NN classifier’s. The sensitive attribute hence is a bad predictor of the class attribute and the anatomized NN classifier approximates a bad classifier which increases its generalization error relative to identifying NN.

Figure 5.4 gives the boxplots of error rates for $k$-anonymized k-NN classifiers in addition to original, identifying and anatomized NN. In the third aspect, we have the expected result for the $k$-anonymized NN classifiers. Their error rates are greater than the anatomized NN classifiers. Due to generalization of the identifying attribute values, the utility loss of the $k$-anonymized training data is more than the anatomized training data’s. In the context of $k$-anonymized NN, note that the error rates are decreased when $k$ is increased. Increasing $k$ for the $k$-anonymized NN classifier is same as increasing the number of neighbors for the k-NN classifier.
Analysis of Error Rates for k-NN on Adult Data

Figure 5.4 shows the results of multiple types of k-NN classifiers on the adult data for 3, 5, 7 and 9 neighbors (k). In the first aspect, we start discussion with $\ell = 2$ and $\ell = 3$ (no suppression cases.) The anatomized k-NN classifiers have higher error rates than the original k-NN classifiers. Since Theorem 5.2.3 tells that the anatomized training data increases the bias terms of the k-NN classifiers’ generalization error while reducing the variance, the anatomized k-NN classifiers suffer from underfitting. Although the former Theorem’s assumptions are violated in cases of $\ell = 4$ and $\ell = 5$, we see that the error rates of anatomized k-NN classifiers are still higher than the
original k-NN classifiers’. We thus can assume that the same underfitting behaviour continues. In the second aspect, the sensitive attribute is a good predictor of the class attribute because the original 3-NN, 5-NN, 7-NN and 9-NN classifiers outperform the identifying 3-NN, 5-NN, 7-NN and 9-NN classifiers in terms of the error rates. When $\ell = 2$ and $\ell = 3$, the anatomized k-NN classifier captures the bias variance tradeoff.
between the identifying and original k-NN classifiers for multiple values of k. The anatomized k-NN classifiers outperforms the identifying k-NN classifiers while it is outperformed by the original k-NN classifiers. Due to suppression, the former bias variance tradeoff conditions don’t hold in cases of $\ell = 4$ and $\ell = 5$. The anatomized k-NN classifiers thus don’t have the former tradeoff.

Figure 5.6.: 10 Cross Validation Errors Rates of Ipums Data
Analysis of Error Rates for k-NN on IPUMS Data

Figure 5.6 shows the results for multiple types of k-NN classifiers on the IPUMS data. The IPUMS data satisfy the $\ell$-diversity condition when $\ell = 2, \ell = 3$ and $\ell = 4$, so the Theorem 5.2.3 and the bound 5.5 are expected to hold in here. In the first aspect, the anatomized 3-NN, 5-NN, 7-NN and 9-NN classifiers are outperformed by the original 3-NN, 5-NN, 7-NN and 9-NN classifiers. From Theorem 5.2.3, the increase in the bias terms yields the underfitting classifiers. In the second aspect, the sensitive attribute is a good predictor of the class attribute because the original k-NN classifiers outperform the identifying k-NN classifier for multiple values of hyperparameter $k$. When $\ell = 3$ and $\ell = 4$, the anatomized k-NN classifiers surprisingly fail to capture the bias variance tradeoff between the original and the identifying k-NN classifiers. From Theorem 5.2.3, the increase in bias is way greater to capture bias variance tradeoff for the anatomized k-NN classifiers (cf. Figure 5.6). The case of $\ell = 2$ is special. Note that the anatomized 5-NN, 7-NN and 9-NN classifiers capture the bias variance trade-off between the original and identifying ones, as expected (cf. Figures 5.6(b), 5.6(c) and 5.6(d)). For anatomized 3-NN classifier, the increase in bias is way greater than the decrease in variance. It thus fails to capture again the bias variance tradeoff between the original 3-NN and the identifying 3-NN classifiers (cf. Figure 5.6(a)). Last, the anatomized 3-NN, 5-NN, 7-NN and 9-NN classifiers under $\ell = 5$ (suppression) show a similar trend to the anatomized 3-NN, 5-NN, 7-NN and 9-NN classifiers under $\ell = 3$ and $\ell = 4$ in terms of the first and second aspects.

Analysis of Error Rates for k-NN on Marketing Data

Figure 5.7 shows the result for multiple types of k-NN classifiers on the marketing data. The marketing data satisfy the $\ell$-diversity condition when $\ell$ is 2-to-4. Theorem 5.2.3 and the bound 5.5 thus are expected to hold in most $\ell$ values. In the first aspect, the anatomized k-NN classifiers outperform the original k-NN classifiers for all combinations of $\ell$ and hyperparameter $k$. This shows that the distortion of $\ell$-
diversity reduces the generalization error by increasing the bias and reducing the variance of the original k-NN classifier (fixing the overfitting issue.) Note that when $\ell$ is increased from 4 to 5, the overfitting issue is fixed less because the increase in bias exceeded the good bias variance tradeoff and the model is directed to the underfitting case. Besides, the training data has suppression in this case. In the second aspect, the
sensitive attribute is a good predictor of 7-NN and 9-NN classifiers since the original 7-NN and 9-NN’s error rates are less than the identifying 7-NN and 9-NN’s (cf. Figures 5.7(c) and 5.7(d)). Surprisingly, the anatomized 7-NN and 9-NN’s error rate are lower than both the original and the identifying 7-NN and 9-NN classifiers. The sensitive attribute, on the other hand, is a bad predictor of 3-NN and 5-NN classifiers since the original 3-NN and 5-NN’s error rates are greater than the identifying 3-NN and 5-NN’s (cf. Figures 5.7(a) and 5.7(b)). Surprisingly, the anatomized 3-NN and 5-NN classifier have again lower error rates than both original and identifying 3-NN and 5-NN classifiers. The most plausible reason is the regularization effect of $\ell$-diversity that results in the lowest generalization error of the anatomized k-NN classifiers for all $\ell$-values. The distortion of $\ell$-diversity increases the bias such that it fixes the overfitting issue of both the original and identifying k-NN classifiers.

Analysis of Error Rates for k-NN on Fatality Data

Figure 5.8 shows the result for multiple types of k-NN classifiers on the fatality data. As the fatality data satisfy the $\ell$-diversity condition for $\ell = 2$, Theorem 5.2.3 thus is expected to hold in the first aspect. Anatomized k-NN classifiers outperform the original k-NN classifiers when $\ell = 2$. From Theorem 5.2.3, the increase in bias reduces the generalization error of the original k-NN classifiers which is overfitting to the original training data. In the second aspect, the sensitive attribute is a bad predictor of k-NN classifiers since the original k-NN’s error rates are greater than the identifying k-NN’s. Although the error rates of anatomized k-NN is less than the original k-NN’s, its error rates are greater than the identifying k-NN’s. This is expected since the anatomized k-NN is trying to capture the bias variance tradeoff between the original and the identifying k-NN classifiers. The anatomized k-NN classifier is estimating the original k-NN classifier’s distribution which is not the best classifier in the existing data. Last, increasing $\ell$ to 3 and 4 increases the error rates of the anatomized k-NN classifiers. Due to suppression, the theoretical analysis does
not hold here. The most plausible reason is that the reduction of the training data size results in the overfitting of the models. This would increase the generalization error.

Figure 5.8.: 10 Cross Validation Errors Rates of Fatality Data

(a) Error Rates of 3-NN
(b) Error Rates of 5-NN
(c) Error Rates of 7-NN
(d) Error Rates of 9-NN
Figure 5.9: Convergence of NN Using Multiple Size Training Sets

Analysis of Convergence Rates

Figure 5.9 shows the error rates of original and anatomized NN classifiers as a function of the increasing training set sizes.

In Figure 5.9(a) we see that the anatomized NN classifier converges faster than the original NN classifier on the adult data when \( \ell = 2 \) and \( \ell = 3 \), as expected from Theorem 5.2.2. The former \( \ell \) values satisfy the \( \ell \)-diversity condition, so there is no
supression and the assumptions of the Theorem 5.2.2 are not violated. Note that the anatomized NN classifier converges slower than the original NN classifier when the assumption of Theorem 5.2.2 is violated under \( \ell = 4 \) and \( \ell = 5 \) (due to suppression.)

In Figure 5.9(b) we see that the anatomized NN classifier converges faster than the original NN classifier on the IPUMS data when \( \ell = 2, \ell = 3 \) and \( \ell = 4 \), as expected from Theorem 5.2.2. The former \( \ell \) values satisfy the \( \ell \)-diversity condition, so there is no supression and the assumptions of the Theorem 5.2.2 are not violated. Note that the anatomized NN classifier still converges faster than the original NN classifier when the assumption of Theorem 5.2.2 is violated under \( \ell = 5 \). The most likely reason is that the number of instances in the IPUMS data is too large and the number of suppressed instances are negligible relative to its size.

In Figure 5.9(c) we see that the anatomized NN classifier converges faster than the original NN classifier on the marketing data for all values of \( \ell \), as expected from Theorem 5.2.2. The marketing data satisfy the \( \ell \)-diversity condition for \( \ell \) values 2-to-5. Hence, the assumptions of the Theorem 5.2.2 are never violated in the experiments.

Last, Figure 5.9(d) shows that the anatomized NN classifier converges faster than the original NN classifier on the fatality data under \( \ell = 2 \). This is again expected from Theorem 5.2.2 since the fatality data satisfy the \( \ell \)-diversity condition for \( \ell = 2 \). It is easy to notice that the convergence of anatomized NN classifier is slower than the original NN classifier under \( \ell = 3 \) and \( \ell = 4 \) due to suppression, as expected.

5.3.3 Student t-test for Anatomized k-NN versus Original k-NN

Tables 5.1, 5.2, 5.3 and 5.4 give the statistical test results for confidence interval 0.95. In all Tables, “P” stands for pass while “F” stands for fail. “N/A” stands for not applicable in cases where the domain size of sensitive attribute is less than the \( \ell \) value. “Org.” stand for the original k-NN whereas “\( \ell \)’’ stand for the anatomized k-NN. Note that we do the test for original k-NN vs anatomized k-NN, because the Theorem 5.2.3’s scope covers this analysis.
Table 5.1.: Anatomized k-NN vs Original k-NN on Adult

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Org. vs $\ell = 2$</th>
<th>Org. vs $\ell = 3$</th>
<th>Org. vs $\ell = 4$</th>
<th>Org. vs $\ell = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>3-NN</td>
<td>F</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>5-NN</td>
<td>P</td>
<td>F</td>
<td>F</td>
<td>P</td>
</tr>
<tr>
<td>7-NN</td>
<td>F</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>9-NN</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
</tbody>
</table>

Table 5.2.: Anatomized k-NN vs Original k-NN on IPUMS

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Org. vs $\ell = 2$</th>
<th>Org. vs $\ell = 3$</th>
<th>Org. vs $\ell = 4$</th>
<th>Org. vs $\ell = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>3-NN</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>5-NN</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>7-NN</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>9-NN</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
</tbody>
</table>

Table 5.3.: Anatomized k-NN vs Original k-NN on Marketing

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Org. vs $\ell = 2$</th>
<th>Org. vs $\ell = 3$</th>
<th>Org. vs $\ell = 4$</th>
<th>Org. vs $\ell = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>3-NN</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>5-NN</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>7-NN</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>9-NN</td>
<td>P</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

From the Tables, the combinations of $\ell$ and hyperparameter k give at least 1 statistically significant comparison when there is no suppression in the creation of the
Table 5.4.: Anatomized k-NN vs Original k-NN on Fatality

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Org. vs $\ell = 2$</th>
<th>Org. vs $\ell = 3$</th>
<th>Org. vs $\ell = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>F</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>3-NN</td>
<td>F</td>
<td>F</td>
<td>P</td>
</tr>
<tr>
<td>5-NN</td>
<td>F</td>
<td>F</td>
<td>P</td>
</tr>
<tr>
<td>7-NN</td>
<td>P</td>
<td>F</td>
<td>P</td>
</tr>
<tr>
<td>9-NN</td>
<td>P</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

anatomized training data. There is at least one statistically significant comparison when the theoretical analysis is supposed to hold under the $\ell$-diversity condition.

In the adult dataset, there is no significant advantage to having original data for NN classifier as $t$-test comparison fails for all $\ell$ values. We can draw the same conclusion in the marketing dataset for 7-NN and 9-NN classifiers, especially when $\ell \geq 3$. In the fatality dataset, 3-NN and 5-NN classifiers do not have significant advantage for using original data when $\ell = 2$ and $\ell = 3$. Last, the results in the ipums dataset show that having original data has significant advantage in all k-NN classifiers and $\ell$ values. However, the results here show that using anatomized data could still give classifiers that could approach to the generalization error of the original k-NN classifiers.

Among the anatomized and the original data, choosing the one which has significant advantage depends on the data domain, the classification task, the model hyperparameter and the privacy protection level ($\ell$). We can conclude that sharing the anatomized data within an organization is a very viable option.
6 SUPPORT VECTOR CLASSIFICATION FOR \( \ell \)-DIVERSITY

In this chapter, we will consider again a theoretical approach for data classification in a non-distributed learning scheme. The data is still assumed to be released to a third party and anybody can train models from the released data. Our theoretical approach will differ from the previous chapter though the learning techniques used to solve the Problem 2.1.1. We will focus on a parametric model here, in particular on support vector classification. A preprocessing algorithm will be proposed for the anatomized training data and the learning theory will be elaborated to derive the generalization ability of the support vector classifier after preprocessing.

The chapter organization will be the following. Section 6.1 will give the notations throughout the chapter and will recap the outcome of the statistical learning theory for support vector classification. Section 6.2 will explain in detail the preprocessing algorithm we propose and its theoretical implications on the generalization of support vector classification. Section 6.3 will conclude the chapter with experiment results on publicly available datasets. The discussion here will follow mostly Mancuhan et al. [76].

6.1 Notations

We now summarize the notations used in this chapter. \( x_i \) will denote a training instance in the original training data \( D \) and pruned training data \( D_P \) interchangeably (We will explain pruning later, this is created from the anatomized data). \( N \) will be the total number of instances in \( D \) and \( D_P \). \( X \) will be a random variable vector in \( D \) and \( D_P \) interchangeably. \( D \subseteq \mathbb{R}^{d+1} \) and \( D_P \subseteq \mathbb{R}^{d+1} \) will hold in Euclidean space (see Section 6.2.2 for practical issues). \( y \) will be the binary class label with values \{0, 1\}. 
$f(X) = wX + b$ will be a linear classifier such that $w \in \mathbb{R}^{d+1}$ and $b \in \mathbb{R}$. $\mathcal{F}$ is the functional space

$$\{ f : \mathbb{R}^{d+1} \to \{0, 1\} : f(X) = wX + b, b \in \mathbb{R}, w \in \mathbb{R}^{d+1}\}. \quad (6.1)$$

We will use $f$ instead of $f(X)$ for shorthand in subsequent parts of this chapter. The risk of a linear classifier $f$, $R(f)$, is (6.2).

$$R(f) = \int |(y - f(X))|p(X, y)dXdy \quad (6.2)$$

In (6.2), $p(X, y)$ is the joint probability density of training instances $X$ with class label $y$. The empirical risk of classifier $f$, $\hat{R}_N(f)$, is (6.3).

$$\hat{R}_N(f) = \frac{1}{N} \sum_{i=1}^{N} I(y \neq f(x_i)) \quad (6.3)$$

In (6.3), $N$ is the number of training instances and $I(*)$ is the indicator function. The linear classifier $f$ is an empirical risk minimizer such that $\hat{f}_N = \arg\min_{f \in \mathcal{F}} \hat{R}_N(f)$.

Given the empirical risk minimizer $\hat{f}_N$ is the SVC with the largest margin, bound (6.4) holds

$$E[R(\hat{f}_N)] \leq E[\frac{(R(||w||)^2}{N}] \quad (6.4)$$

when the training data is linearly separable (77). In (6.4), $R$ stands for the radius of the sphere that the shatterable instances lie on and $w$ stands for the weight vector of hyperplane $f(X)$ in (6.1) (77) (78). Given some $\delta$ such that $0 \leq \delta \leq 1$ and the same SVC, the bound (6.5) of generalization ability holds with probability $1 - \delta$ (77) (78).

$$E[R(\hat{f}_N)] - \inf_{f \in \mathcal{F}} R(f) \leq 4 \sqrt{\frac{(d + 2)log(N + 1) + log(\frac{2}{\delta})}{N}} \quad (6.5)$$

In (6.5), $\inf_{f \in \mathcal{F}} R(f)$ is the minimum possible risk for the SVC $f$. Next, we define our pruning mechanism.
6.2 Pruning Mechanism for Anatomization

6.2.1 Algorithm

We will explain our algorithm ($\sigma_G$ in Definition 3.0.14) through the example in Figure 6.1. The curious reader should visit Figures 6.5 and 6.6 in Section 6.2.2 to see the pseudo code and the complexity. Although the example is for any linear classifier (hyperplane), the pruning mechanism is valid for SVC and SVM. We later define the generalization ability of pruned SVC/SVM (cf. Definition 3.0.15).

Figure 6.1(a) shows the original training data with six instances: two instances of a blue class (on the left side) and four of a red class (on the right side), with two attributes $A_1$ and $A_2$. Here, every instance has a different shape and filling combination since they are unique. Figure 6.1(b) shows the anatomized training data with 12 instances created from pairs $IT(A_1, GID)$ and $ST(GID, A_2)$ when $\ell = 2$ (cf. Definitions 3.0.10 and 3.0.11).

A typical training procedure would be the subtraction of mean from attributes $A_1$ and $A_2$ in the original training data, and solving an objective function of a perceptron

Figure 6.1.: Training Data Example
or SVC (cf. Figure 6.2). In Figure 6.2, the original training data is linearly separable and the instances which are closest to the separating hyperplane lie on the surface of the circle\(^1\). This circle is the key point of linear classification, because the original training data is guaranteed to be linearly separable if the instances that are closest to the decision boundary lie on the surface of a circle\(^7\). This observation lets us define two steps of the pruning mechanism algorithm:

1. **Prerequisite Step**: Estimate the circle of shatterable instances from the anatomized training data (Algorithm in Figure 6.5) (See 78 for a discussion of shatterable instances).

2. **Pruning Step**: For every group in the anatomized training data, pick an instance that is closest to the surface of the estimated circle of shatterable instances (Algorithm in Figure 6.6).

Figure 6.3 show the range of radii for all possible circles of shatterable instances in the prerequisite step. The radius of the original training data must be between the

\(^{1}\)The discussion can be generalized to sphere for 3 or larger dimensions. See Burges [77] and Vapnik [78] for general discussion.
norms of the pair of instances that are closest to \((r_{A_{\text{min}}} \text{ in Figure 6.3})\) and farthest from \((r_{A_{\text{max}}} \text{ in Figure 6.3})\), the origin. Under the random worlds assumption \([\text{11}]\), the prerequisite step assumes that \((r_{A_{\text{min}}}, r_{A_{\text{max}}})\) has uniform distribution and therefore estimates the expected radius \(E[r]\) with \(\frac{r_{A_{\text{min}}} + r_{A_{\text{max}}}}{2}\) (dashed green line in Figure 6.3).

Using the estimated radius from the prerequisite step, the pruning step creates the pruned training data in Figure 6.4. Figure 6.4 also has the hyperplane that is trained from the pruned training data. Although the shatterable instances of the pruned training data (cf. Figure 6.4) are the same as the shatterable instances of the original training data (cf. Figure 6.2), other instances are different. The purpose of the pruning step is to find a linearly separable case instead of distribution reconstruction.

There are two remaining issues to address. First is the application of the pruning algorithm even if the anatomized training data is not linearly separable (cf. Figure 6.1(b)). The instances within each group are not linearly independent from the other \(\ell - 1\) instances and the shattering property is damaged \([\text{77}]\). The second issue is non-separable original and anatomized training data. If the training data is not linearly separable in the original \((d + 1)\) dimensional space, the right approach would
be projecting it into higher dimensional space, apply the pruning algorithm in the projected space, or use a soft margin classifier.

6.2.2 Pseudo code and Complexity

Figures 6.5 and 6.6 give the pseudocodes of the two steps described in Section 6.2.1. In the pseudocodes, the parameter $\overline{D_A}$ signifies the augmented anatomized training data. The augmentation here includes three points:

1. After the inner join operation between the $IT$ and $ST$ tables (cf. Definition 3.0.11), the instances are sorted with respect to the attribute $GID$; and then the attributes $IT.GID$ and $ST.GID$ are dropped.

2. The mean of every numeric and non-numeric ordinal attribute is subtracted in the augmented anatomized training data. If the attribute $A_i$ is non-numeric ordinal, we replaced the non-numeric values with integer values 1 to $|\text{dom}(A_i)|$ according to domain-wise order and set the mode of the discrete values to be the mean.
computePrerequisites \((\mathcal{D}_A)\):

\[
\|R_{\text{min}}\|^2 := +\infty \quad /\text{Squared norm of minimum potential radius}
\]

\[
\|R_{\text{max}}\|^2 := -\infty \quad /\text{Squared norm of maximum potential radius}
\]

\(\text{list}_{\text{sqNorm}} := \emptyset \quad /\text{List of squared norms}\)

\[
\text{for } i = 1 \text{ to } |\mathcal{D}_A|:
\]

\[
\text{list}_{\text{sqNorm}} := \text{list}_{\text{sqNorm}} \cup \|\mathcal{D}_A[i]\|^2
\]

\[
\text{if}(\|\mathcal{D}_A[i]\|^2 \leq \|R_{\text{min}}\|^2) \quad \text{then}
\]

\[
\|R_{\text{min}}\|^2 := \|\mathcal{D}_A[i]\|^2
\]

\[
\text{if}(\|\mathcal{D}_A[i]\|^2 \geq \|R_{\text{max}}\|^2) \quad \text{then}
\]

\[
\|R_{\text{max}}\|^2 := \|\mathcal{D}_A[i]\|^2
\]

//Estimate the expected squared radius from \(U[\|R_{\text{min}}\|^2, \|R_{\text{max}}\|^2]\)

\[
E[\|R\|^2] := \frac{\|R_{\text{min}}\|^2 + \|R_{\text{max}}\|^2}{2}
\]

\[
\text{return } (E[\|R\|^2], \text{list}_{\text{sqNorm}})
\]

---

**Figure 6.5.:** Prerequisite Step Pseudocode

\text{pruneTrainingData} \((\mathcal{D}_A)\):

\[
(\text{E}[\|R\|^2], \text{list}_{\text{sqNorm}}) := \text{computePrerequisites} \((\mathcal{D}_A)\)
\]

\[
\mathcal{D}_p := \emptyset \quad /\text{List holding the pruning result}
\]

\[
i := 0 \quad /\text{index for instances}
\]

\[
j := 0 \quad /\text{index for visited groups}
\]

\[
\text{while } (i < |\mathcal{D}_A|):
\]

\[
d_c := +\infty \quad /\text{distance between the chosen instance in a group and } E[\|R\|^2]
\]

\[
g_c := +\infty \quad /\text{index of the instance in a group with distance } d_c
\]

\[
\text{total}_{\text{groups}} := |G_j| \quad /\text{Total number of instances for all visited groups}
\]

//Look for an instance closest to \(E[\|R\|^2]\) in the current group

\[
\text{while } (i < \text{total}_{\text{groups}}):
\]

\[
d := |\text{list}_{\text{sqNorm}}[i] - E[\|R\|^2]|
\]

\[
\text{if}(d < d_c) \quad \text{then} \quad /\text{a closer instance to } E[\|R\|^2]
\]

\[
d_c := d
\]

\[
g_c := i
\]

\[
i := i + 1
\]

\[
\mathcal{D}_p := \mathcal{D}_p \cup \mathcal{D}_A[g_c]
\]

\[
\text{if}(j < |G|) \quad \text{then} \quad /\text{Update number of instances for the next visited group}
\]

\[
j := j + 1
\]

\[
\text{total}_{\text{groups}} := \text{total}_{\text{groups}} + |G_j|
\]

\[
\text{return } \mathcal{D}_p
\]

---

**Figure 6.6.:** Pruning Step Pseudocode
3. If the attribute $A_i$ is non-numeric nominal, we replaced the mode of $A_i$ with integer 0 and the rest of the of $A_i$ values with integer 1.

Note that the pseudocodes use the squared norm instead of the norm itself, because Theorem 6.2.1 defines the generalization error upper bound with the squared radius of the sphere containing the shatterable instances of the original training data $D$.

The complexity of the algorithm in Figure 6.6 is $O(N(d + 2))$. Note that although there are 2 while loops in the algorithm, every instance is visited once and the algorithm doesn’t go through $d + 1$ attributes due to list sqNorm which makes the execution time of pruning $O(N)$. The prerequisites algorithm need to visit every instance and dimension which makes the execution time $O(N(d + 1))$. So the total execution time is $O(N(d + 2))$. All the groups $(G)$ and the total number of instances within each group $(G_j)$ of the anatomized training data are assumed to be known. In case it is not known, the grouping information can be computed using an inner join operation and group by query on IT and ST tables. Such a nested operation is easily implemented in $O(N \log N)$ execution time.

6.2.3 Privacy Preservation

The preprocessing and pruning steps preserve the $\ell$-diversity condition of anatomization. The algorithm doesn’t estimate the correct matchings between the identifying and the sensitive tables. Instead, it makes a guess within each group which is expected to give some linearly separable training data. It is possible that the original training data isn’t linearly separable or even is a random set of instances without any pattern (see Section 6.3).

6.2.4 Generalization Error of Pruned SVC

We will now give the upper bound on the generalization error of the pruned SVC (cf. Definition 3.0.15).
Theorem 6.2.1 Let $N$ be the number of instances, $d$ be the number of identifying attributes and $d + 1$ be the total number of attributes in the original and the pruned training data. Let $R$ be the radius of sphere containing the shatterable instances of the original training data $D$ and $w$ be the weights of the linear hyperplane resulting from linear SV classifier trained on the original training data $D$. Let $R_p$ and $w_p$ be the symmetric notations for a linear SV classifier trained on the pruned training data $D_p$. Assume that all the training instances are located in an Euclidean space $\mathbb{R}^{d+1}$. Let $\|\cdot\|$ be the Euclidean norm of vector $\cdot$. Let $r^2$ be $(R\|w\|)^2$, $r_p^2$ be $(R_p\|w_p\|)^2$, $[r_p^2]_{\text{min}}$ be $\min\{r_p^2\} > 0$ and $[r_p^2]_{\text{max}}$ be $\max\{r_p^2\} < \infty$. Let $\hat{R}_N(f)$ be the empirical risk of on the original training data $D$ and $\hat{R}_{N_p}(f)$ be the empirical risk on the pruned training data. Let $\mathcal{F}$ be the functional space defining the set of possible linear SV classifiers on the original training data $D$ and $\mathcal{F}_p$ be the functional space of possible linear SV classifiers on the pruned training data $D_p$. Let $\hat{f}_N$ be the empirical risk minimizer such that $\hat{f}_N = \arg\min_{f \in \mathcal{F}} \hat{R}_N(f)$ and $\hat{f}_{N_p}$ be the empirical risk minimizer such that $\hat{f}_{N_p} = \arg\min_{f \in \mathcal{F}_p} \hat{R}_{N_p}(f)$ Last, let $\inf_{f \in \mathcal{F}} \mathbb{E}[\mathbb{R}(f)]$ be the lowest value of the risk of the linear SV classifier $f$ that could be analytically calculated. Given some $\delta$ such that $0 \leq \delta \leq 1$, the expected risk $\mathbb{E}[\mathbb{R}](\hat{f}_{N_p})$ of $\hat{f}_{N_p}$ converges, with probability $1 - \delta$, to $\inf_{f \in \mathcal{F}} \mathbb{E}[\mathbb{R}(f)]$ under the upper bound

$$E[\mathbb{R}(\hat{f}_{N_p})] - \inf_{f \in \mathcal{F}} \mathbb{E}[\mathbb{R}(f)] \leq 4 \sqrt{\frac{(d + 2) \log(N + 1) + \log(\frac{2}{\delta})}{N}} + \frac{[r_p^2]_{\text{max}} - [r_p^2]_{\text{min}}}{N}$$

(6.6)

using only $D_p$.

Proof From 6.4 we have

$$E[\mathbb{R}(\hat{f}_{N_p})] \leq \frac{E[r_p^2]}{N}$$

(6.7)
Let $\epsilon > 0$ be the small change on $r^2$ caused by $D_P$ such that $r_p^2 = r^2 \pm \epsilon$ holds. Using (6.7) we have

$$E[R(\hat{f}_{N_P})] \leq \frac{E[r_p^2]}{N} = \frac{E[r^2]}{N} \pm \frac{\epsilon}{N}$$

(6.8)

From (6.4) we also have $R(\hat{f}_N) \leq \frac{E[r^2]}{N}$. Using this in the second line of (6.8) results in

$$E[R(\hat{f}_{N_P})] \leq \max\{R(\hat{f}_N)\} \pm \frac{\epsilon}{N}$$

(6.9)

Subtracting $\inf_{f \in \mathcal{F}} R(f)$ from both sides of (6.9) gives (6.10).

$$E[R(\hat{f}_{N_P})] - \inf_{f \in \mathcal{F}} R(f) \leq \max\{R(\hat{f}_N)\} - \inf_{f \in \mathcal{F}} R(f) \pm \frac{\epsilon}{N}$$

(6.10)

Using (6.5) in the right-hand side of (6.10) and considering the worst case of $r_p^2 = r^2 + \epsilon$ result in (6.11) with probability $1 - \delta$.

$$E[R(\hat{f}_{N_P})] - \inf_{f \in \mathcal{F}} R(f) \leq 4\sqrt{\frac{(d + 2)\log(N + 1) + \log(\frac{2}{\delta})}{N}}$$

$$+ \frac{\epsilon}{N}$$

(6.11)

Since both $r^2$ and $r_p^2$ is expected to exist in the interval $[r_p^2]_{\min}, [r_p^2]_{\max}$ according to the algorithm in Figure 6.6 and the definition of maximum margin in the linear SV classifier [77], $0 \leq \epsilon \leq [r_p^2]_{\max} - [r_p^2]_{\min}$ holds. Using $\epsilon \leq [r_p^2]_{\max} - [r_p^2]_{\min}$ in the right-hand side of (6.11) gives (6.12)

$$E[R(\hat{f}_{N_P})] - \inf_{f \in \mathcal{F}} R(f) \leq 4\sqrt{\frac{(d + 2)\log(N + 1) + \log(\frac{2}{\delta})}{N}}$$

$$+ \frac{[r_p^2]_{\max} - [r_p^2]_{\min}}{N}$$

(6.12)

This concludes the proof of Theorem 6.2.1. 

The upper bound (6.6) is defined as the function of two terms where the second term is the result of using pruned training data. The former upper bound shows that pruned SVC can be as accurate as the original SVC under two conditions: 1)
Very large training data size \((N \to \infty)\) 2) Small size of sensitive attribute domain or low \(\ell\) value or both \(\([r_p^2]_{\text{max}} - [r_p^2]_{\text{min}} \to 0\)\).

Theorem 6.2.1 holds when the pruned training data is mapped into a higher dimensional space \(d'\) using the kernel trick. Although the generalization ability of SVMs with RBF kernel is not formally defined (invalidating Theorem 6.2.1), SVMs with RBF kernel are expected to work under the conditions of Theorem 6.2.1 in the infinite dimensional space \([77, 78]\).

6.3 Experiments and Results

6.3.1 Prerequisites

Datasets

We tested our algorithm on the adult, IPUMS and marketing datasets of the UCI data repository \([62]\) and the fatality dataset of the Keel data repository \([74]\):

1. **Adult**: The adult dataset is drawn from the 1994 census data of the United States \([62]\). It is composed of 45222 instances after the removal of instances with missing values. The binary classification task is to predict whether a person’s adjusted gross income is \(\leq 50K\) or \(> 50K\). The attribute “final weight” is ignored. Education was treated as the sensitive attribute in the experiments.

2. **IPUMS**: This data is drawn from the 1970, 1980 and 1990 census data of the Los Angeles and Long Beach areas \([62]\). It has 233584 instances in total. We picked the 10 attributes that are included in the adult data. The binary classification task is to predict whether a person’s total income is \(\leq 50K\) or \(> 50K\). The classifiers are expected to show a different behavior from the former adult data since the population (and to some extent, classification task, as it is total income rather than adjusted gross income) are different. Again, education was treated as the sensitive attribute in the experiments.
3. **Marketing Data:** This data is drawn from a phone based marketing campaign of a Portuguese banking institution for long term deposits [62]. We created the following binary classification task which is linearly separable under a soft margin SVC: “among all the people who didn’t submit a long term deposit, predict whether a person has a housing loan or not”. We performed the following preprocessing using Weka filters [63]: 1) pick 39922 instances who didn’t make a long term deposit 2) choose four attributes job, day, month and age using the correlation with the class attribute “housing”. Discretized age is treated as sensitive attribute.

4. **Fatality Data:** This data is a U.S. National Center for Statistics and Analysis compilation of 2001 car accidents. The original class attribute has eight labels indicating the level of injury suffered [74]. We created the binary “Injured” and “No_Injury” in the following way: 1) remove the instances with labels “Injured_Severity_Unknown”, “Died_Prior_to_Accident”, “Unknown” and “Possible_Injury’ from the original data. This results in 91085 instances 2) label “Injured” the instances with labels “Nonincapaciting_Evident_Injury”, “Incapaciting_Injury” and “Fatal_Injury”. No feature selection is applied on this dataset. “POLICE_REPORTED_ALCOHOL_INVOLVEMENT” was treated as sensitive attribute.

Weka was used for attribute selection and discretization where needed [63].

Privacy Setup

The anatomization was done according to Xiao et al.’s bucketization algorithm [11]. When \(\ell\)-diversity is not satisfied, the instances were divided into groups of size \(\ell\) according to the original bucketization algorithm. Leftover instances were suppressed (not used in training models).

\(k\)-Anonymized training data was also created for the adult dataset. We used Inan et al.’s value generalization hierarchies in the experiments. The privacy parameters
were \( k = \ell \) for \( k \)-anonymity and \( \ell \)-diversity to compare the classifiers using the same group sizes in the training data.

Anonymized and anatomized training data had the same identifying and sensitive attributes. The sensitive attributes were chosen such that the \( \ell \)-diversity is satisfied for at least \( \ell = 2 \).

Model Evaluation Setup

LibSVM version 3.21 was used for the support vector classification [79]. We trained the support vector machine with linear (SVC) and RBF kernels (SVM).

10-fold cross validation was used for evaluation. The comparison includes pruned SVC/SVM, original SVC/SVM and identifying SVC/SVM. The comparison on adult dataset also include \( k \)-anonymized SVC/SVM. The \( k \)-anonymized SVC/SVM are not included for other datasets since Inan et al. provided generalization hierarchies only in the adult dataset [25]. Last, the error rates of pruned and original SVC/SVM are compared using the Student \( t \)-test (See Section 6.3.3). Other models are not included, because Theorem 6.2.1 covers only pruned and original SVC/SVM.

6.3.2 Analysis of Results

The first aspect is the comparison between the pruned and original SVC/SVM. From Theorem 6.2.1 we expect that the average error rates of pruned SVC/SVM will be greater then the original SVC/SVM’s if there is no suppression due to the \( \ell \)-diversity constraint. One exceptional case would be the regularization effect where \( \ell \)-diversity and the pruning algorithm reduces either the bias of underfitting SVC/SVM or the variance of overfitting SVC/SVM. Another exceptional case would be the suppression of many instances of the original training data due to \( \ell \)-diversity constraint. This violates the assumption of Theorem 6.2.1. We will refer to this aspect of experiments pruned-to-original in the rest of the chapter.
The second aspect is the comparison between pruned and identifying SVC/SVM. From the shattering properties of the statistical learning theory, the pruned SVC/SVM are expected to outperform the identifying SVC/SVM if the sensitive attribute is a good predictor of the class attribute. If the sensitive attribute is a bad predictor of the class attribute, the opposite of the former behavior is expected to occur. We will refer to this aspect of experiments pruned-to-identifying in the rest of the chapter.

Last, the third aspect is the comparison between the pruned and $k$-anonymized SVC/SVM. The pruned SVC/SVM are expected to outperform the $k$-anonymized SVC/SVM because anatomization preserves the original values for all the attributes. We will refer to this aspect of experiments pruned-to-$k$-anonymized in the rest of the chapter.

![Adult Data Linear SV](image)

**Figure 6.7.** SVC on Adult

Figures 6.7 to 6.14 show the results of all the experiments. In all Figures, “Org.” and “Id.” labels will stand for the original SVC/SVM and the identifying SVC/SVM.
respectively. The pruned and $k$-anonymized SVM will be represented by their respective privacy parameters ($L$ for $\ell$ and $k$ for $k$.)

Figure 6.7 shows a surprising result of pruned SVC in the pruned-to-original and pruned-to-identifying cases. In the pruned-to-original case, increasing $\ell$ reduces the average error rate of the pruned SVC; $\ell$-diversity and the pruning algorithm apparently regularize the underfitting original SVC. Theorem 6.2.1 does not hold as well for $\ell \geq 4$, because some original training data instances are suppressed. In

![Adult Data RBF SVM](image)

Figure 6.8: SVM on Adult

In the pruned-to-identifying case, the sensitive attribute is a bad predictor of the class attribute. Identifying SVC performs better than many pruned SVCs. In the pruned-to-$k$-anonymized case, the average error rate of pruned SVC is less than the average error rate of $k$-anonymized SVC for all $\ell = k$ values.

Figure 6.8 shows the expected results of pruned SVM in all three aspects. Increasing $\ell$ result in the increase of average error rate for pruned SVM and original SVM outperforms the pruned SVM. The expectation from Theorem 6.2.1 occurs here
despite suppression (violation of assumption). We believe that $\ell$-diversity and pruning algorithm act as a regularizer for SVMs with RBF kernel which tend to overfit to the training data. Notice that the sensitive attribute is a good predictor of the class attribute in the infinite dimensional space since average error rate of pruned SVM is less than the average error rate of identifying SVM. Last, the average error rate of the pruned SVM is less than the $k$-anonymized SVM by 0.1.

Figure 6.9 shows in general the expected result of pruned SVC in the pruned-to-original. $\ell = 4$ is a special case where its average error rate is greater than the pruned SVC’s that is trained on 5-diverse data. Theorem 6.2.1’s assumption is violated again when $\ell = 5$ because some original training data instances are suppressed. In the pruned-to-identifying case, the pruned SVC cannot capture the good shattering property that the sensitive attribute provide in the original dimensional space.

Figure 6.10 shows in general the expected result of pruned SVM for pruned-to-original and pruned-to-identifying aspects. In the pruned-to-original aspect, the pruned SVM outperforms the original SVM when $\ell = 2$. This shows that the pruning algorithm and $\ell$-diversity has the regularization effect even if the sensitive attribute is a good predictor according to the pruned-to-identifying aspect. The regularization
case could occur in general, because it is statistically significant for confidence interval 0.95 (See Table 6.2).

Figure 6.10.: SVM on IPUMS

Figure 6.11.: SVC on Marketing

Figure 6.11 show the expected behavior of pruned SVC in the pruned-to-original aspect. One thing to emphasize is the surprising spike in the error rate distribution when \( \ell = 5 \). The reason is that the original training data satisfies the \( \ell \)-diversity
condition when $\ell = 2$ and $\ell = 3$. When $\ell = 5$, almost half of the training instances are suppressed. This strongly violates the assumption of Theorem 6.2.1 and the result is also not statistically significant (cf. Table 6.1). We should note that the sensitive attribute is a bad predictor since the average error rates of pruned SVC are greater than the identifying SVC’s.

Figure 6.12.: SVM on Marketing

Figure 6.12 show the expected behavior of pruned SVC in the pruned-to-original aspect. The pruned SVC also show the expected result in the pruned-to-identifying aspect. The sensitive attribute is not a good predictor in the infinite dimensional space.

In Figure 6.13, the pruned SVC gives an interesting and surprising result in the pruned-to-original case. The average error rate of pruned SVC is approximately same as the average error rate of original SVC for all $\ell$ values. We believe that this would only occur in this dataset because the results are not statistically significant (cf. Table 6.1). When $\ell = 3$ and $\ell = 4$, the assumption in Theorem 6.2.1 is violated because most of the training instances are suppressed. In the pruned-to-identifying case, sensitive attribute is bad (insignificant) predictor since the pruned SVC does not reduce the error rate of the identifying SVC.
Figure 6.13.: SVC on Fatality

Figure 6.14.: SVM on Fatality

Figure 6.14. show the expected results of pruned SVM in the pruned-to-original and pruned-to-identifying aspects (despite violating the assumption of Theorem 6.2.1). In the pruned-to-identifying aspect, note that the sensitive attribute is a bad predictor in the infinite dimensional space. The average error rate of pruned SVM is greater than the identifying SVM’s.
6.3.3 Student $t$-test for Pruned SVC/SVM versus Original SVC/SVM

Tables 6.1 and 6.2 give the statistical test results for confidence interval 0.95. In all Tables, “P” stands for pass while “F” stands for fail. “N/A” stands for not applicable in cases where the domain size of the sensitive attribute is less than the $\ell$ value. “Org.” stand for the original SVC/SVM whereas “$\ell$” stand for the pruned SVC/SVM. Note that we do the test for original SVC/SVM vs pruned SVC/SVM, because the Theorem 6.2.1’s scope covers this analysis.

Table 6.1.: Pruned SVC vs Original SVC

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\ell=2$</th>
<th>$\ell=3$</th>
<th>$\ell=4$</th>
<th>$\ell=5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>P</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>Fatality</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>N/A</td>
</tr>
<tr>
<td>IPUMS</td>
<td>F</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Adult</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

In Section 6.3.2, we saw the theoretically expected results for pruned SVC vs original SVC when they are trained on IPUMS dataset (cf. Figure 6.9). Table 6.1 shows that the difference between the pruned and original SVC is statistically significant for almost all $\ell$ values. We saw, in contrast, theoretically unexpected results in case of pruned SVC vs. original SVC on the adult and fatality datasets. Table 6.1 shows that the difference between the pruned and original SVC are statistically insignificant in adult and fatality datasets. As such, the theoretically unexpected results are likely to occur by just random chance, rather than suggesting an issue with Theorem 6.2.1. In the marketing dataset, the difference between the pruned and the original SVC is statistically insignificant for $\ell$ values 3-to-5. When $\ell = 5$, most of the training instances were suppressed. Note that Theorem 6.2.1 holds if and only if both the
pruned and the original training dataset have the same number of instances (no or negligible suppression.) (See Theorem 6.2.1)

Table 6.2 shows that the difference between pruned SVM and original SVM are statistically significant in almost all datasets for multiple \( \ell \) values. The expectation from Theorem 6.2.1 occurred in all the datasets. (cf. Figures 6.8, 6.10, 6.12 and 6.14) The good results are therefore very unlikely to occur by random chance, suggesting Theorem 6.2.1 applies to infinite dimensions. This result concurs with Vapnik and

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Org. vs ( \ell=2 )</th>
<th>Org. vs ( \ell=3 )</th>
<th>Org. vs ( \ell=4 )</th>
<th>Org. vs ( \ell=5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Fatality</td>
<td>F</td>
<td>P</td>
<td>P</td>
<td>N/A</td>
</tr>
<tr>
<td>IPUMS</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Adult</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
</tbody>
</table>

Burges’ claim for the original SVM when there is no suppression [77, 78]. Last, we observe surprisingly significant results when the assumption of Theorem 6.2.1 is violated. We believe that the \( \ell \)-diversity and pruning acts as a regularizer since SVMs with RBF kernel tend to overfit to the training data.

In summary, we measured the statistically significant error rates when the pruned SVC/SVM show the expectation from Theorem 6.2.1. The error rates were statistically insignificant when they don’t respect the expected result of Theorem 6.2.1 or when the pruned training data violates the assumption of Theorem 6.2.1.
7 SUMMARY

This dissertation proposes the following problem in the context of data classification under the privacy standard \( \ell \)-diversity:

Define heuristics to train classifiers on anatomized data without violating \( \ell \)-diversity while using the sensitive information, with a theoretical guarantee of good generalization under reasonable assumptions.

As the problem description states, \( \ell \)-diversity is respected in the context of the anatomization scheme. This problem is tackled following both empirical and theoretical methodology. The empirical approach is set in a distributed environment which makes the solution of the problem relatively easy, because the owner of the data is involved in the learning scheme. The purpose of the empirical approach is to investigate the feasibility of the problem for further theoretical elaboration. Provided the interesting outcome of the empirical methods in the distributed learning, the theoretical analysis is done for non-parametric and parametric classifiers in a more challenging non-distributed setting. In this case, the challenge is in building the classifiers without involving the owner of the data. All the work is achieved for a real world prediction task that is defined for anatomization.

Chapter 4 explains the first proposed method in the distributed setting, a distributed decision tree classifier. The proposed method is shown to preserve the privacy constraint \( \ell \)-diversity. The distributed decision tree classifier is tested on various datasets and the results show that fairly accurate decision trees can be built whereas the learning cost is reduced remarkably for the owner of the data.

Chapter 5 demonstrates the feasibility of non-parametric classification in non-distributed setting. The chapter mainly focuses on the k-nearest neighbor classification (k-NN). We show that the asymptotic error bounds are the same for anatomized
data as for the original data. Perhaps surprisingly, the proposed 1-NN classifier has a faster convergence to the asymptotic error rate than the convergence of 1-NN classifier using the training data without anatomization. In addition, the analysis suggests that any non-parametric classifier using the anatomized training data has the variance term of generalization error that is less than the non-parametric classifiers’ using the original training data. In contradiction, any non-parametric classifier using the anatomized training data has bias terms of generalization error that are greater than the non-parametric classifiers’ using the original training data. The anatomized training data thus pushes the optimum point of bias variance tradeoff towards the bias terms. Experiments on multiple datasets confirm the theoretical convergence rates. These experiments also demonstrate that proposed k-NN on anonymized data can outperform k-NN on the original data. In particular, the experiments on well known Adult data show that 1-NN on anatomized data outperforms learning on data anonymized to the same anonymity levels using generalization.

Chapter 6 investigates the parametric classification in a non-distributed setting. The chapter mainly focuses on the support vector classification. We propose a preprocessing algorithm for anatomization. Our algorithm estimates a linearly separable training data from the anatomized training data. We define the generalization ability of support vector classifiers when they are trained on the former preprocessed data. The key point to remember is that our algorithm gives good generalization guarantees to support vector classifiers. The proposed mechanism is evaluated on multiple publicly available datasets and accurate models are observed in most cases while $\ell$-diversity is preserved.

This dissertation has significant impact on applied privacy. Iyengar claims that the right metric choice and the consideration of classification task give $k$-anonymized data that results in both high utility and high privacy guarantee. Although he shows empirically the effect of the group size on the classification error, he doesn’t give the theoretical justification about how and why the increasing group size (and eventually stricter generalization and suppression) impacts the classification error.
Nergiz et al. later show empirically that the small sized groups (small \( k \)) result in low classification errors under certain anonymization techniques and metrics \[80\]. However, their work still excludes the theoretical justification about the grouping impact onto classification error. This dissertation fills this theoretical gap through a similar standard \( \ell \)-diversity which is also based on groups. The anatomization scheme here is used to show the impact of the group size.

Anatomization is preferred in this work, because it provides higher data utility and interpretability than the generalization based \( k \)-anonymity and \( \ell \)-diversity methods \[11\]. At the same time, it also provides some level of protection for the sensitive attribute. Through the choice of anatomization, this work also shows that the anatomization is a viable option for data publishing within a company or an institution. The results convey a strong argument against giving everybody access to the original data within a company to achieve predictive tasks. Such argument is valuable as more and more data leaks occur over the years because of insider attacks.

Future work exists in the model, the privacy standard and the data publishing scheme choice. In the model choice, extending the theoretical analysis of anatomization based \( \ell \)-diversity to the other modeling techniques such as the decision trees, neural networks and ensemble methods would be significant contributions. Provided the same models in this dissertation, comparing theoretically the generalization ability on anatomization based \( \ell \)-diversity with the generalization ability on generalization based \( \ell \)-diversity, \( k \)-anonymity and differential privacy would also be other valuable directions. In particular, showing the generalization impact on the \( \ell \)-diverse groups would provide valuable insights about the specific data publishing method choice.

We personally recommend that all these potential directions should include the scalability challenges as we are currently living in the big data era. We also advise to the followers of this work to consider first finding the right type of personal data which has an interesting learning task. This is often ignored due to the theoretical nature of the topic although the search of the data is one of the biggest challenges for the privacy work in academia.
REFERENCES
REFERENCES


VITA
VITA

Koray Mancuhan received his BS in computer engineering at Galatasaray University in 2010 and his MS in computer science at Purdue University in 2014. His main research areas are applied machine learning and data mining. His interest is spread across different applications such as discrimination detection/prevention, privacy and personality recognition. He has been a student member of professional associations such as IEEE, SIAM and ACM. During his PhD work, he received two research grant awards from the Purdue Research Foundation and the Northrop Grumman Cybersecurity Consortium in collaboration with his advisor Chris Clifton. He received his PhD degree in December 2017 from the Department of Computer Science at Purdue University.