Unrealization Approaches for Privacy Preserving Data Mining

by

James Williams
B.A., University of British Columbia, 1999
B.Sc., University of British Columbia, 2002
J.D., University of Victoria, 2008

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

Unrealization Approaches for Privacy Preserving Data Mining

by

James Williams
B.A., University of British Columbia, 1999
B.Sc., University of British Columbia, 2002
J.D., University of Victoria, 2008

Supervisory Committee

Dr. Valerie King, Co-supervisor, (Department of Computer Science).

Dr. Jens Weber, Co-supervisor, (Department of Computer Science).

Dr. Imir (Alex) Thomo, Departmental Member, (Department of Computer Science).

Supervisory Committee

Dr. Valerie King, Co-supervisor, (Department of Computer Science).

Dr. Jens Weber, Co-supervisor, (Department of Computer Science).

Dr. Imir (Alex) Thomo, Departmental Member, (Department of Computer Science).

ABSTRACT

This thesis contains a critical evaluation of the unrealization approach to privacy preserving data mining. We cover a fair bit of ground, making numerous contributions to the existing literature. First, we present a comprehensive and accurate analysis of the challenges posed by data mining to privacy. Second, we put the unrealization approach on firmer ground by providing proofs of previously unproven claims, using the multi-relational algebra. Third, we extend the unrealization approach to the C4.5 algorithm. Fourth, we evaluate the algorithm's space requirements on three representative data sets. Lastly, we analyse the unrealization approach against various issues identified in the first contribution. Our conclusion is that the unrealization approach to privacy preserving data mining is novel, and capable of addressing some of the major challenges posed by data mining to privacy. Unfortunately, its space and time requirements vitiate its applicability on real-world data sets.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

Jens Weber and Valerie King, for their patience.

I do not pretend to start with precise questions. I do not think you can start with anything precise. You have to achieve such precision as you can, as you go along.
Bertrand Russell

Big Brother in the form of an increasingly powerful government and in an increasingly powerful private sector will pile the records high with reasons why privacy should give way to national security, to law and order, to efficiency of operation, to scientific advancement and the like.
Justice William O. Douglas

DEDICATION

This work is dedicated to several influential mentors in my undergraduate programs: Alan Richardson, Paul Bartha and Will Evans.

Chapter 1

Introduction

This thesis contains an evaluation of a new method in privacy preserving data mining –the *unrealization* approach to decision tree induction discovered by Pui Fong [20]. Given the growing proliferation of databases, as well as the increasing sophistication of data mining methods, new approaches to privacy preservation are desperately needed if informational privacy interests are to be protected. Although data protection law was designed to safeguard privacy in the face of advancing technology, the advent of data mining poses unique challenges that cannot be solved by legal means alone.

Fong's unrealization approach presents a novel method for preserving privacy in data mining. Concentrating on classification scenarios, he showed how one can construct decision trees for a database by using a *data complementation* approach that hides the original training data. Instead of creating a decision tree from a training set directly, Fong uses the training set to create two *unreal* data sets, each of which contains spurious information. Given that these unreal data sets contain false information, they can be safely released to a data recipient, in place of the original (possibly sensitive) training data. This provides a degree of security against data recipients who may wish to use the information for secondary purposes.

Although useless on their own, the unreal data sets are very useful when combined with a modified ID3 decision tree inducer. In his thesis, Fong shows that the decision tree that results from using his modified ID3 algorithm on the unreal data sets is the same tree that would have been generated from using the standard ID3 algorithm on the original training set.

Fong's approach leads to an obvious usage scenario, in which a data custodian releases unreal data sets to a data recipient, in place of sensitive data sets. Since the data in the unreal data sets is useless to anyone who does not use the modified ID3 decision tree algorithm, the original data would be safeguarded against secondary uses. If feasible, this approach would revitalise privacy protection, as many data sharing arrangements could be addressed using this type of model.

Apart from a background section that provides a solid introduction to privacy and data mining, the bulk of this work is devoted to a critical examination of Fong's unrealization approach. Our contribution in this thesis consists of:

1. Providing the most up-to-date and accurate analysis of the challenges that data mining poses to modern data protection regimes.

2. Putting the unrealization approach outlined in Fong [20] on more mature footing by: a) providing an axiomatization of the multi-relational calculus, and; b) proving claims that were merely asserted in the original presentation.

3. Extending the unrealization approach to the industry-standard C4.5 algorithm, from the rarely-used ID3 approach.

4. Evaluating the unrealization approach against several real-world data sets.

5. Providing an evaluation of the merits of the unrealization approach, with respect to both privacy preservation and space/time requirements.

We begin with a background section that is designed to appeal to readers from a variety of disciplines. Without a thorough understanding of privacy, data mining and technical approaches to privacy, the average reader will have a difficult time following the material contained in this thesis.

Chapter 2

Background

In this chapter, we introduce some of the key concepts underlying the field of privacy preserving data mining. We begin by discussing the concept of *privacy*, emphasising its amorphous nature, its rationales, and its instantiation in modern legal regimes. Our main interest in this work is *informational privacy* –the category of privacy interest that focuses on an individual's control over personal information. We claim that: a) technological advances have created new risks to informational privacy interests, and; b) these risks require corresponding technological advances in data protection; existing safeguards are simply inadequate to deal with the implications of improved processing capacity on the part of private and public sector organisations.

Following the introductory section on privacy, we discuss one of the aforementioned technological advances –namely, *data mining*. Keeping the discussion at an introductory level, we recap some of the key concepts in data mining, including its use in prediction. Of critical importance is the impact of data mining techniques on privacy interests. We provide an accurate and rigorous assessment of the major challenges to privacy that arise from the use knowledge discovery techniques.

The last section of the chapter contains a brief exposition of the work performed in the data mining and database communities on privacy protection. Without delving into exquisite detail, we recount the major approaches that have been explored by various research communities. A comprehensive overview of *privacy preserving data mining* ("PPDM") is required in order for the reader to assess the merits of the unrealization approach.

2.1 Privacy

We begin our review of basic material with a brief (but rigorous) discussion of privacy. As stated above, we claim that the body of privacy law that has developed over the course of the last century cannot adequately address certain threats that arise from the growing sophistication of information technology. In order to convince the reader of this assertion, it is necessary to outline the major features of traditional data protection regimes, as well as the recent technological advances that have called them into question. In particular, Section 2.1 is partitioned into these sub-sections:

1. **Fundamental Concepts:** This sub-section contains a discussion of the basic concepts of privacy. We cover the different categories of privacy interests, including territorial, bodily, informational and communications privacy.

2. **The Normative Basis of Privacy:** We subsequently undertake a brief treatment of the importance of privacy. Accounts of the value of privacy interests are typically grounded in utilitarian or deontological reasoning, and we mention key examples in each category.

3. **Informational Privacy:** The next sub-section discusses the formulation of privacy that is most affected by information technology. We discuss the traditional dynamic in which technological innovation spurs (sometimes belated) calls for increased privacy protection.

4. **Data Protection Regimes:** Following the discussion of informational privacy, we recount the main legal tools used to provide privacy protection in the face of advancing technological developments. We present one of the most common formulations of the *fair information practises*, which will figure prominently in the pages to follow.

5. **New Challenges to Informational Privacy:** This sub-section introduces five examples of recent technological innovations that are causing problems for data protection regimes, namely: a) increased storage capacity; b) automated decision support; c) social networking; d) ubiquitous computing, and; e) data mining.

6. **An Overview of Technical Approaches to Privacy:** We subsequently present a very short overview of the work that the computer science research community has performed in respect of privacy. We introduce statistical control in databases, data sanitization and other exciting areas of research. The main purpose of this sub-section is to give the reader a sense of the technical tools available.[1]

7. **Measuring Privacy Protection:** The last sub-section discusses mathematical metrics for measuring privacy protection. We present the concept of *differential privacy*, which we use later in this work in evaluating the unrealization approach.

As the reader can discern from our outline, the background section on privacy is quite verbose. Although not immune to the charms of brevity, I believe that a rigorous discussion of privacy-preserving data mining ("PPDM") techniques must be firmly grounded in both the law and history of privacy protection, as well as the technical aspects of data mining. Without a solid understanding of the concepts, rationales and traditional approaches to privacy, a researcher in the sciences may wander in the wrong direction.

In addition, the depth of treatment offered in this section has one happy side effect: it enables us, in subsection 2.2.3, to give one of the most accurate accounts of the challenges posed by knowledge discovery techniques to existing data protection regimes. Many of the works on privacy and data mining in both the legal and computer science communities are imprecise at best, since there are few researchers with the requisite skills to span both areas.

With this outline of the current section in hand, we turn to our first task –namely, providing an introduction to the basics of privacy.

[1]Section 2.2 of this Chapter will engage in a more lengthy discussion of data mining, its impact on data protection regimes, and privacy-preserving variants of traditional data mining algorithms.

2.1.1 Fundamental Concepts

Judging by its prominence in both legal systems and common discourse, privacy is regarded as an important norm in most (if not all) of the world's countries and cultures. As noted by Swire and Bermann [52], the concept of privacy is found in some of the oldest written texts. Practises subsumed by the privacy concept are found in the Qur'an, the Talmud, and the New Testament, and in ancient Chinese and Greek law.[2] In modern times, privacy has been recognised as a human right by the General Assembly of the United Nations,[3] and rights to privacy have been either explicitly stated or implicitly recognised in the Constitutions of various nation states.

Although undoubtedly a concept of great importance, privacy has proven notoriously difficult to define. In the words of Daniel Solove, privacy appears to be a sweeping concept, encompassing *"freedom of thought, control over one's body, solitude in one's home, control over personal information, freedom from surveillance, protection of one's reputation, and protection from searches and interrogations."*[48] Privacy has been approached by scholars from a variety of disciplines, including economics, law, sociology, political science and computer science. A brief survey of the literature on privacy will reveal a great diversity of opinion about not only the meaning of the term, but of its status as a normative concept.

Skipping a detailed treatment of these issues for the sake of brevity, we feel that it is sufficient to point out that privacy is a multifaceted concept that faces a number of challenges in terms of vagueness, ambiguity, and reductionism. In this thesis, we have opted to sidestep these issues, concentrating on privacy norms as enunciated in the legal systems of Europe and North America. [4]

As related by Swire and Bermann [52] the legal protection of privacy in Anglo-American law dates to the *Justices of the Peace Act*[5], which included provisions intended to stop 'peeping toms' and eavesdroppers. In 1675, Lord Camden struck down a warrant to enter and seize papers from a home, declaring that no law could

[2] For a discussion of privacy as enunciated in religious texts, see also [18].

[3] *"No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence"* Article 12, Universal Declaration of Human Rights

[4] Following Guarda [23], we regard the legal dimension as fundamental in the context of issues relating to data processing. Not only does the legal dimension set the responsibilities for both private and public sector organisations, but it is the most widely discussed dimension in both the practical and academic literature.

[5] *Justice of the Peace Act*, England, 1361.

justify such an act. If there were, Camden stated, "it would destroy all the comforts of society, for papers are often the dearest property any man can have". Various European countries followed by passing legislation that endowed individuals with privacy rights. The first (and perhaps most pithy) definition of privacy in modern Anglo-American law was due to Cooley, who defined privacy as the *"right to be let alone."*[15] Jurisprudence in Europe and North America continued to build on these advances, as courts grappled with the issue of privacy protection in a number of contexts, including search, seizure and surveillance.

Despite the apparent simplicity of the "right to be let alone", it quickly became apparent that privacy was a broad concept. The various legal instruments and court judgements evidenced a wide variety of interests that fell under the rubric of privacy. Commentators have partitioned these interests into the following categories:[6]

- **Territorial privacy:** this type of privacy interest relates to control over one's spatial environment. Claims of this sort have been regulated in the western legal tradition by rules relating to property. Violations of territorial privacy can result from trespass, video surveillance and remote or hidden listening devices.
- **Privacy of the body:** this type of privacy interest relates to control over one's person. Claims of this sort are typically addressed in law through prohibitions against unlawful confinement, assault, battery, and unwarranted search and seizure. Violations can arise through these means, as well as more subtle acts, such as genetic testing.
- **Informational privacy:** this type of privacy interest relates to an individual's control over information relating to them. It is based on the idea that information about an individual is in a fundamental way her own, for her to communicate or retain as she sees fit.
- **Communications privacy:** this type of privacy interest involves protection of the means and content of correspondence, including mail, email, and telephone.

It is the information privacy interest that is of interest in this thesis. Before discussing informational privacy in more detail, we quickly recount the rationale for the legal recognition of privacy interests.

[6]See, for instance, Swire and Bermann [52], or the Commission on Freedom of Information and Individual Privacy [39].

2.1.2 The Normative Basis for Privacy

The normative basis of privacy interests have been explored by scholars from a variety of disciplines, including law, philosophy, sociology and history. In general, there are two categories of justification for the importance of privacy interests: *utilitarian*, and *deontological*. Utilitarian accounts of privacy are by far the most numerous, and focus on the effects that privacy interests have on the utility of individuals or groups. Deontological arguments ground privacy interests in other norms that individuals or groups possess. We briefly present examples of each type of argument, in an effort to show the importance of privacy claims.

As an empirical matter, individuals have a number of practical interests which may be seriously harmed by invasions of their privacy, including social standing, employment prospects and the maintenance of relationships. In addition, privacy can have great utility for social groups; according to Shafer [46], a wide variety of groups require a kind of nutritive privacy to protect their organisational life.

Utilitarian arguments for the value of privacy justify privacy on the basis of these interests. Examples of utilitarian arguments include:

- **Personal Development:** As an example, John Stuart Mill argued that there is a close correlation between the availability of a protected zone of privacy, and an individual's ability to freely develop her *individuality* and *creativity*.
- **Integrity and Identity:** Some commentators believe that an individual's integrity (and the development and preservation of *personal identity*) require the protection of a zone of privacy within which the ultimate secrets of one's "core" self remain inviolable against unwanted intrusion or observation.
- **Alleviating Stress:** Other scholars have claimed that social life is frequently stressful, and generates tensions which would be unmanageable unless the individual had opportunities for periods of privacy [56].
- **Enabling Social Relations:** Lastly, some have argued that privacy is valuable because it provides the rational context of a number of *ends*, including love, trust, friendship and self-respect. It is a necessary element of these ends, and not an ancillary one.

In addition to these utilitarian arguments, some commentators have advanced non-utilitarian grounds for the importance of privacy. For instance, some commentators have argued that to respect someone as a person is to concede that one ought to take account of the way in which his enterprise might be affected by one's own decisions.

As the purpose of this thesis does not involve an analysis of the normative basis of privacy relationships, it is sufficient to point out that privacy seems to have significant ramifications for both individuals and groups. Even in a paternalistic legal system, regulators must take care to explicitly balance privacy interests with other social values. As stated by the Canadian *Commission on Freedom of Information and Individual Privacy*, at least two aspects of personal autonomy are threatened by privacy invasions:[7]

1. our relationships with other individuals, and;
2. our relationships with institutions.

Of these, the latter is of the utmost importance for our purposes. Given the growing numbers of databases (and growing interest in using data)[8], the ability of institutions to view the intimate details of an individual's life may be increasingly steadily.

In the next section, we turn our attention to the topic of informational privacy interests, and the tools that have been developed in the last century to sustain them within the context of the modern liberal state. As mentioned above, our claim is that these existing safeguards may not be sufficient to deal with emerging trends in information technology.

[7] See [39] at p.501.

[8] For a recent example on the increasing number of government requests for data held by social networking websites, see R. Lardner, Break the law and your new friend may be the FBI, Associated Press, March 16, 2010.

2.1.3 Informational Privacy and the Challenge of Technology

As stated above, this thesis concentrates on the *informational* aspect of privacy, which concerns an individual's control over information relating to them.[9] One of the earliest statements of informational privacy in the common law is due to Samuel Warren and Louis Brandeis. In a seminal paper (prompted by disgust at encroachments by members of the local media) the two jurists stated that:

"[t]he intensity and complexity of life...have rendered necessary some retreat from the world, and man, under the refining influence of culture has become more sensitive to publicity, so that solitude and privacy have become more essential to the individual; but modern enterprise and invention have, through invasions upon his privacy, subjected him to mental pain and distress, far greater than could be inflicted by mere bodily injury."[45]

Warren and Brandeis were concerned with several new technologies that made the dissemination of personal information feasible on a broad scale - namely, portable photographic equipment and improved printing presses. The age of the "pen and brush" caricature and political cartoon had yielded to technology that could produce a black and white approximation of a photographic image on any paper surface.

The concern over the rapid growth of information technology was picked up in the mid 20th century by Alan Westin, the father of modern data protection law. In Westin's formulation, informational privacy is the *"claim of individuals, groups or institutions to determine for themselves when, how, and to what extent information about them is communicated to others"* [56]. Taking a cue from Warren and Brandeis, Westin stated that technological advances *"now make it possible for government agencies and private persons to penetrate the privacy of homes, offices and vehicles; to survey individuals moving about in public places; and to monitor the basic channels of communication by telephone, telegraph, radio and television."*

In addition to new technologies, the growing power of the state was a major contributor to privacy concerns. Governments began reaching into more and more aspects of life, offering programs such as welfare, workers compensation, auto insur-

[9] Although a discussion of the merits of various conceptual approaches to privacy is beyond the scope of this work, one advantage of informational privacy is that it enables us to understand how the concept of privacy can compass both "being let alone", as well as communicating with others. See [46] for more details.

ance, subsidised housing and other hallmarks of modern liberalism. In so doing, they became party to a growing collection of information on citizens. In the words of the Commission on Freedom of Information and Individual Privacy, *"[t]he development of modern forms of social organisation of increasing size and complexity, and the corresponding growth of large public and private institutions have given rise to an unprecedented growth in the collection, analysis and use of information. This increase in institutional needs for information has been coupled with remarkable gains in the sophistication and capacities of technologies used in the gathering, storage, analysis and dissemination of information. It is often said, with good reason, that we are living in an 'information age'. Personal information concerning individuals is now collected and used by large institutions to an extent that would have been inconceivable to previous generations."* [39, at p.495]

As a result of this increasing accumulation of information, both private and public sector organizations hold extensive dossiers on individual citizens. In a similar fashion to the 19th-century technological innovations that vexed Warren and Brandeis, modern information technology can capture, process and transmit data about individuals far beyond the reach of their local social networks. According to Shafer, the *"ordinary citizen who, in earlier times, would have been known only in his or her own community, now leaves a 'trail of data' behind with almost every project undertaken: the tax form completed; the social welfare claimed; the application for credit, insurance or a drivers license; or the purchase of consumer goods."*[46]

The potential impacts of misuse of personal information can be severe. In the words of one commentator, *"[t]he accumulation of personal information on an individual enables the creation of a composite image of that person that is often false and reductionist. More and more one hears of the electronic identity of a person... It becomes a determining factor of the individual's potential for action and development. That identity could be stolen or appropriated. It serves to categorise a person. When doubt is cast- even if it is unfounded- on his or her integrity, that identity can prevent a person from travelling, from finding a place to live or a job, or to obtain insurance. The closer personal information comes to the biographical heart of a person, the more that information can have significant consequences on the shaping of identity and on imposing serious limitations."*[10]

[10]DAoust R, The Proliferation of Data Banks, Speech at the National Forum on Criminal Records: Economic, Social, Legal and Political Issues, November 29, 2004.

Indeed, the general public seems to be aware of the risks that accompany the growing number of databases containing personal information. The latest Equifax/Harris Consumer Privacy Survey showed that over 78% of respondents see computer technology as a threat to personal privacy. Furthermore, almost 76% believe that they have lost control over their personal information. Stories in the media about the use of information by governments and large companies[11] undoubtedly play a role in this perception. If the use of technology to manage and process personal information is to have a less sinister reputation among the general public, the potential risks to personal privacy must be mitigated. Mechanisms to accomplish this very task are the subject of our next sub-section.

2.1.4 Data Protection Regimes

In the previous section, we outlined the role played by information technology in posing challenges to privacy interests.[12] The 20th century solution to this issue (as urged by Westin and other scholars) involves the creation of a *data protection regime* –a regulatory regime that: a) subjects information systems to a regulatory oversight, and; b) grants individuals legal rights that are intended to afford them a degree of control over their personal information.

Since privacy interests can conflict with other public policy goals, the task of developing a data protection regime is inherently complex. In the words of the Commission on Freedom of Information and Individual Privacy, the issue involves *"striking appropriate balances between [organisational interests] in the collection and use of personal information, and the interests of the individual in reducing the impact of data collection, in participating in decisions with respect to subsequent use, and in ensuring fairness in decision-making based on personal files."*[39]

[11] See, for example, Amy S. Clark, Employers Look At Facebook Too: Companies Turn To Online Profiles To See What Applicants Are Really Like, CBS News, June 20, 2006.

[12] As stated by DeVries, *"[t]he modern evolution of the privacy right is closely tied to the story of industrial-age technological development - from the telephone to flying machines. As each new technology allowed new intrusions into things intimate, the law reacted - slowly - in an attempt to protect the sphere of the private."* [18, at p.285]

A competing view is offered by Taipale, who states that *"[s]ecurity and privacy are not a balancing act but rather dual obligations of a liberal democracy that present a wicked problem for policy makers. Wicked problems are well known in public policy and are generally problems with no correct solution."*[54]

As a matter of the historical record, the development of data protection regimes began with the use of *soft* legal mechanisms. The response of privacy advocates and legislators to the increasing sophistication of institutional data collection was to promulgate sets of *guidelines* around the collection, use and disclosure of personal information. The first steps in this direction were contained in a report of a committee of the United States Department of Health, Education and Welfare [35]. In the report, the committee clearly articulated five fundamental principles of *fair information practise*:

1. There must be no personal data record-keeping systems whose very existence is secret.

2. There must be a way for an individual to find out what information about him is in a record, and how it is used.

3. There must be a way for an individual to prevent information about him that was obtained for one purpose from being used or made available for other purposes without his consent.

4. There must be a way for an individual to correct or amend a record of identifiable information about him.

5. Any organization creating, maintaining, using or disseminating records or identifiable personal data must assure the reliability of data for their intended use, and must take precautions to prevent misuse of the data.

These principles became the basis of modern data protection regimes. For instance, they were explicitly used as a model for the influential Organization for Economic Cooperation and Development ("OECD") guidelines [21], which promulgated eight core principles of fair information practise:

1. **The Collection Limitation Principle:** There should be limits to the collection of personal data, and any such data should be obtained by lawful and fair means, and, where appropriate, with the knowledge or consent of the data subject.

2. **The Data Quality Principle:** Personal data should be relevant to the purposes for which they are to be used, and, to the extent necessary for those purposes, should be accurate, complete and kept up-to-date.

3. **The Purpose Specification Principle:** The purposes for which personal data are collected should be specified not later than at the time of data collection and the subsequent use limited to the fulfilment of those purposes or such others as are not incompatible with those purposes and as are specified on each occasion of change of purpose.

4. **The Use Limitation Principle:** Personal data should not be disclosed, made available or otherwise used for purposes other than those specified in accordance with [the purpose specification principle], except: a) with the consent of the data subject, or; b) by the authority of law.

5. **The Security Safeguards Principle:** Personal data should be protected by reasonable security safeguards against such risks as loss or unauthorised access, destruction, use, modification or disclosure of data.

6. **The Openness Principle:** There should be a general policy of openness about developments, practises and policies with respect to personal data. Means should be readily available of establishing the existence and nature of personal data, and the main purposes of their use, as well as the identity and usual residence of the data controller.

7. **The Individual Participation Principle:** An individual should have the right: a) to obtain from a data controller, or otherwise, confirmation of whether or not the data controller has data relating to him; b) to have data relating to him communicated to him, within a reasonable time, at a charge, if any, that is not excessive; in a reasonable manner; and in a form that is readily intelligible to him; c) to be given reasons if a request made under sub-paragraphs (a) and (b) is denied, and to be able to challenge such denial; and d) to challenge data

relating to him, and, if the challenge is successful to have the data erased, rectified, completed or amended.

8. **The Accountability Principle**. A data controller should be accountable for complying with measures which give effect to the principles stated above.

In turn, the OECD principles became the basis for a number of influential legislative instruments and standards, including the European Union *Directive on Information Processing*[13] and the Canadian Standards Association ("CSA") *Model Code for the Protection of Personal Information.*[14] While not all countries have established comprehensive data protection regimes, the transition of fair information practises from a form of soft law into explicit statutory obligations is likely to continue.

The features of a data protection regime differ between jurisdictions. Some bind private sector entities, while others affect only the public sector. At their heart, these regulatory frameworks seek to safeguard privacy interests by imposing constraints on an organisation's ability to collect, use, disclose and retain personal information.[15] Organisations are typically obligated to provide policies, procedures, human resources and administrative mechanisms to meet the requirements of the fair information practises listed above. Many data protection regimes also create an *administrative officer* with the power to investigate complaints, interpret legislation, compel production of documents and make decisions on particular cases.

Despite the increasing sophistication of data protection law, recent developments are posing challenges for existing regulatory frameworks. In our next section, we turn to a discussion of some the issues raised by modern information technology.

[13]This directive is the European Union's over arching data protection law. It was passed in 1995, as the *Directive 95/46/EC of the European Parliament and of the Council on the protection of individuals with regard to the processing of personal data and on the free movement of such data*

[14]The CSA Model Code was incorporated into the Canadian *Protection of Personal Information and Electronic Documents Act* ("PIPEDA"), a federal statute that regulates a portion of the private sector.

[15]A thorough discussion of data protection regimes can be found in [52].

2.1.5 Current Challenges to Informational Privacy

As we have seen above, a recurring theme in the development of privacy protection has been the inability of existing safeguards to deal with advances in technology. The simple physical protections existing during the time of Warren and Brandeis were threatened by the advent of the hand-held camera and improving printing technology. In the mid 20th century, Alan Westin suggested that the protections developed after Warren and Brandeis were inadequate to meet the challenges afforded by computers and other forms of information technology.

In a similar fashion, recent advances in information technology threaten modern data protection regimes. These developments include:

1. **Surging repositories:** The amount of information at the disposal of private and public sector organisations has grown significantly. Rapid technological advances in storage capacity have enabled organisations to routinely manage databases that are inconceivably larger than the simple tools available in Alan Westin's 1960. In the words of one commentator, *"[u]ntil recently, data sets were small in size, typically containing fewer than ten variables. Data analysis traditionally revolved around graphs, charts and tables. But the real-time collection of data, based on thousands of variables, is practically impossible for anyone to analyze today without the aid of information systems. With such aid, however, the amount of information you can mine is astonishing."*[14]

2. **Automated decision-making:** Second, an increasing amount of processing is happening in the absence of a relational setting between the individual and the institution in question. Government offices, banks and insurance agencies make decisions on eligibility for housing and other benefits at a distance, often with the use of automated decision support systems. In the words of the United States Privacy Protection Study Commission, *"[t]he substitution of records for face-to-face contact in these relationships is what makes the situation today dramatically different from the way it was even as recently as 30 years ago. It is now commonplace for an individual to be asked to divulge information about himself for use by unseen strangers who make decisions about him that directly affect his everyday life. Furthermore, because so many of the services offered by organisations are, or have come to be considered, necessities, an individual has*

little choice to but submit to whatever demands for information about him an organisation may make."[13][16]

3. **Social networking:** A new generation of Internet applications has radically changed the way in which individuals maintain an online presence. Social networking applications such as Facebook and MySpace allow individuals to create personal profiles containing a wide variety of personal information. The privacy implications of having vast amounts of personal data stored in social networking applications are significant.[17] This sort of data would not have been available years ago, and there is evidence that employers and other organisations are actively seeking it.[18] As a result, social networking applications are being investigated by regulatory authorities in several countries.[19]

4. **Ubiquitous computing:** The growing sophistication and miniaturisation of computing hardware has led to the development of the field of *ubiquitous computing*. Ubicomp, as the field is known to practitioners, envisions the integration of small computing devices with buildings, clothing, appliances, and a host of other artifacts. Communication between these devices can facilitate new interactions that provide efficiency. However, ubicomp also has the potential for creating major privacy risks.

[16] See also [47], in which Daniel Solove states that privacy issues in databases involve a *"process of bureaucratic indifference, arbitrary errors, and dehumanization, a world where people feel powerless and vulnerable, without meaningful form of participation in the collection and use of their information."*

[17] More recently, this type of application architecture has started to become more common in sensitive domains such as health care, raising concerns about privacy and security. For more information, see [57].

[18] *Supra*, note 8

[19] For instance, the Information and Privacy Commissioner of Canada has conducted a review of Facebook's privacy practises. In that work, the Commissioner noted that fair information practises were not designed to deal with information systems in which users voluntarily contributed information: *"The purpose of the Act is to balance an organisation's need to collect, use and disclose personal information for appropriate purposes with the individual's right to privacy... In the off-line world, organisations may collect particular personal information, and use and disclose such personal information, in order to provide a specific service. On Facebook, users decide what information they provide in order to meet their own needs for social networking."* (PIPEDA Case Summary 2009-008)

5. **Knowledge discovery / data mining:** Lastly, the recent emergence of knowledge discovery in databases ("KDD") has raised a host of new problems relating to privacy. In later sections of this document, we will present a comprehensive and novel analysis of the impact of KDD on traditional approaches to informational privacy.

Without adequate means of addressing the risks entailed by these developments, the privacy protection could be significantly compromised.

2.1.6 Technical Approaches to Privacy Protection

To close this Section, we briefly survey some of the technical research on privacy safeguards that has taken place in computer science. The treatment of privacy within computer science has been quite broad, and we are only capable of presenting a small sample of work. Nevertheless, we will outline some of the key areas of research, in an attempt to position the unrealization approach [20] within the larger context.

Securing Statistical Databases

The first technical work on safeguards for personal information was undertaken in the database research community. A *statistical database* ("SDB") is a database system that allows queries to return only aggregate statistics. *Security* in an SDB means preserving the ability of users to retrieve accurate aggregate statistics, while preventing the same users from being able to infer confidential information about any individual whose data is contained in the database.[20] *Compromise* (or *disclosure*) occurs when a user infers (from one or more queries) confidential information of which she was previously unaware. In particular:[21]

- *Positive exact compromise* occurs whenever the user discovers that an individual belongs to a particular category, or holds a particular data value.

[20] See, for example [1]. For a slightly different formulation of these definitions, see [27].
[21] For detailed accounts of compromise in statistical database systems, see [25], [24] and [1].

- *Negative exact compromise* occurs whenever the user determines that the individual does not belong to a particular category, or does not hold a particular data value.
- *Positive compromise* occurs whenever the user discovers information that gives them a more accurate estimate as to whether an individual belongs to a particular category, or holds a particular data value.
- *Negative compromise* occurs whenever the user discovers information that gives them a more accurate estimate as to whether the individual does not belong to a particular category, or does not hold a particular data value.

Simple approaches to protecting individual information in statistical databases were not successful.[22] Adam and Wortmann [1] group existing approaches under four headings:

1. *Conceptual:* these approaches concentrate on the security problem at the level of the data model.
2. *Query restriction:* these approaches attempt to provide security by controlling queries. Examples of control mechanisms include: a) restriction on query set size; b) controlling overlap between successive queries through audit trails; c) partitioning the database, and; d) making 'cells' of small size unavailable.
3. *Data perturbation:* these approaches introduce noise into the data, resulting in a database that has been modified. Queries proceed as normal on the modified data.
4. *Output perturbation:* these approaches perturb the results of queries, introducing noise into the results. As opposed to data perturbation approaches, the underlying data itself is not modified.

These approaches have also been highly seminal, spawning similar techniques in other research areas.

[22]For an example of a tool to defeat simple protection schemes, see the 'tracker' outlined in [17].

Privacy Policy Languages

Another area of research concerns providing tools for facilitating the exchange of information between disparate information systems, through the development of *formal languages* that represent privacy policies/preferences. When information is transferred from one information system to another, it is suddenly subjected to a new range of organisational policies concerning security and privacy. If a well-defined language was available to annotate personal information with policy directives, the receiving information system could respect the privacy and security commitments in place at the disclosing organisation.[23]

Privacy Access Control

Having a language for specifying privacy preferences is undoubtedly useful for transferring data between information systems; however, organisations have to enforce these preferences within their own boundaries. *Privacy-aware access control* mechanisms attempt to address this issue, by formalizing the obligations incumbent on an organisation managing personal data [23, at p.17]. Examples of active research projects include E-P3P [28], EPAL [6], and XACML.

Privacy Requirements Engineering

Privacy requirements engineering involves the integration of privacy concerns into the software development life cycle. The main issue is to provide tools and methodologies for modelling the *"organisational context of a system along with the goals of environmental and system actors and the social relationships among them."* [23, at p.18] By capturing privacy requirements in the early stages of development, systems designers can avoid expensive rework, and reduce privacy risks.[24]

[23] Current efforts in the area of privacy policy languages include the P3P Preference Language ("APPEL") [16] and XPref [3].

[24] This coheres with the well-known *privacy-by-design* approach advocated by the Ontario Information and Privacy Commissioner, available at http://www.privacybydesign.ca.

Privacy in Social Networks

Recently, various research groups have opted to study the privacy and security issues that arise in social networking applications such as Facebook. Researchers have addressed topics ranging from providing real-time anonymity for users [30] to new access control models that take advantage of features of the social networking domain [9]. A recent survey of the field can be found in [57].

Privacy and Ubiquitous Computing

Researchers in the field of ubiquitous computing are acutely aware of the privacy implications of embedding computational devices in everyday objects and living environments.[25] Work on privacy protection in ubiquitous environments includes prototypes of privacy-aware architectures [26] and location anonymization [8] [29].

Privacy Preserving Data Publishing

Organisations routinely exchange data sets containing sensitive personal information.[26] According to Chen et al. [11], the approaches used in practise primarily rely on: a) policies and guidelines to restrict the types of publishable data, and; b) agreements on the use and storage of sensitive data. The problem with this approach, according to the same authors, is that it *"either distorts data excessively or requires a trust level that is impractically high in many data-sharing scenarios."* Contracts and agreements by themselves cannot guarantee that sensitive data will not be misplaced, disseminated or used for secondary purposes.

Privacy preserving data publishing ("PPDP") is concerned with the development of algorithms and software tools for use in the context of *data publication* - namely, exporting data from a *data publisher* to a *data recipient*, such that: a) the data remains useful, and; b) individual privacy is preserved. The recipient is always regarded as an adversary, while the publisher may be either trustworthy or non-trustworthy.

[25] See, for example, [7] and [31].

[26] As an example, health authorities routinely submit information to government public health agencies, for purposes of statistical analysis.

According to Chen et al [11], one of the differences between work in PPDP and work in statistical database security concerns the larger set of threats considered by the PPDP community, including *"background attacks, inference of sensitive attributes, generalization, and various notions of data utility measures."* Many PPDP algorithms proceed by way of *anonymization* or *pseudonymization*. Influential approaches for tabular data sets include k-anonymity [51] and l-diversity [32]. A survey of data publishing methods for graph data can be found in [58].

Privacy Preserving Data Mining

Privacy preserving data mining ("PPDM") involves modifying data mining approaches to account for privacy concerns. According to [11], PPDM researchers must carefully craft data modification methods that preserve individual privacy, while maintaining the utility of the data sets at an aggregate level. Unless a privacy-preserving approach can support useful data mining results, it is unlikely to be adopted in practise. We will discuss PPDM more thoroughly in Section 2.3, once we have introduced the basic concepts of data mining.

2.1.7 Quantifying Privacy Protection

One of the key tasks in designing technical safeguards for privacy risks involves creating metrics to measure privacy loss. One of the most influential measures in existence is *differential privacy*, which was an outgrowth on privacy protection work in the statistical databases community.[27] In order to understand the following definition, assume that we have a data custodian (or 'curator') who releases information to a data recipient. The database holds sensitive information pertaining to individuals, and the recipient performs a processing task on any data that she receives. We model the processing task as a randomized algorithm $\mathcal{A}$.

Definition 1. *We say that algorithm $\mathcal{A}$ gives ϵ-**differential privacy** if for all datasets D_1, D_2 differing on a single element (and for all $S \subseteq Range(\mathcal{A})$):*

$$P(\mathcal{A}(D_1) \in S) \leq exp(\epsilon)P(\mathcal{A}(D_2) \in S)$$

The value ϵ is, of course, a parameter. Typical examples are 0.01 or $ln2$. As stated by Dwork, an algorithm satisfying this definition addresses concerns that an individual might have about the leakage of her personal information. For an appropriate value of ϵ, even if her information is in the database D_1, removing her record from D_1 (resulting in database D_2) will not significantly affect the output of the algorithm [19].

Some observations are in order:

1. Differential is what Dwork calls an *ad omnia* guarantee, in contrast to an *ad hoc* definition that provides protection only against a specific set of threats/attacks. She notes that it is also quite rigid, as the claim is independent of the computational power and auxiliary information available to an attacker.

2. Achieving differential privacy is typically performed by adding *noise* to the data. Algorithms ($\mathcal{A}$) vary in their sensitivity to noise.

3. The probability space of interest is over coin-flips of the mechanism, and not over sampling of the data. As a result, privacy comes from the *process*.

[27] Our main reference for this section is Cynthia Dwork's unpublished paper, available online at http://research.microsoft.com/en-us/projects/databaseprivacy/dwork.pdf.

4. Also according to Dwork, differential privacy may be achieved not only by reducing the probability of a true positive, but also by increasing the probability of a false positive. That is, by providing erroneous data for people who are not in the data set, we can provide 'cover' for the individuals whose data was released.

5. The differential privacy concept embodies a *composability property*, where parameters on consecutive queries can be accumulated, in order to provide a differential privacy bound over the aggregate of the queries.

There have been numerous applications and refinements of the differential privacy concept. For our purposes, it is sufficient to use differential privacy as an example of a concept that attempts to provide bounds on the probability of a privacy loss. Without an accurate way of estimating such losses, it is difficult to provide formal arguments about the sufficiency of privacy-preserving algorithms.

2.1.8 Section Summary

To summarize, we partitioned our discussion of privacy as follows:

1. **Fundamental Concepts:** We introduced the basic concepts of privacy, beginning with the classic formulation of privacy as the 'right to be let alone'. We distinguished between territorial, bodily, informational and communications privacy interests, with the observation that *informational privacy* will form the basis for the approach in this thesis.

2. **The Normative Basis of Privacy:** We briefly discussed the importance of privacy, including utilitarian and deontological approaches. Examples of utilitarian approaches included the importance of privacy interests to personal development, integrity and identity. We stated that our particular emphasis in this thesis is on the individual's relationship with institutions, including the administrative arm of the state.

3. **Informational Privacy:** Our treatment of informational privacy centred around a major theme: that of technological advancements outstripping traditional mechanisms for fostering privacy. In addition, we discussed the increasing reach of modern liberal governments, and their tendency towards data collection. We closed with a discussion of risks posed by increased data collection on the part of governments and large organisations.

4. **Data Protection Regimes:** Following our discussion of informational privacy, we reviewed the traditional approach to privacy protection. As a concrete example, we presented the OECD Guidelines, which will appear again in later portions of this document.

5. **New Challenges to Informational Privacy:** This sub-section introduced several innovations that are causing problems for data protection regimes, namely: a) increased storage capacity; b) automated decision support; c) social networking; d) ubiquitous computing, and; e) knowledge discovery / data mining.

6. **An Overview of Technical Approaches to Privacy:** We introduced some of the research areas in computer science that directly address the privacy risks raised by new technology. We introduced statistical database security, privacy in ubiquitous environments and social networks, privacy requirements engineering, and privacy preserving data mining/publishing.

7. **Measuring Privacy Protection:** To cap the section off, we discussed the *differential privacy* metric for privacy protection. We mentioned that technical metrics of this sort are essential for dealing with privacy issues in a formalised manner.

One of our main claims in this work is that traditional approaches to informational privacy protection make assumptions that are vitiated by data mining and knowledge discovery techniques. To that end, we now turn to a discussion of data mining, including the particular challenges that it poses to data protection regimes.

2.2 Data Mining

In this section, we quickly recap the basic concepts of data mining. Although readers with a technical background are undoubtedly well-acquainted with data mining techniques, a brief introduction would be useful for legal researchers, political scientists and other non-technical academics. Our discussion is partitioned into the following sections:

1. **Basic Concepts and Applications:** This section introduces data mining, with an emphasis on the basic steps involved in the data mining process. Different types of data mining algorithms are discussed, including classification and prediction.

2. **Decision Trees:** The next section introduces decision trees as an example of a data mining approach. We give a very brief introduction to decision trees, deferring the technical details until they are required in later sections.

3. **Data Mining and its Impact on Privacy:** The last section discusses the problems that data mining poses for privacy. In particular, we concentrate on the challenges that arise for data protection regimes. Using the OECD principles as a motivating example, we demonstrate that data mining has vitiated some of the safeguards that form the basis of modern privacy law.

By the end of this section, we will have covered the basics of privacy and data mining. The last section in the chapter discusses the new discipline of *privacy preserving data mining*. The work performed in [20] and this thesis are a contribution to this relatively young research area.

2.2.1 Basic Concepts

Knowledge Discovery and Data Mining

The first issue to discuss is that of terminology. The term 'data mining' has a variety of definitions in the research literature, often appearing beside the concept of 'knowledge discovery'. Establishing a definition of both of these terms is therefore of some importance. In this thesis, we follow the example set by Maimon and Rokach [33], who define *knowledge discovery in databases* ("KDD") as the automatic exploration, analysis and modelling of large databases. According to these authors, KDD is the process of identifying valid, novel, useful and understandable patterns from large data sets. The same authors define *data mining* ("DM") as the *"core of the KDD process"*, involving:

1. the construction/inference of *algorithms* that explore the data;
2. the development of a *model*, and;
3. the discovery of previously unknown *patterns*.[28]

A model created by data mining procedures can be used for a number of purposes, including:

1. Characterisation of trends;
2. Association analysis;
3. Classification and prediction;
4. Cluster analysis, and;
5. Outlier analysis.

It is important to distinguish KDD and DM activities from data warehousing and traditional statistical analysis:

[28] See [33]. The combined term *knowledge discovery and data mining* ("KDDM") is also common in the literature. According to Sumathi, KDDM is an *"umbrella term describing several activities and techniques for extracting information from data and suggesting patterns in very large databases."* [50, at p.275]

- *Data warehousing* is an activity consisting of the extraction and transformation of data from operational databases[29] into specialised repositories that serve to facilitate decision support. Data warehousing efforts aim at amalgamating data from disparate sources into a central data store (the 'warehouse') that can be used for strategic business functions. Data warehouses are often used as a source of information for knowledge discovery activities.[30]

- *Statistical analysis* is concerned with the analysis of data. As stated by Quinlan [41, at p.15], in some cases there is no difference between methods invoked in statistics and those used in knowledge discovery and data mining. However, on a general level, statistical techniques tend to involve tasks in which all the attributes have continuous or ordinal values. Many traditional statistical methods also assume that the data fits a particular model; analyses of this sort generally proceed by searching for parameters that will make the model fit the data. In contrast, KDD techniques place an emphasis on discovering novel and unanticipated models that explain patterns in the data.

The Rationale for Knowledge Discovery

Having briefly introduced the basics of knowledge discovery, we turn our attention to a brief discussion of the major drivers for this relatively young research discipline:

1. **Growth of data:** Our first rationale for the use of KDD techniques concerns the rapid growth of commercial data collection efforts. As noted in Sub-section 2.1.5 above, the accumulation of data has become much easier of late. In fact, the amount of stored information is said to double every 20 months [33].

   In contrast, the ability of humans to understand and make use of this data is not growing exponentially. This widening gap calls for the development of technology that can sift through large databases for interesting patterns and relationships.

[29] Operational databases are those that support an organisation's operational (routine) activities. For instance, a bank will have several databases devoted to processing transactions.

[30] For a discussion of data warehousing, see [49] at p.461.

2. **Novel discoveries:** A second rationale for KDD techniques concerns the utility of the inferred models themselves. The ability to create models of data is incredibly useful, even in the context of small data-sets that humans can comprehend and manipulate. The results of KDD procedures may be novel, surprising, and unpredictable to human analysts. In short, automated analysis can turn data into higher forms of knowledge that can be more compact, more abstract, or more useful [54, at p.164].

3. **Resource constraints:** A third rationale for KDD concerns *resource scarcity*. While decision support systems have proven themselves to be useful in a wide variety of contexts, they are not particularly easy to construct. Many organizations that could use decision support face *resource shortages*; it is often difficult for subject matter experts to have time to sit down with knowledge engineers, let alone participate in a thorough requirements analysis effort. [31]

   As a result of these logistical and practical difficulties, automated knowledge representation approaches have a great deal of appeal for many organizations. Not only can the organization receive a model that may help their bottom line, but the demands on subject matter experts are greatly reduced.

With this brief discussion in hand, we turn to a brief discussion of the knowledge discovery process. Not only will an understanding of the KDD process aid the reader in comprehending the work that has been performed for this thesis, but a clear picture of the various stages is crucial to understanding the impact of KDD and data mining on privacy.[32]

[31] The inability of organizations to adequately resource decision support efforts is known as the *knowledge elicitation bottleneck*.

[32] Many of the existing works on data mining and privacy (e.g., [12], [50], [10], [38], [37]) have done a poor job of identifying the real issues that KDD poses to informational privacy interests. Most of these efforts examined the impact of data protection regimes on data mining efforts, instead of the more interesting question as to the impact of data mining on privacy interests.

The Knowledge Discovery Process

According to Maimon and Rokach [33], the knowledge discovery process is both iterative and interactive, involving the following steps:

1. **Understanding the Application Domain:** This step involves understanding the goals of the effort, as well as the environment in which the KDD effort will take place.

2. **Selecting a Data Set:** The next step involves selecting the data set to be mined. In keeping with the 'garbage in, garbage out' principle, the selection of the data set is of paramount importance.

3. **Preprocessing and Cleansing:** In this step, the data is processed to enhance its reliability. Activities undertaken at this stage may include: a) handling missing values; b) removing noise, and; c) dealing with outliers.

4. **Data Transformation:** In the next step, the data is treated to make it more amenable for the data mining algorithm. Processing steps may include: a) dimension reduction; b) record reduction, and; c) attribute transformation.

5. **Decide upon the Task:** This step involves determining the type of task, such as classification, regression or clustering.

6. **Pick the Algorithm:** Once the task is decided, the next step involves the selection of a specific method. Each algorithm has parameters, methods of training, and particular types of data sets for which it is more accurate.

7. **Employ the Algorithm:** In this step, the algorithm is run on the data. Typically this is an iterative process, since the algorithm's control parameters may require tuning.

8. **Evaluate Results:** The next step is to interpret the model with respect to the goals identified above.

9. **Use the Model:** The last step is to use the knowledge, perhaps for prediction or classification of previously unseen data sets.

We will make use of this idealized process in the chapters to follow. Our next task is to present a high-level view of the varieties of knowledge discovery approaches that appear both in the literature and in practice.

Types of Knowledge Discovery Activities

In one of their recent papers [33], Maimon and Rokach present a taxonomy of knowledge discovery methods. A simplified version of their diagram appears below:

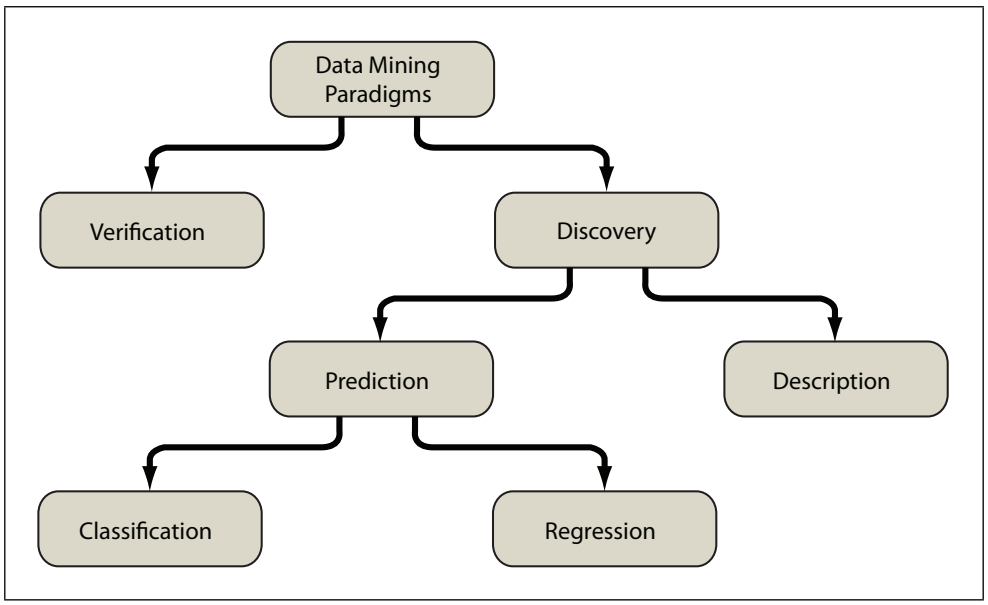

Figure 2.1: Knowledge discovery hierarchy.

At the highest level of abstraction, knowledge discovery methodologies can be partitioned into *verification* and *discovery* methods:

1. **Verification:** These methodologies involve the evaluation of a hypothesis proposed by an external source. Traditional statistical tests such as goodness of fit and analysis of variance fit into this class.

2. **Discovery:** These methodologies automatically identify patterns in the, without the need for external provision of hypotheses.

In turn, Maimon and Rokach partition discovery methods into two sub-categories:

1. **Description:** These methods involve data interpretation, which focuses on understanding the way the data set relates to its own parts. Examples of *description methods* include clustering, summarization, linguistic summary and visualization.

2. **Prediction:** These methods build a model that is able to make predictions about the values of attributes for new (unseen) samples.

Our focus in this work is not on descriptive methods, but on predictive ones. *Predictive data mining* is also known as *supervised learning*, in which a model is built on the basis of a target attribute whose value is known. In contrast, *unsupervised learning* concerns techniques that group objects (as represented in a data set) without a pre-specified target attribute whose value is known. At the risk of repetition, there are two major categories of predictive methods:

1. **Classification:** These methods map the data set into predefined classes. For instance, a classification method might be used to figure out the risk category for a mortgage applicant.

2. **Regression:** These methods map the data set into a real-valued domain. For instance, a regression method might attempt to predict the amount of time needed to heat a chemical in an industrial process.

In the next section, we briefly examine data mining methods from the perspective of machine learning. This treatment will give us a chance to discuss training methods and a few other details that are important for understanding the work that we are presenting in this thesis.

Knowledge Discovery as Learning

A profitable means of interpreting KDD methodologies involves adopting the perspective of a *learning agent*. Many KDD problems can be regarded as a form of instruction in which a human operator asks a machine to learn one or more concepts from a data set. The *classification problem*, for example, is a classic instance of this type of *concept learning*. In a classification problem, the learner must search for a

function that maps the set of all possible input values into a set of class labels.[33] At a high level, there are three types of learning approaches for this problem:[34]

1. **Supervised Learning:** In supervised learning, the learner has access to both the training set, as well as the correct (or approximately correct) value of the function for particular elements of the training set.

2. **Reinforcement Learning:** In reinforcement learning, the learner receives some evaluation of its actions, but is not told what the correct value of the function is for a particular element of the training set. For instance, the learner might receive a reward or punishment, instead of an assessment of accuracy.

3. **Unsupervised Learning:** In unsupervised learning, the learner receives no information about the correct value of the function for any of the training set elements. An unsupervised learner can infer relationships among the inputs, but not with the aid of any information about correct values.

The choice of *representation* for the desired function is one of the most important issues facing the designer of a classifier. In learning, as in reasoning, there is a trade-off between *expressiveness* (the ability of the formalism to represent the learning function) and *efficiency*.[35] A representation that is highly expressive is unlikely to be very efficient, and vice versa.

Thankfully, we have opted to concentrate on a formalism that is not only highly efficient, but also possessed of enough representation power for a wide variety of applications –namely, the *decision tree*. We introduce the decision tree in more detail in a later sub-section. Our next task is to briefly recount a few more details about supervised learning.

[33] Given a training set S with input attributes $A = \{a_1, a_2, \ldots, a_n\}$ and a nominal target attribute y from an unknown fixed distribution D over the labelled instance space, the goal is to induce an optimal classifier with minimum generalization error [34, at p.151].

[34] See [44] at p.529.

[35] First order logic, for instance, is a highly expressive language that (unfortunately) suffers from major issues in terms of tractability.

Supervised Learning

Supervised learning is a form of *inductive learning*, which involves constructing a description of a function (hypothesis) from a set of input/output examples.[36] The underlying assumption of inductive learning is that generalization from past examples is an adequate basis for predicting the value of unseen (future) instances.[37] The *basis* for the induction is a set of *objects*, which are described in terms of a collection of *attributes*.[38] Each attribute measures some important feature of an object.[39] The key feature of interest (the *target* attribute) is identified in advance of the induction. In a supervised learning setting, the *training set* for the induction is a collection of objects whose attribute values are known to the learning algorithm.

As mentioned above, our interest in this work lies in classification. The *induction task for classification* involves constructing a function that categorizes objects into a set of mutually exclusive, pre-defined classes. The end goal is to develop a *classification rule* that can determine the class of any object (including objects that were not in the training set) from the values of its (non-target) attributes. There are numerous algorithms for forming classification rules, including:

1. Decision trees.
2. Neural networks.
3. Genetic algorithms.
4. Instance classifiers.
5. Support vector machines.
6. Bayesian networks.

These approaches each have various strengths and weaknesses, with respect to efficiency, comprehensibility, robustness and other criteria.

[36] See [44, at p.525].

[37] For more information on induction in KDD, see [33] and [40]. The original source for the problem of induction is David Hume's Treatise of Human Nature (Book I, Part III, section VI), first published in 1739.

[38] For more on this topic, see Quinlan [40].

[39] Attributes are typically limited to taking a set of discrete, mutually exclusive values.

The main issue in forming a classification rule from a training set is whether or not the attributes of the objects in the training set provide sufficient information.[40] For instance, the following difficulties may exist in the training data:

- **Missing Values:** The training set may be missing values for key attributes, perhaps as a result of systematic errors in data collection.
- **Noise:** The training set may exhibit a degree of randomness. Noise will impede the ability of the classifier to identify patterns and trends.
- **Duplicate Objects with Differing Classes:** The training set may contain examples which match on every attribute, save the target attribute. In this case, the training set contains objects with inconsistent class labels. This situation will confuse classifiers.

Classifiers may suffer from two drawbacks, with respect to the information contained in the training set:

- **Under-fitting:** The classification rule may not have enough information to perform well on the training set, even though the value of the target attribute is known for all objects in that collection.
- **Over-fitting:** The classification rule may perform too well on the training set, and not well enough on unseen examples. In such a case, the rule may have memorised the training set, at the cost of committing itself to patterns that are not found in general.

Lastly, any preference for one hypothesis over another (beyond mere consistency with the examples) is called a *bias*. Since there are usually a large number of consistent hypotheses for any learning task, all learning algorithms evidence some sort of bias.

[40] Material in this section is drawn from [40].

2.2.2 A Motivating Example: Decision Trees

A *decision tree* is a predictive model that can be used to represent both classifiers and regression models. As mentioned above, our choice of decision trees as a data mining algorithm was not made casually. Not only are decision trees intuitive, simple and transparent [43, at p.7], but they are computationally efficient. As a result, decision trees are one of the most widely used data mining approaches.

As noted by Rokach and Maimon [43, at p.5], when a decision tree is used for classification tasks, it is more appropriately referred to as a *classification tree*.[41] A classification tree is used to categorise an *object* (also called an *instance*) with respect to a predefined set of classes. The classification is made on the basis of the object's attributes, which are usually accessible to the classification algorithm.

Classification trees are frequently used in applied fields such as finance, marketing, engineering and medicine [43, at p.6]. As an example, classification trees might be used to classify applicants for a mortgage. Each applicant for a mortgage fills out an application that includes relevant details on their financial and credit history, including yearly income, assets, liabilities, and credit rating. A simple classification tree for a mortgage application is shown below:

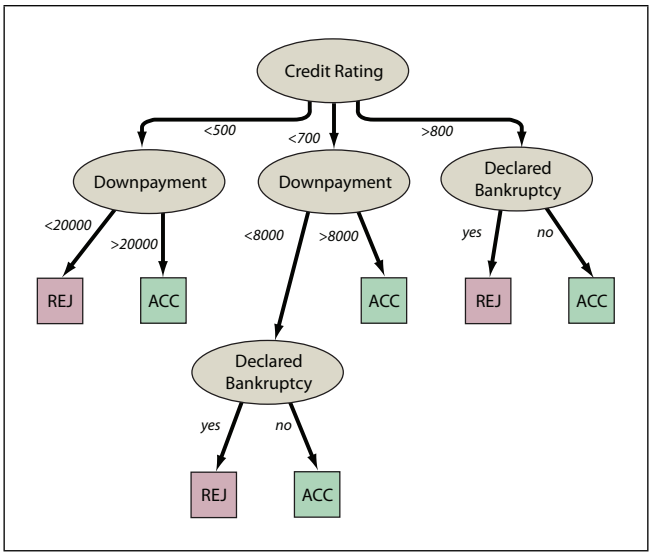

Figure 2.2: A sample decision tree.

[41] As one might predict, when a decision tree is used for regression tasks, it is most appropriately called a *regression tree*.

Administrative staff can use the decision tree to classify mortgage applicants. In the following diagram, Fred Wilson's application is classified using the tree presented above. In our idealized example, there are only three attributes relevant to the classification procedure: credit rating, downpayment, and bankruptcy history.

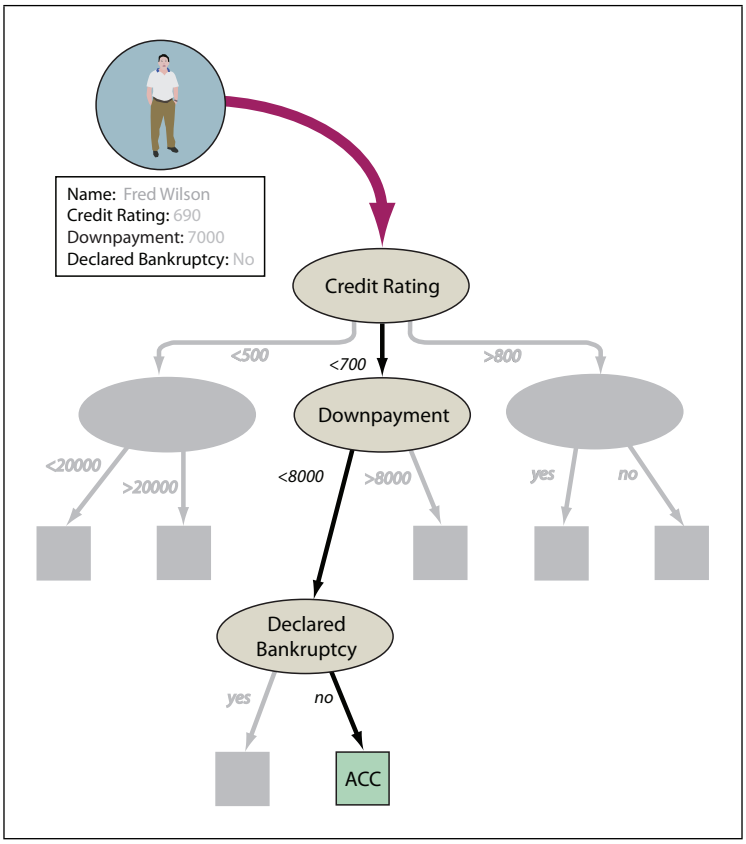

Figure 2.3: Classifying an applicant.

As one can see from the diagram, the application was successful. Fred Wilson turned out to have a credit rating that was moderately good. Although he did not have a large downpayment, the fact that he had never declared bankruptcy was enough to earn him the approval of the lender. In contrast, if Fred had declared bankruptcy previously, his application would have been rejected.

We will engage in a more detailed discussion of decision trees in a subsequent section of this thesis. Having introduced knowledge discovery at a high level, we are ready to examine the impact of these techniques on informational privacy regimes.

2.2.3 Data Mining and its Impact on Privacy

As we discussed in subsection 2.1.5, a major theme in the development of privacy consists of technological advances in information technology spurring subsequent developments in law. Modern data processing techniques are no exception to this trend, as numerous commentators have raised concerns about the use of automated tools to process stores of personal information. For instance, the Commission on Freedom of Information and Individual Privacy stated that *"[s]everal privacy costs can be identified. First, an invasion of privacy results from loss of control over the dissemination of PI concerning oneself. Second, the growth of documentation may create a pressure to perform 'for the record', with a subsequent loss in personal autonomy. Third, the persistence of [a] record makes it more difficult for an individual to 'make a fresh start', to resolve to conduct himself in new and more responsible ways. Fourth, the growth of decision making 'by the record' increases the danger that the determinations will be made on the basis of erroneous information, with consequent unfairness to the individual."* [39, at p.503]

It is, however, important to distinguish the risks arising from data mining from those arising from the use of other technologies, such as decision support or statistical analysis. According to Tailpale, privacy concerns around databases are of two kinds:

1. those that arise from the aggregation/integration of data itself, and;
2. those that arise from the automated analysis of data that may not be based on any individualized suspicion [54, at p.210].

Taipale calls the former issue the *database* problem, and the latter issue the *mining* problem. Clearly, integration and aggregation of data can be done without recourse to knowledge discovery techniques.[42] One of our tasks in this work is to pin down the precise threat to privacy arising from data mining, at a greater level of precision than other efforts in the research literature. To this end, we have identified the following challenges posed by KDD techniques to informational privacy regimes:

[42]To give a simple example, individuals can be manually 'matched' between different databases through the use of strong identifiers such as social insurance or health card numbers.

Challenge 1: Secondary Uses

According to Olivera and Zaiane [38], the major factor contributing to privacy violation in data mining is misuse of data. As we noted in the discussion above, data protection regimes based on the OECD guidelines explicitly prohibit uses of personal information that have not been specified at the time of collection.[43] As a result, many data mining efforts will lead to breaches of the fair information principles, and therefore of information privacy interests.

We are not, however, interested in risks that arise merely on account of the fact that KDD involves processing personal information. Rather, we are interested in privacy risks that arise due to the unique nature of KDD activities themselves. Thankfully, a connection between data misuse, KDD and privacy has already been identified. As noted by the Privacy Commissioner of Ontario [10], a data mining program cannot delineate what the primary purpose of the mining will be in advance. The role of data mining is to sift through all the information available to unearth the unknown. As stated by the Commissioner, *"[t]he discovery model upon which it builds has no hypothesis - this is precisely what distinguishes it from traditional forms of analysis."* [10, at p.12] Identifying a primary purpose at the beginning of the process, then restricting one's use of the data to that purpose are the antithesis of a data mining exercise.

Challenge 2: The Legal Status of Intermediate Work Products

Our next issue arises on account of the unique nature of the KDD process, which involves a succession of data transformation activities (including cleaning, coding and reduction) that have no correlate in mere database amalgamation, aggregation or querying. As noted by Taipale [54, at p.190], it is not clear as to whether *derived data* (generated from the query or analysis process) or *meta-data* becomes part of the record. These intermediate work products are derived from the original data sets, but they have been transformed in various ways. At some point, transformations will alter them so significantly as to escape copyright and privacy law.

[43] In particular, the OECD "Use Limitation" principle holds that personal data should not be used for purposes other than those specified, except with consent of the individual or the blessing of law. The OECD "Purpose Specification" principle holds that subsequent uses must be limited to the fulfilment of those purposes which were specified upon collection.

This issue is not merely academic. For instance, if a health authority discloses information on patients to a research institution that has an active data mining program, the research institution will come into possession of a set of intermediate work products that may bear little resemblance to the original data sets. The processes of transformation, cleaning and coding may yield information that does not appear to be the same as that which was originally disclosed. If this information is not covered by copyright, private contract or privacy law, the research institute may do as it pleases with the data (including selling it to pharmaceutical companies).

Challenge 3: The Legal Status of Models

This issue is strongly related to the legal status of intermediate work products. In the data mining process, not only does the miner perform a series of steps to clean, filter, recode and transform the original data, but she creates a *model* that can be used for prediction. The model is a separate work product that can be distributed, reused, and transformed into other representations as required.[44]

The key difference between statistical analysis and data mining is that many of the models in the data mining domain are very powerful; in fact, some can even encode the training set itself, particularly if the learning process involved over-training [43]. This means that the model itself may contain quite accurate information about the data, including sensitive information.

As in the case of intermediate work products, it is not clear that the model is caught by existing legal regimes. Since the model is not straightforwardly personal information, it may escape the reach of privacy law. Since the model may not be a representation or reproduction of the data set, it may be the case that it also avoids oversight from copyright law. Given that a model can effectively encode a large portion of the training set, it can serve as a slightly inaccurate substitute for the original data set. If a company with access to data could merely form over-trained models on various training sets drawn from the data, it would be a loophole by which the company could use, disseminate and modify personal information, without oversight from regulatory authorities or the courts.

[44]For instance, a decision tree may be converted into a set of rules.

Challenge 4: Defeating Anonymization

Existing data protection regimes deal exclusively with personal information, as opposed to information about aggregates or corporate entities. Personal information in modern data protection law is typically defined as information relating to an *identifiable individual*.[45] As stated by Oliveira and Zaiane, [38] information is considered *personally identifiable* if it can be linked, directly or indirectly, to an individual person.

Repositories of personal information are often *de-identified* by organisations that wish to share data. For instance, a common practise in the health care sector is to 'anonymize' data by removing key identifying information, such as names and health card numbers. Since privacy law only attaches to information relating to identifiable individuals, once a data set has been de-identified, it can be shared freely.

As was amply demonstrated by computer scientists, de-identification is not as simple as legislators assumed [36]. With appropriate background knowledge, it is possible to reconstruct personal information. Even worse, new data mining algorithms permit partial reconstruction of data even in the absence of data matching between previously separated data sets. As an empirical matter, data sets are routinely anonymized in a naive fashion, and released to third parties for processing. With appropriate data mining tools, these latter organisations can obtain far more information from the data set than the disclosing organisation would have anticipated.

Challenge 5: Probabilistic Inferences

As mentioned above, one of the major issues in privacy protection is the ability of an adversary to use background knowledge to make inferences. As stated by Clifton [12], the problem lies in ensuring that we cannot infer 'private' data from 'public' data. KDD raises the risk factor in such situations, since it allows for the automation of the inference progress. In addition, KDD also poses a new problem. The inferences that we find are not specific. The discovery that 'A implies B with confidence 25%' reveals information, but it is only probabilistic. The difficulty arises when one considers that our data protection schemes are not designed with probabilistic inferences in mind.

[45] See, for example Section 2 of PIPEDA.

Challenge 6: Comprehensibility

One of the issues with data mining is that some algorithms represent models in forms that are extremely difficult for a layperson (or even an expert) to interpret. As noted by O'Leary [37], the *Openness* principle of the OECD guidelines may require individuals to understand the uses to which their data is put. Data mining models (unlike standard database systems) may be difficult to explain and justify.

As a related difficulty, the *Individual Participation* principle stipulates that individuals have a right to have data related to them communicated to them in a form that is readily intelligible. In the case of intermediate work products and induced models, this latter condition is unlikely to be satisfied.

Challenge 7: Existence, Access and Correction

One of the most difficult challenges posed by data mining to data protection regimes based on the OECD guidelines involves the *Individual Participation* principle, which dictates that a data controller must disclose whether or not it has control of data relating to an individual. If the data controller holds intermediate work products (or a model that encodes the training set, as discussed above) this is a particularly difficult question to answer. Figuring out if an individual appears in a transformed data set or induced model is much more difficult.

Second, the right of an individual to correct or amend data is made more complicated by the presence of KDD activities. The individual's data may exist in a series of intermediate work products that may not be amenable to correction or updates.

Challenge 8: Records in the Training Set

As noted in [50], a particularly interesting situation arises when someone has: 1) a classifier; 2) a record (lacing the value of the target attribute), and; 3) knowledge that the record was in the training set. Classifiers (particularly ones that have been over-trained) are very accurate on their training set. The knowledge that a record was in the training set provides strong inductive support for the proposition that the classifier is correct with respect to the classification of the record.

Analysis of the Challenges

As we demonstrated in the preceding pages, KDD poses some unique problems for data protection law. In particular, KDD undermines some of the key *assumptions* of the fair information practises, including:

- That there is a clear distinction between identifiable and non-identifiable information. KDD techniques can perform novel and surprising matches between data sets, identifying individuals on the basis of patterns that human analysts would not predict.
- That there are simple and indefeasible methods for securely de-identifying information. KDD techniques can defeat many of the anonymization routines that currently exist [36].
- That regulators have a good idea of the sorts of artefacts that contain personal information. We have seen that KDD activities involve a host of intermediate work products (and resulting models) that appear to be outside the scope of legal regimes; nevertheless, these work products can contain valuable personal information. KDD may provide loopholes by which organisations can exchange information that would be otherwise protected by privacy law.
- That an organisation will be able to tell if it holds personal information. The intermediate work products and models producing in a KDD effort are often incomprehensible to humans, but can nevertheless be saturated with valuable personal information.
- That organisations are capable of specifying their uses of personal information in advance of collection. KDD activities are based on a model of discovery that does not identify hypotheses in advance. An organisation with an active KDD program may have no clue as to the patterns and results that will arise.

The next section of this thesis contains a discussion of some of the key technical responses to the challenges outlined above. In the remainder of this section, we quickly outline some of the legal and administrative recommendations made by academics and regulators.

Among the solutions urged by academics and regulators are:

- **Distributed database architectures:** According to Taipale [53], such an architecture protects privacy by diversifying control and eliminating a single point of attack. A distributed system *"permits local institutional control over individual databases and, to some extent, local accountability. Local access control and individual privacy rules can be negotiated, enforced, and tracked at many points in the system. There is no single point of control to be exploited either by attack or by misuse."*[46]

- **Privacy by design:** Taipale also argued that privacy interests were best protected by employing "*value sensitive technology development strategies*" in combination with policy implementation. These strategies take privacy concerns into account during design and development. The Information and Privacy Commissioner of Ontario has also taken this approach, with her *privacy-by-design* advocacy.

- **Diversified authorisation and oversight:** According to Taipale, diversification can make misuse and abuse "*difficult to achieve and easy to uncover*".

Taipale and other commentators have also made specific recommendations on technological responses, including rule-based processing technologies, data labelling, digital rights management, and selective revelation. In the next section, we discuss the work carried out in the computer science community related to countering the negative implications of KDD on informational privacy interests.

[46]Taipale was citing the Public Policy Committee of the ACM on this issue.

2.2.4 Section Summary

Although perhaps familiar to most readers, the background information related in this section is needed to understand the work that was performed for this thesis. As a bonus, we also presented the most accurate and current account of data mining's impact on informational privacy regimes. In particular, we covered:

1. **Basic Concepts and Applications:** We contrasted KDD and data mining with data warehousing and statistical analysis. We discussed the rationale for KDD activities, and presented an overview of the 9-step KDD process. Following Maimon and Rokach, we introduced a taxonomy of data mining approaches, and discussed the case of supervised learning in depth.

2. **Decision Trees:** As an example of a classification approach, we introduced decision trees. We presented a motivating example, deferring a rigorous presentation on decision trees until later in this work.

3. **Data Mining and its Impact on Privacy:** Lastly, we discussed data mining's impact on privacy protection. We saw that data mining has some unique features that cause problems for current data protection law. We briefly outlined some non-technical responses to these issues, as presented by academics and regulators.

The last section of this chapter introduces the technical tools that computer scientists have brought to bear on the issue of privacy and data mining.

2.3 Privacy Preserving Data Mining

As we saw in the last section, data mining techniques pose some interesting problems for data protection regimes. Some of the solutions suggested by academics and regulators have been administrative and operational in nature. In this section, our emphasis is on the tools developed by the computer science research community. Our discussion is partitioned into the following sections:

- **Basic Concepts of PPDM**: We discuss the basics of privacy preserving data mining, including the initial research agenda that was outlined in a seminal paper by Clifton and Marks [12]. We discuss the trade-off between *information loss* and *privacy protection*, as well as the *inference problem* that plagues data protection schemes.
- **Overview of Approaches**: We discuss the overarching concepts that have been used to classify the multitude of approaches, including data modification schemes and selective modification techniques.
- **Past work on Decision Trees**: We briefly introduce a few examples of past work on decision trees. Our goal is not to provide a comprehensive overview, but to show that decision trees are an active area of research in privacy preserving data mining.
- **Our Contribution**: Lastly, we discuss the contribution made in this paper towards the advancement of decision tree algorithms for privacy preserving data mining. Our efforts are conservative, in that it seeks to solidify the earlier work presented in [20]. In the course of this thesis, however, we manage to show how the unrealization approach can be implemented on an industry-standard decision tree algorithm, as opposed to the simple ID3 version that was treated in the original presentation.

With this outline in hand, we turn to an introductory discussion of privacy preserving data mining.

2.3.1 Basic Concepts of PPDM

As one might expect, computer scientists have formulated their own definitions of privacy, in order to make feasible the process of proving claims about protocols and algorithms. Privacy preserving data mining ("PPDM") is new research area in which knowledge discovery algorithms are modified in light of their effects on informational privacy. According to Verykios et. al. [55], there are two main methods for combatting privacy breaches in KDD applications:

1. sensitive information like identifiers and names should be modified or excised, in order for the recipient of the data not to be able to compromise another person's privacy;

2. sensitive information which can be mined from a database by using KDD algorithms should be protected, because such knowledge can equally well compromise data privacy.

The same authors state that the main objective in PPDM is to "*develop algorithms for modifying the original data in some way, so that the private data and private knowledge remain private even after the mining process.*" As we outlined above, the problem of providing access to aggregate information (while restricting access to sensitive personal information) was studied extensively by the statistical database community. Many of the controls proposed have suffered from the *inference problem* –the challenge of ensuring that a data recipient cannot infer the value of sensitive data from the apparently innocuous data set.

The seminal paper on PPDM was written by Clifton and Marks [12], who proposed several methods to deal with the inference problem. A robust response would provide data custodians with criteria that accurately predict when a data set is not amenable to data mining. According to Clifton and Marks, the answer lies in: a) understanding data mining algorithms, and; b) providing a set of counter-measures. The authors made the following list of suggestions:

- **Limit access:** One can protect data by limiting access to it. This is the approach studied in the secure database management system community.
- **'Fuzz' the data:** One can protect data by altering its values.
- **Eliminate unnecessary groupings:** Groupings in data allow adversaries to find similarities that are not otherwise available.
- **Augment the data:** One can add misleading data to a data set, in order to confuse an adversary. Ideally, the misleading information will only be retrieved by 'inappropriate' queries, and not legitimate ones.
- **Audit:** A thorough auditing program will not prevent inappropriate inferences from being made, but it may detect misuse after the fact.

According to Aggarwal and Yu [2], most PPDM methods use some form of *transformation* on the data in order to provide privacy protection. These transformations typically reduce the granularity of the representation in order to protect sensitive information. The loss in granularity results in decreased knowledge discovery effectiveness. Hence, there is a *trade-off* between information loss and privacy. In order to quantify the degradation of the data, two metrics are commonly used: 1) a metric for data protection, and; 2) a metric for loss of functionality [55].

In the next sub-section, we offer a cursory overview of the various approaches to PPDM in the research literature.

2.3.2 A Taxonomy of Privacy Preserving Data Mining

Unfortunately, an accurate taxonomy of PPDM methods has yet to be published in the literature. Verykios et. al [55] note that existing approaches can be classified according to a number of dimensions:

1. **Data distribution:** Some methods have been developed by data in a centralised repository, while some have been developed for distributed data. For instance, *horizontal distribution* of data refers to those cases where different database records reside in different locations.

2. **Data modification scheme:** Most approaches modify the original values in a database. Methods of modification include:
   - *Perturbation*: altering attribute values, perhaps by adding noise.
   - *Blocking*: replacing existing attribute values with a 'null' value.
   - *Aggregation/merging*: combining several values into a coarser category.
   - *Swapping*: interchanging values between records in the data set.
   - *Sampling*: releasing data for only a sample (proper subset) of the population.
3. **Hiding data:** Some methods hide data, leading a data miner to produce weaker inference rules that will not allow the inference of confidential values. This process is known as *rule confusion*.
4. **Mining algorithm:** Some techniques only work with particular types of algorithms, such as decision trees or neural networks.
5. **Selective modification technique:** Selective modification is used to provide higher utility for the modified data set. Techniques include heuristic, cryptographic and reconstruction approaches.

As the reader can see, a classification scheme based on five dimensions is not amenable to a concise presentation. Nevertheless, it is worthwhile to present several important examples of approaches.

Important families of PPDM methods include:[47]

- **Randomization methods:** This family of methods concentrates on perturbation of the data set. Particular algorithms include approaches that work with data streams, as well as static tabular datasets.
- **Group-based anonymization:** This family of methods provides privacy protection by merging individual records into groups. Examples include the *k-anonymity* framework, the *l-diversity* framework, and the *t-closeness* model.
- **Distributed methods:** As mentioned above, this family concentrates on distributed data, whether partitioned vertically or horizontally.

[47] See [2] for more details on these methods.

2.3.3 Past Work on Decision Tree Algorithms

There are a number of papers that use decision trees, including:

- **Perturbation Approaches**: In one of the first efforts in the field of PPDM, Agrawal and Srikant [4] built decision-tree classifiers from data that had been perturbed. They introduced a procedure to reconstruct the original distribution of data values from the modified distribution. Using the reconstructed distribution, they were able to build classifiers whose accuracy was roughly comparable to those built with the original training data. However, their approach did not support categorical variables.
- **Unrealization**: In a recent paper, Fong [20] introduces a new scheme for PPDM that uses decision tree classifiers on manipulated datasets. The author implemented his scheme on the classical ID3 algorithm, which is modified to deal with data sets containing spurious information.

2.3.4 Our Contribution

Our work extends that of Fong [20] by implementing his approach on one of the most modern, popular and robust algorithms: the C4.5 classification tree algorithm from Quinlan [41]. In contrast to ID3, C4.5 has support for missing values in the dataset and numerical attributes. It also provides a pruning stage that produces trees with much greater accuracy than ID3.

We believe that showing the feasibility of the unrealization approach within a popular (and industrially accepted) framework for classification is of great utility to data mining practitioners. In addition, one of the secondary contributions in this thesis is to provide a much more rigorous, mathematically grounded presentation of the unrealization approach. We present full proofs for the many unproven assertions that occur in Fong's original thesis. In doing so, we firmly ground the unrealization approach in an axiomatization of the multi-relational algebra. We hope that this more mature treatment of the work will be useful to researchers who are curious about the unrealization approach.

2.4 Chapter Summary

Although lengthy, this Chapter has provided the background that is required to understand the work performed for this thesis. Since the reader may come from either a technical or legal background, we have placed equal emphasis on explaining the basic concepts of privacy law and knowledge discovery. In particular, the chapter contained sections on:

1. **Privacy:** We introduced the basic concepts of privacy, including the rationales that have been given for the importance of privacy interests. We detailed the constant dynamic in which technological advances spur (belated) developments in privacy law. We introduced modern data protection regimes by recounting the OECD Guidelines. We also gave an account of the technological developments that are threatening modern privacy law regimes, including growing repositories of data, automated decision-making, social networking, ubiquitous computing and data mining. We closed with a brief treatment of the technical approaches to privacy that have been developed by the computer science community.

2. **Data Mining:** In the next section, we related the basic concepts of data mining. We discussed the stages of the knowledge discovery processing, as well as a taxonomy of techniques. Decision trees were illustrated by means of a simple example, as they are the basis of our work in this thesis. Lastly, we gave the most comprehensive and accurate account in the literature of the challenges that data mining poses to modern data protection regimes.

3. **Privacy preserving data mining:** We provided an introduction to privacy preservation in knowledge discovery and data mining, placing an emphasis on the trade-off between information loss and privacy protection. We introduced key concepts, including the various dimensions used to categorise existing approaches. Finally, we discussed decision tree algorithms, and highlighted the contributions made in this paper.

With this background material behind us, we turn to the chapter that contains the bulk of our contribution.

Chapter 3

The New Approach and Solution

This Chapter contains much of the novel work contained in this thesis. Specifically, our contributions to the *unrealization* approach to privacy preserving data mining include:

1. Providing a solid explanation of the requirements and limitations of decision tree induction.
2. Introducing an axiomatization of multi-relational algebra, to make up for the lack of formalism in the original paper.
3. Adding more detail on impurity measures, and their use in decision tree algorithms.
4. Presenting a rigorous treatment of the unrealization approach in [20], with missing proofs and deficiencies from the original work corrected.
5. Demonstrating that the unrealization approach can work with the C4.5 algorithm.

In later Chapters, we provide an evaluation of the storage requirements for the unrealization approach on three real-world data sets. We also evaluate the unrealization approach against the core material on privacy introduced in Chapter 2.

In order to accomplish these tasks, however, we first need to obtain a solid understanding of the unrealization approach.

The *first* Section of this chapter introduces some of the required *formal background* in decision tree induction. First, we outline the requirements and limitations of the inductive paradigm in detail. Second, we provide a fully formalised account of the *multi-relational algebra*, suitable for performing proofs of the claims made in the original paper on unrealization approaches. Since multi-set algebra has markedly distinct properties from set theory and relational algebra, it is important to prove that the manipulations involved in the unrealization approach are actually permissible in a multi-relational setting.[1] Third, we give a formal account of training sets, using the multi-relational algebra introduced previously. Fourth, we present a *general framework* for top-down decision tree algorithms, including the key *impurity measure* that forms the basis of the ID3 algorithm used in [20].

The *second* Section of this chapter discusses the unrealization approach developed in [20]. First, we introduce the basic concept behind the *unreal* data sets that form the basis of the algorithm. Second, we provide an explanation of how the author modified the standard ID3 algorithm to work with the unreal data sets. Third, we engage in a lengthy and highly mathematical treatment of the unrealization approach, proving key claims that were taken for granted in the original presentation. Given the thorough nature of our treatment, this portion of the chapter is quite dense.

The *third* Section of this chapter extends the unrealization approach to the industry-standard C4.5 algorithm. We show that the C4.5 *gain ratio* impurity measure can be treated in a similar fashion to the *information gain* impurity measure used in ID3. We also show that the C4.5 *pruning procedure* can work with unreal data sets.

[1] A formal treatment of the multi-relational algebra is of the utmost importance, as there are key claims in [20] that were never proved.

3.1 Background

This section contains the following sub-sections:

1. **Decision Tree Induction**: We recap the basic features of decision tree induction, at a level of detail appropriate for computer science. In contrast to the previous chapter, we discuss some of the limitations and assumptions of decision tree induction.

2. **Multi-relational Algebra**: We introduce an axiomatization of the multi-relational algebra. The original presentation on unrealization suffered greatly from a lack of rigour, which was partially due to an absence of a formal theory of multi-sets. We aim to overcome this limitation, in order to see if the claims made in the original paper can be supported.

   In general, treating multi-sets as sets is a sure path to disaster. The properties of multi-sets differ drastically from those of sets. Some key properties, such as DeMorgan's laws, do not hold.

3. **A Formal Account of Training Sets**: We apply the multi-relational algebra to the concrete setting of training sets for induction. We introduce the vocabulary and key symbols that will be used throughout the text.

4. **A Framework for Decision Tree Algorithms**: We present a framework for understanding top-down decision tree algorithms, including ID3 and C4.5. We discuss the recursive tree growth algorithm, pruning, and impurity metrics.

   In particular, the portion on metrics is vital for understanding the *information gain* and *gain ratio* metrics that are used in ID3 and C4.5. Calculating the information gain of a split is a crucial component in the unrealization algorithm presented in [20]. Without a detailed understanding of the mechanism, we cannot extend that work to the C4.5 algorithm.

3.1.1 Decision Trees Induction

As we mentioned in the previous chapter, a *decision tree* is a tree structure formed from a set of *nodes*. One of the nodes is called the *root* of the tree; as one might guess, it has no incoming edges. The rest of the nodes in the tree have exactly one incoming edge, and zero or more outgoing edges. A node with outgoing edges (*branches*) is referred to as an *internal* (or *test*) node. A node with zero outgoing edges is called a *leaf* (also known as a *terminal* or *decision* node). The function of each of these components is listed below:

1. **Internal nodes**: Each internal node represents a *test* that splits the instance space into a partition of sub-spaces. The split is made according to a discrete function of the input attribute values. In the simplest case, each test considers a single attribute, such that the instance space is partitioned according to the attribute's value [43, at p.8].

2. **Branches**: Each branch from an internal node represents an outcome of the test at that node. An arbitrary number of branches may grow from each internal node. Each node corresponds with a certain characteristic (typically a single attribute) and the branches correspond with a range of values for that characteristic. The range of values used in the branches must induce a partition of the set of values for the given characteristic [43, at p.9].[2]

3. **Leaves**: Each leaf represents a class to which the incoming object is assigned. In particular, each leaf is typically assigned to a class that represents the most appropriate target value.[3]

Instances are classified by navigating from the root of the tree down to a leaf, choosing the appropriate branches according to the outcome of the tests encountered. For each internal node, the classification algorithm considers the characteristic tested by the node, finding the branch that matches the characteristic expressed by the instance.

[2] If this condition is not met, then there will be gaps in the decision tree. Some instances may not be classifiable.

[3] Alternatively, the leaf may hold a probability vector (affinity vector) indicating the probability of the target attribute having a certain value. See [43] at p.9.

The *size* of a decision tree is very important. It may be measured by a number of metrics, including: a) total number of nodes; b) total number of leaves; c) tree depth, and; d) number of attributes used. Smaller trees are generally more comprehensible. In addition, tree size has an effect on accuracy. In top-down tree generation methods, the size of a tree is controlled by the *stopping criteria* and *pruning method* [43, at p.9].

Decision Tree Inducers

The underlying strategy of decision tree induction is a non-incremental learning from examples. The induction algorithm is presented with a set of examples which it uses to develop a decision tree. Typically, trees are developed "top down", guided by frequency information in the examples, but not by the particular order in which the examples are given [40]. The examples are known only through their values of a set of properties/attributes. The decision trees are also expressed in terms of these attributes.

There are many classification tree algorithms in the literature.[4] We restrict ourselves to a brief discussion of two examples:

- **ID3:** This algorithm uses a measure called *information gain* as a splitting criteria. The tree ceases to grow when all instances belonging to a single value of a target feature, or when best information gain is not greater than zero. There is no pruning procedure, and the algorithm does not handle missing values or numeric attributes [43, at p.71]
- **C4.5:** This algorithm uses a measure called the *gain ratio* as a splitting criteria. The splitting cases when the number of instances to be split is below a certain threshold. Error-based pruning is performed after the growing phase. C4.5 can handle numeric attributes. It can also induce form a training set that incorporates missing values by using corrected gain ratio criteria.

These algorithms are quite relevant to this thesis. ID3 was the algorithm chosen for the first work on privacy protection using unrealized datasets [20]. One of our goals in this paper is to extend that work to the C4.5 framework.

[4] See [43] for a comprehensive survey.

Limitations

As described in [41], not all classification tasks lend themselves to the inductive decision tree approach. The following requirements must be met in order for the method to work:

1. **Attribute-value description**: The data to be analyzed must be in a *flat file*, in which all information about an object is expressible in terms of a fixed collection of properties or attributes. Each attribute may have either discrete or numeric values, but the attributes used to describe a case must not vary from one case to another.

2. **Predefined classes**: The categories to which objects are to be assigned must have been established beforehand.

3. **Discrete classes**: The classes must be sharply delineated. An object either does or does not belong to a particular class. Prediction of continuous values or vague categories are not in scope.

4. **Sufficient data**: Inductive generalization proceeds by identifying patterns in data. The approach does not work if valid patterns cannot be distinguished from mere coincidences. As the differentiation typically depends on statistical tests, there must be sufficient cases to allow these tests to be effective.

5. **Logical classification**: The decision tree method constructs classifiers that can be represented as decision trees. It cannot represent other types of classifiers, such as linear discriminants.

An important concept is that of *distribution*. As stated by Quinlan, a fully distributed classifier's performance does not depend critically on any small part of the model. Some examples can be forgotten, some branches corrupted, or some elements discarded, without destroying the performance of the model [41, at p.15]. Unfortunately, decision trees are quite susceptible to small alterations.

Lastly, the problem of finding the smallest decision tree consistent with a training set is NP-complete. As a result, most decision tree methods are non-backtracking, greedy algorithms. [41, at p.20]

3.1.2 The Multi-Relational Algebra

In this section, we provide background on the formalism that provides the foundation for decision tree algorithms. In the theory of databases, the *relational algebra* provides the theoretical basis for a variety of practical operations, including projection, selection, and joins. Unfortunately, the relational algebra suffers from a major deficiency –namely, it deals with *sets* of tuples. Since sets do not allow duplicates, this means that relational algebra does not provide the tools required to deal with data sets containing duplicate records.

In contrast, the various theories of *multi-relational algebra* were explicitly designed to deal with duplicate records. The basic structure in a multi-relational algebra is a *bag* (or *multi-set*) that can contain more than one copy of a given object. (When storing only one copy of any object, bags are equivalent to sets).

Sadly, the theory of multi-relational algebra is not nearly as well developed as that of relational algebra. Unlike the ZF axiomatization of set theory, there is no commonly accepted axiomatization for the theory of multi-sets. Even worse, Joseph Albert [5] has proven that many of the algebraic properties of sets fail for multi-sets.

Our goal in this section is to provide an account of multi-sets that avoids these difficulties, enabling the work in Fong [20] to be placed on firmer bedrock than that provided by a nave approach to the algebra of bags.[5] We will introduce only the machinery necessary for accomplishing this purpose, avoiding superfluous operators, theorems and meta-mathematical results in this fairly technical subject area.

[5]Our treatment is based on [22] and [5]. A thorough treatment of the issues involved in the multi-relational algebra is beyond the scope of this work.

Basic Structures

In this section, we detail the basic structures of our sub-set of multi-relational algebra:

- **Objects**: our first concept consists of a countable set of *primitive objects* $O = \{o_1, o_2, \dots\}$, where o_i is an object. All of the objects in O are *atomic*, meaning that each object is an indivisible unit. Objects may have a *type*.

- **Domain**: a *domain* d is a subset of O, together with a name and a type. In this work, we assume that our domains are finite. We also assume that they contain values of the same type. A domain may consist of a set of strings or a set of integers; however, it may not consist of a set of strings and integers.

  For instance, the set of real numbers $\mathbb{R}$ is a domain, as is the set of values $\{high, medium, low\}$.

- **Multi-set**: a *multi-set* (also called a *bag*) is a collection of objects that permits duplicates. The number of occurrences of a given object o in a multi-set T is called its *multiplicity*, denoted by $o \in\in T$.

  Multi-sets have two *representations*:

  1. They can represented as a *collection of individual objects*, possibly containing duplicates. We use *square brackets* to define multi-sets in this manner, to distinguish them from regular sets.

     As a first example, we can represent a multi-set T containing three copies of a and two copies of b as $T = [a, a, a, b, b]$. As a second example, the multi-set $[o_1, o_2, o_2, o_3, o_4]$ contains two copies of object o_2.

  2. Alternatively, a multi-set T can represented as a *set of pairs*. Each pair is $(o, T(o))$, where o is an object, and $T(o)$ represents the number of occurrences of object o in T. An object in the domain $dom(T)$ that is not contained in the multi-set may be represented by the pair $(o_i, 0)$, indicating that there are zero copies of o_i present.[6]

     In this approach, the multi-set $T = [a, a, a, b, b]$ can be represented by the set $\{(a, 3), (b, 2)\}$. The multi-set $[o_1, o_2, o_2, o_3, o_4]$ above is represented as the set $\{(o_1, 1), (o_2, 2), (o_3, 1), (o_4, 1)\}$.

[6] This formalizes the intuition that a multi-set T can be equated with a function $f_T : dom(T) \rightarrow \mathbb{N}$.

We say that an object o is 'in' a multi-set T, denoted $o \in T$, if $k > 0$ copies of it appear in T. That is, the pair (o, k) appears in the set pair representation of T. In contrast, if the pair $(o, 0)$ appears in the set pair representation of T, then $o \notin T$.

- **Relation schema**: a relation schema $\mathcal{R}$ (also called a *bag schema*) is a list of attributes $\langle a_1, a_2, \ldots, a_n \rangle$. Each attribute a_i takes values from a single domain $dom(a_i)$. The *type* of a relation schema $\mathcal{R}$ is defined as $dom(\mathcal{R}) = dom(a_1) \times dom(a_2) \times \cdots \times dom(a_n)$.

- **Tuples**: A tuple $t = (t_1, t_2, \ldots, t_n)$ on relation schema $\mathcal{R}$ is an element in $dom(\mathcal{R}) = dom(a_1) \times dom(a_2) \times \cdots \times dom(a_n)$. We indicate the i-th element of the tuple t_i by $t[i]$. The number of attributes in the tuple is denoted by $\#(t)$. Lastly, we denote the number of times that a tuple t appears in multi-set T by $x \in\in T$.[7]

  The *equality relation on tuples* is an important concept. If two tuples are defined on a different relation schema, they are not equal. If two tuples t_1 and t_2 are defined on the same relation schema, $(t_1 = t_2)$ if and only if $(t_1[i] = t_2[i])$ for $1 \leq i \leq n$.

- **Universal instance space**: the universal instance space U for a relation schema $\mathcal{R}$ is the set $dom(\mathcal{R}) = dom(a_1) \times dom(a_2) \times \cdots \times dom(a_n)$. The universal instance space contains all possible tuples drawn from the domain of $\mathcal{R}$, without duplicates.

- **Relation instance**: a relation instance R (also called a *bag instance*) on relation schema $\mathcal{R}$ is a multi-set of elements in $dom(\mathcal{R})$. In other words, a relation instance consists of a multi-set containing tuples drawn from $U_{\mathcal{R}}$.

Predicates

In addition to objects and multi-sets, our small version of the multi-relational algebra includes predicates.

[7] Although cumbersome, this notation is used frequently in the literature.

- **Atomic Predicates**: In any given interpretation, the set *Pred-A* is composed of *atomic predicates* on the set of objects O. An n-place predicate is a relation on $O^n \times \{0, 1\}$.

  As an example, imagine the objects are natural numbers, so $O = \mathbb{N}$. Then $(x = 2)$ is a unary predicate.

- **Compound Predicates**: In any given interpretation, the set *Pred-C* consists of predicates that are built up from those in *Pred-A*, in the following manner:

  1. If ψ is an atomic predicate, it is in *Pred-C*.
  2. If ψ_1 is a predicate in *Pred-C*, $\neg\psi_1$ is a predicate in *Pred-C*. [8]
  3. If ψ_1 and ψ_2 are predicates in *Pred-C*, $\psi_1 \wedge \psi_2$ is a predicate in *Pred-C*. [9]
  4. If ψ_1 and ψ_2 are predicates in *Pred-C*, $\psi_1 \vee \psi_2$ is a predicate in *Pred-C*. [10]

  It is easy to show that *Pred-C* is denumerable, containing the closure of the atomic predicates under the boolean operators $\neg, \vee, \wedge$. Note that a compound predicate may have greater *arity* than its atomic components. For example, $(x = 2)$ and $(y) = 4$ are both atomic predicates of arity 1, while $(x = 2) \wedge (y = 4)$ is a compound predicate of arity 2

In this work, predicates will be used in the *selection* operator to be introduced below.

Basic Operators

In this section, we present some important operators that are vital to understanding the various proofs in this work. In the interests of brevity, we present only those operators that appear in the proofs. Assume T_1 and T_2 are relation instances (multi-sets) on $\mathcal{R}$:

- **Containment ($T_1 \subseteq T_2$)**: This is the analogue of the subset operator in set theory:

$$T_1 \subseteq T_2 \text{ iff } (\forall t)(t \in dom(\mathcal{R}) \rightarrow (t \in\in T_1) \leq (t \in\in T_2)) \tag{3.1}$$

[8] The symbol $\neg$ stands for the *negation* operator from boolean algebra.
[9] The symbol $\wedge$ stands for the *conjunction* operator from boolean algebra.
[10] The symbol $\vee$ stands for the *disjunction* operator from boolean algebra.

That is, T_1 is contained in T_2 if: a) it is based on the same schema, and; b) every tuple that occurs in T_1 k times occurs in T_2 at least k times.

- **Equality ($T_1 = T_2$)**: Equality is a fundamental notion:

$$T_1 = T_2 \text{ iff } (\forall t)(t \in dom(\mathcal{R}) \rightarrow (t \in\in T_1) = (t \in\in T_2)) \tag{3.2}$$

- **Strict Containment ($T_1 \subset T_2$)**: Akin to a proper subset in set theory:

$$T_1 \subset T_2 \text{ iff } T_1 \subseteq T_2 \text{ and } T_1 \neq T_2 \tag{3.3}$$

We pause for a moment to prove an identity that is used frequently in this work:

Lemma 1. *$T_1 = T_2$ iff $T_1 \subseteq T_2$ and $T_2 \subseteq T_1$.*

Proof. Assume $T_1 = T_2$. We begin with the forward direction of the biconditional. Take any $t \in dom(\mathcal{R})$. We know $(t \in\in T_1) = (t \in\in T_2)$, so $(t \in\in T_1) \leq (t \in\in T_2)$. Hence, $T_1 \subseteq T_2$. Similarly, we know that $(t \in\in T_2) \leq (t \in\in T_1)$. Hence, $T_2 \subseteq T_1$.

For the other direction of the biconditional, assume that $T_1 \subseteq T_2$ and $T_2 \subseteq T_1$. From $T_1 \subseteq T_2$ we know that $(t \in\in T_1) \leq (t \in\in T_2)$. From $T_2 \subseteq T_1$ we know that $(t \in\in T_2) \leq (t \in\in T_1)$. The only way these statements can be true is if $(t \in\in T_1) = (t \in\in T_2)$. Hence $T_1 = T_2$. □

The next operations that we introduce are integral to the work performed in this paper. As above, we assume that T_1 and T_2 are relation instances on $\mathcal{R}$. (If they are not, type mismatches will arise). For convenience, we use the set of pairs representation for a multi-set that was outlined above.

- **Union ($T_1 \cup T_2$)**:

$$T_1 \cup T_2 = \{(x, max(x \in\in T_1, x \in\in T_2))|x \in dom(\mathcal{R})\} \tag{3.4}$$

The union operation results in the smallest multiset that contains both T_1 and T_2. For instance, if $T_1 = [a, a, b]$ and $T_2 = [a, b, b, c]$, then $T_1 \cup T_2 = [a, a, b, b, c]$.

- **Concatenation** ($T_1 \uplus T_2$):

$$T_1 \uplus T_2 = \{(x, x \in\in T_1 + x \in\in T_2) | x \in dom(\mathcal{R})\} \tag{3.5}$$

The concatenation operation results in a new multi-set that has a copy of each of the tuple instances from T_1 and T_2. For instance, if $T_1 = [a, a, b]$ and $T_2 = [a, b, b, c]$, then $T_1 \cup T_2 = [a, a, a, b, b, b, c]$.

- **Difference** ($T_1 - T_2$):

$$T_1 - T_2 = \{(x, max(x \in\in T_1 - x \in\in T_2, 0)) | x \in dom(\mathcal{R})\} \tag{3.6}$$

The multi-set difference operation results in a new multi-set that has a copy of each tuple instance from T_1, but with each tuple instance from T_2 removed. For instance, if $T_1 = [a, a, b]$ and $T_2 = [a, b, b, c]$, then $T_1 \cup T_2 = [a]$.

- **Selection** (σ): Using the selection operator σ on a multiset T is a means of retrieving only those elements of T that satisfy a predicate. In particular, $\sigma_\phi(T_1)$ selects those tuples in T that satisfy predicate ϕ (which is defined on the attributes in $\mathcal{R}$). The resulting multi-set is also defined on $\mathcal{R}$.

$$\sigma_\phi(T) = \{(x, x \in\in T) | x \in dom(\mathcal{R}) \wedge \phi(x)\} \cup \{(x, 0) | (x \notin dom(\mathcal{R}) \vee (x \in dom(\mathcal{R}) \wedge \neg\phi(x))\} \tag{3.7}$$

- **Size** ($|T|$): The size of a set is the number of elements contained in it. Since a multi-set has duplicates, the size of a multi-set is the number of copies contained within it.

$$|T| = \sum_{t \in \mathcal{R}} t \in\in T \tag{3.8}$$

Since we are dealing with multi-sets containing finite numbers of elements, we can set aside worries about computing the size of multi-sets containing countably infinite numbers of elements.

The proofs that these operations are well defined can be found in Albert [5]. Since that paper also demonstrates that multi-sets lack some of the algebraic properties of sets, we will be careful to restrict ourselves to operations that are legitimate for multi-sets. We do so by proving key properties as needed.

Derived operators

In this section, we introduce some derived operators that will come in handy during our exposition:

- **Iterated Concatenation**: We can apply a union to the same set multiple times, as follows. Let T be a multi-set on relation schema $\mathcal{R}$. We define the iterated union operator on T as follows:

$$T \uplus_1 T = T \uplus T \tag{3.9}$$
$$T \uplus_n T = T \uplus_{n-1} T \tag{3.10}$$

  We can show by induction that

$$T \uplus_n T = \{(x, n(x \in\in T)) | x \in dom(\mathcal{R})\} \tag{3.11}$$

- **Multiplication by a Constant**: let q be a constant and T a multi-set. We call $q(T) = T \uplus_q T$ the *q-multiple of* T. Using set pair notation:

$$q(T) = T \uplus_q T = \{(x, q(x \in\in T)) | x \in dom(\mathcal{R})\} \tag{3.12}$$

  Note that this definition implies that

$$|q(T)| = q|T| \tag{3.13}$$

Meta-mathematical notation

In addition to the structures and operators of our (admittedly crippled) multi-relational algebra, we introduce some meta-mathematical symbols.

- **Replacement**: We use the symbol $\leftarrow$ to denote replacement on multi-sets. That is, if T_1 and T_2 are multi-sets based on the same relation schema $\mathcal{R}$, $T_1 \leftarrow T_2$ means that we remove all tuples from T_1, replacing them with a copy of the contents of T_2.
- **The Empty Multi-Set** $\varnothing$: We use the symbol $\varnothing$ to denote an empty multi-set. In particular, we say that a multi-set T is *empty* iff $\forall x \in dom(\mathcal{R}))((x, 0) \in T)$. By convention, we assume there is a unique empty multi-set, referred to as $\varnothing$.

Limitations of the Multi-Relational Algebra

One must be careful in using set theoretic operations on multi-sets. As noted by Albert [5], the key point regarding union, intersection and complement for sets is that these operations satisfy the axioms of a Boolean algebra. As he shows in the same paper, no Boolean algebra structure is available for multi-sets, if the semantics of bag operations are defined similarly to those of multi-sets. In addition, there are other properties of multi-set operations that have no set-theoretic counterpart.

As a result of these concerns, we will be exceedingly careful in our use of multi-relational algebra expressions, making sure to prove theorems before we use them. In our next section, we show how we will adapt our carefully chosen subset of relational algebra to the definition of training sets.

3.1.3 A Formal Account of Training Sets

We follow Rokach and Maimon [42] in characterising a training set as an instance of a *relation schema*. We diverge from them in terms of notation, and in terms of the precision with which we define key terms. The following concepts are key:

- **Examples and attributes:** Data sets used for classification take the form of an arbitrary number of *examples*, each of which is described by an set of *attributes*. Interpreting this state of affairs in terms of multi-relational algebra, each example is a *tuple*. The set of n possible attributes is denoted by $A = \{a_1, a_2, \ldots, a_n\}$.

  In the *unrealization* approach, attributes are assumed to have *nominal* values, instead of numeric ones. We can denote the set of k_i possible domain values for a nominal attribute a_i by $dom(a_i) = \{v_1^i, v_2^i, \ldots, v_{k_i}^i\}$. Another way to represent the k_i possible values for attribute a_i is by using the set cardinality expression $|dom(a_i)|$.

- **Target / class variable:** We denote the *target attribute* of the training set by the letter y. The target attribute (or class variable) is the feature of interest from the standpoint of classification. As above, the k_y possible values for y are represented by $dom(y) = \{v_1^y, v_2^y, \ldots, v_{k_y}^y\}$. Then $|dom(y)| = k_y$.

- **Instance spaces:** The set of all possible (unclassified) examples is called the *instance space*. The instance space therefore consists of all combinations of non-target attributes. Formally, the instance space is

$$X = dom(a_1) \times dom(a_2) \times \cdots \times dom(a_n) = \prod_{i=1}^{n} dom(a_i) \tag{3.14}$$

  In contrast, the *universal instance space* (or *labelled instance space*) is the set of all possible (classified) examples. Formally, the *universal instance space* is:

$$U = X \times dom(y) = dom(a_1) \times dom(a_2) \times \cdots \times dom(a_n) \times dom(y) \tag{3.15}$$

  Where the matter is not likely to cause confusion, we will sometimes treat the target variable y as a regular attribute. In such a case, the universal instance space is defined as in the multi-relational algebra above:

$$U = \prod_{i=1}^{n} dom(a_i) \tag{3.16}$$

However, we may only treat the universal instance space in this manner if we point out that the attributes include the target variable y.

- **Relation schemas:** A *relation schema* provides a description of the attributes and their domains. Formally, a relation schema is denoted by $\mathcal{R}(A \cup y) = \mathcal{R}(\{a_1, a_2, \ldots, a_n\} \cup \{y\})$.

- **Training Set:** A *training set* is a relation instance consisting of a set of m tuples. Formally, a training set T on bag schema S is a finite set of tuples $T(S) = \{t_1, t_2, \ldots, t_m\}$. Each tuple t_i is drawn from the universal instance space U.

  Tuples represent *classified* examples, since they include the target attribute y. In particular, each tuple t_i can be viewed as a pair (x_i, v_i^y), where $x_i \in X$ and $v_i^y \in dom(y)$.

An Example

To provide an illustration of the basic formalism, we present an example training set. The issue in question is whether a hiking trip is desirable, given certain environmental conditions. To provide a hint sense of realism, we assume that the training data is drawn from the notebooks of a park ranger, who has observed numerous hiking outings end in disaster. In our contrived description, there are three attributes of interest: a) the temperature; b) the chance of precipitation, and; c) whether it is bear season.

- **Attributes**: $A = \{\text{temperature}, \text{bear-season}, \text{precipitation}\}$.
- **Target**: $y = \{\text{hike}\}$
- **Domains**: The following domain values are present:
  - $dom(\text{temperature}) = \{\text{hot}, \text{cold}, \text{warm}\}$
  - $dom(\text{precipitation}) = \{\text{low}, \text{high}\}$
  - $dom(\text{bear-season}) = \{\text{yes}, \text{no}\}$
  - $dom(\text{hike}) = \{\text{yes}, \text{no}\}$
- **Instance space**: the following table describes the instance space:

temperature	bear-season	precipitation
hot	no	low
hot	no	high
hot	yes	low
hot	yes	high
warm	no	low
warm	no	high
warm	yes	low
warm	yes	high
cold	no	low
cold	no	high
cold	yes	low
cold	yes	high

- **Universal instance space**: To form U, we merely add the target attribute 'hike':

temperature	bear-season	precipitation	*hike*
hot	no	low	*yes*
hot	no	high	*yes*
hot	yes	low	*yes*
hot	yes	high	*yes*
warm	no	low	*yes*
warm	no	high	*yes*
warm	yes	low	*yes*
warm	yes	high	*yes*
cold	no	low	*yes*
cold	no	high	*yes*
cold	yes	low	*yes*
cold	yes	high	*yes*
hot	no	low	*no*
hot	no	high	*no*
hot	yes	low	*no*
hot	yes	high	*no*
warm	no	low	*no*
warm	no	high	*no*
warm	yes	low	*no*
warm	yes	high	*no*
cold	no	low	*no*
cold	no	high	*no*
cold	yes	low	*no*
cold	yes	high	*no*

- **Training set**: Recall that a training set is a bag instance drawn from the universal instance space. A sample training set is shown below:

temperature	bear-season	precipitation	*hike*
hot	no	low	*yes*
hot	no	low	*yes*
warm	yes	low	*yes*
warm	yes	low	*no*
cold	no	low	*no*
cold	yes	high	*no*

  Recall that a training set may contain duplicate records. In the example above, the first two rows are identical. Even worse, there are two entries for the values (warm, yes, low), each of which has a different target attribute associated with it. This sort of inconsistency can confuse an inductive learner.

In the next section, we provide an example of a key operation –that of *selection*.

Selection of Tuples

Recall that the *selection* operation, when applied to a training set T, returns a training set T' that contains only those records from T that satisfy a given *predicate*. As we discussed above, the projection operation is denoted by σ, and the predicate is given as a subscript. To provide a concrete example, we illustrate projection using an atomic predicate. Let T be the following training set:

temperature	bear-season	precipitation	*hike*
hot	no	low	*yes*
warm	yes	low	*no*
cold	no	low	*no*

The projection $\sigma_{a_i=v}(T)$ denotes a multi-set that contains all those tuples (including duplicates) from T in which attribute a_i takes the value v. In particular, $\sigma_{\text{temperature=warm}}(T)$ is:

temperature	bear-season	precipitation	*hike*
warm	yes	low	*no*

This definition can be expanded to cover non-atomic predicates formed from boolean operators. For instance, consider the following training set T:

temperature	bear-season	precipitation	*hike*
hot	no	low	*yes*
warm	yes	low	*no*
cold	no	low	*no*

We wish to select only those rows in which the temperature is hot or bears are in season. We denote this operation by the formula $\sigma_{\text{temperature=hot}\vee\text{bear-season=yes}}(T)$. The end result of the projection is:

temperature	bear-season	precipitation	*hike*
hot	no	low	*yes*
warm	yes	low	*no*

3.1.4 A Framework for Decision Trees

In a typical supervised learning scenario, a training set is given, and the goal is to form a description that can be used to predict previously unseen examples. Each record in the training set is described by a set of attributes, each of which takes values from a finite domain. The target attribute is typically demarcated in a special way, even if it is merely one of many attributes in the training set.

As described in [42] and [43], top-down decision tree induction algorithms can be thought of as constructing a tree by growing it from data, and then (optionally) pruning the resulting tree to remove nodes. Following [42], we show the high-level control structure for this process in the table below:

Algorithm:	*DecisionTreeInducer*(T, A, y)
Inputs:	T: the training set. A: the list of attributes. y: the target attribute.
Outputs:	D: a decision tree.
1: 2:	DecisionTree d = *GrowTree*(T,A,y); return *PruneTree*(d,T);

Table 3.1: Top-Down Decision Tree Induction Algorithm

Growing

Recall that the internal nodes in a decision tree impose attribute-based tests on incoming examples. As pointed out by Quinlan [41], any internal node that divides the training set in a nontrivial way (so that at least two of the subsets are not empty) will eventually yield a partition of the instance space, even if all of the subsets in the

partition contain a single training case. That is, a decision tree can (at the risk of over-training) always induce a partition of the training set.

Despite this result, the process of building a decision tree is not merely intended to find any such partition; instead, the goal is to build a *model* that reveals the structure of the application domain. This sort of model will have greater predictive power than one that merely fits the training set. In order to represent this structure at a higher level, we need a significant number of cases at each leaf. As Quinlan puts it, we would ideally like to choose a test at each stage so that the final tree is small.

Pseudocode for the growing algorithm is presented below:

Algorithm:	*GrowTree*(T,A,y))
Inputs:	T: the training set. A: the list of attributes. y: the target attribute.
Outputs:	D: a decision tree.
1:	DecisionTree d = new Tree;
2:	if (*StoppingCriteriaMet*(T,A,y))
3:	{
4:	d.type = "leaf";
5:	d.class = *SelectRepresentativeClass*(T);
6:	return d;
7:	}
8:	Select $a_i \in A$ that maximizes *SplitCriterion*(a_i, T);
9:	d.type = "internal";
10:	d.class = "a_i";
11:	for each outcome v^j of a_i:
12:	{
13:	DecisionTree c = *GrowTree*$(\sigma_{a_i=v_j}(T), A, y)$;
14:	d.*AddLabeledBranchToSubtree*(c, v_j);
15:	}
16:	return d;

Table 3.2: Decision Tree Growth Algorithm

It is important to note that the training set T is partitioned in each recursive call, instead of being passed in its entirety. In particular, the recursive call to generate a subtree corresponding to the value v_j of the chosen split attribute a_i only passes those training examples in which the value of a_i is v_j.[11]

The TreeGrow algorithm uses several helpers, some of which will be discussed in more detail when we present the ID3 algorithm in the context of Fong's unrealization algorithm [20]:

- *StoppingCriteriaMet*(T, A, y): This routine determines when the recursion should terminate. The stopping criterion is a major determinant of tree size.
- *SelectRepresentativeClass*(T): This routine is called when the stopping criteria is reached, in order to determine how a leaf should be categorised. The most expedient means of determining the class of a leaf upon termination is to select the classification that occurs most frequently in the remaining training instances T.
- *SplitCriterion*(a_i, T): This helper determines which attribute offers the best split of the training set.
- *AddLabeledBranchToSubtree*(c, v_j): This helper merely adds the tree c constructed during the recursive call as a child of the tree d, using a labelled branch. Each subtree has a value label v_j.

Pruning

After the growing phase is completed, the resulting tree can be reduced in size through the application of tree *pruning* techniques. As we noted above, there are several advantages to smaller decision trees. Since obtaining the smallest decision tree is intractable, heuristic methods are used to reduce tree size incrementally.

Since the ID3 algorithm used in the unrealization paper by Fong does not involve pruning, we will defer a discussion of pruning until the section on the C4.5 algorithm.

[11] Our presentation of the algorithm is based on [43] (at p.20), but we have made several improvements.

3.1.5 Splitting Criteria

In this section, we present some common choices for splitting criteria.[12] Most of the popular decision tree induction algorithms use *univariate* splitting criteria, in which the test at each internal node is based on the values of a single attribute. Although multivariate splitting criteria have been studied extensively, the simplicity of univariate criteria helps to explain their popularity.

According to Rokach and Maimon [43], splitting criteria can be characterised in a number of ways, including: a) the origin of the metric used, and; b) the structure of the metric. The table below lists their categories for each of these concepts:

Origin of Measure	Measure Structure
information theory	impurity-based criteria
dependence	normalised impurity-based criteria
distance theory	binary criteria

For reasons of brevity, we will not launch into a discussion of the various splitting criteria in the literature. We will, however, detail the background theory behind two important criteria, namely: a) *information gain*, used by the ID3 algorithm that is the basis of Fong and Weber's paper, and; b) the *gain ratio* criteria, which is a refinement of information gain that was introduced by Quinlan for the C4.5 algorithm. In order to accomplish this goal, we must introduce the concept of *information entropy*.

[12] A comprehensive survey is presented in [42].

Entropy

A central concept in the mathematical discipline of *information theory*, entropy provides us with a way of measuring the *uncertainty* associated with a *random variable*. Assume that a discrete random variable $\mathcal{X}$ has n possible outcomes $\{k_1, k_2, \ldots, k_n\}$. Let the probability mass function $f_{\mathcal{X}}$ assign probabilities $P(\mathcal{X} = k_i)$ to each outcome k_i. Representing this function as a *stochastic vector*, we have

$$f_{\mathcal{X}} = (P(k_1), P(k_2), \ldots, P(k_n))$$

The following definitions will come in handy:

1. The *shortest* stochastic vector has the value $1/n$ as each component of the vector, and has a length of $\frac{1}{\sqrt{n}}$.
2. The *longest* stochastic vector has the value 1 in a single component and 0 in all others, and has a length of 1.

We can provide a measure of the uncertainty surrounding the random variable X by measuring the information that is conveyed by the occurrence of an event. For instance, if we know that $P(\mathcal{X} = k_i) = 1$, the occurrence of the event $\{k_i\}$ provides us with no information. (In fact, we could predict the outcome with certainty). In contrast, knowing that a fair coin has landed heads is highly revealing, since we had no information on which to favour one outcome (heads) over the other (tails).

The *information entropy* $\mathcal{H}$ associated with a probability density function $f_{\mathcal{X}}$ (for a discrete random variable $\mathcal{X}$ with outcomes $\{k_1, k_2, \ldots, k_n\}$) quantifies this intuition. Formally, information entropy is defined as:

$$\begin{aligned} \mathcal{H}(\mathcal{X}) &= \text{InformationContent}(f_{\mathcal{X}}) \\ &= \text{InformationContent}(P(k_1), P(k_2), \ldots, P(k_n)) \\ &= -\sum_{i=1}^{n} P(\mathcal{X} = k_i) \log_2 \left(P(\mathcal{X} = k_i)\right) \end{aligned} \tag{3.17}$$

The *information content* of an outcome is a measure of how much information we obtain by finding out if an outcome obtains. This value lies within a closed, bounded subset of $\mathbb{R}$, namely $[0, \log_n(\frac{1}{n})]$. In particular:

- InformationContent(k_i)=0 means that event $\{k_i\}$ is completely uninformative
- InformationContent(k_i) = $\log_2(\frac{1}{n})$ means that we had no reason to prefer any outcome in advance (and therefore, the occurrence of $\{k_i\}$ is maximally informative).

It turns out that the long stochastic vectors have low information content. In turn, short stochastic vectors have high information content. We will see this theme play out in the next sections.

Impurity-based Criteria

Impurity-based criteria are presented in detail by Rokach and Maimon in [42] and [43]. Given a random variable $\mathcal{X}$ with k discrete values distributed according to stochastic vector $P = (p_1, p_2, \ldots, p_k)$, an *impurity measure* is a function $\phi : [0,1]^k \to \mathbb{R}$ that satisfies the following conditions:

1. $\phi(P) \geq 0$
2. $\phi(P)$ is *minimum* if $(\exists i)(p_i = 1)$.
3. $\phi(P)$ is *maximum* if $(\forall i)((1 \leq i \leq k) \to (p_i = 1/k))$.
4. $\phi(P)$ is symmetric with respect to components of P.
5. $\phi(P)$ is differentiable everywhere in its range.

If the stochastic vector P is *long*, (i.e., has a single component of 1), then $\phi(P)$ is minimal, and the random variable $\mathcal{X}$ is defined as *pure*. If the stochastic vector is *shortest*, (i.e., the components have the same value), then $\phi(P)$ is maximum. In the latter case, we say that random variable $\mathcal{X}$ is at the *maximum level of impurity*.[13]

[13]For instance, $P = (0, 0, 1, 0, 0)$ indicates a pure variable, while $P = (1/5, 1/5, 1/5, 1/5, 1/5)$ is the maximum level of impurity for a variable with five outcomes.

Given a training set T, the stochastic vector of the target attribute (discrete random variable) y is defined as:

$$\begin{aligned} P_y(T) &= (P(y = c_1), P(y = c_2), \ldots, P(y = c_{|dom(y)|})) \\ &= \left(\frac{|\sigma_{y=c_1}(T)|}{|T|}, \frac{|\sigma_{y=c_2}(T)|}{|T|}, \ldots, \frac{|\sigma_{y=c_{|dom(y)|}}(T)|}{|T|} \right) \end{aligned} \tag{3.18}$$

In other words, $P_y(T)$ gives us the frequency of the target attribute y as represented in the training data T.

Given this interpretation, one can define the *goodness-of-split* $\Delta\Phi(a_i, T)$ due to the attribute a_i as the reduction in impurity of the target attribute y after partitioning the training set T according to the values $\{v_1^i, v_2^i, \ldots, v_j^i\}$ of a_i:

$$\begin{aligned} \Delta\Phi(a_i, T) &= \text{(impurity on T)} - \text{(impurity on partitions of T induced by } a_i) \\ &= \phi(P_y(T)) - \sum_{j=1}^{|dom(a_i)|} P(a_i = v_j^i)\phi(P_y(\sigma_{a_i=v_j^i}(T))) \\ &= \phi(P_y(T)) - \sum_{j=1}^{|dom(a_i)|} \frac{|\sigma_{a_i=v_j^i}(T)|}{|T|}\phi(P_y(\sigma_{a_i=v_j^i}(T))) \end{aligned} \tag{3.19}$$

Two examples of impurity criteria built from this framework are: a) *information gain*, and; b) *gain ratio*. These approaches are used in the ID3 and C4.5 algorithms, respectively. We discuss these measures in later sections.

3.1.6 Section Summary

In the previous pages, we introduced some of the required *formal background* in decision tree induction:

1. **Induction**: we outlined the requirements and limitations of the inductive paradigm in detail.

2. **Algebra**: we provided a fully formalised account of the *multi-relational algebra*, suitable for performing proofs of the claims made in the original paper on unrealization approaches. Since multi-set algebra has markedly distinct properties from set theory and relational algebra, it is important to prove that the manipulations involved in the unrealization approach are actually permissible in a multi-relational setting.[14]

3. **Training sets**: we gave a formal account of training sets, using the multi-relational algebra introduced previously.

4. **Algorithms**: we presented a *general framework* for top-down decision tree algorithms, including the key *impurity measure* that forms the basis of the ID3 algorithm used in [20].

In the next section, we discuss the basics of the unrealization approach developed by Fong [20].

[14] A formal treatment of the multi-relational algebra is of the utmost importance, as there are key claims in [20] that were never proved.

3.2 The Unrealization Approach

In this section, we present a more rigorous version of the algorithm developed by Fong [20]. We have taken the time to augment the original treatment with more details, providing proofs of various claims that were merely asserted in Fong's thesis.

Unrealization is intended to preserve privacy by hiding the original training set from the entity conducting the data mining. Instead of releasing a modified data set (as in the various data modification schemes outlined in Section 2.3), an unrealization approach releases *unreal* data sets to the data miner. A modified decision tree algorithm allows the data miner to construct the same decision tree from the unreal data that would have been constructed from the training set with a standard ID3 algorithm. The following diagram details the high-level strategy:

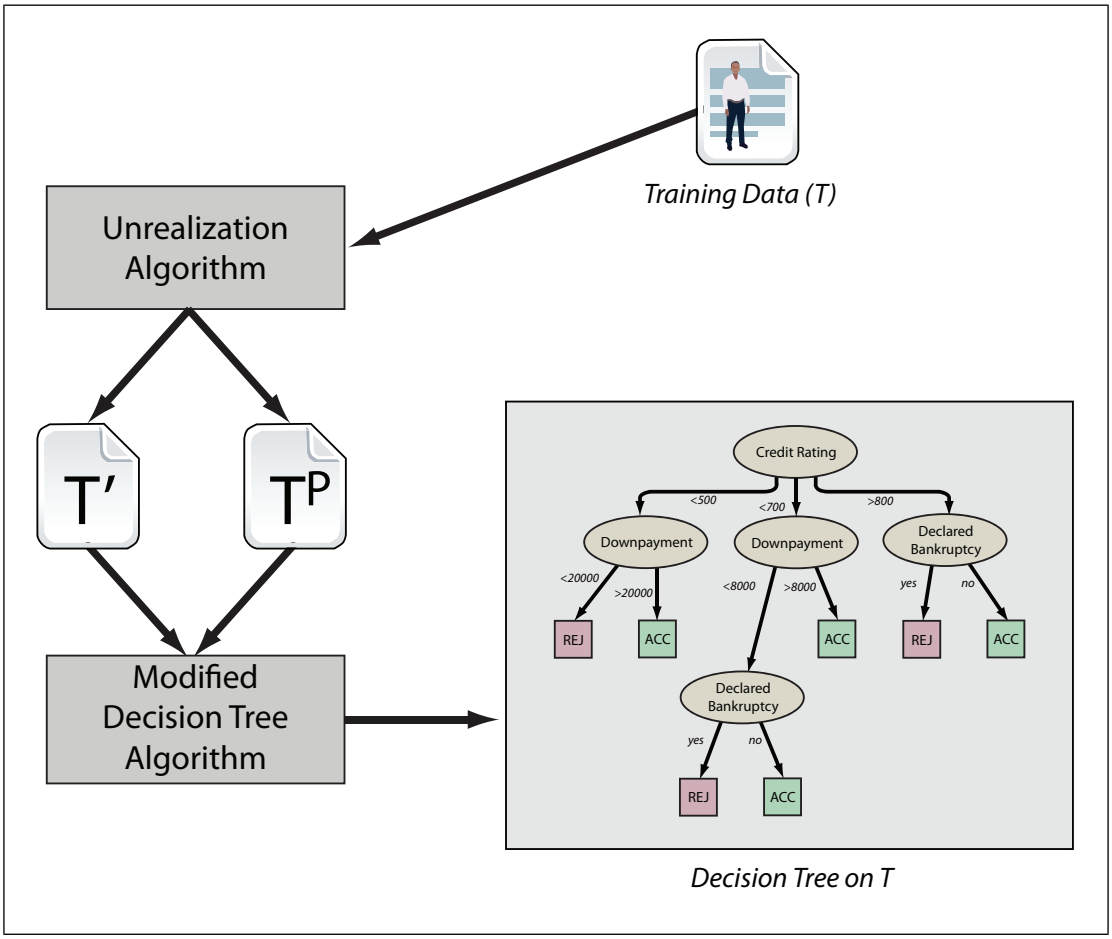

Figure 3.1: An Overview of Unrealization.

The *basic usage scenario* for the data mining is the following: imagine a data custodian (e.g., a health authority) holds a large amount of sensitive data about individuals. The custodian wishes to release information to a third party for information processing; for instance, it wishes to release information to a data analysis firm that will assemble a classification schema that determines which patients are likely to avoid paying their medical bills.

Since the data custodian does not have staff members capable of performing the analysis, it must find a method of releasing data to a third party in a manner that protects privacy interests. While it could attempt to protect the data by *contract*, contractual protections are only useful if the parties trust one another to honour them, or if defections from the contract can be detected. In the case of secondary uses of data sets, detection of unauthorised uses is next to impossible.

The unrealization approach allows the data custodian to take the database T, and process it into an unreal data set T' and a perturbing data set T^P. Instead of giving T to the third party, it may: (a) select a subset $t \subseteq T$; (b) process it, and; (c) release $t' \subseteq T'$ and $t^P \subseteq T^P$. The third party may use the modified data mining algorithm to build the same decision tree from t' and t^P that it would have constructed from t.

The problem of protecting the information in t has now become the problem of ensuring that the third party cannot use t' and t^P to make inferences about the original data subset t, apart from what it can learn by constructing a decision tree. Although a reverse transformation scheme can allow a third party to reconstruct T from T' and T^P, it only works if the data custodian has released a perturbing and unreal data set corresponding to the full contents of T. For $t' \subset T'$ and $t^P \subset T^P$ (proper subsets), the reverse transformation will not work.

In particular, the unrealization approach enables *secure multiparty computation* ("SMC") schemes. If the custodian doesn't trust one party with both data sets T' and T^P, it may give the perturbing set to one party, and the unrealized set to another. Protocols can then be developed to facilitate data mining using an exchange, where no one party is privy to both the full unrealized and perturbing data sets.

Now that we have discussed the basic usage scenarios, we turn to an examination of the method. The rest of this section consists of the following sub-sections:

1. **Multiplication and Complementation**: In this sub-section, we introduce a high-level overview of two basic manipulations involved in the unrealization approach. While some readers may be familiar with these operations, the use of examples provides a bit of insight for those individuals who are not familiar with dataset manipulations.

2. **Unrealizing Data**: This sub-section introduces the mechanism by which Fong creates secondary data sets from the incoming training set. We detail his unrealization algorithm in *recursive* form, as it is the variant presented in his thesis. We give a concrete example (spanning several pages) of this algorithm in operation. Following this exposition, we present the algorithm in *iterative* form. We prove various properties of the algorithm that were merely asserted in his thesis.[15]

3. **Tree Induction**: The next sub-section presents Fong's modified ID3 decision tree induction algorithm. We outline the algorithm, noting the subroutines that make the unrealization approach work. We demonstrate how one of these subroutines works in the unrealization setting, leaving the issue of the splitting criteria for the next sub-section.

4. **Information Gain**: This portion of the chapter contains a fair bit of mathematics, as we tighten up Fong's proofs that one can calculate information gain with the secondary (unreal) data sets, instead of with the training set.

Our aim in this section is to provide an upgraded account of the unrealization framework, supplying missing proofs and cleaning up errors and oversights. We begin this task on the next page, with a discussion of two operations that figure prominently.

[15]Some of the proofs are non-trivial, as one has to be careful to use the multi-relational algebra properly.

3.2.1 Multiplication and Complementation

Multiplication

In this part of the chapter, we provide an example of *training set multiplication* –an operation that is used frequently in the work performed in [20]. Let U be a universal instance space, $T \subseteq U$ a multi-set on relation schema $\mathcal{R}$, and $q \in \mathbb{N}$ a positive integer. Recall from equation 3.12 that $q(T)$ (the *q-multiple of* T) is a multi-set of records on $\mathcal{R}$ containing q copies of each tuple in T. That is:

$$q(T) = T \uplus_q T = \{(x, q(x \in\in T)) | x \in dom(\mathcal{R})\}$$

As an example of training set multiplication, consider the training set T below:

temperature	bear-season	precipitation	*hike*
hot	no	low	*yes*
warm	yes	low	*no*
cold	no	low	*no*

The 3-multiple of T is:

temperature	bear-season	precipitation	*hike*
hot	no	low	*yes*
hot	no	low	*yes*
hot	no	low	*yes*
warm	yes	low	*no*
warm	yes	low	*no*
warm	yes	low	*no*
cold	no	low	*no*
cold	no	low	*no*
cold	no	low	*no*

Complements

One of the key concepts in Fong's work is that of of *complementation*. Unfortunately, for reasons we discuss below, complementation in the multi-relational algebra is problematic. To avoid some of the pitfalls, we introduce a limited form of complementation.

Definition If T is a bag instance, we say that the *absolute complement* of T ($AC(T)$) is the set of all records in the universal instance space U that are not in T. From equation 3.6, we have:

$$AC(T) = U - T = \{(x, max(x \in\in U - x \in\in T, 0))|x \in dom(\mathcal{R})\} \tag{3.20}$$

Let us illustrate these concepts by means of an example. Imagine that we have the following universal instance space U:

temperature	precipitation	*hike*
hot	low	*yes*
hot	low	*no*
hot	high	*yes*
hot	high	*no*
cold	low	*yes*
cold	low	*no*
cold	high	*yes*
cold	high	*no*

Now suppose our bag T contains the first three records from U:

temperature	precipitation	*hike*
hot	low	*yes*
hot	low	*no*
hot	high	*yes*

The absolute complement $AC(T)$ is therefore:

temperature	precipitation	*hike*
hot	high	*no*
cold	low	*yes*
cold	low	*no*
cold	high	*yes*
cold	high	*no*

In this case, T was contained in U. Let us choose an example where T contains more copies of a given tuple than U. Let T' be the following multi-set:

temperature	precipitation	*hike*
hot	low	*yes*
hot	low	*no*
hot	high	*yes*
hot	high	*yes*
cold	low	*yes*
cold	low	*yes*
cold	low	*yes*

The absolute complement of T' is displayed below:

temperature	precipitation	*hike*
hot	high	*no*
cold	low	*yes*
cold	high	*yes*
cold	high	*no*

So far we have not introduced anything novel, with respect to complementation. Our next operator, however, manages to put a twist on complementation by adding in the multiplication operator.

Definition If T is a multi-set, we say that $(q(U)-T)$ (the *q-th absolute complement of* T) is the multi-set containing those tuples in the q-multiple of the universal instance space U that are not in T. That is:

$$(q(U) - T) = \{(x, max(x \in\in q(U) - x \in\in T, 0))|x \in dom(\mathcal{R})\} \tag{3.21}$$

To illustrate this concept, we return to our previous examples. We can see that $(2(U) - T)$ (the 2nd absolute complement of T) is:

temperature	precipitation	*hike*
hot	low	*yes*
hot	low	*no*
hot	high	*yes*
hot	high	*no*
hot	high	*no*
cold	low	*yes*
cold	low	*yes*
cold	low	*no*
cold	low	*no*
cold	high	*yes*
cold	high	*yes*
cold	high	*no*
cold	high	*no*

In contrast, $(2(U) - T')$ (the 2nd absolute complement of T') is:

temperature	precipitation	*hike*
hot	low	*yes*
hot	low	*no*
hot	high	*no*
hot	high	*no*
cold	low	*no*
cold	low	*no*
cold	high	*yes*
cold	high	*yes*
cold	high	*no*
cold	high	*no*

Now that the reader has a sense of these basic operations, we turn to a discussion of the unrealization approach.

3.2.2 Unrealizing Data

The main thrust of the unrealization approach outlined in Fong's thesis [20] can be described as follows:

1. **Unrealize**: Generate secondary (unreal) data sets from the training data.
2. **Alter**: Modify the ID3 algorithm to work with the secondary data sets, instead of the training data.
3. **Induce**: Use the secondary data sets to form a decision tree.

Modification of the ID3 algorithm involves two tasks:

1. Changing the *default rule*, which is activated when there is not enough information to categorize a leaf through other means.
2. Changing the *split criterion subroutine*, which chooses an attribute to split on. In this case, the split criterion routine must be changed to work with the unreal data sets, instead of the actual training data from which they were derived.

Unreal Data Sets

We begin by detailing the approach that Fong uses for creating unreal data sets. It is this step that produces privacy preservation. In later sections, we will adapt his method to the industry-standard C4.5 algorithm as explained in Quinlan [41].

As mentioned above, instead of training a classification tree from the training data T, Fong creates an extra relation instances (multi-sets) T' and T^P. These *unreal* data sets are then used by a modified data mining algorithm to construct a decision tree. We detail the unrealization algorithm below, followed by a simple illustration.

Algorithm: *UnrealizeTrainingSet*(T, U, T', T^P)	
Inputs:	T: the training bag. U: the universal instance space. T': an output bag of unrealized records. T^P: a perturbing bag of unreal records.
Outputs:	T': an bag of unrealized records. T^P: a perturbing bag of unreal records.
1:	**if** $(\|T\| = 0)$
2:	**then return** $\langle T', T^P \rangle$;
3:	Tuple $t =$ *SelectTuple*(T);
4:	**if** $((t \notin T^P)$ or $(T^P = [t]))$
5:	**then** $T^P = T^P \uplus U$;
6:	$T^P = T^P - [t]$;
7:	Tuple $s =$ *SelectMostFrequentTuple*(T^P);
8:	**return** *UnrealizeTrainingSet*$(T - [t], U, T' \uplus [s], T^P - [s])$;

Table 3.3: The Recursive Unrealization Algorithm

Constructing a dataset begins with the function call *UnrealizeTrainingSet*$(T, U, \varnothing, \varnothing)$. That is, the unreal relation instances T' and T^P are initially empty.

To ease the burden on the reader, we provide an example of the algorithm in operation. As before, imagine that we have the following universal instance space U:

$U =$

temperature	precipitation	*hike*
hot	low	*yes*
hot	low	*no*
hot	high	*yes*
hot	high	*no*
cold	low	*yes*
cold	low	*no*
cold	high	*yes*
cold	high	*no*

Suppose the training set T contains the first three records from U (with a duplicate):

$T =$

temperature	precipitation	*hike*
hot	low	*yes*
hot	low	*no*
hot	high	*yes*
hot	high	*yes*

Step 1: The invocation begins as follows (omitting U for brevity):

$T =$

temp	prec	*hike*
hot	low	*yes*
hot	low	*no*
hot	high	*yes*
hot	high	*yes*

$T' = \{\} \quad T^P = \{\}$

In this step, T is not empty. The algorithm selects (hot, low, no) in T and assigns it to variable t. Since T^P is empty, t cannot be an element of T^P. The algorithm then adds the universal instance space to T^P. It then removes t from T^P. Next, it selects the most frequent tuple in T^P. At this point, all tuples in T^P are equally likely, so assume the algorithm chooses (cold, high, yes). The algorithm then recurses.

Step 2: The first recursive call is executed, with all parameters apart from U having been modified from the previous step:

$T =$

temp	prec	*hike*
hot	low	*yes*
hot	high	*yes*
hot	high	*yes*

$T' =$

temp	prec	*hike*
cold	high	*yes*

$T^P =$

temp	prec	*hike*
hot	low	*yes*
hot	high	*yes*
hot	high	*no*
cold	low	*yes*
cold	low	*no*
cold	high	*no*

In this step, T is not yet empty. The algorithm selects the tuple (hot, high, yes) from T and assigns it to variable t. In this case, t is contained in T^P. The algorithm removes t from T^P, and assigns the most frequent tuple in T^P to the variable s. Again, the options are equally likely, so assume the algorithm chooses $s =$ (cold, low, no).

Step 3: The second recursive call is executed.

$T =$

temp	prec	*hike*
hot	low	*yes*
hot	high	*yes*

$T' =$

temp	prec	*hike*
cold	low	*no*
cold	high	*yes*

$T^P =$

temp	prec	*hike*
hot	low	*yes*
hot	high	*no*
cold	low	*yes*
cold	high	*no*

Again, T is not empty. The algorithm selects (hot, high, yes) from T and assigns it to t. In this case, t is not in T^P. The contents of the universal instance space are then added to T^P. After that is accomplished, t is removed from T^P. The algorithm selects the most frequent tuple in T^P, which is now a bit more interesting since there are four tuples with duplicate entries. Assume the algorithm picks $s =$(cold, high, no).

Step 4: The third recursive call is executed.

$T =$

temp	prec	*hike*
hot	low	yes

$T' =$

temp	prec	*hike*
cold	low	*no*
cold	high	*yes*
cold	high	*no*

$T^P =$

temp	prec	*hike*
hot	low	*yes*
hot	low	*yes*
hot	low	*no*
hot	high	*no*
hot	high	*no*
cold	low	*yes*
cold	low	*yes*
cold	low	*no*
cold	high	*yes*
cold	high	*no*

T is almost empty. There is only one choice for t from T, namely (hot, low, yes). This is obviously in T^P (in fact, twice). t is removed (once) from T^P. The algorithm chooses $s =$ (hot, high, no) as the most frequent.

Step 5: The last recursive call is executed.

$T =$

temp	prec	*hike*

$T' =$

temp	prec	*hike*
hot	high	*no*
cold	low	*no*
cold	high	*yes*
cold	high	*no*

$T^P =$

temp	prec	*hike*
hot	low	*yes*
hot	low	*no*
hot	high	*no*
cold	low	*yes*
cold	low	*yes*
cold	low	*no*
cold	high	*yes*
cold	high	*no*

The algorithm returns T' and T^P up the chain of function invocations.

If we compare the various bag instances, we notice something interesting:

$T =$

temp	prec	*hike*
hot	low	*yes*
hot	low	*no*
hot	high	*yes*
hot	high	*yes*

$T' =$

temp	prec	*hike*
hot	high	*no*
cold	low	*no*
cold	high	*yes*
cold	high	*no*

$T^P =$

temp	prec	*hike*
hot	low	*yes*
hot	low	*no*
hot	high	*no*
cold	low	*yes*
cold	low	*yes*
cold	low	*no*
cold	high	*yes*
cold	high	*no*

The training set T and unreal relation instance T' are the same size. Furthermore, there appears to be a relation between the size of T, the size of the universal instance space U, and the size of the permuting multi-set T^p. We explore these and other issues in the next section.

Analysis of the Unrealization Algorithm

In this section, we prove certain properties about the unrealization algorithm presented above. The original work in which this algorithm was presented [20] skipped over formal proofs of these properties. One of our contributions in this work is to put the earlier work on more sure footing. In order to provide a different perspective on the algorithm, we present it below in an *iterative* form, rather than the *recursive* form favoured by Fong.

Algorithm:	*UnrealizeTrainingSet*(T, U)
Inputs:	
	T: the training bag.
	U: the universal instance space.
Outputs:	
	T': an bag of unrealized records.
	T^P: a perturbing bag of unreal records.
Local Variables:	
	T': an bag of records, initially empty.
	T^P: a perturbing bag of records, initially empty.
1:	**while** $(\|T\| > 0)$
2:	{
3:	Tuple $t =$ *SelectTuple*(T);
4:	**if** $((t \notin T^P)$ **or** $(T^P - [t] = \varnothing))$
5:	$T^P \leftarrow T^P \uplus U$;
6:	$T^P \leftarrow T^P - [t]$;
7:	Tuple $s =$ *SelectTuple*(T^P);
8:	$T \leftarrow T - [t]$;
9:	$T_P \leftarrow T^P - [s]$;
10:	$T' \leftarrow T' \uplus [s]$;
11:	}
12:	return $\langle T', T^P \rangle$

Table 3.4: Unrealization algorithm in iterative form

We prove several results about this algorithm in the following pages.

Theorem 1. *UnrealizeTrainingSet(T,U) halts for all relation instances T in $|T|$ iterations of the while loop.*

Proof. The proof is exceptionally simple. We use induction on the size of T.

- For the *base case*, if $|T| = 0$ the control flow will not enter the while loop on line 1. Instead, control will jump to line 13, where the algorithm terminates. The number of iterations of the while loop is $0 = |T|$.
- For the *inductive step*, assume that the while loop terminates for sets T of size $k \geq 0$ in $|T|$ iterations. Show that it terminates for sets T of size $k+1$ in $k+1$ iterations of the loop.

  Assume $|T| = k + 1$. Then the control flow enters the while loop. A tuple t is selected from T on line 3, and it is subsequently deleted from both T^P and T. No other modifications are made to T. At the end of this iteration of the while loop, the size of T is now k. By inductive assumption, the loop will terminate for T of this size in k iterations. Therefore the algorithm terminates for sets of size $k + 1$ in $k + 1$ iterations of the loop.

□

Lemma 2. *When the algorithm finishes and returns a value $\langle T', T^P \rangle$, the size of the unreal relation instance T' is $|T|$.*[16]

Proof. We use theorem 1. The algorithm halts for all inputs T in $|T|$ iterations of the loop. The bag instance T' is initially empty. The only update to T' occurs within the loop at line 11, where a tuple is inserted into T'. Since T' is a multi-set (and not a set), each insertion operation increases its size by 1. For inputs of size $|T|$, there are $|T|$ iterations of the loop, and therefore $|T|$ insertions into T'. This entails that $|T| = |T'|$. □

[16]Note that we are not claiming that this is an invariant. In fact, the size of T' is quite different from that of T during the various iterations of the 'while' loop.

Theorem 2. *When the algorithm finishes and returns a value* $\langle T', T^P \rangle$, $T \uplus T' \subseteq q(U)$ *for some* $q \in \mathbb{N}$. *That is, the concatenation of the training set and unreal data set* T' *are subsets of some* q*-multiple of the universal instance space.*

Proof. Recall from theorem 1 that the while loop executes $|T|$ times, with the assignment of the universal instance space U to T^P on line 5 occurring $q \leq |T|$ times. Every tuple in T' comes from T^P, and T^P only receives tuples from being assigned a copy of U on line 5. Hence, every tuple in T' comes from a q-multiple of U, and $T' \subseteq q(U)$.

To prove $T \subseteq q(U)$, see the guard condition on line 4. Every tuple $t \in T$ selected in an iteration of the loop has a corresponding tuple in T^P. (This tuple was either in T^P already, or it is supplied on line 5 by an infusion of tuples from U). Since T^P draws its tuples exclusively from U, it follows that $T \subseteq q(U)$. Hence (by the definition of $\uplus$) $T \uplus T' \subseteq q(U)$. □

Lemma 3. *When the algorithm finishes and returns a value* $\langle T', T^P \rangle$, $T' \uplus T^P \subseteq q(U)$ *for some* $q \in \mathbb{N}$.

Proof. The proof is similar to theorem 2. T' receives all of its tuples from T^P. In turn, T^P receives all of its tuples from infusions of U on line 5. Thus, every tuple in $T' \uplus T^P$ comes from a q-multiple of U. □

Our next result (appearing on the following page) is fundamental to the work performed in Fong's thesis.

Theorem 3. *After the execution of the algorithm, the input set T is the q-absolute complement of $T \uplus T^P$ for some $q \in \mathbb{N}$.*[17] *That is,*

$$T = q(U) - (T \uplus T^P)$$

Proof. So we wish to show that for some $q \in \mathbb{N}$,

$$T = q(U) - (T' \uplus T^P)$$

We begin to prove this claim by showing that $T^P = q(U) - (T \uplus T')$. As we know from Theorem 1, the loop is executed $|T|$ times. Inside the loop, T^P receives tuples in only one instance - namely, on line 5. Since T^P begins the algorithm initialized to the empty set, the only way it can receive tuples is through this assignment. Instead of trying to determine tight bounds on the number of times line 5 is executed, we merely assume that it is reached by the control flow $q \leq |T|$ times. This means that T_P receives q copies of U during execution of the algorithm.

Despite this result, T^P also loses tuples through lines 6 and 9. Each line is executed $|T|$ times. Examining line 6, we see that a unique tuple t of T is removed from T^P each iteration. (The guard condition on line 4 ensures that there is such a t to remove from T^P). Hence, the entire contents of the training set are withdrawn from T^P during the course of the algorithm. Additionally, $|T|$ arbitrary tuples are removed from T^P and placed in T'. This means that T^P contains q copies of U, minus the contents of T and T'. So we have shown that $T^P = q(U) - (T \uplus T')$.

If we were working in set theory, it would be easy to prove Lemma 3 from this result. We would simply proceed as follows:

$$\begin{aligned} T^P &= q(U) - (T \cup T') \\ T^P \cup (T \cup T') &= q(U) \\ T \cup (T' \cup T^P) &= q(U) \\ T &= q(U) - (T' \cup T^P) \end{aligned}$$

(QED)

Sadly, we cannot assume that the same laws hold for multi-sets. As we mentioned above, multi-relational algebra lacks many of the properties attributed to sets.

[17] This claim is asserted by Fong on p.60. However, it is not proven in that work.

A simple example, based on the putative proof above, will suffice to convince the reader of the danger. In this example, we prove that it is not a law of the multi-relational algebra that $A = (A - B) \uplus B$. Let multi-set $A = [a, a, b, b, c]$ and multi-set $B = [b, b, b]$. Consider $(A - B)$:

$$\begin{aligned}(A - B) &= \{(x, max(x \in\in A - x \in\in B, 0)\} \\ &= \{(a, max(2 - 0, 0)), (b, max(2 - 3, 0)), (c, max(1 - 0, 0))\} \\ &= \{(a, 2), (b, 0), (c, 1)\} \\ &= [a, a, c]\end{aligned}$$

Now calculate $(A - B) \uplus B$:

$$\begin{aligned}(A - B) \uplus B &= \{(x, x \in\in (A - B) + x \in\in B\} \\ &= \{(a, 2 + 0), (b, 0 + 3), (c, 1 + 0)\} \\ &= \{(a, 2), (b, 3), (c, 1)\} \\ &= [a, a, b, b, b, c]\end{aligned}$$

As the reader can see, $((A - B) \uplus B) \neq A$. We interrupt our proof of lemma 3 in order to prove several results that will enable us to get past this obstacle. □

Lemma 4. *Let A and B be relation instances on relation schema $\mathcal{R}$. Then,*

$$(A \uplus B) - B = A$$

Proof. We prove this claim directly:

1)	$A \uplus B = \{(x, x \in\in A + x \in\in B) \mid x \in dom(\mathcal{R})\}$	(def. $\uplus$)
2)	$(A - B) = \{(x, max(x \in\in A - x \in\in B, 0)) \mid x \in dom(\mathcal{R})\}$	(def. $-$)
3)	$(A \uplus B) - B = \{(x, max(x \in\in (A \uplus B) - x \in\in B, 0)) \mid x \in dom(\mathcal{R})\}$	(subst., 2)
4)	$= \{(x, max((x \in\in A + x \in\in B) - x \in\in B, 0)) \mid x \in dom(\mathcal{R})\}$	(subst.,1,3)
5)	$= \{(x, max(x \in\in A, 0)) \mid x \in dom(\mathcal{R})\}$	(line 6)
7)	$= \{(x, x \in\in A) \mid x \in dom(\mathcal{R})\}$	($x \in\in A \geq 0$)
8)	$= A$	(def. $\uplus$)

□

Lemma 5. *Let A and B be relation instances on relation schema $\mathcal{R}$. If $B \subseteq A$ then,*

$$((A - B) \uplus B) = A$$

Proof. We prove this result directly. Recall that $B \subseteq A$ iff $(\forall x \in dom(\mathcal{R}))(x \in\in B \leq x \in\in A)$. Since $x \in\in B \leq x \in\in A$, we know that $0 \leq x \in\in A - x \in\in B$. This means that $max(x \in\in A - x \in\in B, 0) = (x \in\in A - x \in\in B)$. So $(A - B) = \{(x, x \in\in A - x \in\in B) | x \in dom(\mathcal{R})\}$.

From the definition of $A \uplus B$ we have:

1)	$x \in\in B \leq x \in\in A$		(def. $\subseteq$)
2)	$0 \leq x \in\in A - x \in\in B$		(line 1)
3)	$(A - B) = \{(x, max(x \in\in A - x \in\in B, 0) \mid x \in dom(\mathcal{R})\}$		(def. $-$)
4)	$= \{(x, x \in\in A - x \in\in B) \mid x \in dom(\mathcal{R})\}$		(line 2)
5)	$(A - B) \uplus B = \{(x, x \in\in (A - B) + x \in\in B) \mid x \in dom(\mathcal{R})\}$		
6)	$= \{(x, (x \in\in A - x \in\in B) + x \in\in B) \mid x \in dom(\mathcal{R})\}$		(line 4)
7)	$= \{(x, x \in\in A) \mid x \in dom(\mathcal{R})\}$		
8)	$= A$		

□

Lastly, we prove two useful properties of the multi-relational algebra:

Lemma 6. *(Commutativity of Concatenation) Let A and B be relation instances on relation schema $\mathcal{R}$. Then,*

$$A \uplus B = B \uplus A$$

Proof. This result follows from the definition of $\uplus$ and commutativity of addition:

$A \uplus B =$	$\{(x, x \in\in A + x \in\in B) \mid x \in dom(\mathcal{R})\}$	(def. of $\uplus$)
$=$	$\{(x, x \in\in B + x \in\in A) \mid x \in dom(\mathcal{R})\}$	(assoc. of $+$)
$=$	$B \uplus A$	(def. of $\uplus$)

□

Lemma 7. *(Associativity of Concatenation) Let A, B and C be relation instances on relation schema $\mathcal{R}$. Then*

$$(A \uplus B) \uplus C = A \uplus (B \uplus C)$$

Proof. We prove the result directly, using the definition of $\uplus$ and associativity of $+$:

$$\begin{aligned}
A \uplus B &= \{(x, x \in\in A + x \in\in B) | x \in dom(\mathcal{R})\} && \text{(def. of } \uplus) \\
(A \uplus B) \uplus C &= \{(x, x \in\in (A \uplus B) + x \in\in C) | x \in dom(\mathcal{R})\} && \text{(def. of } \uplus) \\
&= \{(x, (x \in\in A + x \in\in B) + x \in\in C) | x \in dom(\mathcal{R})\} && \text{(line) 1} \\
&= \{(x, x \in\in A + (x \in\in B + x \in\in C)) | x \in dom(\mathcal{R})\} && \text{(assoc. of } +) \\
&= \{(x, x \in\in A + x \in\in (B \uplus C)) | x \in dom(\mathcal{R})\} && \text{(def. of } \uplus) \\
&= A \uplus (B \uplus C) && \text{(def. of } \uplus)
\end{aligned}$$

□

Returning to the proof of lemma 3, we had proven that $T^P = q(U) - (T \uplus T')$. We had, of course, been seeking to prove that $T = q(U) - (T' \uplus T^P)$. Given some of our new results, we can finish this proof.

1)	$T^P = q(U) - (T \uplus T')$	(proved in first half)
2)	$T^P \uplus (T \uplus T') = (q(U) - (T \uplus T')) \uplus (T \uplus T')$	(line 1)
3)	$(T \uplus T') \subseteq q(U)$	(lemma 2)
4)	$q(U) = (q(U) - (T \uplus T')) \uplus (T \uplus T')$	(lemma 5, subst.)
5)	$q(U) = T^P \uplus (T \uplus T')$	(line 2 and 4)
6)	$q(U) = T^P \uplus (T' \uplus T)$	(comm., line 5)
7)	$q(U) = (T^P \uplus T') \uplus T$	(assoc., line 6)
8)	$q(U) = (T' \uplus T^P) \uplus T$	(comm., line 7)
9)	$q(U) = T \uplus (T' \uplus T^P)$	(comm., line 8)
10)	$q(U) - (T' \uplus T^P) = (T \uplus (T' \uplus T^P)) - (T' \uplus T^P)$	(line 9)
11)	$q(U) - (T' \uplus T^P) = T$	(lemma 4 and line 10)

Hence, we have proved Theorem 3. Although perhaps tedious, a rigorous proof of this result is central to the work performed in Fong's paper. In particular, this result allows us to prove the key claim that makes unrealization possible. First, unfortunately, we introduce two more useful results about the multi-relational algebra.

Lemma 8. *Let A and B be multi-sets on relation schema $\mathcal{R}$. If $B \subseteq A$, then*

$$|A - B| = |A| - |B|$$

Proof. We use direct proof.

1)	$B \subseteq A$ iff $(\forall x \in dom(\mathcal{R}))(x \in\in A \leq x \in\in B)$	(def. $\subseteq$)
2)	$A - B = \{(x, max(x \in\in A - x \in\in B, 0))\|x \in dom(\mathcal{R}\}$	(def. $-$)
3)	$= \{(x, x \in\in A - x \in\in B)\|x \in dom(\mathcal{R}\}$	($B \subseteq A, 1$)
4)	$\|A - B\| = \sum_{x \in dom(\mathcal{R})} x \in\in (A - B)$	(def. of size)
5)	$= \sum_{x \in dom(\mathcal{R})} x \in\in A - x \in\in B$	(line 3)
6)	$= \sum_{x \in dom(\mathcal{R})} x \in\in A - \sum_{x \in dom(\mathcal{R})} x \in\in B$	
7)	$= \|A\| - \|B\|$	(def. of size)

□

Lemma 9. *If A and B are multi-sets on relation schema $\mathcal{R}$, then*

$$|A \uplus B| = |A| \uplus |B|$$

Proof. We use direct proof.

1)	$A \uplus B = \{(x, x \in\in A + x \in\in B)\|x \in dom(\mathcal{R}\}$	(def. $\uplus$)
2)	$\|A \uplus B\| = \sum_{x \in dom(\mathcal{R})} x \in\in (A \uplus B)$	(def. of size)
3)	$= \sum_{x \in dom(\mathcal{R})} x \in\in A + x \in\in B$	(line 3)
4)	$= \sum_{x \in dom(\mathcal{R})} x \in\in A + \sum_{x \in dom(\mathcal{R})} x \in\in B$	(line 5)
5)	$= \|A\| + \|B\|$	(def. of size)

□

Theorem 4. *Let T be the training set, and $\langle T', T^P \rangle$ be the pair returned from the unrealization algorithm. Then:*

$$|q(U)| = 2|T| + |T^P| = 2|T'| + |T^P|$$

Proof.

1)	$T = q(U) - (T' \uplus T^P)$	(Theorem 3)
2)	$\lvert T\rvert = \lvert q(U) - (T' \uplus T^P)\rvert$	(line 1)
3)	$(T' \uplus T^P) \subseteq q(U)$	(lemma 3)
4)	$\lvert T\rvert = \lvert q(U)\rvert - \lvert T' \uplus T^P\rvert$	(lemma 8; line 2,3)
5)	$\lvert T\rvert = \lvert q(U)\rvert - (\lvert T'\rvert + \lvert T^P\rvert)$	(lemma 9; line 4)
6)	$\lvert T\rvert = \lvert q(U)\rvert - \lvert T'\rvert - \lvert T^P\rvert$	(line 5)
7)	$\lvert T\rvert = \lvert T'\rvert$	(lemma 2)
8)	$\lvert T\rvert = \lvert q(U)\rvert - \lvert T\rvert - \lvert T^P\rvert$	(line 6,7)
9)	$2\lvert T\rvert + \lvert T^P\rvert = \lvert q(U)\rvert$	(line 8)

□

In the next section, we show how the modified ID3 algorithm works. The results introduced in this section will find immediate application, when we explain the process by which unrealization preserves privacy.

3.2.3 Tree Induction

Pseudocode for the standard ID3 decision tree induction algorithm appears below.

Algorithm:	*ID3-GrowTree*(T,A,y, default)
Inputs:	T: the training set. A: the list of attributes. y: the target attribute. default: the default value.
Outputs:	D: a decision tree.

```
1:  Tree d = new Tree;
2:  if (|T| = 0 or |A| = 0 )
3:   {
4:     d.type = "leaf";
5:     d.class = default;
6:     return d;
7:   }
8:  default ← MajorityValue(T);
9:  if (H(y, T) = 0)
10:  {
11:    d.type = "leaf";
12:    d.class = default;
13:    return d;
14:  }
15: Attribute b ← ChooseBestAttribute(A, T);
16: d.type = "internal";
17: d.class = b;
18: for each value v^j of b:
19:  {
20:    Tree s = ID3-GrowTree(σ_{b=v_j}(T), A − {b}, y, default);
21:    d.AddSubtreeWithLabel(s, v_j);
22:  }
23: return d;
```

Table 3.5: ID3 Algorithm

Fong's variant is displayed below:

Algorithm:	*GrowFTree*(T', T^P, A, size, default)
Inputs:	T': the unreal training set. T^P: the unreal perturbing set. A: the list of attributes. size: the size of $q(U)$ for these unreal sets. default: the default value.
Outputs:	D: a decision tree.

```
1:  Tree d = new Tree;
2:  if (|T ⊎ T^P| = 0 or |A| = 0 )
3:   {
4:     d.type = "leaf";
5:     d.class = default;
6:     return d;
7:   }
8:  default ← MinorityValue(T' ⊎ T^P);
9:  if (H(y, q(U) − (T' ⊎ T^P)) = 0)
10:  {
11:    d.type = "leaf";
12:    d.class = default;
13:    return d;
14:  }
15: Attribute b ← ChooseBestAttributeUnreal(A, size, T', T^P);
16: d.type = "internal";
17: d.class = b;
18: for each value v^j of b:
19:  {
20:    Tree s = GrowFTree(σ_{b=v_j}(T'), σ_{b=v_j}(T^P), A − {b}, size/|dom(b)|, default);
21:    d.AddSubtreeWithLabel(s, v_j);
22:  }
23: return d;
```

Table 3.6: Fong's Modified ID3 Algorithm

We have called the algorithm *GrowFTree* in reference to Fong. A summary of the major changes between it and the standard ID3 algorithm appears below:

1. **Input Sets**: The Fong variant receives input sets T' and T^P, instead of T. In this way, a decision tree can be formed on the basis of unreal data sets, instead of on the original (sensitive) data.

2. **Size**: The Fong variant passes the size of the universal instance space corresponding to the current unreal sets T' and T^P. Since these are whittled down by one dimension in every recursive call, the size parameter is also reduced correspondingly in line 20.

3. **Majority vs. Minority**: On line 8, we see that the *MajorityValue*(T) function call has been replaced with *MinorityValue*$(T' \uplus T^P)$. We will explain this majority/minority switch below.

4. **Entropy**: On line 9, the algorithm calculates the entropy of target attribute y on $q(U) - (T' \uplus T^P)$. In the standard algorithm (table 3.2), this is calculated on T. However, it turns out that $\mathcal{H}(y, q(U) - (T' \uplus T^P)) = \mathcal{H}(y, T)$. We already proved this fact in Theorem 3, where we showed that $T = q(U) - (T' \uplus T^P)$. Hence we can calculate the entropy for y on T without any access to T.

5. **Choosing the Best Attribute**: The Fong variant has a new function, *ChooseBestAttributeUnreal*$(A, \text{size}, T', T^P)$. This replaces the *ChooseBestAttribute*(A, T) function of the standard ID3 algorithm. As in the case of entropy, the idea is to calculate the best attribute (using the *information gain* criterion) without actually using the sensitive training data T. Proving that this can be done is the goal of the next sub-section.

The basic idea of the unrealization approach developed in [20] is to calculate the same decision tree that would be calculated by the ID3 algorithm, but with the use of the unreal data sets T' and T^P. In Fong's words, the records in T' and T^P are unreal individually, but meaningful when they are used together.

Before turning to an examination of how we can calculate the best splitting attribute from the unreal data sets, we explain the majority/minority issue noted above.

Majority vs. Minority

As we noted on the previous page, on line 8 of the *GrowFTree* algorithm, we see that the *MajorityValue*(T) function call has been replaced with *MinorityValue*($T' \uplus T^P$). In fact, both of these functions accomplish the same goal: to select the least frequent value in the domain of the decision attribute y.

We can show this easily, provided we prove a few facts about the multi-relational algebra:

Lemma 10. *Let A and B be multi-sets with $B \subseteq A$, and let ϕ be a predicate. Then:*

$$\sigma_\phi(A - B) = \sigma_\phi(A) - \sigma_\phi(B) \tag{3.22}$$

Proof. We use direct proof. Recall from equation 3.7 that $\sigma_\phi((A - B)) = \{(x, x \in\in (A-B))|x \in dom(\mathcal{R}) \wedge \phi(x)\} \cup \{(x, 0)|(x \notin dom(\mathcal{R}) \vee (x \in dom(\mathcal{R}) \wedge \neg\phi(x))\}$. Recall from equation 3.6 that $(A-B) = \{(x, max(x \in\in A - x \in\in B, 0)|x \in dom(\mathcal{R}\}$. Since $B \subseteq A$, we know that $(A - B) = \{(x, x \in\in A - x \in\in B)| \in dom(\mathcal{R}\}$. But then, $\sigma_\phi((A-B)) = \{(x, x \in\in A-x \in\in B)|x \in dom(\mathcal{R}) \wedge \phi(x)\} \cup \{(x, 0)|(x \notin dom(\mathcal{R}) \vee (x \in dom(\mathcal{R}) \wedge \neg\phi(x))\}$. We can form this same expression from $\sigma_\phi(A) - \sigma_\phi(B)$.[18] □

Lemma 11. *Let A and B be multi-sets on relation schema $\mathcal{R}$, and let ϕ be a predicate. Then:*

$$\textit{If } B \subseteq A \textit{ then } \sigma_\phi(B) \subseteq \sigma_\phi(A) \tag{3.23}$$

Proof. We prove this claim by contradiction. Assume $B \subseteq A$ but $\sigma_\phi(B) \subsetneq \sigma_\phi(A)$. Then by the former, for all $t \in dom(\mathcal{R})$, $t \in\in B \leq t \in\in A$. But by the latter, we know that there is a $s \in dom(\mathcal{R})$ such that $s \in\in \sigma_\phi(B) > s \in\in \sigma_\phi(A)$. Since $s \in \sigma_\phi(B)$, we know three things: $s \in dom(\mathcal{R})$, $s \in\in B = s \in\in \sigma_\phi(B)$ and $\phi(s)$. It follows that $s \in\in B > s \in\in \sigma_\phi(A)$. Since $s \in dom(\mathcal{R})$ and $\phi(s)$, $s \in\in \sigma_\phi(A) = s \in\in A$. So $s \in\in B > s \in\in A$. This contradicts the claim that for all $t \in dom(\mathcal{R})$, $t \in\in B \leq t \in\in A$. □

[18] This claim can also be proved by an explicit proof using containment in both directions. Due to brevity, we do not reproduce this proof here.

Lemma 12. *Let A and B be multi-sets on relation schema $\mathcal{R}$, which has attributes $A = \{a_1, a_2, \ldots, a_n\}$. Let ϕ be a predicate, and let $B \subseteq A$. Then:*

$$|\sigma_\phi(A - B)| = |\sigma_\phi(A)| - |\sigma_\phi(B)|$$

Proof. This claim is easily proven, with the help of previous results.

1)	$\sigma_\phi(A - B) = \sigma_\phi(A) - \sigma_\phi(B)$	(lemma 10)
2)	$\lvert\sigma_\phi(A - B)\rvert = \lvert\sigma_\phi(A) - \sigma_\phi(B)\rvert$	(line 1)
3)	$\sigma_\phi(B) \subseteq \sigma_\phi(A)$	(lemma 11)
4)	$\lvert\sigma_\phi(A) - \sigma_\phi(B)\rvert = \lvert\sigma_\phi(A)\rvert - \lvert\sigma_\phi(B)\rvert$	(lemma 8)
5)	$\lvert\sigma_\phi(A - B)\rvert = \lvert\sigma_\phi(A)\rvert - \lvert\sigma_\phi(B)\rvert$	

□

Returning to our discussion of the majority/minority issue, we note that the unreal data sets are a subset of the universal instance space. So we have:

1)	$T = q(U) - (T' \uplus T^P)$	(Theorem 3)
2)	$\sigma_{y=k}(T) = \sigma_{y=k}(q(U) - (T' \uplus T^P))$	
3)	$\lvert\sigma_{y=k}(T)\rvert = \lvert\sigma_{y=k}(q(U) - (T' \uplus T^P))\rvert$	
4)	$\lvert\sigma_{y=k}(T)\rvert = \lvert\sigma_{y=k}(q(U))\rvert - \lvert\sigma_{y=k}(T' \uplus T^P))\rvert$	(lemma 12)
5)	$\lvert\sigma_{y=k}(T)\rvert + \lvert\sigma_{y=k}(T' \uplus T^P))\rvert = \lvert\sigma_{y=k}(q(U))\rvert$	

Fong notes that $|\sigma_{y=k}(q(U))|$ (the number of tuples in which the target attribute y takes the value v in the q-multiple of the universal instance space) can be treated as a constant, since it does not vary from training set to training set. Therefore, the default decision value for T (which is normally computed by a majority function) can be computed from $(T' \uplus T^P)$ using a minority function.[19]

In the next section, we turn our attention to the last remaining issue in understanding why Fong's modified ID3 algorithm produces the same tree (using T' and T^P) that the standard ID3 algorithm does using T –namely, the calculation of the *best splitting attribute*.

[19] See [20] at p.60 for details.

3.2.4 Information Gain

The Information Gain Metric

Information gain is an *impurity measure*[20] that uses information entropy, as defined above in section 3.1.5. As described in Fong, there is a relationship between information entropy and the selection criteria of a test attribute. One can view a classification as a type of event that corresponds to a discrete random variable. If the outcome of a classification is meaningful, then some information content should be delivered from the event. This heuristic entails that an attribute should be picked as a splitting criteria if and only if it makes the decision more certain.

Recalling Section 3.1.5, one can define the *goodness-of-split* $\Delta\Phi(a_i, T)$ due to the attribute a_i as the reduction in impurity of the target attribute y after partitioning the training set T according to the values $\{v_1^i, v_2^i, \ldots, v_j^i\}$ of a_i:

$$\begin{aligned}
\Delta\Phi(a_i, T) &= \text{(impurity on T)} - \text{(impurity on partitions of T induced by } a_i) \\
&= \phi(P_y(T)) - \sum_{j=1}^{|dom(a_i)|} P(a_i = v_j^i)\phi(P_y(\sigma_{a_i=v_j^i}(T))) \\
&= \phi(P_y(T)) - \sum_{j=1}^{|dom(a_i)|} \frac{|\sigma_{a_i=v_j^i}(T)|}{|T|}\phi(P_y(\sigma_{a_i=v_j^i}(T)))
\end{aligned}$$

We can create the information gain metric from this definition by using *entropy* as our impurity measure ϕ. Recall from equation 3.17 that the *information entropy* $\mathcal{H}$ associated with a probability density function $f_{\mathcal{X}}$ (for a discrete random variable $\mathcal{X}$ with outcomes $\{k_1, k_2, \ldots, k_n\}$) is:

$$\begin{aligned}
\mathcal{H}(\mathcal{X}) &= \text{InformationContent}(f_{\mathcal{X}}) \\
&= \text{InformationContent}(P(k_1), P(k_2), \ldots, P(k_n)) \\
&= -\sum_{i=1}^{n} P(\mathcal{X} = k_i)\log_2\left(P(\mathcal{X} = k_i)\right)
\end{aligned}$$

[20] See Section 3.1.5

In the context of training sets T, the probability distribution P is calculated directly from the training data. The entropy of target attribute y on training set T is therefore:

$$\mathcal{H}(y,T) = -\sum_{v \in dom(y)} \left(\frac{|\sigma_{y=v}(T)|}{|T|} \log_2 \frac{|\sigma_{y=v}(T)|}{|T|} \right) \tag{3.24}$$

Using entropy on a training set as our impurity measure,[21] the *information gain* associated with splitting the training set T by attribute a_i is:

$$\begin{aligned}
\mathrm{IG}(a_i,T) &= \mathcal{H}(y,T) - \mathcal{H}(y,T|a_i) \\
&= \mathcal{H}(y,T) - \sum_{j=1}^{|dom(a_i)|} \frac{|\sigma_{a_i=v_j^i}(T)|}{|T|} \mathcal{H}(y, \sigma_{a_i=v_j^i}(T)) \\
&= -\sum_{v \in dom(y)} \left(\frac{|\sigma_{y=v}(T)|}{|T|} \log_2 \frac{|\sigma_{y=v}(T)|}{|T|} \right) \\
&\quad - \sum_{j=1}^{|dom(a_i)|} \frac{|\sigma_{a_i=v_j^i}(T)|}{|T|} \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v}(\sigma_{a_i=v_j^i}(T))|}{|\sigma_{a_i=v_j^i}(T)|} \log_2 \frac{|\sigma_{y=v}(\sigma_{a_i=v_j^i}(T))|}{|\sigma_{a_i=v_j^i}(T)|} \right)
\end{aligned}$$

This is the information gain measure used in the ID3 algorithm, which was utilized by Fong in the work that this thesis aims to extend. When using entropy as the impurity measure, one can interpret the components as follows:

1. $\mathcal{H}(y,T)$ is the information content associated with treating the value of y as the outcome of a random experiment on data set T. It is the information content of y before any choice of a splitting attribute is chosen.

2. $\mathcal{H}(y,T|a_i)$ is the information content associated with treating the value of y as the outcome of a random experiment on data set T, but conditioned on the partitions induced by the k_i outcomes of attribute a_i. It is the information content of y given the choice of splitting attribute a_i.

The higher the information gain of an attribute test, the lower the uncertainty contained in its decision.

[21] A proof that entropy is a satisfactory impurity measure, according to the criteria listed in Section 3.1.5, is beyond the scope of this work.

The Goal of Unrealization

The main goal in using information gain as a metric for unrealized data sets is to show that we can calculate $\mathcal{H}(y, T)$ and $\mathcal{H}(y, T|a_i)$ by using unreal data sets T' and T^P alone. If we can achieve this, we can effectively complete the decision tree induction algorithm in table 3.6. At that point, we will be able to induce the same decision tree that ID3 would construct for training set T, but with the use of unreal sets T' and T^P.

In order to accomplish this, we need to show that we can compute not just the entropies $\mathcal{H}(y, T)$ and $\mathcal{H}(y, T|a_i)$ for the training set T, but the entropies $\mathcal{H}(y, \sigma_\phi(T))$ and $\mathcal{H}(y, \sigma_\phi(T)|a_i)$ for any predicate-based selection operation σ_ϕ.

The reason is that a top-down decision tree algorithm such as ID3 involves recursive calls that impose successive selections on the training set T. For instance, the training data passed through successive invocations of the recursive procedure might look like:

Recursive Depth 0 :	T
Recursive Depth 1 :	$\sigma_{a_1=v_2^1}(T)$
Recursive Depth 2 :	$\sigma_{a_1=v_2^1 \wedge a_5=v_4^5}(T)$
Recursive Depth 3 :	$\sigma_{a_1=v_2^1 \wedge a_5=v_4^5 \wedge a_9=v_1^9}(T)$
Recursive Depth 4 :	$\sigma_{a_1=v_2^1 \wedge a_5=v_4^5 \wedge a_9=v_1^9 \wedge a_2=v_3^2}(T)$

It is not enough merely to be able to compute $\mathcal{H}(y, T)$ and $\mathcal{H}(y, T|a_i)$ from T' and T^P. We must be able to compute these values for all possible selections on the training set T. This is the problem to which we now turn.

Information Gain on the Universal Instance Space

We begin by showing that the universal instance space has unique properties with respect to entropy. In particular, we show that:

$$\begin{aligned}
\mathcal{H}(y, U) &= -\log_2\left(\frac{1}{k_y}\right) \\
\mathcal{H}(y, q(U)) &= -\log_2\left(\frac{1}{k_y}\right) \\
\mathcal{H}(y, \sigma_{a=w}(q(U))) &= -\log_2\left(\frac{1}{k_y}\right) \\
(\forall a_i \in A)\mathrm{IG}(a_i, T) &= 0
\end{aligned}$$

That is, no information about y can be gleaned from the universal instance space, making the occurrence of y equivalent to a fair coin toss, from an information theory standpoint. Furthermore, the information gain expected from choosing any attribute a_i as a splitting attribute on the universal instance space U is precisely zero.

We can explain these results from a different perspective. Consider an attribute $a \in A$ with domain $\{v_1^a, v_2^a, \ldots, v_n^a\}$. The universal instance space U contains all possible tuples of attributes. As a result, the following equation holds:

$$(\forall i \in [1, n])(\forall j \in [1, n])\left(\frac{|\sigma_{a=v_i^a}|}{|U|} = \frac{|\sigma_{a=v_j^a}|}{|U|}\right)$$

That is, the values for an attribute are present in equal proportions in the universal instance space. This means that there is no good choice of test attribute. What this shows is that the information content of an attribute in the universal set is maximal, corresponding to a toss of a fair coin. We had no reason to prefer any particular value for the attribute, so an outcome indicating a value is quite informative.

The key insight for the unrealization approach is this: the closer a training set gets to a universal set (or a q-multiple thereof) the smaller becomes the quantity of information content of an attribute that can be retrieved from another.

As usual, we begin by proving some helpful results, compensating for the lack of proofs in the original work.

Lemma 13. *Let U be the universal instance space corresponding to attributes $A = \{a_1, a_2, \ldots, a_n\}$. Then the size of U is:*

$$|U| = |dom(a_1)||dom(a_2)| \ldots |dom(a_n)|\} = \prod_{i=1}^{n} |dom(a_i)|$$

Proof. This result follows from definition of the domain of U as $dom(a_1) \times dom(a_2) \times \cdots \times (a_n)$. The finite cardinality of a Cartesian product is the product of the finite cardinalities of its components. Alternatively, one may use a counting argument, wherein one has $|dom(a_i)|$ choices for each attribute i. □

Lemma 14. *Let U be the universal instance space corresponding to attributes $A = \{a_1, a_2, \ldots, a_n\}$. Then the size of the projection $\sigma_{a_i=v_j^i}(U)$ is:*

$$|\sigma_{a_i=v_j^i}(U)| = \frac{|U|}{|dom(a_i)|} = \frac{\prod_{j=1}^{n} |dom(a_j)|}{|dom(a_i)|}$$

Proof. This result can be proved by a counting argument. A projection that fixes a value for a single attribute effectively reduces the dimension of the universal instance space by 1. For each attribute $a_j \neq a_i$, we have $|dom(a_j)|$ values. This gives us $\prod_{a_j \in A \wedge a_j \neq a_i} |dom(a_j)| = \frac{\prod_{j=1}^{n} |dom(a_j)|}{|dom_(a_i)|}$. □

Lemma 15. *Let U be the universal instance space corresponding to attributes $A = \{a_1, a_2, \ldots, a_n\}$. Let $B = \{b_1, b_2, \ldots, b_m\} \subseteq A$ be a subset of the attributes. Then the size of the projection $\sigma_{b_1=v_1 \wedge b_2=v_2 \wedge \cdots \wedge b_m=v_m}(U)$ is:*

$$|\sigma_{b_1=v_1 \wedge b_2=v_2 \wedge \cdots \wedge b_m=v_m}(U)| = \frac{\prod_{a \in A} |dom(a)|}{\prod_{b \in B} |dom(b)|}$$

Proof. As above, we use a counting argument. Each attribute in b effectively reduces the dimension of the universal instance space by 1. The remaining choices are made on the attributes not in b. That is, $|U| = \prod_{a \in (A-B)} |dom(a)|$. Multiplying this fraction by $\frac{\prod_{b \in B} |dom(b)|}{\prod_{b \in B} |dom(b)|}$ gives the result above. □

Lemma 16. *Let y be the target attribute, with $dom(y) = \{v_1^y, v_2^y, \ldots, v_{k_y}^y\}$. Let $A = \{a_1, a_2, \ldots, a_n\}$ be the complete set of attributes (including the target y) with domain $dom(a_i) = \{v_1^i, v_2^i, \ldots, v_{k_i}^i\}$. Let U be the universal instance space on A. Then the entropy of y on U is:*

$$\mathcal{H}(y, U) = -log_2\left(\frac{1}{k_y}\right)$$

Proof.

$$\begin{aligned}
\mathcal{H}(y,U) &= \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v}(U)|}{|U|} \log_2 \frac{|\sigma_{y=v}(U)|}{|U|} \right) && \text{(Eq. 3.24)} \\
&= \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v}(U)|}{\prod_{p=1}^{n} k_p} \log_2 \frac{|\sigma_{y=v}(U)|}{\prod_{p=1}^{n} k_p} \right) && \text{(lemma 13)} \\
&= \sum_{v \in dom(y)} \left(-\frac{(\prod_{p=1}^{n} k_p)/k_y}{\prod_{p=1}^{n} k_p} \log_2 \frac{(\prod_{p=1}^{n} k_p)/k_y}{\prod_{p=1}^{n} k_p} \right) && \text{(lemma 14)} \\
&= \sum_{v \in dom(y)} -\frac{1}{k_y} \log_2 \frac{1}{k_y} \\
&= -\frac{1}{k_y} \sum_{v \in dom(y)} \log_2 \frac{1}{k_y} \\
&= -\log_2 \frac{1}{k_y}
\end{aligned}$$

□

Next, we show that this result also holds for any q-multiple of the universal set U. To do so, we need an intermediate result:

Lemma 17. *Let T be a multi-set on relation schema $\mathcal{R}$, q be a positive integer, and ϕ a predicate. Then:*

$$\sigma_\phi(q(U)) = q(\sigma_\phi(U))$$

Proof. We use lemma 1. For the first direction, take $t \in q(\sigma_\phi(U))$. Then $t \in\in q(\sigma_\phi(U)) = qk > 0$, where $k = (t \in\in \sigma_\phi(U))$. Since $qk > 0$ we know that $k > 0$. So $t \in (\sigma_\phi(U))$, which entails that $t \in dom(\mathcal{R})$ and $\phi(t)$. But by the definition of selection in equation 3.7, this means that $t \in\in U = k$. But if $t \in\in U = k$, then $t \in\in q(U) = qk$, by equation 3.12. Since we already proved that $t \in dom(\mathcal{R})$ and $\phi(t)$, it follows from equation 3.7 that $t \in \sigma_\phi(q(U))$. So we have shown that $q(\sigma_\phi(U)) \subseteq \sigma_\phi(q(U))$.

For the opposite direction, we prove the contrapositive. Assume $t \notin q(\sigma_\phi(U))$. Then $t \in\in q(\sigma_\phi(U)) = 0$. But by equation 3.12 this means that $t \in\in \sigma_\phi(U) = 0$. This could occur in a number of ways. First, it could be that $t \notin dom(\mathcal{R})$. If so, then $t \notin \sigma_\phi(q(U))$. Second, t might be in $dom(\mathcal{R})$, but the predicate ϕ might not hold of t. Likewise, $t \notin \sigma_\phi(q(U))$. □

Lemma 18. *Let y be the target attribute, with $dom(y) = \{v_1^y, v_2^y, \ldots, v_{k_y}^y\}$. Let $A = \{a_1, a_2, \ldots, a_n\}$ be the complete set of attributes (including the target y) with domain $dom(a_i) = \{v_1^i, v_2^i, \ldots, v_{k_i}^i\}$. Let U be the universal instance space on A and q a positive integer. Then:*

$$\mathcal{H}(y, q(U)) = -log_2\left(\frac{1}{k_y}\right)$$

Proof.

$$\begin{aligned}
\mathcal{H}(y, q(U)) &= \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v}(q(U))|}{|q(U)|} \log_2 \frac{|\sigma_{y=v}(q(U))|}{|q(U)|} \right) && \text{(eq. 3.24)} \\
&= \sum_{v \in dom(y)} \left(-\frac{|q(\sigma_{y=v}(U))|}{|q(U)|} \log_2 \frac{|q(\sigma_{y=v}(U))|}{|q(U)|} \right) && \text{(lemma 17)} \\
&= \sum_{v \in dom(y)} \left(-\frac{q|\sigma_{y=v}(U)|}{q|U|} \log_2 \frac{q|\sigma_{y=v}(U)|}{q|U|} \right) && \text{(eq. 3.13)} \\
&= \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v}(U)|}{|U|} \log_2 \frac{|\sigma_{y=v}(U)|}{|U|} \right) \\
&= \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v}(U)|}{\prod_{p=1}^{n} k_p} \log_2 \frac{|\sigma_{y=v}(U)|}{\prod_{p=1}^{n} k_p} \right) && \text{(lemma 13)} \\
&= \sum_{v \in dom(y)} \left(-\frac{(\prod_{p=1}^{n} k_p)/k_y}{\prod_{p=1}^{n} k_p} \log_2 \frac{(\prod_{p=1}^{n} k_p)/k_y}{\prod_{p=1}^{n} k_p} \right) && \text{(lemma 14)} \\
&= \sum_{v \in dom(y)} -\frac{1}{k_y} \log_2 \frac{1}{k_y} \\
&= -\frac{1}{k_y} \sum_{v \in dom(y)} \log_2 \frac{1}{k_y} \\
&= -\log_2 \frac{1}{k_y}
\end{aligned}$$

□

Continuing on, we show that the same result holds for a univariate projection on a q-multiple of the universal set U. Again, an intermediate result helps us:

Lemma 19. *Assume T is a set with attributes $A = \{a_1, a_2, \ldots, a_n\}$ on domain schema $\mathcal{R}$. Let each a_i have $dom(a_i) = \{v_1^i, v_2^i, \ldots, v_{k_i}^i\}$. Then:*

$$\sigma_{a_i=v_j^i \wedge a_s=v_t^s}(T) = \sigma_{a_i=v_j^i}(\sigma_{a_s=v_t^s}(T))$$

Proof. We use lemma 1. For the first half, take $t \in \sigma_{a_i=v_j^i \wedge a_s=v_t^s}(T)$. By equation 3.7, we know from this that $t \in T$, $t \in dom(\mathcal{R})$, $t[a_i] = v_j^i$ and $t[a_s] = v_t^s$. But then we also know that $t \in \sigma_{a_s=v_t^s}(T)$. However, we can also infer from the information at hand and equation 3.7 that $t \in \sigma_{a_i=v_j^i}(\sigma_{a_s=v_t^s}(T))$. Hence, $\sigma_{a_i=v_j^i \wedge a_s=v_t^s}(T) \subseteq \sigma_{a_i=v_j^i}(\sigma_{a_s=v_t^s}(T))$.

For the opposite direction, assume that $t \in \sigma_{a_i=v_j^i}(\sigma_{a_s=v_t^s}(T))$. Then we know that $t \in \sigma_{a_s=v_t^s}(T)$, $t[i] = v_j^i$ and $t \in dom(\mathcal{R}$. But from this information, we also know that $t \in T$, $t[s] = v_t^s$. We can infer that $t[i] = v_j^i \wedge t[s] = v_t^s$. But then we can conclude that $t \in \sigma_{a_i=v_j^i \wedge a_s=v_t^s}(T)$. Hence, $\sigma_{a_i=v_j^i}(\sigma_{a_s=v_t^s}(T)) \subseteq \sigma_{a_i=v_j^i \wedge a_s=v_t^s}(T)$. □

Lemma 20. *Let y be the target attribute, with $dom(y) = \{v_1^y, v_2^y, \ldots, v_{k_y}^y\}$. Let $A = \{a_1, a_2, \ldots, a_n\}$ be the complete set of attributes (including the target y) with domain $dom(a_i) = \{v_1^i, v_2^i, \ldots, v_{k_i}^i\}$. Let U be the universal instance space on A and q a positive integer. If $a \in A$ and $w \in dom(a)$, then:*

$$\mathcal{H}(y, \sigma_{a=w}(q(U))) = -log_2\left(\frac{1}{k_y}\right)$$

Proof.

$$\begin{aligned}
\mathcal{H}(y, \sigma_{a=w}(q(U))) &= \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v}(\sigma_{a=w}(q(U)))|}{|\sigma_{a=w}(q(U))|} \log_2 \frac{|\sigma_{y=v}(\sigma_{a=w}(q(U)))|}{|\sigma_{a=w}(q(U))|} \right) && \text{(eq. 3.24)} \\
&= \sum_{v \in dom(y)} \left(-\frac{|q(\sigma_{y=v}(\sigma_{a=w}(U)))|}{|q(\sigma_{a=w}(U))|} \log_2 \frac{|q(\sigma_{y=v}(\sigma_{a=w}(U)))|}{|q(\sigma_{a=w}(U))|} \right) && \text{(lemma 17)} \\
&= \sum_{v \in dom(y)} \left(-\frac{q|\sigma_{y=v}(\sigma_{a=w}(U))|}{q|\sigma_{a=w}(U)|} \log_2 \frac{q|\sigma_{y=v}(\sigma_{a=w}(U))|}{q|\sigma_{a=w}(U)|} \right) && \text{(eq. 3.13)} \\
&= \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v}(\sigma_{a=w}(U))|}{|\sigma_{a=w}(U)|} \log_2 \frac{|\sigma_{y=v}(\sigma_{a=w}(U))|}{|\sigma_{a=w}(U)|} \right) \\
&= \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v \wedge a=w}(U)|}{|\sigma_{a=w}(U)|} \log_2 \frac{|\sigma_{y=v \wedge a=w}(U)|}{|\sigma_{a=w}(U)|} \right) && \text{(lemma 19)} \\
&= \sum_{v \in dom(y)} \left(-\frac{(\prod_{p=1}^{n} k_p)/(k_a k_y)}{(\prod_{p=1}^{n} k_p)/k_a} \log_2 \frac{(\prod_{p=1}^{n} k_p)/(k_a k_y)}{(\prod_{p=1}^{n} k_p)/k_a} \right) \\
&= \sum_{v \in dom(y)} \left(-\frac{1}{k_y} \log_2 \frac{1}{k_y} \right) \\
&= \frac{1}{k_y} \sum_{v \in dom(y)} -\log_2 \frac{1}{k_y} \\
&= -\log_2 \frac{1}{k_y}
\end{aligned}$$

□

We use these result to prove a useful result about the information gain on any multiple of the universal instance space:

Theorem 5. *The information gain of any attribute a_i on any q-multiple of the universal instance space U is 0.*

Proof.

$$
\begin{aligned}
\mathrm{IGain}(a_i, q(U)) &= \mathcal{H}(y, q(U)) - \mathcal{H}(y, q(U)|a_i) \\
&= \mathcal{H}(y, q(U)) - \sum_{j=1}^{|dom(a_i)|} \frac{|\sigma_{a_i=v_j^i}(q(U))|}{|q(U)|} \mathcal{H}(y, \sigma_{a_i=v_j^i}(q(U)) \\
&= \mathcal{H}(y, q(U)) - \sum_{j=1}^{|dom(a_i)|} \frac{|q(\sigma_{a_i=v_j^i}(U))|}{|q(U)|} \mathcal{H}(y, \sigma_{a_i=v_j^i}(q(U)) \quad \text{(lemma 17)} \\
&= \mathcal{H}(y, q(U)) - \sum_{j=1}^{|dom(a_i)|} \frac{q|\sigma_{a_i=v_j^i}(U)|}{q|U|} \mathcal{H}(y, \sigma_{a_i=v_j^i}(q(U)) \quad \text{(eq 3.13)} \\
&= -\log_2\left(\frac{1}{k_y}\right) - \sum_{j=1}^{|dom(a_i)|} \frac{|\sigma_{a_i=v_j^i}(U)|}{|U|} \mathcal{H}(y, \sigma_{a_i=v_j^i}(q(U))) \quad \text{(lemma 18)} \\
&= -\log_2\left(\frac{1}{k_y}\right) - \sum_{j=1}^{|dom(a_i)|} \frac{|\sigma_{a_i=v_j^i}(U)|}{|U|} \left(-\log_2\left(\frac{1}{k_y}\right)\right) \quad \text{(lemma 20)} \\
&= -\log_2\left(\frac{1}{k_y}\right) + \log_2\left(\frac{1}{k_y}\right) \sum_{j=1}^{|dom(a_i)|} \frac{|\sigma_{a_i=v_j^i}(U)|}{|U|} \\
&= -\log_2\left(\frac{1}{k_y}\right) + \log_2\left(\frac{1}{k_y}\right) 1 \\
&= 0
\end{aligned}
$$

□

Entropy on a Complement of the Universal Instance Space

In the preceding pages, we proved that:

$$\begin{aligned}
\mathcal{H}(y, U) &= -\log_2\left(\frac{1}{k_y}\right) \\
\mathcal{H}(y, q(U)) &= -\log_2\left(\frac{1}{k_y}\right) \\
\mathcal{H}(y, \sigma_{a=w}(q(U))) &= -\log_2\left(\frac{1}{k_y}\right) \\
(\forall a_i \in A)\mathrm{IG}(a_i, T) &= 0
\end{aligned}$$

Our next task is to take these observations and see if we can find expressions that don't merely involve the universal instance space U. If we can find a means of computing $\mathcal{H}(y, T)$ and $\mathcal{H}(y, T|a_i)$ from T and T', we have fulfilled our obligation to provide a rigorous and formal footing for Fong's unrealization approach.

What we accomplish in this section is to provide expressions for the following:

1. $\mathcal{H}(y, q(U) - T)$
2. $\mathcal{H}(y, \sigma_{a_i=v_j^i}(q(U) - T))$
3. $\mathcal{H}(y, q(U) - T|a_i)$

As usual, we start by proving a useful result.

Lemma 21. *Let T be a multi-set, and let ϕ be a predicate. Then:*

$$\sigma_\phi(T) \subseteq T$$

Proof. Assume that $\sigma_\phi(T)$ is not contained in T. Then, using set pair notation and equation 3.1, we know there is a tuple $t \in dom(\mathcal{R})$ such that $t \in\in \sigma_\phi(T) > t \in\in T$. But by equation 3.7, we know that $\sigma_\phi(T) = \{(x, x \in\in T)|x \in dom(\mathcal{R}) \wedge \phi(x)\} \cup \{(x, 0)|(x \notin dom(\mathcal{R}) \vee (x \in dom(\mathcal{R}) \wedge \neg\phi(x))\}$. We can assume that $t \in\in \sigma_\phi(T) > 0$, so we know from the definition that t occurs in T exactly $t \in\in \sigma_\phi$ times. That is, $t \in\in \sigma_\phi(T) = t \in\in T$. This contradicts the statement that $t \in\in \sigma_\phi(T) > t \in\in T$. □

Lemma 22. *Let U the universal instance space on relation schema $\mathcal{R}$, involving attributes $A = \{a_1, a_2, \ldots, a_n\}$. (A includes the target atttribute). Let $|dom(a_i)| = k_i$. Let T be a multi-set, q be a positive integer, and let y denote the target attribute with domain size k_y. Assume $T \subseteq q(U)$. Then:*

$$\mathcal{H}(y, q(U) - T) = -\sum_{v \in dom(y)} \left(\frac{\frac{q}{k_y}\prod_{i=1}^{n} k_i - |\sigma_{y=v}(T)|}{q\prod_{i=1}^{n} k_i - |T|} log_2 \frac{\frac{q}{k_y}\prod_{i=1}^{n} k_i - |\sigma_{y=v}(T)|}{q\prod_{i=1}^{n} k_i - |T|} \right)$$

Proof.

$$\begin{aligned}
\mathcal{H}(y, q(U) - T) &= \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v}(q(U) - T)|}{|q(U) - T|} \log_2 \frac{|\sigma_{y=v}(q(U) - T)|}{|q(U) - T|} \right) \\
&= \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v}(q(U))| - |\sigma_{y=v}(T)|}{|q(U) - T|} \log_2 \frac{|\sigma_{y=v}(q(U))| - |\sigma_{y=v}(T)|}{|q(U) - T|} \right) && \text{lemma 12} \\
&= \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v}(q(U))| - |\sigma_{y=v}(T)|}{|q(U)| - |T|} \log_2 \frac{|\sigma_{y=v}(q(U))| - |\sigma_{y=v}(T)|}{|q(U)| - |T|} \right) && \text{lemma 8} \\
&= \sum_{v \in dom(y)} \left(-\frac{|q(\sigma_{y=v}(U))| - |\sigma_{y=v}(T)|}{|q(U)| - |T|} \log_2 \frac{|q(\sigma_{y=v}(U))| - |\sigma_{y=v}(T)|}{|q(U)| - |T|} \right) && \text{lemma 17} \\
&= \sum_{v \in dom(y)} \left(-\frac{q|\sigma_{y=v}(U)| - |\sigma_{y=v}(T)|}{q|U| - |T|} \log_2 \frac{q|\sigma_{y=v}(U)| - |\sigma_{y=v}(T)|}{q|U| - |T|} \right) && \text{eq. 3.13} \\
&= \sum_{v \in dom(y)} \left(-\frac{q\frac{\prod_{i=1}^{n} k_i}{k_y} - |\sigma_{y=v}(T)|}{q\prod_{i=1}^{n} k_i - |T|} \log_2 \frac{q\frac{\prod_{i=1}^{n} k_i}{k_y} - |\sigma_{y=v}(T)|}{q\prod_{i=1}^{n} k_i - |T|} \right) && \text{lem. 13, 14} \\
&= -\sum_{v \in dom(y)} \left(\frac{\frac{q}{k_y}\prod_{i=1}^{n} k_i - |\sigma_{y=v}(T)|}{q\prod_{i=1}^{n} k_i - |T|} \log_2 \frac{\frac{q}{k_y}\prod_{i=1}^{n} k_i - |\sigma_{y=v}(T)|}{q\prod_{i=1}^{n} k_i - |T|} \right)
\end{aligned}$$

□

Lemma 23. *Let U the universal instance space on relation schema $\mathcal{R}$, involving attributes $A = \{a_1, a_2, \ldots, a_n\}$. (A includes the target atttribute). Let $|dom(a_i)| = k_i$. Let T be a multi-set, q be a positive integer, and let a_i be a non-target attribute. Finally, let y denote the target attribute with domain size k_y. Assume $T \subseteq q(U)$. Then:*

$$\mathcal{H}(y, \sigma_{a_i=v_j^i}(q(U) - T)) =$$

$$- \sum_{v \in dom(y)} \left(\frac{\frac{q}{k_y k_i} \prod_{h=1}^{n} k_h - |\sigma_{y=v \wedge a_i=v_j^i}(T)|}{\frac{q}{k_i} \prod_{h=1}^{n} k_h - |\sigma_{a_i=v_j^i}(T)|} log_2 \frac{\frac{q}{k_y k_i} \prod_{h=1}^{n} k_h - |\sigma_{y=v \wedge a_i=v_j^i}(T)|}{\frac{q}{k_i} \prod_{h=1}^{n} k_h - |\sigma_{a_i=v_j^i}(T)|} \right)$$

Proof.

$$\mathcal{H}(y, \sigma_{a_i=v_j^i}(q(U) - T)) = \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v \wedge a_i=v_j^i}(q(U) - T)|}{|\sigma_{a_i=v_j^i}(q(U) - T)|} \log_2 \frac{|\sigma_{y=v \wedge a_i=v_j^i}(q(U) - T)|}{|\sigma_{a_i=v_j^i}(q(U) - T)|} \right)$$

To save on space, consider one of the components of this expression:

$$\begin{aligned}
\frac{|\sigma_{y=v \wedge a_i=v_j^i}(q(U) - T)|}{|\sigma_{a_i=v_j^i}(q(U) - T)|} &= \frac{|\sigma_{y=v \wedge a_i=v_j^i}(q(U))| - |\sigma_{y=v \wedge a_i=v_j^i}(T)|}{|\sigma_{a_i=v_j^i}(q(U))| - |\sigma_{a_i=v_j^i}(T)|} && \text{lemma 12} \\
&= \frac{|q(\sigma_{y=v \wedge a_i=v_j^i}(U))| - |\sigma_{y=v \wedge a_i=v_j^i}(T)|}{|q(\sigma_{a_i=v_j^i}(U))| - |\sigma_{a_i=v_j^i}(T)|} && \text{lemma 17} \\
&= \frac{q|(\sigma_{y=v \wedge a_i=v_j^i}(U))| - |\sigma_{y=v \wedge a_i=v_j^i}(T)|}{q|(\sigma_{a_i=v_j^i}(U))| - |\sigma_{a_i=v_j^i}(T)|} && \text{eq. 3.13} \\
&= \frac{\frac{q}{k_y k_i} \prod_{h=1}^{n} k_h - |\sigma_{y=v \wedge a_i=v_j^i}(T)|}{\frac{q}{k_i} \prod_{h=1}^{n} k_h - |\sigma_{a_i=v_j^i}(T)|} && \text{lem. 13, 14}
\end{aligned}$$

□

Lemma 24. *Let U the universal instance space on relation schema $\mathcal{R}$, involving attributes $A = \{a_1, a_2, \ldots, a_n\}$. (A includes the target atttribute). Let $|dom(a_i)| = k_i$. Let T be a multi-set, q be a positive integer, and let a_i be a non-target attribute. Finally, let y denote the target attribute with domain size k_y. Assume $T \subseteq q(U)$. Then:*

$$\mathcal{H}(y, q(U) - T|a_i) = \sum_{j=1}^{|dom(a_i)|} \frac{\frac{q}{k_i}\prod_{h=1}^{n} k_h - |\sigma_{a_i=v_j^i}(T)|}{q\prod_{h=1}^{n} k_h - |T|} \left(- \sum_{v \in dom(y)} (x log_2 x) \right)$$

where

$$x = \frac{\frac{q}{k_y k_i}\prod_{h=1}^{n} k_h - |\sigma_{y=v \wedge a_i=v_j^i}(T)|}{\frac{q}{k_i}\prod_{h=1}^{n} k_h - |\sigma_{a_i=v_j^i}(T)|}$$

Proof.

$$\mathcal{H}(y, q(U) - T|a_i) = \sum_{j=1}^{|dom(a_i)|} \frac{|\sigma_{a_i=v_j^i}(q(U) - T)|}{|q(U) - T|} \mathcal{H}(y, \sigma_{a_i=v_j^i}(q(U) - T)$$

We computed the value of $\mathcal{H}(y, \sigma_{a_i=v_j^i}(q(U) - T)$ previously, so let us examine the other part of the expression.

$$\begin{aligned}
\sum_{j=1}^{|dom(a_i)|} \frac{|\sigma_{a_i=v_j^i}(q(U) - T)|}{|q(U) - T|} &= \sum_{j=1}^{|dom(a_i)|} \frac{|\sigma_{a_i=v_j^i}(q(U))| - |\sigma_{a_i=v_j^i}(T)|}{|q(U) - T|} && \text{lemma 12} \\
&= \sum_{j=1}^{|dom(a_i)|} \frac{|q(\sigma_{a_i=v_j^i}(U))| - |\sigma_{a_i=v_j^i}(T)|}{|q(U) - T|} && \text{lemma 17} \\
&= \sum_{j=1}^{|dom(a_i)|} \frac{|q(\sigma_{a_i=v_j^i}(U))| - |\sigma_{a_i=v_j^i}(T)|}{|q(U)| - |T|} && \text{lemma 8} \\
&= \sum_{j=1}^{|dom(a_i)|} \frac{q|\sigma_{a_i=v_j^i}(U)| - |\sigma_{a_i=v_j^i}(T)|}{q|U| - |T|} && \text{eq. 3.13} \\
&= \sum_{j=1}^{|dom(a_i)|} \frac{\frac{q}{k_i}\prod_{h=1}^{n} k_h - |\sigma_{a_i=v_j^i}(T)|}{q\prod_{h=1}^{n} k_h - |T|} && \text{lem. 13, 14}
\end{aligned}$$

□

Entropy of T Computed with T' and T^P

In the preceding pages, we provided expressions for the following:

1. $\mathcal{H}(y, q(U) - T)$
2. $\mathcal{H}(y, \sigma_{a_i=v_j^i}(q(U) - T))$
3. $\mathcal{H}(y, q(U) - T|a_i)$

In this section, we push this result further by proving similar expressions (not involving data in T) for:

1. $|\sigma_\psi(q(U) - T)|$
2. $\mathcal{H}(y, T)$
3. $\mathcal{H}(y, \sigma_\psi(T))$
4. $\mathcal{H}(y, \sigma_\psi(T)|a_i)$

where ψ is a predicate composed from applying the '$\wedge$' operator to atomic predicates composed from the attributes of A.

The important point is that the expressions will involve calculations based on T' and T^P, and not the original data set T. Once we have finished this task, we will have shown that the unrealization approach can allow us to build the same decision tree that ID3 would construct from the training data T, using only the unreal data sets T' and T^P.

In this section, we use ψ to stand for a predicate composed from applying the logical 'and' operator to atomic predicates. Examples of this type of predicate include:

- $a_1 = v_5^1$
- $a_1 = v_1^1 \wedge a_2 = v_1^2 \wedge \cdots \wedge a_n = v_n^n$

Recall that $A = \{a_1, a_2, \ldots, a_n\}$. Let $[1, r] = \{1, 2, \ldots, r\}$ be a finite set of consecutive natural numbers, with $r \leq n$. We then define an injective function f that maps $[1, r]$ to A . The idea is that f picks out a subset A' of the attributes in A. We can conceive of A' as the set $\{f(1), f(2), \ldots, f(r)\}$. The predicate ψ_r is a predicate on A', with each attribute being assigned some value in its domain:

$$\psi_r = (f(1) = v^{f(1)}) \wedge (f(2) = v^{f(2)}) \wedge \cdots \wedge (f(r) = v^{f(r)}) = \bigwedge_{i=1}^{r} (f(i) = v^{f(i)}) \quad (3.25)$$

We use this formulation to prove a useful result:

Lemma 25. *Let U be the universal instance space, q be a positive integer, T be a multi-set such that $T \subseteq q(U)$, and ψ be a predicate as defined in equation 3.25.*

$$|\sigma_\psi(q(U))| = \frac{2|T'| + |T^P|}{\prod_{i=1}^{r} |dom(f(i))|}$$

Proof.

1)	$\lvert q(U)\rvert$	$= 2\lvert T'\rvert + \lvert T^P\rvert$	(Theorem 4)
2)	$\lvert\sigma_\psi(q(U))\rvert$	$= \lvert q(\sigma_\psi(U))\rvert$	(Lemma 17)
3)		$= q\lvert(\sigma_\psi(U))\rvert$	
4)		$= q\frac{\prod_{i=1}^{n} \lvert dom(a_i)\rvert}{\prod_{i=1}^{r} \lvert dom(f(i))\rvert}$	(Lemma 15)
5)		$= \frac{q\lvert U\rvert}{\prod_{i=1}^{r} \lvert dom(f(i))\rvert}$	
6)		$= \frac{\lvert q(U)\rvert}{\prod_{i=1}^{r} \lvert dom(f(i))\rvert}$	
7)		$= \frac{2\lvert T'\rvert + \lvert T^P\rvert}{\prod_{i=1}^{r} \lvert dom(f(i))\rvert}$	(lines 1,6)

□

Lemma 26. *Let U be the universal instance space, q be a positive integer, T be a training set with $T \subseteq q(U)$, and T', T^P be unreal data sets formed by the unrealization procedure. Then we can calculate $\mathcal{H}(y,T)$ by using information about U, T' and T^P alone.*

Proof.

$$\begin{aligned}
\mathcal{H}(y,T) &= \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v}(T)|}{|T|} \log_2 \frac{|\sigma_{y=v}(T)|}{|T|} \right) && \text{(Eq. 3.24)} \\
\frac{|\sigma_{y=v}(T)|}{|T|} &= \frac{|\sigma_{y=v}(q(U) - (T' \uplus T^P))|}{|q(U) - (T' \uplus T^P)|} && \text{(Thm. 3)} \\
&= \frac{|\sigma_{y=v}(q(U))| - |\sigma_{y=v}(T' \uplus T^P))|}{|q(U) - (T' \uplus T^P)|} && \text{(Lem. 12)} \\
&= \frac{|\sigma_{y=v}(q(U))| - |\sigma_{y=v}(T' \uplus T^P))|}{|q(U)| - |T' \uplus T^P|} && \text{(Lem. 8)} \\
&= \frac{|q(\sigma_{y=v}(U))| - |\sigma_{y=v}(T' \uplus T^P))|}{|q(U)| - |T' \uplus T^P|} && \text{(Lem. 17)} \\
&= \frac{q|(\sigma_{y=v}(U))| - |\sigma_{y=v}(T' \uplus T^P))|}{q|(U)| - |T' \uplus T^P|} \\
&= \frac{\frac{q \prod_{i=1}^{n} |dom(a_i)|}{|dom(y)|} - |\sigma_{y=v}(T' \uplus T^P))|}{q|U| - |T' \uplus T^P|} \\
&= \frac{\frac{q|U|}{|dom(y)|} - |\sigma_{y=v}(T' \uplus T^P))|}{q|U| - |T' \uplus T^P|}
\end{aligned}$$

So in the end, we can show that $\mathcal{H}(y,T) =$

$$\sum_{v \in dom(y)} \frac{\frac{q|U|}{|dom(y)|} - |\sigma_{y=v}(T' \uplus T^P))|}{q|U| - |T' \uplus T^P|} \log_2 \frac{\frac{q|U|}{|dom(y)|} - |\sigma_{y=v}(T' \uplus T^P))|}{q|U| - |T' \uplus T^P|}$$

Since T appears nowhere in this expression, we have proved our claim. □

Lemma 27. *Let U be the universal instance space, q be a positive integer, T be a training set with $T \subseteq q(U)$, ψ be a predicate, and T', T^P be unreal data sets. Then we can calculate $\mathcal{H}(y, \sigma_\psi(T))$ by using information about U, T' and T^P alone.*

Proof.

$$\begin{aligned}
\mathcal{H}(y, \sigma_\psi(T)) &= \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v}(\sigma_\psi(T))|}{|\sigma_\psi(T)|} \log_2 \frac{|\sigma_{y=v}(\sigma_\psi(T))|}{|\sigma_\psi(T)|} \right) && \text{(Eq. 3.24)} \\
\frac{|\sigma_{y=v}(\sigma_\psi(T))|}{|\sigma_\psi(T)|} &= \frac{|\sigma_{y=v}(\sigma_\psi(q(U) - (T' \uplus T^P)))|}{|\sigma_\psi(q(U) - (T' \uplus T^P))|} && \text{(Thm. 3)} \\
&= \frac{|\sigma_{y=v \wedge \psi}(q(U) - (T' \uplus T^P))|}{|\sigma_\psi(q(U) - (T' \uplus T^P))|} && \text{(Lem. 19)} \\
&= \frac{|\sigma_{y=v \wedge \psi}(q(U))| - |\sigma_{y=v \wedge \psi}(T' \uplus T^P))|}{|\sigma_\psi(q(U))| - |\sigma_\psi(T' \uplus T^P)|} && \text{(Lem. 12)} \\
&= \frac{|q(\sigma_{y=v \wedge \psi}(U))| - |\sigma_{y=v \wedge \psi}(T' \uplus T^P))|}{|q(\sigma_\psi(U))| - |\sigma_\psi(T' \uplus T^P)|} && \text{(Lem. 17)} \\
&= \frac{q|\sigma_{y=v \wedge \psi}(U)| - |\sigma_{y=v \wedge \psi}(T' \uplus T^P))|}{q|\sigma_\psi(U)| - |\sigma_\psi(T' \uplus T^P)|} \\
&= \frac{\frac{q \prod_{i=1}^{n} |dom(a_i)|}{|dom(y)| \prod_{i=1}^{r} |dom(f(i))|} - |\sigma_{y=v \wedge \psi}(T' \uplus T^P))|}{\frac{q \prod_{i=1}^{n} |dom(a_i)|}{\prod_{i=1}^{r} |dom(f(i))|} - |\sigma_\psi(T' \uplus T^P)|} \\
&= \frac{\frac{q|U|}{|dom(y)| \prod_{i=1}^{r} |dom(f(i))|} - |\sigma_{y=v \wedge \psi}(T' \uplus T^P))|}{\frac{q|U|}{\prod_{i=1}^{r} |dom(f(i))|} - |\sigma_\psi(T' \uplus T^P)|}
\end{aligned}$$

So in the end, we can show that $\mathcal{H}(y, \sigma_\psi(T)) =$

$$\sum_{v \in dom(y)} \frac{\frac{q|U|}{|dom(y)| \prod_{i=1}^{r} |dom(f(i))|} - |\sigma_{y=v \wedge \psi}(T' \uplus T^P))|}{\frac{q|U|}{\prod_{i=1}^{r} |dom(f(i))|} - |\sigma_\psi(T' \uplus T^P)|} \log_2 \frac{\frac{q|U|}{|dom(y)| \prod_{i=1}^{r} |dom(f(i))|} - |\sigma_{y=v \wedge \psi}(T' \uplus T^P))|}{\frac{q|U|}{\prod_{i=1}^{r} |dom(f(i))|} - |\sigma_\psi(T' \uplus T^P)|}$$

Since T appears nowhere in this expression, we have proved our claim. □

Lemma 28. *Let U be the universal instance space, q be a positive integer, T be a training set with $T \subseteq q(U)$, a_i be a (non-target) attribute, ψ be a predicate (not involving a_i) and T', T^P be unreal data sets formed by the unrealization procedure. Then we can calculate $\mathcal{H}(y, \sigma_\psi(T)|a_i)$ by using information about U, T' and T^P alone.*

Proof.

$$\mathcal{H}(y, \sigma_\psi(T)|a_i) = \sum_{j=1}^{|dom(a_i)|} \frac{|\sigma_{a_i=v_j^i}(\sigma_\psi(T))|}{|\sigma_\psi(T)|} \mathcal{H}(y, \sigma_{a_i=v_j^i}(\sigma_\psi(T)))$$

We break this proof up, as it is cumbersome to do in LaTeX otherwise. First, we tackle a familiar expression:

$$\begin{aligned}
\frac{|\sigma_{a_i=v_j^i}(\sigma_\psi(T))|}{|\sigma_\psi(T)|} &= \frac{|\sigma_{a_i=v_j^i}(\sigma_\psi(q(U) - (T' \uplus T^P)))|}{|\sigma_\psi(q(U) - (T' \uplus T^P))|} && \text{(Thm. 3)} \\
&= \frac{|\sigma_{a_i=v_j^i \wedge \psi}(q(U) - (T' \uplus T^P))|}{|\sigma_\psi(q(U) - (T' \uplus T^P))|} && \text{(Lem. 19)} \\
&= \frac{|\sigma_{a_i=v_j^i \wedge \psi}(q(U))| - |\sigma_{a_i=v_j^i \wedge \psi}(T' \uplus T^P))|}{|\sigma_\psi(q(U))| - |\sigma_\psi(T' \uplus T^P)|} && \text{(Lem. 12)} \\
&= \frac{|q(\sigma_{a_i=v_j^i \wedge \psi}(U))| - |\sigma_{a_i=v_j^i \wedge \psi}(T' \uplus T^P))|}{|q(\sigma_\psi(U))| - |\sigma_\psi(T' \uplus T^P)|} && \text{(Lem. 17)} \\
&= \frac{q|\sigma_{a_i=v_j^i \wedge \psi}(U)| - |\sigma_{a_i=v_j^i \wedge \psi}(T' \uplus T^P))|}{q|\sigma_\psi(U)| - |\sigma_\psi(T' \uplus T^P)|} \\
&= \frac{\frac{q \prod_{h=1}^{n} |dom(a_h)|}{|dom(a_i)| \prod_{h=1}^{r} |dom(f(h))|} - |\sigma_{a_i=v_j^i \wedge \psi}(T' \uplus T^P))|}{\frac{q \prod_{h=1}^{n} |dom(a_h)|}{\prod_{h=1}^{r} |dom(f(h))|} - |\sigma_\psi(T' \uplus T^P)|} \\
&= \frac{\frac{q|U|}{|dom(a_i)| \prod_{h=1}^{r} |dom(f(h))|} - |\sigma_{a_i=v_j^i \wedge \psi}(T' \uplus T^P))|}{\frac{q|U|}{\prod_{h=1}^{r} |dom(f(h))|} - |\sigma_\psi(T' \uplus T^P)|}
\end{aligned}$$

Next, we move to tackle $\mathcal{H}(y, \sigma_{a_i=v_j^i}(\sigma_\psi(T)))$:

$$\mathcal{H}(y, \sigma_{a_i=v_j^i}(\sigma_\psi(T))) = \sum_{v \in dom(y)} \left(-\frac{|\sigma_{y=v}(\sigma_{a_i=v_j^i}(\sigma_\psi(T)))|}{|\sigma_{a_i=v_j^i}(\sigma_\psi(T))|} \log_2 \frac{|\sigma_{y=v}(\sigma_{a_i=v_j^i}(\sigma_\psi(T)))|}{|\sigma_{a_i=v_j^i}(\sigma_\psi(T))|} \right) \quad \text{(eq. 3.24)}$$

We work with half of the expression:

$$\begin{aligned}
\frac{|\sigma_{y=v}(\sigma_{a_i=v_j^i}(\sigma_\psi(T)))|}{|\sigma_{a_i=v_j^i}(\sigma_\psi(T))|} &= \frac{|\sigma_{y=v \wedge a_i=v_j^i \wedge \psi}(T)|}{|\sigma_{a_i=v_j^i \wedge \psi}(T)|} && \text{(Lem. 19)} \\
&= \frac{|\sigma_{y=v \wedge a_i=v_j^i \wedge \psi}(q(U) - (T' \uplus T^P))|}{|\sigma_{a_i=v_j^i \wedge \psi}(q(U) - (T' \uplus T^P))|} && \text{(Thm. 3)} \\
&= \frac{|\sigma_{y=v \wedge a_i=v_j^i \wedge \psi}(q(U))| - |\sigma_{y=v \wedge a_i=v_j^i \wedge \psi}(T' \uplus T^P))|}{|\sigma_{a_i=v_j^i \wedge \psi}(q(U))| - |\sigma_{a_i=v_j^i \wedge \psi}(T' \uplus T^P))|} && \text{(Lem. 12)} \\
&= \frac{|q(\sigma_{y=v \wedge a_i=v_j^i \wedge \psi}(U))| - |\sigma_{y=v \wedge a_i=v_j^i \wedge \psi}(T' \uplus T^P))|}{|q(\sigma_{a_i=v_j^i \wedge \psi}(U))| - |\sigma_{a_i=v_j^i \wedge \psi}(T' \uplus T^P))|} && \text{(Lem. 17)} \\
&= \frac{q|(\sigma_{y=v \wedge a_i=v_j^i \wedge \psi}(U))| - |\sigma_{y=v \wedge a_i=v_j^i \wedge \psi}(T' \uplus T^P))|}{q|(\sigma_{a_i=v_j^i \wedge \psi}(U))| - |\sigma_{a_i=v_j^i \wedge \psi}(T' \uplus T^P))|} \\
&= \frac{\frac{q \prod_{h=1}^{n} |dom(a_h)|}{|dom(a_i)||dom(y)| \prod_{h=1}^{r} |dom(f(h))|} - |\sigma_{y=v \wedge a_i=v_j^i \wedge \psi}(T' \uplus T^P))|}{\frac{q \prod_{h=1}^{n} |dom(a_h)|}{|dom(a_i)| \prod_{h=1}^{r} |dom(f(h))|} - |\sigma_{a_i=v_j^i \wedge \psi}(T' \uplus T^P))|} \\
&= \frac{\frac{q|U|}{|dom(a_i)||dom(y)| \prod_{h=1}^{r} |dom(f(h))|} - |\sigma_{y=v \wedge a_i=v_j^i \wedge \psi}(T' \uplus T^P))|}{\frac{q|U|}{|dom(a_i)| \prod_{h=1}^{r} |dom(f(h))|} - |\sigma_{a_i=v_j^i \wedge \psi}(T' \uplus T^P))|}
\end{aligned}$$

Since we can express all the components of $\mathcal{H}(y, \sigma_{a_i=v_j^i}(\sigma_\psi(T)))$ in terms of U, T' and T^P, we have proved our claim.[22] □

[22]The size of these equations makes displaying them on a single page rather difficult.

To summarize, we have shown that we can calculate the following values by the unreal data sets T' and T^P alone:

1. $|\sigma_\psi(q(U) - T)|$
2. $\mathcal{H}(y, T)$
3. $\mathcal{H}(y, \sigma_\psi(T))$
4. $\mathcal{H}(y, \sigma_\psi(T)|a_i)$

This accomplishment means that the decision tree modifications introduced by Fong preserve the decision tree that would have been constructed from the training data T. Instead of using this data, we can use T' and T^P, knowing that the decision tree that results is the same.

Of course, the original intellectual effort in deriving this scheme was due to Fong. Our role in this section of the chapter was to upgrade his work by:

1. Providing a formal basis in a variant of the multi-relational algebra.
2. Proving key claims that were left unaddressed in Fong's work.
3. Providing a more clear explanation of the workings of the algorithm.
4. Proving a large number of helpful results in the multi-relational algebra.
5. Analyzing the unrealization algorithm, which we presented in iterative form.
6. Generalizing some of Fong's concepts, as in the case of the general predicates ϕ and ψ.

We deviated from his original presentation in just about every way possible. In the next section, we show that the unrealization approach can be extended to the industry-standard C4.5 algorithm.

3.3 Extending the Unrealization Approach

In this section, we demonstrate that the unrealization approach outlined in Fong's thesis [20] is not limited to the ID3 decision tree induction algorithm. It can be extended to cutting-edge algorithms, since the general strategy is quite generic. We demonstrate this claim by showing how the unrealization method can be adapted to the industry-standard C4.5 algorithm of Quinlan [41].

3.3.1 The C4.5 Algorithm Explained

Quinlan's C4.5 algorithm is a refinement of various developments in decision tree learning that followed his seminal ID3 algorithm. The main changes over the ID3 approach are:

1. **Split Criteria**: The C4.5 algorithm uses a new split criteria, called the *Gain Ratio*. The new approach is intended to address weaknesses in the *Information Gain* criterion used in the ID3 algorithm.

2. **Numeric Values**: The C4.5 algorithm supports numeric values, as well as categorical ones.

3. **Missing Values**: The C4.5 algorithm provides mechanisms for dealing with missing data.

4. **Pruning**: Most importantly, the C4.5 algorithm supports the two-phased approach outlined in Section 3.1.4 above. It includes a *pruning* phase that modifies the decision tree produced by a top-down induction algorithm.

In order to show that the unrealization approach can be extended from ID3 to C4.5, it is sufficient to show that:

1. We can calculate the *Gain Ratio* for attribute a on training set T by using only the information contained in the unreal data sets T' and T^P.

2. We can similarly support the *pruning* phase of the algorithm by means of T' and T^P, without using the data set T explicitly.

In the next two sections, we show how to fulfil both of these conditions.

Splitting Criteria

Recall from Section 3.2.4 that the ID3 algorithm used by Fong in the unrealization algorithm involved an impurity measure known as the *information gain*. In particular, the information gain associated with splitting the training set T by attribute a_i is:

$$\mathrm{IG}(a_i, T) = \mathcal{H}(y, T) - \mathcal{H}(y, T|a_i)$$

In this equation, $\mathcal{H}$ represents the information theoretic concept of entropy, as defined in Section 3.1.5. In Section 3.2.4, we showed that we can calculate entropies on the training set T from the unreal sets T' and T^P alone. For instance, we showed that for the unrealization approach,

$$\mathcal{H}(y, T) = \sum_{v \in dom(y)} \frac{\frac{q|U|}{|dom(y)|} - |\sigma_{y=v}(T' \uplus T^P)|}{q|U| - |T' \uplus T^P|} \log_2 \frac{\frac{q|U|}{|dom(y)|} - |\sigma_{y=v}(T' \uplus T^P)|}{q|U| - |T' \uplus T^P|}$$

We also proved that we can calculate the entropy of arbitrary selection operations on T, by using only the information in T' and T^P. For instance, $\mathcal{H}(y, \sigma_\psi(T)) =$

$$\sum_{v \in dom(y)} \frac{\frac{\frac{q|U|}{|dom(y)|\prod_{i=1}^{r}|dom(f(i))|} - |\sigma_{y=v\wedge\psi}(T' \uplus T^P)|}{\frac{q|U|}{\prod_{i=1}^{r}|dom(f(i))|} - |\sigma_{\psi}(T' \uplus T^P)|} \log_2 \frac{\frac{\frac{q|U|}{|dom(y)|\prod_{i=1}^{r}|dom(f(i))|} - |\sigma_{y=v\wedge\psi}(T' \uplus T^P)|}{\frac{q|U|}{\prod_{i=1}^{r}|dom(f(i))|} - |\sigma_{\psi}(T' \uplus T^P)|}$$

Moving on to a discussion of C4.5, Quinlan [41] suggested the *Gain Ratio* metric as an improvement over the information gain measure, which suffers from several key defects.[23] The Gain Ratio normalizes the information gain measure as follows:

$$\mathrm{GainRatio}(a_i, T) = \frac{\mathrm{IG}(a_i, T)}{\mathcal{H}(a_i, T)} = \frac{\mathcal{H}(y, T) - \mathcal{H}(y, T|a_i)}{\mathcal{H}(a_i, T)} \tag{3.26}$$

By using a simple substitution of variables, (a_i for y), we know that:

$$\mathcal{H}(a_i, T) = \sum_{v \in dom(a_i)} \frac{\frac{q|U|}{|dom(a_i)|} - |\sigma_{a_i=v}(T' \uplus T^P)|}{q|U| - |T' \uplus T^P|} \log_2 \frac{\frac{q|U|}{|dom(a_i)|} - |\sigma_{a_i=v}(T' \uplus T^P)|}{q|U| - |T' \uplus T^P|}$$

[23] One of the issues with the information gain is that it has a strong bias in favour of attributes with many values.

Since we can express both the numerator and denominator of equation 3.26 as a function of $|U|$, T' and T^P, we know we can express the Gain Ratio of a_i and T without using T. Hence, the Gain Ratio metric is compatible with the unrealization approach.

Our next (and final) task is to show that the C4.5 pruning method can be accomplished with the use of the unreal data sets T' and T^P alone.

Pruning

As stated in Rokach [43], employing tight stopping criteria usually leads to small and underfitted decision trees. In contrast, loose stopping criteria leads to large trees that are overfitted to the training set. One approach to rectifying this solution is to take the tree generated by an induction algorithm and modify it to remove paths that do not contribute to the generalization accuracy. Not only can this improve the performance of the tree, but the pruning also aids in increasing the comprehensibility of the resulting model.[24]

Most pruning techniques work by performing top-down or bottom-up traversal of the nodes. An internal node will be pruned if the pruning operation improves a chosen metric. In the C4.5 approach, pruning a node from a tree results in replacing the node with a suitable leaf. This occurs even if the node has children, so that an entire subtree can be removed from a tree with a pruning operation.[25] We use the following function, which returns the tree d with node n replaced by a suitable leaf:

Algorithm:	$Prune(d, n)$
Inputs:	d: a decision tree. n: a node in the decision tree.
Outputs:	d': a new decision tree formed from d by replacing n with a leaf.

Table 3.7: Interface for the Prune Subroutine

[24] Among other reasons, large trees are much more difficult to understand.
[25] We assume some sort of garbage collection.

The pruning function (which we do not show explicitly) removes a node (potentially with subtrees as children) from the tree, replacing it with the most appropriate leaf value.

However, we still require an algorithm that makes the decision as to whether a given node should be pruned. That is the job of our next routine. The pruning procedure in the C4.5 algorithm performs a *bottom-up traversal* on all nodes in the decision tree. At each internal node, the pruning algorithm invokes a subroutine to determine whether pruning the node will result in a better classifier.

Algorithm:	*PruneAtNode*(d, n, T^S)
Inputs:	d: a decision tree. n: a node in the decision tree. S: the data that reaches node n.
Outputs:	none
1:	if (n.type = "leaf") return;
2:	Edge maxChildEdge = *BranchWithLargestInstanceCount*(n);
3	Node maxChild = *GetNode*(maxChildEdge);
4:	DTree pruned = *PruneTree*(d, n);
5:	Float $v1 = \varepsilon_{UB}(n, S)$;
6:	Float $v2 = \varepsilon_{UB}(pruned, S)$;
7:	Float $v3 = \varepsilon_{UB}$(d, max child,$\sigma_{maxChild=maxChildEdge.value}(S)$);
8:	$v = argmin(v1, v2, v3)$;
9:	if $(v = v1)$
10	return;
11:	if $(v = v2)$
12:	d $\leftarrow$ pruned;
13:	return;
14:	else
15:	n $\leftarrow$ maxChild;
16:	return;

Table 3.8: The C4.5 Tree Pruning Evaluation Algorithm

Our presentation of this algorithm is based on [43] at p.67. The procedure *Branch-WithLargestInstanceCount* returns the branch (edge) that accommodates the largest number of tuples from the training set. Using this routine returns the edge to the child node that takes on the largest number of tuples.

The routine evaluates three error metrics at each node. According to the lowest value, the routine either:

1. leaves the tree as is;
2. prunes the node n (and hence any subtree rooted at n) from the tree, or;
3. replaces n with the child node (subtree) that receives the most instances from n.[26]

In order to modify C4.5 to work with the unrealization approach to decision tree induction, it is sufficient to ensure that the error calculation routine ε_{UB} can be calculated from T', T and $|U|$ alone. In order to accomplish this task, we briefly discuss the concepts of *training error* and *generalization error*.

Training Error and Generalization Error

The *generalization error* of a decision tree $I(T)$ induced from training set T is its propensity to misclassify an instance, selected according to the distribution D of the universal instance space U [43, at p.21].

In contrast, the *training error* of a decision tree $I(T)$ induced from training set T is defined as the percentage of examples in the training set correctly classified by the tree. Formally:

$$\hat{\varepsilon}(I(T), T) = \frac{1}{|T|} \sum_{(x,y)\in T} L(y, I(S)(x)) \tag{3.27}$$

[26]Note that in our code we have assumed that the parameter representing the tree is passed by reference, so that it can be modified the routine.

Here, $I(S)(x)$ represents the predicted target attribute generated by decision tree $I(T)$ on input example $x \in X$. Recall that when we are classifying instances, we draw tuples from the instance space X, and not the universal instance space U.[27] The reason is that the instance space does not contain the target attribute, while the universal instance space does. The instance space is therefore the set of all possible unclassified examples.

The function $L(y, y') : dom(y) \times dom(y) \rightarrow \{0, 1\}$ is a binary function that calculates whether expected value y and predicted value y' agree. It is defined as:

$$L(y, y') = \begin{cases} 0 & \text{if } y = y', \\ 1 & \text{if } y \neq y'. \end{cases} \tag{3.28}$$

Unfortunately, a low training error does not guarantee low generalization error. As Rokach points out, there is usually a trade-off between the two.

The Error Metric in C4.5 Pruning

In the C4.5 algorithm, the error rate is estimated using the upper bound of the statistical confidence interval for proportions [43, at p.66]:

$$\varepsilon_{UB}(d, T) = \hat{\varepsilon}(d, T) + Z_{\alpha}\sqrt{\frac{\hat{\varepsilon}(d, T)(1 - \hat{\varepsilon}(d, T))}{|T|}} \tag{3.29}$$

Here, d is a node (representing a tree), T is the training set, Z is the inverse of the standard normal cumulative distribution, and α is the desired significance level.

As the reader can see, the C4.5 error metric is based on the training set error $\hat{\varepsilon}$ listed above in equation 3.27. In order to show that the unrealization approach can be extended to C4.5, all we need to do is show that we can calculate the training error $\hat{\varepsilon}$ for a tree using only T' and T^P, and not the actual training set T.

[27] For refreshers on these terms, see Section 3.1.3.

Calculating the Training Error Using Unreal Data

Earlier in this thesis, we showed that we can calculate entropies (and hence, *information gain* and *gain ratio*) for an attribute in a training set T using only the unreal data sets T' and T^P. For instance:

$$\mathcal{H}(y,T) = \sum_{v \in dom(y)} \frac{\frac{q|U|}{|dom(y)|} - |\sigma_{y=v}(T' \uplus T^P)|}{q|U| - |T' \uplus T^P|} \log_2 \frac{\frac{q|U|}{|dom(y)|} - |\sigma_{y=v}(T' \uplus T^P)|}{q|U| - |T' \uplus T^P|}$$

This expression involves a number of conceptual elements. The size of the universal instance space U can be stored in a global variable (or passed as a parameter), as it is calculated during execution of the unrealization algorithm in table 3.3. Likewise, the sizes of T', T^P and $T' \uplus T^P$ can also be calculated once, by the same unrealization algorithm.

However, the projection $\sigma_{y=v}(T' \uplus T^P)$ is impossible compute without the actual data in the multi-sets T' and T^P. The situation is obvious when one considers complex predicates, such as $|\sigma_{a_1=v_4^1 \wedge a_3=v_2^3 \wedge a_5=v_2^5}(T)|$. Calculating the size of such a multi-set requires access to the multi-set T.

Similarly, calculating the training error $\hat{\varepsilon}$ (equation 3.27) requires that we access each tuple in the training set. The problem we face in making the unrealization approach work in the C4.5 setting is that we do not have access to the training set; instead, we merely have access to the unreal data sets T' and T^P. This leaves us with two options:

1. Recreate the training data T prior to pruning the tree, using the reconstruction procedure detailed in Fong's thesis [20].

2. Compute the elements of T 'on the fly', in the course of executing the error calculation.

Since the first option would defeat the purpose of the unrealization approach (which is to keep the training set out of the hands of the data recipient), we describe how we can accomplish the second option.

Returning again to the C4.5 error metric in equation 3.29, we have:

$$\varepsilon_{UB}(d,T) = \hat{\varepsilon}(d,T) + Z_\alpha \sqrt{\frac{\hat{\varepsilon}(d,T)(1-\hat{\varepsilon}(d,T))}{|T|}}$$

We also have, from equation 3.27, the following expression for the error of the decision tree on the training set:

$$\hat{\varepsilon}(I(T),T) = \frac{1}{|T|} \sum_{(x,y)\in T} L(y, I(S)(x))$$

From lemma 2, we know that in the unrealization approach $|T| = |T'|$. This is good news, as we know the size of T'. Since we have the decision tree formed in the growing phase, it is also easy to calculate the loss function $L(y, I(S)(x))$ for any input tuple x. The problem is finding (x, y), given that we are not allowed to access the training set T.

Thankfully, we can reconstruct values of T on the fly, from T' and T^P. We know from theorem 3 that $T = q(U) - (T' \uplus T^P)$. If we could find q, we stand a chance of reconstructing T.

In fact, Fong [20] shows that:

$$q = \frac{2|T'| + |T^P|}{|U|}$$

As discussed, we have easy access to these values, making q trivial to compute. Recall that $q(U) - (T' \uplus T^P) = \{(x, max(x \in\in q(U) - x \in\in T' \uplus T^P, 0)) | x \in dom(\mathcal{R})\}$. By lemma 3, $T' \uplus T^P \subseteq q(U)$. That means that:

$$T = q(U) - (T' \uplus T^P) = \{(x, x \in\in q(U) - x \in\in T' \uplus T^P) | x \in dom(\mathcal{R})\}$$

Assume that the various multi-sets T, T', T^P are represented in set/pair fashion, with each tuple t that exists in multi-set S with frequency $t \in\in S = c$ appearing in the data structure for S as (t, c).[28]

[28]For instance, if $[a, a, a, b, b, c]$ is a multi-set S, then the data structure for S is a list $((a, 3), (b, 2), (c, 1))$.

In order to compute the training error $\hat{\varepsilon}$ on the set T, we simply walk each entry in the data structure for $T' \uplus T^P$. For each entry (t, c), we see if $q - c > 0$. If it is, we know that $t \in T$. In fact, t will appear in T exactly $q - c$ times. We use the tuple $t = (x, y)$ to calculate the loss function for the prediction of the tree on attribute array $x \in X$, by comparing the prediction $I(T)(x)$ to the observed target value y.

We can also use this approach to compute arbitrary predicates σ_ψ on T. In this case, we walk the data structure for $T' \uplus T^P$, selecting only those entries (t, c) for which: a) $q - c > 0$, and; b) $\phi(t[x])$ =true.

Algorithm:	*CalculateTrainingError*(d, T', T^P, ψ)
Inputs:	
	d: a decision tree for training set T. T': the unreal data set. T^P: the unreal perturbation set. ψ: an arbitrary predicate.
Outputs:	
	e: the training error.
1:	int errors = 0;
2:	int q = $(2\|T'\| + \|T^P\|)/\|U\|$;
3:	for each $((t = (x, y), c) \in (T' \uplus T^P))$
4:	{
5:	if $(\psi(t[x]) \wedge (q - c > 0))$
6:	{
7:	errors ← errors + (q-c)(*LossFunction*(*Classify*(d,t[x]),t[y]));
8:	}
9:	}
10:	return errors;

Table 3.9: Training Error Calculation

This algorithm returns the sum of the errors generated for those tuples in $T' \uplus T^P$ that: a) meet the predicate ψ, and; b) exist in T. The algorithm computes the number of instances of the tuple that exist in T 'on the fly', without recreating an explicit reconstruction of T.

With this result, we have shown that we can compute the error for the C4.5 pruning approach using only T' and T^P. This finishes our demonstration that the unrealization approach can be extended to the C4.5 algorithm.

Chapter 4

Evaluation, Analysis and Comparisons

This Chapter contains two Sections. The first evaluates the unrealization approach on real-world data sets. The second evaluates the unrealization approach on its space/time requirements, and privacy preservation abilities.

4.1 Illustrations on Real-Life Data Sets

In this section, we review three data sets against the unrealization algorithm outlined above. Our focus is on the storage requirements for the universal instance space U, since that set dominates both the training set T and the unreal data set T'. Since the reader is already aware that the storage requirements for U are exponential, the fact that the unrealization algorithm has prohibitive storage costs will not be a surprise. However, a few choice illustrations will drive home the point of how bad the situation really is, with respect to using the algorithm on real world data sets.

Two of our examples are drawn from the Machine Learning Repository maintained by the Center for Machine Learning and Intelligent Systems (CMLIS) at the University of California, Irvine. The third was derived from the British Columbia ("BC") Surgical Wait Times database. Each of these examples is a real-world data set that a machine learning algorithm should be able to handle.

4.1.1 Breast Cancer Data

The Breast Cancer Data Set is one of many health-related data sets maintained by CMLIS. It contains 286 examples, on a schema containing 9 attributes. Provided by the Institute of Oncology at the University Medical Center in Ljubljana, Yugoslavia, this data set is a staple in the machine learning literature. Since this data set would have originally contained PHI, the original version would have been highly sensitive. The following table details its schema:

Attribute	$\|dom(\text{Attribute})\|$	Bits Required
Age	9	4
Menopause	3	2
Tumour Size	12	4
Inv-Nodes	13	4
Node-Caps	2	1
Deg-Malig	3	2
Breast	2	1
Breast Quadrant	4	2
Irradiat	2	1
Target (y)	2	1

Table 4.1: The Breast Cancer Schema from the CMLIS.

For sake of brevity, we only detail the domain size for each attribute. We also give the size of the integer (in bits) that is required to encode the values for a given domain. We use these values as a lower bound in determining the storage requirements of a tuple from this data set.[1]

The size of the universal instance space U for this data set is $|U| = \prod_{i=1}^{10} |dom(a_i) = 808,704$. The number of bits required to represent all attributes via 'bit packing' is 22. Note that if each instance is stored as a separate integer, the minimum number of bits would be 32. We stick with 22 to provide an extremely aggressive lower bound.

At 22 bits per row, this dataset will require around under 2 Megabytes to store. This is an entirely feasible storage requirement, since modern computers possess gigabytes of memory. Unfortunately, we will see in our next example that reasonable requirements do not arise from the other real-world data sets.

[1]The actual storage requirements in a real database management system would be much higher, due to minimum sizes for integer attributes, block sizes on hard disk drives, and other factors.

4.1.2 Audiology Data

The Audiology database contains the responses to a diagnostic hearing test. Larger than the breast cancer schema, it is depicted below:[2]

A_i	$\|dom(A_i)\|$	Bits Required	A_i	$\|dom(A_i)\|$	Bits Required
age_gt_60	2	1	m_sn_2_3k	2	1
air	5	3	m_sn_gt_1k	2	1
airBoneGap	3	2	m_sn_gt_2k	2	1
ar_c	3	2	m_sn_gt_3k	2	1
ar_u	3	2	m_sn_gt_4k	2	1
bone	3	2	m_sn_gt_500	2	1
boneAbnormal	2	1	m_sn_gt_6k	2	1
bser	4	2	m_sn_lt_1k	2	1
history_buzzing	2	1	m_sn_lt_2k	2	1
history_fluctuating	2	1	m_sn_lt_3k	2	1
history_fullness	2	1	middle_wave_poor	2	1
history_heredity	2	1	mod_gt_4k	2	1
history_nausea	2	1	mod_mixed	2	1
history_noise	2	1	mod_s_mixed	2	1
history_recruitment	2	1	mod_s_sn_gt_500	2	1
history_ringing	2	1	mod_sn	2	1
history_roaring	2	1	mod_sn_gt_1k	2	1
history_vomiting	2	1	mod_sn_gt_2k	2	1
late_wave_poor	2	1	mod_sn_gt_3k	2	1
m_at_2k	2	1	mod_sn_gt_4k	2	1
m_cond_lt_1k	2	1	mod_sn_gt_500	2	1
m_gt_1k	2	1	notch_4k	2	1
m_m_gt_2k	2	1	notch_at_4k	2	1
m_m_sn	2	1	o_ar_c	3	2
m_m_sn_gt_1k	2	1	o_ar_u	3	2
m_m_sn_gt_2k	2	1	s_sn_gt_1k	2	1
m_m_sn_gt_500	2	1	s_sn_gt_2k	2	1
m_p_sn_gt_2k	2	1	s_sn_gt_4k	2	1
m_s_gt_500	2	1	speech	6	3
m_s_sn	2	1	static_normal	2	1
m_s_sn_gt_1k	2	1	tymp	5	2
m_s_sn_gt_2k	2	1	viith_nerve_signs	2	1
m_s_sn_gt_3k	2	1	wave_V_delayed	2	1
m_s_sn_gt_4k	2	1	waveform_ItoV_prol.	2	1

Table 4.2: The Audiology Schema from the CMLIS.

[2]We have used two-column format in order to fit the entire schema on one page.

This schema is fairly representative of clinical health information surveys. The audiology diagnostic test obviously included a large number of 'true/false' questions, yielding many attributes with a binary domain. The target attribute y of the data set[3] indicates the diagnosis of the test. It consists of 19 possible outcomes, instead of the simple 'yes/no' answer involved in the breast cancer database.

The size of the universal instance space U on this data set is around 2.13×10^{24}. At 85 bits per tuple, the following table indicates the storage space required to represent the universal instance space (rounded to two significant digits):

Unit	Units Required
Bit	1.81×10^{26}
Byte	2.26×10^{25}
KiloByte	2.26×10^{22}
MegaByte	2.26×10^{19}
GigaByte	2.26×10^{16}
TeraByte	2.26×10^{13}

Table 4.3: Storage Requirements for Audiology Schema

As one can see from the table, the storage requirements for this relatively intuitive data set are astronomical.[4] In order to show that this result is not unique, we present a final data schema drawn from the Surgical Wait Times database that is in current use in British Columbia.

[3]We did not show the target in the table above, as it would have made the last line of the table somewhat awkward.

[4]We use the term 'astronomical' literally, as the life age of the universe is a low number compared to the size of the universal instance space.

4.1.3 Surgical Wait Times Data

In this section, we present a schema that we drew from the BC Surgical Wait Times database.[5] The goal of the classification task is to predict which patients will cancel their surgeries, based on the values of attributes relating to surgical procedures, wait times and geographical location. The schema appears in the following table:

Attribute	$\|dom(\text{Attribute})\|$	Bits Required
Health Authority	5	3
Postal FSA	190	8
Referring Facility	300	9
Primary Procedure	20	5
Secondary Procedure	20	5
Procedure Wait Time (months)	12	4
Months Since Booking	12	4
Assessment	2	1
Cancer Proven or Suspected	2	1
Sex	4	2
Age	10	4
Target (y): Cancels	2	1

Table 4.4: Surgical Wait Times

Unlike the CMLIS examples presented above, we possess a full data dictionary for this schema. As a result, we are in a position to make some clarifying remarks about the attributes:

- There are five regional *health authorities* in British Columbia. Each governs hospitals, clinics and other facilities.
- A postal code *forward sorting area* ("FSA") corresponds to the first three digits of a Canadian postal code. Since there are 850,000 active postal codes in Canada, categorising a location by a full postal code is costly. As an alternative, using an FSA is a means of assigning individual postal codes into a smaller set of categories.
- The *primary and secondary procedure* attributes give the type of procedures being performed on the patient: (i.e., cardiac stent insertion, angioplasty). Values

[5] I would like to thank Dr. Erdem Yazganoglou of the PHSA for allowing us access to the database schema.

here come from the ICD-9 coding system.

- The *assessment* attribute indicates whether the patient has undergone an assessment, prior to scheduling their surgery.
- The *sex* attribute has four possible values, since BC recognizes alternative forms of sexuality beyond male and female.

In this schema, the size of the universal instance space is approximately 2.00×10^{15}. Each tuple requires 56 bits to store. As before, we represent the size of the universal instance space in a table:

Unit	Units Required
Bit	1.12×10^{17}
Byte	1.40×10^{16}
KiloByte	1.40×10^{13}
MegaByte	1.40×10^{10}
GigaByte	1.40×10^{7}
TeraByte	1.40×10^{4}

Table 4.5: Storage Requirements for Surgical Wait Times

In short, we could store the universal instance space for this schema on 14,000 one-terabyte hard disk drives. It is obvious that this sort of storage requirement is outside the realm of practical application, on today's hardware. Even worse, the exponential growth in storage requirements means that advances in storage methods can be quickly wiped out by a small increase in the size of the schema.

In the next Section, we discuss some important consequences of the unrealization approach invented by Fong [20]. We detail *resource requirements*, and discuss the algorithm's impact on the various challenges posed by data mining to informational privacy interests. While the unrealization approach mitigates some of the problems caused by data mining, it suffers from some severe problems that make it unusable in all but a small number of circumstances.

4.2 Resource Requirements

We discuss two types of resource requirements: *time* and *space*.

4.2.1 Time Complexity

The main issue with the unrealization approach from a practical perspective is its brutally high time and storage costs. The worst case time complexity for the *UnrealizeTrainingSet*[6] function (which creates the unreal data sets T and T^P from the training data T) is $O(m^n)$, where:

1. $n \in \mathbb{N}$ is the finite cardinality of the set of attributes $A = \{a_1, a_2, \ldots, a_n\}$
2. $m = max(|dom(a_1)|, |dom(a_2)|, \ldots, |dom(a_n)|)$

In the later chapters of his thesis [20], Fong introduces modifications to the unrealization algorithm that permit the algorithm to work with dummy attributes. The time complexity is still $O(m^n)$.

4.2.2 Storage Requirements

In a similar fashion, the storage requirements are quite exacting. Although it may be possible to avoid storing the universal instance space U explicitly (by computing it 'on the fly'), the unreal data sets T' and T^P can approach U in size. Even worse, Fong recommends later in his thesis that the training set T be larger than the size of the universal instance space U. Hence, the storage requirements for the data set are $\Omega(m^n)$, where:

1. $n \in \mathbb{N}$ is the finite cardinality of the set of attributes $A = \{a_1, a_2, \ldots, a_n\}$
2. $m = min(|dom(a_1)|, |dom(a_2)|, \ldots, |dom(a_n)|)$

[6] See table 3.3

4.2.3 Impact

Algorithms with exponential time complexity are considered to be *intractable*. It is clear that without a means of avoiding exponential time complexity, the unrealization approach faces fatally demanding storage and time demands.

Approaching the same issue from a practical standpoint, in the previous Section we computed storage requirements for three real-world data sets. Only one of these data sets had a representation that was feasible, while the other two involved storage requirements that were astronomical. It is clear that the unrealization approach is only applicable to data sets with small footprints.

4.3 Privacy Preservation

On page 72 of his thesis [20], Fong introduces a metric to calculate the impact of privacy losses. He notes that there are at least two potential privacy issues of the unrealization approach. First, privacy is not preserved well if the variance of the frequency of a tuple in the training set T is low. Second, unrealization actually increases the possibility of privacy loss for tuples that appear in the training set with low frequency. Fong introduces dummy attributes / attribute values to deal with this issue.[7]

Since Fong has already discussed privacy loss from a mathematical standpoint, we extend his work by discussing unrealization within the context of our previous analysis of the challenges that data mining poses for informational privacy interests. We finish the chapter with a discussion of a major weakness of the unrealization approach from a privacy standpoint –the presence of a data reconstruction algorithm to recover the original training set T from unreal data sets T' and T^P.

[7]Due to time constraints, we will not be able to put Fong's analysis of the privacy preserving properties of unrealization on firmer ground.

4.3.1 Mitigating the Privacy Risks of Data Mining

In Section 2.2.3, we gave a novel and accurate account of the challenges that data mining poses for modern data protection regimes. We mirror the presentation style used in that portion of this thesis, analyzing the mitigating effect that the unrealization approach has on each privacy risk.

Challenge 1: Secondary Uses

As we noted above, the primary challenge that data mining poses to data protection regimes arises from the exploratory nature of data mining. As stated by the Ontario Privacy Commissioner [10], identifying a primary purpose at the beginning of the process, (and then restricting one's use of the data to that purpose), is the "antithesis" of data mining, which aims to discover previously unknown patterns in the data. As a result of this goal, secondary use violations are to be expected when data is released for data mining

The unrealization approach provides a partial answer to the question of secondary uses. The unreal data sets that are released to the organization performing data mining do not contain actual data. A third party (adversary) would not be able to figure out what information in the unreal data sets was in the original training set. The unrealization approach therefore provides protection from secondary uses by a nave user.

In contrast, a non-nave user (who understands the unrealization process, and who possesses software implementing a modified decision tree algorithm that can deal with unrealized data) may obtain the same decision tree from the unreal data that they would have obtained from the actual data set. The method does not limit what can be discovered from the sensitive data; it merely ensures that we don't have to have the sensitive data on hand to perform discovery.

Lastly, the sensitive data can be recovered from the unreal data sets by means of a simple data reconstruction transformation. The unreal data sets provide a certain measure of security against nave users, but not against those who know about the reconstruction transformation. We will have more to say about this issue below.

Challenge 2) The Legal Status of Intermediate Work Products

We noted above that the intermediate work products (cleansed, transformed and modified data sets) created in the data mining process can hold sensitive information. The legal status of these work products is important, as an organization can take advantage of a loophole in current privacy and copyright law by the use of sufficiently clever transformations.

The unrealization process actually counters this challenge, provided that the receiving organization is not aware of the data reconstruction transformation that can be performed on the unreal data to recover the original sensitive data set. The unreal data sets are useless to a nave user, in the absence of this reconstruction transformation. As a result, any further transformations of the unreal data sets (through coding, projection, selection and other means) are also useless.

Challenge 3) The Legal Status of Models

This challenge concerned the status that data mining models have under privacy and copyright law. The issue is not merely academic, as the ability of a data mining model to overfit the training set means that the model can actually represent the original training data to some degree.

Unfortunately, unrealization does not help with this issue. The model obtained from the unreal data sets is the same as would have been obtained from the sensitive data. Both the standard ID3 algorithm and its data mining variant face the same set of problems, in this respect.

Challenge 4) Defeating Anonymization

Unrealization helps somewhat with this issue, assuming that the data recipient (the person performing the mining) lacks knowledge of the reconstruction transformation. Since one means of using background knowledge to defeat anonymization consists of linking data between different databases, the unrealization approach provides some protection. A user who is looking to match data between an existing database and one of the unreal data sets will be frustrated beyond belief.

Challenge 5)

Unrealization does not have much of an impact on this challenge. The probabilistic nature of inferences made from a data mining model are not altered by the presence of unreal data sets.

Challenge 6) Comprehensibility

Unrealization does not change the model that is constructed from the data, so the comprehensibility of the model is the same under unrealization as it would be in the standard data mining paradigm.

Despite this observation, the unrealization approach may actually reduce comprehensibility in some ways. Given that individuals have a right to know if an organization has data on them, the unreal data sets pose a bit of a problem. Unreal data sets T' and T^P can contain information about individuals, even though that information is not represented explicitly. Instead, the information may be extracted during the decision tree induction process, or by using the data reconstruction transformation to reconstruct the original data set. While transparent to data recipients who are familiar with the unrealization approach, this situation is likely incomprehensible to the layperson.

Challenge 7) Existence, Access and Correction

Unrealization exacerbates difficulties with respect to this challenge. As we noted above, a data controller must disclose whether or not it has control of data relating to an individual. Unreal data sets T' and T^P can contain information about individuals, even though that information is not represented explicitly.[8] An organization cannot tell from looking at the unreal data sets whether it has information pertaining to a given individual. Instead, it must perform the data reconstruction transformation. This extra step could cause headaches.

[8]The data can be obtained from reconstruction, or by building an overfitted model.

Second, an individual has a right to correct or amend personal information in the custody and control of an organization. Correcting information in an unreal data set is difficult, since the data is not represented explicitly. In the absence of further research on the topic, the most obvious way to correct data is to amend it in the original training set, and run the unrealization algorithm again. If the organization in possession of the unreal data sets is a third party that does not have access to (or knowledge of) either the unrealization software or the original data set, this approach will be impossible.

Challenge 8) Records in the Training Set

Recall that possession of the model and knowledge that a tuple was in the training set gives an adversary knowledge about the classification of that tuple. Unrealization does not help with this issue. Since the decision tree produced from the unreal data sets is the same as the one that would be produced from the sensitive data, the same problem arises.

4.3.2 Reconstruction

The potential privacy pitfalls noted by Fong (as well as those hilighted in our discussion above) appear benign when compared to one of the most salient features of the unrealization approach. As shown by Fong on page 70 of this thesis, the original data set may be reconstructed from the universal instance space U, the unreal data set T and the perturbing data set T^P. The complexity of the transformation is approximately $O(m^n)$, where:

1. $n \in \mathbb{N}$ is the finite cardinality of the set of attributes $A = \{a_1, a_2, \ldots, a_n\}$
2. $m = max(|dom(a_1)|, |dom(a_2)|, \ldots, |dom(a_n)|)$

As we noted, this sort of complexity is prohibitive. However, for small data sets, the transformation is quite painless, as it involves simple arithmetic operations on the data set. This means that the goal of preserving privacy by giving recipient organizations only the unreal data sets is made much more complex. If the recipient organization has the requisite knowledge about the reconstruction procedure, (as well as the full data sets T' and T^P it can perform the transformation to recover the original data set T.

This feature of the unrealization approach means that additional safeguards are needed. One option consists of releasing only a portion of the data sets T' and T^P to an attacker.[9] In a more speculative vein, future research could be performed on the use of secure multi-party computation mechanisms within the context of the unrealization framework.

[9] In such a case, there would obviously be an impact on the basic usage scenario that we have assumed throughout the entirety of this thesis.

Chapter 5

Conclusions

As stated in the Introduction, our contribution in this thesis consists of:

1. Providing the most up-to-date and accurate analysis of the challenges that data mining poses to modern data protection regimes.
2. Putting the unrealization approach outlined in Fong [20] on more mature footing, by providing an axiomatization of the multi-relational calculus, as well as proofs of claims that were merely asserted in the original presentation.
3. Extending the unrealization approach to the industry-standard C4.5 algorithm, from the rarely-used ID3 approach.
4. Evaluating the unrealization approach against several real-world data sets.
5. Providing an evaluation of the merits of the unrealization approach, with respect to both privacy preservation and space/time requirements. In particular, we analyzed the unrealization approach against our formulation of the difficulties that data mining poses to privacy.

In this closing chapter, we summarize the work performed in this thesis. The next pages contain a brief discussion of the key achievements.

5.1 Summary of the Results

We have covered a fair bit of ground in this work. This section summarizes our discussion.

Background: Privacy and Data Mining

We began with a discussion of the basics of *informational privacy*, including its rationales and key concepts. Introducing the fair information practices that form the heart of modern data protection, we outlined some of the key challenges to privacy in the 21st century. We saw that data mining is one of several developments in information technology that poses issues for data protection regimes.

After the introductory section on privacy, we provided an overview of *data mining*, with an emphasis on inductive learning techniques. We detailed the data mining process, including the various artifacts that are created in the course of data mining activities. After introducing decision trees, we engaged in a presentation of the key challenges that data mining poses to informational privacy regimes. As we noted, many of the treatments in the literature fail to deliver on their promises; instead of analyzing the impact of data mining on privacy, they discuss the difficulties that data protection regimes pose for data mining activities. We rectified this deficiency by means of a thorough discussion that was informed by a solid understanding of both fair information principles and data mining.

We followed this with a short section on *privacy preserving data mining*. Introducing the key concepts that various commentators have used to classify privacy preservation techniques, we provided a brief overview of the field.

The Unrealization Approach

While this background chapter contained largely non-technical content, we introduced a significant number of proofs in the chapter on the unrealization approach to privacy preservation created by Fong [20]. We began by shoring up the fundamentals, introducing a formalization of *multi-relational algebra* sufficient to support explicit proofs. We also introduced the basic framework for top-down induction of decision trees.

Next, we moved to a treatment of the *unrealization approach* to decision-tree induction. Our main goal was to substantially improve the presentation in [20], by providing rigorous proofs of key claims that were unproven (or proven badly) in the original work. In the end, we showed that the unrealization approach constructs the same decision tree from the unreal data sets that a standard ID3 algorithm would create from the original training set.

The next section explored the possibility of extending the unrealization approach from the infrequently-used ID3 algorithm to the C4.5 approach that is popular in practise. We saw that it was fairly easy to extend the unrealization algorithm to C4.5. This result likely holds of other state-of-the-art decision tree approaches.

Evaluation

In order to provide an evaluation of the unrealization approach, we introduced three real-world data sets. Two were drawn from the machine learning repository at the University of California, Irvine, while the third was taken from the British Columbia Surgical Scheduling database program. We showed that the storage demands imposed by the unrealization approach for two of the data sets are astronomical.

In addition, the evaluation section included a discussion of privacy preservation, as well as a brief treatment of space and time requirements. We evaluated the unrealization approach against the issues raised in our analysis (Section 2.2.3) of the challenges that data mining poses to modern data protection regimes. We found that unrealization does help to address several of these issues, particularly those involving risks arising from intermediate work products.

However, our discussion of time and space requirements made it clear that the unrealization approach is completely infeasible for all but the most concise data sets. With exponential time complexity, and exponential data set sizes, two of the sample data sets introduced in Chapter 4 were utterly impossible to treat with an unrealization approach.

We also noted that even if the space and time requirements were reasonable, the presence of a data reconstruction transformation makes life difficult for data custodians who wish to disclose information in confidence. Further research is required to evaluate solutions to this problem.

5.2 Concluding Remarks

The presentation of the unrealization algorithm that appears in this work is much more rigorous and readable than the one contained in Fong's thesis [20]. It may be of use to researchers who are hoping to follow up on Fong's innovative work. We supplied many proofs of key claims that were merely asserted in the original presentation of the unrealization approach. While the exposition may have been tedious at times, I felt it was important to put this approach on firm theoretical ground.

Due to time constraints, we did not discuss all of the content of Fong's thesis. In particular, we avoided a discussion of the privacy preservation metric, and of the dummy attribute approach. Thankfully, these features do not change our ultimate conclusions regarding the merits of the unrealization approach. In the end, the unrealization algorithm is a novel and interesting idea that (while not feasible in its current form on any but small data sets) may spur the development of a new family of approaches to privacy preserving data mining.

Bibliography

[1] Nabil R. Adam and John C. Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Comput. Surv.*, 21(4):515–556, 1989.

[2] Charu C. Aggarwal and Philip S. Yu. A general survey of privacy-preserving data mining models and algorithms. In Ahmed K. Elmagarmid, Amit P. Sheth, Charu C. Aggarwal, and Philip S. Yu, editors, *Privacy-Preserving Data Mining*, volume 34 of *Advances in Database Systems*, pages 11–52. Springer US, 2008.

[3] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. An xpath-based preference language for p3p. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 629–639, New York, NY, USA, 2003. ACM.

[4] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. *SIGMOD Rec.*, 29(2):439–450, 2000.

[5] Joseph Albert. Algebraic properties of bag data types. In *VLDB '91: Proceedings of the 17th International Conference on Very Large Data Bases*, pages 211–219, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

[6] Michael Backes, Günter Karjoth, Walid Bagga, and Matthias Schunter. Efficient comparison of enterprise privacy policies. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 375–382, New York, NY, USA, 2004. ACM.

[7] Victoria Bellotti and Abigail Sellen. Design for privacy in ubiquitous computing environments. In *ECSCW'93: Proceedings of the third conference on European Conference on Computer-Supported Cooperative Work*, pages 77–92, Norwell, MA, USA, 1993. Kluwer Academic Publishers.

[8] Alastair R. Beresford and Frank Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[9] Barbara Carminati, Elena Ferrari, and Andrea Perego. Private relationships in social networks. In *ICDEW '07: Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, pages 163–171, Washington, DC, USA, 2007. IEEE Computer Society.

[10] Ann Cavoukian. *Data Mining: Staking a Claim to your Privacy.* Office of the Information and Privacy Commissioner of Ontario, 1996.

[11] Bee-Chung Chen, Daniel Kifer, Kristen LeFevre, and Ashwin Machanavajjhala. Privacy-preserving data publishing. *Found. Trends databases*, 2(1–2):1–167, 2009.

[12] Chris Clifton and Don Marks. Security and privacy implications of data mining. In *In ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 15–19, 1996.

[13] United States Privacy Protection Study Commission. *Personal Privacy in an Information Society.* USGPO, Washington: DC, 1977.

[14] Jack S. Cook and Laura L. Cook. Social, ethical and legal issues of data mining. In *Data Mining: Opportunities and Challenges.* Idea Group Publishing, Hershey ; PA:, 2003.

[15] Judge Thomas M Cooley. *Treatise on Torts 2e.* Cambridge University Press, New York ; NY:, 1888.

[16] L Cranor, M Langheinrich, and M Marchiori. *A P3P Preference Exchange Language 1.0.* W3C Working Draft, 2002.

[17] Dorothy E. Denning and Peter J. Denning. The tracker: a threat to statistical database security. *ACM Trans. Database Syst.*, 4(1):76–96, 1979.

[18] Will Thomas DeVries. Annual review of law and technology: Privacy - protecting privacy in the digital age. *Berkeley Technology Law Journal*, 18:283, 2003.

[19] Cynthia Dwork and Adam Smith. Differential privacy for statistics: What we know and what we want to learn. *Journal of Privacy and Confdentiality*, 1(2), 2009.

[20] Pui Fong. Privacy preservation for training datasets in database:application to decision tree learning. Master's thesis, University of Victoria, 2008.

[21] Organization for Economic Cooperation and Development. *Guidelines Governing the Protection of Privacy and Transborder Data Flows of Personal Data*. OECD ; Switzerland:, 1980.

[22] Paul W. P. J. Grefen and Rolf A. de By. A multi-set extended relational algebra - a formal approach to a practical issue. In *Proceedings of the Tenth International Conference on Data Engineering*, pages 80–88, Washington, DC, USA, 1994. IEEE Computer Society.

[23] Paolo Guarda and Nicola Zannone. Towards the development of privacy-aware systems. *Inf. Softw. Technol.*, 51(2):337–350, 2009.

[24] Carl S. Guynes, Glenn E. Maples, and Victor R. Prybutok. Privacy issues in statistical database environments. *SIGCAS Comput. Soc.*, 25(4):3–5, 1995.

[25] Carl Stephen Guynes. Protecting statistical databases: a matter of privacy. *SIGCAS Comput. Soc.*, 19(1):15–20, 1989.

[26] Jason I. Hong and James A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 177–189, New York, NY, USA, 2004. ACM.

[27] John B. Kam and Jeffrey D. Ullman. A model of statistical database their security. *ACM Trans. Database Syst.*, 2(1):1–10, 1977.

[28] Günter Karjoth, Matthias Schunter, and Michael Waidner. Platform for enterprise privacy practices: privacy-enabled management of customer data. In *PET'02: Proceedings of the 2nd international conference on Privacy enhancing technologies*, pages 69–84, Berlin, Heidelberg, 2003. Springer-Verlag.

[29] Ling Liu. Privacy and location anonymization in location-based services. *SIGSPATIAL Special*, 1(2):15–22, 2009.

[30] Matthew M. Lucas and Nikita Borisov. Flybynight: mitigating the privacy risks of social networking. In *WPES '08: Proceedings of the 7th ACM workshop on Privacy in the electronic society*, pages 1–8, New York, NY, USA, 2008. ACM.

[31] Tinghuai Ma, Shin-Dug Kim, Jun Wang, and Yawei Zhao. Privacy preserving in ubiquitous computing: Challenges & issues. In *ICEBE '08: Proceedings of the 2008 IEEE International Conference on e-Business Engineering*, pages 297–301, Washington, DC, USA, 2008. IEEE Computer Society.

[32] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1):3, 2007.

[33] Oded Maimon and Rokach Lior. Introduction to knowledge discovery in databases. In *Data Mining and Knowledge Discovery Handbook*, page 1. Springer Berlin / Heidelberg, 2005.

[34] Oded Maimon and Rokach Lior. Introduction to supervised methods. In *Data Mining and Knowledge Discovery Handbook*, page 149. Springer Berlin / Heidelberg, 2005.

[35] Department of Health Education and Welfare. *Records, Computers and the Rights of Citizens*. Massachusetts Institute of Technology, Cambridge: MA, 1973.

[36] Paul Ohm. Broken promises of privacy: Responding to the surprising failure of anonymization. *UCLA Law Review*, 57:1701, 2010.

[37] Daniel E. O'Leary. Some privacy issues in knowledge discovery: The oecd personal privacy guidelines. *IEEE Expert: Intelligent Systems and Their Applications*, 10(2):48–52, 1995.

[38] Stanley R. M. Oliveira and Osmar R. Zaane. Toward standardization in privacy-preserving data mining. In *In Proc. of the 3nd Workshop on Data Mining Standards (DM-SSP 2004), in conjuction with KDD 2004*, pages 7–17, 2004.

[39] Commission on Freedom of Information and Individual Privacy. *Report on the Commission on Freedom of Information and Individual Privacy*. Queen's Pritner for Ontario, Toronto ; ON:, 1980.

[40] J. R. Quinlan. Induction of decision trees. *Mach. Learn*, pages 81–106, 1986.

[41] J. Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[42] Lior Rokach and Oded Maimon. Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on Systems, Man and Cybernetics*, 35(4):476, 2005.

[43] Lior Rokach and Oded Maimon. *Data Mining with Decision Trees: Theory and Applications*. World Scientific, 2007.

[44] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition, 2003.

[45] Warren S.D. and Brandeis L.D. The right to privacy. *Harvard Law Review*, 4(5):193–220, 1890.

[46] Arthur Shafer. Privacy: A philosophical overview. In *Aspects of Privacy Law*. Butterworths, Toronto ; ON:, 1981.

[47] Daniel J. Solove. Privacy and power: Computer databases and metaphors for information privacy. *Stanford Law Review*, 53(3):1393, 2001. GWU Law School Public Law Research Paper No. 129.

[48] Daniel J. Solove. *Understanding privacy*. Harvard University Press, Cambridge ; MA:, 2008.

[49] S. Sumathi and S. Sivanandam. Data mining and data warehousing. In *Introduction to Data Mining and its Applications*, volume 47 of *Studies in Computational Intelligence*, pages 415–475. Springer Berlin / Heidelberg, 2007.

[50] S. Sumathi and S. Sivanandam. Major and privacy issues in data mining and knowledge discovery. In *Introduction to Data Mining and its Applications*, volume 47 of *Studies in Computational Intelligence*, pages 271–291. Springer Berlin / Heidelberg, 2007.

[51] Latanya Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, 2002.

[52] Peter P. Swire and Sol Bermann. *Information Privacy: Official Reference for the Certified Information Privacy Professional*. International Association of Privacy Professionals, York ; ME:, 2007.

[53] Kim Taipale. Data mining and domestic security: Connecting the dots to make sense of data. *Columbia Science and Technology Law Review*, 5, 2003.

[54] Kim Taipale. Technology, security and privacy: the fear of frankenstein, the mythology of privacy and the lessons of king ludd. *Yale Journal of Law and Technology*, 124, January 2004.

[55] Vassilios S. Verykios, Elisa Bertino, Igor Nai Fovino, Loredana Parasiliti Provenza, Yucel Saygin, and Yannis Theodoridis. State-of-the-art in privacy preserving data mining. *SIGMOD Rec.*, 33(1):50–57, 2004.

[56] Alan Westin. *Privacy and Freedom*. Atheneum, New York ; NY:, 1967.

[57] James Williams. Social networking applications in health care: threats to the privacy and security of health information. In *SEHC '10: Proceedings of the 2010 ICSE Workshop on Software Engineering in Health Care*, pages 39–49, New York, NY, USA, 2010. ACM.

[58] Bin Zhou, Jian Pei, and WoShun Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *SIGKDD Explor. Newsl.*, 10(2):12–22, 2008.