Security of Genetic Databases

Helen Patricia Giggins

B. Comp. Sci.(Hons)

A thesis submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

School of Electrical Engineering & Computer Science
University of Newcastle
Callaghan, 2308
Australia

April 2009

THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

Statement of Originality

The thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying subject to the provisions of the Copyright Act 1968.

(Signed) _____

Helen Patricia Giggins

Acknowledgements

Having invested such a large amount of time and energy into the production of this thesis, I have come to realise that it has also encroached upon the lives of many of my colleagues, friends and family. I want to acknowledge their support and extend my deepest gratitude to everyone who has helped this little dream of mine become a reality.

To my close friends and family, I thank you for your patience and understanding of my long absences and frequently patchy contact. Knowing that you were there when I needed you made the process that much less daunting and painful, indeed it often felt like you were all along for this roller-coaster ride with me.

I want to thank the many academic staff in our department who have offered me advice and employment over the period of my candidature. I would also like to thank the many support staff in the department, especially Aaron Scott, David Montgomery, Geoff Martin and Trevor Nelson, who have always been so willing and happy to help me on many occasions. Many of the staff and my friends from Uni were always on hand to provide some much welcome distraction or advice when needed, in particular I'd like to mention Drew Mellor, Lee-Anne Marsh and Paul CayfordHowell who performed these duties with finesse.

To my academic siblings Zahid, Mousa, Tanya and Luke. The knowledge that you were traveling the same road as myself was a big comfort and support to me. I extend my special appreciation to Mousa and Zahid whom I shared office space with during various stages of my candidature. Your company, support and understanding were invaluable to me and I hope that we remain strong friends and colleagues for many years to come.

To my co-supervisor Prof. Elizabeth Chang, I extend my warm thanks for your support and encouragement during my candidature. I would also like to acknowledge the financial support provided by the ARC Discovery project (DP0452182,"Privacy in Genetic Databases" L. Brankovic). Last, but by no means least, my deepest and most heartfelt gratitude goes to my supervisor Assoc. Prof. Ljiljana Brankovic who has been a mentor, supervisor and a dear friend to me over the last six years. Your patience and guidance were invaluable and I look forward to our continued collaboration and friendship long into the future.

This thesis is gratefully dedicated to

Michael James

Thank you for giving me the freedom to find my way.

"Bear in mind that the wonderful things you learn in your schools are the work of many generations. All this is put in your hands as your inheritance in order that you may receive it, honor it, add to it, and one day faithfully hand it on to your children."
(Albert Einstein)

List of publications arising from this thesis

1. H. Giggins, L. Brankovic, Protecting Privacy in Genetic Databases, *Proceedings of the Sixth Engineering Mathematics and Applications Conference*, 7-11 July, Sydney, Australia, pp. 73-78, 2003.

2. L. Brankovic and H. Giggins, Statistical Database Security, in *Security, Privacy and Trust in Modern Data Management*, M. Petrovic and W. Jonker (eds), Springer, 2007

3. H. Giggins and L. Brankovic. Statistical Disclosure Control: To Trust or Not to Trust, *Proceedings of International Symposium on Computer Science and its Applications*, 13-15 October, Hobart, Australia, 2008.

List of other publications arising during PhD canditure

1. M. Alfalayleh, L. Brankovic , H. Giggins, M. Z. Islam, Towards the Graceful Tree Conjecture: A Survey, *Proceedings of Fifteenth Australasian Workshop on Combinatorial Algorithms (AWOCA2004)*, 6-9 July, Ballina, Australia, pp. 329-247, 2004.

2. L. Brankovic, Z. Islam and H. Giggins, Privacy-Preserving Data Mining, in *Security, Privacy and Trust in Modern Data Management*, M. Petrovic and W. Jonker (eds), Springer, 2007

Contents

Abstract **xiii**

1 Introduction **1**
1.1 Genetic Information Challenges 2
1.2 Research Questions 5
1.3 Thesis Overview 6

2 Trust in Genetic Databases **9**
2.1 Introduction 10
2.2 Trust 13
2.2.1 Related Concepts 15
2.2.2 Trust Considered 16
2.2.3 Types of Trust 19
2.2.4 Trust and Distrust 21
2.2.5 Discussion 22
2.3 Statistical Disclosure Control 23
2.3.1 Data Collection and Management 23
2.3.2 SDC Problem 25
2.3.3 Trust Relationships in the SDC Context 27
2.4 Model of Trust for SDC 30
2.4.1 Modelling Trust Relationships 30
2.4.2 Previous Trust Models 33
2.4.3 Trust Relationships Re-examined 35
2.4.4 Our Model 38
2.5 Challenges in Quantifying Trust 47
2.5.1 Evaluating Trustee Reputation 47
2.5.2 Evaluating Risk and Benefits 49

2.5.3 Privacy Protection Framework 50
2.6 Conclusion 56

3 Comparative Study **57**
3.1 Introduction 58
3.2 A Closer Look 61
3.2.1 Abstract Model 61
3.2.2 Attribute Types 61
3.2.3 Supplementary Knowledge 64
3.3 Restriction 65
3.3.1 Global Recoding 66
3.3.2 Suppression 66
3.3.3 Query Restriction 68
3.4 Noise Addition 69
3.4.1 Additive Data Perturbation 69
3.4.2 Probability Distribution 70
3.4.3 Matrix Masking 71
3.4.4 Categorial Techniques 71
3.5 Information Loss and Disclosure Risk 72
3.6 Software Packages 74
3.7 Conclusion 75

4 PRAM Framework **76**
4.1 Introduction 77
4.2 Privacy Protection Techniques 79
4.3 Post RAndomisation Method (PRAM) 80
4.4 Applying PRAM to Genetic Databases 85
4.5 Clustering Categorical Attributes 87
4.6 Framework for a Genetic Database Security System 90
4.7 Conclusion 92

5 Similarity Measure for Categorical Values **93**
5.1 Motivating Example 94
5.2 Evaluating Similarity 97
5.2.1 Similarity Measure - S_{ij} 98
5.3 Experiments - Similarity 101
5.3.1 Data Sets 101

5.3.2 Parameter Selection . . . 103
5.3.3 Results . . . 104
5.3.4 Numerical Attribute Phenomenon . . . 121
5.3.5 Ordering Categorical Values . . . 121
5.4 Conclusion . . . 122

6 *VICUS* - Noise Addition 124
6.1 Motivating Example . . . 125
6.2 *VICUS* - Noise Addition Technique . . . 125
6.2.1 Graph Partitioning . . . 126
6.2.2 Transition Probability Matrix . . . 129
6.2.3 Perturbing Microdata File . . . 131
6.3 Evaluation Methods . . . 131
6.3.1 Security Measure . . . 132
6.3.2 Data Quality . . . 134
6.4 Experiments - Noise Addition . . . 139
6.4.1 Security . . . 144
6.4.2 Data Quality . . . 156
6.5 Conclusion . . . 176

7 Conclusion 177
7.1 Summary . . . 177
7.2 Contributions . . . 178
7.3 Future Work . . . 181

A Experiments - Detailed 183
A.1 S-Secundum Algorithms . . . 184
A.2 Motivating Example . . . 185
A.3 Mushroom Data Set . . . 186
A.4 Contraceptive Method Choice . . . 195
A.5 Wisconsin Breast Cancer . . . 197
A.5.1 Security Measure Figures . . . 202
A.6 ACS PUMS . . . 211

List of Figures

1.1 Association between genetic components of the cell. 3

2.1 Interdisciplinary model of trust constructs 20
2.2 Strategic Dependency (SD) Model of a Statistical Data Warehouse System 31
2.3 Trust model for statistical data warehouse system. 39
2.4 Propensity to Trust and Distrust Density Functions. 41
2.5 Security Threshold for Data Source. 51
2.6 Security Threshold for Data User. 52
2.7 Security Framework for Statistical Data Warehouse system. . 54

4.1 A Sample data file and the corresponding hypergraph. 88

5.1 Motivating example database represented as a graph. 97
5.2 Similarity color map for Motivating Example. 104
5.3 S'' Threshold T comparison for motivating example. 106
5.4 S' and S'' weighting comparison for Motivating Example. . . 109
5.5 A close look at S_{ij} values for selected attributes in Mushroom. ($T = 0.6$, $c_1 = 0.6$, $c_2 = 0.4$). 110
5.6 Comparing the simple and multigraph representation of the Mushroom data set, all attributes. 111
5.7 Comparing the simple and multigraph representation of the Mushroom data set (attributes 10 and 16 S-Secundum $T = 0.6$). 113
5.8 S'' Threshold T comparison for Contraceptive Method Choice data set. 114
5.9 S' and S'' weighting comparison for Contraceptive Method Choice data set. 116

5.10 Wisconsin Breast Cancer data set, total similarity for all attributes ($T = 0.50, c_1 = 0.6, c_2 = 0.4$). 118
5.11 Sample results for Census PUMS data set ($T = 0.75, c_1 = 0.6, c_2 = 0.4$). 119
5.12 ACS PUMS Attribute *World area of birth* original and reordered. 122

6.1 Comparing record entropy to the number of attributes known by intruder, Mushroom data set. 145
6.2 Comparing confidential attribute entropy to the number of attributes known by intruder, Mushroom data set. 146
6.3 Entropy sensitivity for when the user knows 1 particular attribute, Mushroom data set. 147
6.4 Record entropy sensitivity for when the user knows 1 particular attribute, Mushroom data set. 148
6.5 Confidential entropy sensitivity for when the user knows 1 particular attribute, Mushroom data set. 149
6.6 Distribution of record entropies when user knows 3 attributes for each individual attribute combination, Mushroom data set. 150
6.7 Distribution of record entropies when user knows 2 attributes for each individual attribute combination, Mushroom data set. 151
6.8 Distribution of record entropies when user knows 1 attribute, averaged over the 30 perturbed files, Mushroom data set. . . 152
6.9 Distribution of record entropies when user knows 1, 2 and 3 attributes, averaged over the 30 perturbed files, Mushroom data set. 153
6.10 Record entropy comparison for WBC and Mushroom data sets. 155
6.11 Decision tree for original Mushroom data set. 159
6.12 Classification error distribution comparison for Mushroom - Mush5 - 30 perturbation files. 161
6.13 Decision tree for original Wisconsin Breast Cancer data set. 163
6.14 Decision tree classification error comparison for WBC perturbed files when $k_1 \times k_2 = 100$. 164

6.15 Decision tree classification error comparison for WBC perturbed files when $k_2 = 2$. 166
6.16 Classification error distribution comparison for WBC2 perturbation files. 167
6.17 Sample decision tree for *VICUS* perturbed Wisconsin Breast Cancer data set. ($k_1 = 2$, $k_2 = 50$) 168
6.18 Sample decision tree for *Random* perturbed Wisconsin Breast Cancer data set. ($k_1 = 2$, $k_2 = 50$) 169
6.19 Distribution of chi-square statistic for 30 files perturbed via *VICUS* method. (Attribute 4 and 14 Mushroom data set). . 172
6.20 Distribution of chi-square statistic for 30 files perturbed via *Random* method. (Attribute 4 and 14 Mushroom data set). 173
6.21 Distribution of chi-square statistic for 30 files perturbed via *VICUS* and *Random* methods. (Attribute 4 and 14 Mushroom data set). 174

A.1 Decision tree classification error comparison for WBC and Mushroom data sets. 188
A.2 Classification error distribution comparison for WBC3 perturbation files. 199
A.3 Comparing record entropy to the number of attributes known by intruder, WBC data set. 202
A.4 Comparing confidential entropy to the number of attributes known by intruder, WBC data set. 203
A.5 Entropy sensitivity for when the user knows 1 particular attribute, WBC data set. 204
A.6 Record entropy sensitivity for when the user knows 1 particular attribute, WBC data set. 205
A.7 Confidential entropy sensitivity for when the user knows 1 particular attribute, WBC data set. 206
A.8 Distribution of record entropies when user knows 3 attributes for each individual attribute combination, WBC data set. . . 207
A.9 Distribution of record entropies when user knows 2 attributes for each individual attribute combination, WBC data set. . . 208

A.10 Distribution of record entropies when user knows 1, 2 and 3 attributes, averaged over the 30 perturbed files, WBC data set. 209
A.11 Distribution of record entropies when user knows 1 attribute, averaged over the 30 perturbed files, WBC data set. 210

List of Tables

2.1 An abstract model of a data warehouse 25

3.1 Census Database for Town X 62
3.2 Domains of attributes for Census Database in Table 3.1 . . . 63
3.3 *Total income* summary table of Census Database for *HOH gen* and *NoC* 64
3.4 Redesigned Table 3.3 after global recoding. 66
3.5 Table 3.3 after primary cell suppression. 67
3.6 Table 3.3 after secondary cell suppression. 67

4.1 Sample Genetic Database. 81

5.1 Lecturer Microdata File - Sample 94
5.2 Vertex labeling for Table 5.1 in Figure 5.1 96
5.3 Data Set Summary. 101
5.4 S'_{ij} values for *Lecturer* attribute in Motivating Example. . . . 107
5.5 S''_{ij} values for *Lecturer* attribute in Motivating Example with T=0.4. 107
5.6 S_{ij} values for *Lecturer* attribute in Motivating Example with $T = 0.4$, $c_1 = 0.6$ and $c_2 = 0.4$. 107
5.7 Attribute names and ordering for Mushroom data set. 108
5.8 Attribute names and ordering for Wisconsin Breast Cancer data set. 117

6.1 Lecturer Microdata File - Sample 137
6.2 Probability parameters for perturbations on Mushroom data set 141

6.3 Probability parameters for perturbations on Wisconsin Breast Cancer data set 143
6.4 Average record entropy for Mushroom perturbations, intruder knows 1 attribute. 144
6.5 Average confidential attribute entropy for Mushroom perturbations, intruder knows 1 attribute. 144
6.6 Average record entropy for WBC perturbations, intruder knows 1 attribute.. 156
6.7 Average confidential attribute entropy for WBC perturbations, intruder knows 1 attribute. 156
6.8 Confusion matrix for original Mushroom data set 158
6.9 Average percentage of incorrectly classified instances for Mushroom perturbations, when tested against the original data set. 160
6.10 Selection of k-fold cross-validation parameter, showing k versus percentage of incorrectly classified instances over all folds. 162
6.11 Confusion matrix for original WBC data set 162
6.12 Average percentage of incorrectly classified instances for WBC perturbations. 165
6.13 χ^2 and associated p-value summary for Mushroom data set . 171
6.14 χ^2_{LR} and associated p-value summary for Mushroom data set 175
6.15 χ^2 and associated p-value summary for Wisconsin Breast Cancer data set 176
6.16 χ^2_{LR} and associated p-value summary for Wisconsin Breast Cancer data set 176

A.1 S'_{ij} values for attributes 2-4 of Motivating Example 186
A.2 S''_{ij} values for attributes 2-4 of Motivating Example, T=0.4 . 186
A.3 S_{ij} values for attributes 2-4 of Motivating Example, T=0.4, $c_1 = 0.6$ and $c_2 = 0.4$ 187
A.4 Vertex labeling for Mushroom data set 190
A.5 Record entropies on perturbed files for Mushroom data set, intruder knows 1 attribute 191
A.6 Confidential attribute entropies on perturbed files for Mushroom data set, intruder knows 1 attribute 192

A.7 Percentage of incorrectly classified instances for J48 decision tree builder on perturbed files for Mushroom data set, when tested against original microdata file 193
A.8 Average percentage of incorrectly classified instances for Mushroom perturbations, when the perturbed file is used for testing. 193
A.9 Percentage of incorrectly classified instances for J48 decision tree builder on perturbed files for Mushroom data set 194
A.10 Vertex labeling for Contraceptive Method Choice data set . . 196
A.11 Record entropies on perturbed files for Wisconsin Breast Cancer data set, intruder knows 1 attribute. 197
A.12 Confidential attribute entropies on perturbed files for Wisconsin Breast Cancer data set, intruder knows 1 attribute . . 198
A.13 Percentage of incorrectly classified instances for J48 decision tree builder on perturbed files for Wisconsin Breast Cancer data set, when the perturbed file is tested using the original file. 200
A.14 Percentage of incorrectly classified instances for J48 decision tree builder on perturbed files for Wisconsin Breast Cancer data set, when the perturbed file is used for both training and testing. 201
A.15 Average percentage of incorrectly classified instances for WBC perturbations, when the perturbed file is used for both training and testing. 201
A.16 Vertex labeling for PUMS data set, Attribute 1 WAGP (*Wage or salary income past 12 months*, rounded to nearest 5,000 . . 212
A.17 Vertex labeling for PUMS data set, Attribute 2 PINCP (*Total person's income*), rounded to nearest 5,000 213
A.18 Vertex labeling for PUMS data set, Attribute 3 WKHP (*Usual hours worked per week past 12 months*) top-coded at 99 hours 214
A.19 Vertex labeling for PUMS data set, Attribute 4 WAOB (*World area of birth*) 215
A.20 Vertex labeling for PUMS data set, Attribute 5 RACE1P (*Recoded detailed race code*) 215
A.21 Vertex labeling for PUMS data set, Attribute 6 JWTR (*Means of transportation to work*) 215
A.22 Vertex labeling for PUMS data set, Attribute 7 ST (*State code*) 216

A.23 Vertex labeling for PUMS data set, Attribute 8 ANC1P (*Recoded detailed ancestry*) Part I 217
A.24 Vertex labeling for PUMS data set, Attribute 8 ANC1P (*Recoded detailed ancestry*) Part II 218
A.25 Vertex labeling for PUMS data set, Attribute 8 ANC1P (*Recoded detailed ancestry*) Part III 219
A.26 Vertex labeling for PUMS data set, Attribute 9 AGEP (*Age*) 220
A.27 Vertex labeling for PUMS data set, Attribute 10 CIT (*Citizenship status*) 221
A.28 Vertex labeling for PUMS data set, Attribute 11 COW (*Class of Worker*) 221
A.29 Vertex labeling for PUMS data set, Attribute 12 MAR (*Marital status*) 221
A.30 Vertex labeling for PUMS data set, Attribute 13 SCHL (*Educational attainment* 222
A.31 Vertex labeling for PUMS data set, Attribute 14 SEX (*Sex*) . 222

List of Algorithms

5.1 Calculating S'' values for graph G 100
6.1 Calculating entropy of a record in perturbed microdata file. . . 133
6.2 Calculating entropy of confidential attribute in perturbed microdata file. 134
A.1 Calculating S'' values for graph G, as implemented 184
A.2 Calculating neighbour pairs list, used in Algorithm A.1 185

Abstract

The rapid pace of growth in the field of human genetics has left researchers with many new challenges in the area of security and privacy. To encourage participation and foster trust towards research, it is important to ensure that genetic databases are adequately protected. This task is a particularly challenging one for statistical agencies due to the high prevalence of categorical data contained within statistical genetic databases. The absence of natural ordering makes the application of traditional Statistical Disclosure Control (SDC) methods less straightforward, which is why we have proposed a new noise addition technique for categorical values.

The main contributions of the thesis are as follows.

We provide a comprehensive analysis of the trust relationships that occur between the different stakeholders in a genetic data warehouse system. We also provide a quantifiable model of trust that allows the database manager to granulate the level of protection based on the amount of trust that exists between the stakeholders. To the best of our knowledge, this is the first time that trust has been applied in the SDC context.

We propose a privacy protection framework for genetic databases which is designed to deal with the fact that genetic data warehouses typically contain a high proportion of categorical data. The framework includes the use of a clustering technique which allows for the easier application of traditional noise addition techniques for categorical values.

Another important contribution of this thesis is a new similarity measure for categorical values, which aims to capture not only the direct similarity between values, but also some sense of transitive similarity. This novel measure also has possible applications in providing a way of ordering categorical

values, so that more traditional SDC methods can be more easily applied to them. Our analysis of experimental results also points to a numerical attribute phenomenon, whereby we typically have high similarity between numerical values that are close together, and where the similarity decreases as the absolute value of the difference between numerical values increases. However, some numerical attributes appear to not behave in a strictly 'numerical' way. That is, values which are close together numerically do not always appear very similar.

We also provide a novel noise addition technique for categorical values, which employs our similarity measure to partition the values in the data set. Our method - *VICUS* - then perturbs the original microdata file so that each value is more likely to be changed to another value in the same partition than one from a different partition. The technique helps to ensure that the perturbed microdata file retains data quality while also preserving the privacy of individual records.

Chapter 1

Introduction

"The essence of life is statistical improbability on a colossal scale."

–Richard Dawkins

As members of the human race we all share a common genetic bond; each of our cells contains the necessary information needed to create and sustain life. This fundamental truth hints at the potential benefits to be obtained by gaining a better insight into the workings of the human cell and the genome in particular. The undertaking of such work requires large amounts of genetic sequencing data, along with other supplementary information about the individual, stored in large genetic data warehouses. Increasingly, the collection of such information is being carried out on a larger scale, indeed in some cases on whole populations, and the prospective insights and benefits to be acquired from these genetic databases should not be under-estimated. Indeed advances in both computing power and bioinformatics algorithms has led to an acceleration in the amount of insight gained from genetic data warehouses. However, we must also bear in mind that there are many legal and ethical issues that need to be considered, especially when the potential damage to the individual's privacy is so great. In this chapter we discuss these important issues and give insight into why it is so important to adequately secure genetic information, but also why it can be so technically difficult to do so.

The organisation of this chapter is as follows. We first give a brief primer on genetics so as to better understand the nature of the information that can be stored in genetic data warehouses and discuss the reasons why genetic information differs from other types of data, and in particular its inherently predictive nature and the potential for genetic discrimination. We next outline specific research questions addressed by the thesis and finally provide an overview of the structure of the thesis and in particular discuss the contribution that the thesis makes in the areas of Statistical Disclosure Control.

1.1 Genetic Information Challenges

To fully understand the fundamental importance of genetic information we must first understand what constitutes genetic data. As members of the human race we all share a collective narrative of how we came to be the unique individuals that we are. Through the coming together of a human egg and sperm, i.e. the process of fertilisation, an embryo is formed containing all of the genetic material needed to create life. Egg and sperm cells are a special type of sex cell (*gamete*), and each contain a single set of 23 chromosomes [101]. The other type of cells in our body are known as somatic cells and each such cell contains 46 chromosomes, half from our mother's egg (*ovum*) and the other half from our father's sperm. We each have over 100 trillion somatic cells in our body and each one contains all the instructions needed to maintain life, encoded in our DNA (deoxyribonucleic acid) in the form of genes [101].

Looking at Figure 1.1 we can see an overview of how the main players in this genetic game fit together. Within the nucleus of each somatic cell are the 23 pairs of chromosomes mentioned above. A chromosome is a single strand or molecule of DNA, and each chromosome contains a certain number of *genes* within its structure, with each gene situated at a specific position (*locus*) on the DNA molecule. So in essence a gene is just a long sequence of DNA. The chemical structure of DNA is in the form of the famous double helix, comprising a hydrogen bonding of the four nucleotide bases: *adenine* (A), *guanine* (G), *cytosine* (C) and *thymine* (T)[101]. Hence our DNA can

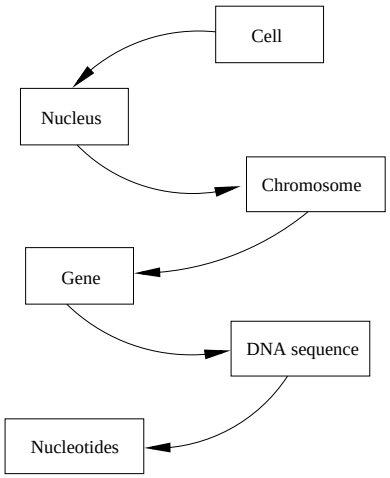

Figure 1.1: Association between genetic components of the cell.

in effect be written in a language comprising only four letters.

A *genome* refers to the sum total of DNA within an organism, comprising both genes and other DNA sequences [111]. Although the majority of our DNA is contained within the structure of chromosomes, we also have a small amount of DNA contained in the cell's mitochondrion: around 13 genes, encoded in a DNA molecule consisting around 16,500 bases [73]. Each gene contains the instructions for the creation and operation of proteins in the body. The details of how this process occurs is not important to our discussion here, so we refer interested readers to any standard introductory biology text for more information on this subject [73, 101]. What we focus on here is that our chemical basis of heredity is contained in our genome.

To this point we have discussed the fundamental genetic building blocks that we all share; in fact the ordering of the nucleotide bases in the genome is over 99% the same for all of us [63, 51]. So, if we're all 'made of the same stuff' what sets us apart from one another? In a word: mutations. The process of protein synthesis that occurs within a cell involves making copies of sequences of DNA, namely via the processes of *transcription* and *translation*. During these processes an error can occur that causes a change in the ordering of the bases, this is known as a mutation, and it is such variations in the genome that introduces diversity into the population [106]. Such a variation in our DNA at a particular location (*locus*) in the genome is known

as a *polymorphism* [106]. The collection of all of these polymorphisms within a population, at a particular locus, are termed *alleles* [106]. Ultimately it is our particular genetic traits, our *genotype*, and the environment in which we live that sets us apart from the rest of the human species. The physical manifestations of our combined genotype and environmental factors is termed our *phenotype* [73, 106]. A mutation only becomes a problem for us when it results in a change in the way a gene produces a protein, or indeed prevent the protein production completely [101].

There are many issues arising from the storage and analysis of genetic information, mainly due to its inherently predictive nature [51, 52]. While some would argue that genetic information requires no further protection than traditional medical data [25], we subscribe to the view that genetic data poses exceptional challenges that warrant additional protection. Several of the potential misuses arising from the predictive nature of genetic information include possible discriminatory effects and family or ethnic group correlations [52]. These potential pitfalls point to an increased need for the proper application of ethics in relation to the collection, storage and analysis of genetic information. Ethics is defined in the Concise Oxford Dictionary as "the moral principles governing or influencing conduct" [102, p.490]. So what sort of moral principles are we dealing with in the context of genetic databases? We summarise them as follows:

- Privacy, informed consent and intellectual property [111]
- Autonomy: ability, security, knowledge, freedom, opportunity and resources [95]
- Duty to know: does a person who finds out they have a genetic disorder, or are a genetic carrier for one, have an obligation to inform relatives? [109]

One recent example of where appropriate application of ethics to genetic database has come into question is with the Icelandic Health Sector database proposed by *deCode* genetics [51, 52]. In contrast to the commercial endeavours of *deCode* Genetics with the Icelandic Health Sector Database, the UK's Biobank project is a non-profit research initiative aimed at improving

disease diagnosis, treatment and prevention in the population at large [117].

One of the key challenges faced by a data manager wanting to provide adequate protection for a genetic database is which parts of the genetic information can they 'safely' modify. As we will see in Chapter 3 the most commonly applied statistical disclosure control mechanisms involve the modification of certain values in the original data file to incorporate a notion of uncertainty in the event where sensitive values are discovered by an intruder. Due to the complexity and lack of total understanding of the actual underlying meaning of genetic sequences we advice against directly modifying this data. Instead we advocate the modification of the supplementary information, such as medical history and diagnosis that is stored alongside the genetic sequencing data in the genetic data warehouse. It is also important to note that much of this additional information is traditionally categorical in nature, i.e., its values do not exhibit a natural ordering. As we will see later, this makes the challenge of protecting such information from an intruder all the more difficult.

1.2 Research Questions

Recent work in the area of computer security has seen a move away from traditional 'hard' security measures to emphasise the role trust and risk play in cooperative relationships [74, 68]. To the best of our knowledge there has not yet been a comprehensive analysis of the role that trust plays in a statistical data warehouse system. Our aim is to not only examine the complex trust relationships that exist between the various stakeholders in such a system, but also model what role trust plays in successful collaboration.

As shown in Section 1.1 genetic data warehouses typically contain a lot of supplementary data which is categorical in nature, that is, attributes whose values exhibit no natural ordering. Traditional Statistical Disclosure Control mechanisms are generally not very successful at dealing with this type of data. In this thesis we examine which techniques may be applicable for use with categorical data, and propose a privacy protection framework aimed specifically for use with genetic statistical data warehouse systems.

We now summarise our main research problem into the following two research questions which we address in the remainder of the thesis.

- *What role does trust play in the relationships between the stakeholders in a genetic statistical data warehouse system and how best can it be modelled?*
- *How can we successfully apply statistical disclosure control measures to a genetic statistical data warehouse systems when a large proportion of the data is categorical in nature?*

1.3 Thesis Overview

We now outline the content of each chapter and comment on the contribution from each.

Chapter 2 - Trust in Genetic Databases

Trust is part of the fabric of our everyday life, it is so intrinsic to our relationship that society would not function so well without it [81]. The area of computer security has recently experienced a shift away from more traditional 'hard' security measures, such as access control, towards the application of social control mechanisms such as trust [74]. In Chapter 2 we investigate the trust relationships that exist between the stakeholders in a Statistical Disclosure Control setting, and present a quantifiable model for trust. A framework is also provided to instruct data managers on how best to foster relationships of trust amongst the system stakeholders, and ultimately achieve the highest level of security and data quality possible. To the best of our knowledge, this is the first time that trust has been so extensively applied in the statistical database context.

Chapter 3 - Comparative Study of Relevant Techniques

In this chapter we provide a comprehensive introduction to the Statistical Disclosure Control (SDC) problem. The difficult task faced by data managers is to balance the conflicting goals of data quality and disclosure protection. We present the ways in which this problem has traditionally been approached and discuss why a users supplementary knowledge about certain records contained in a statistical database poses such a challenge to privacy protection.

Chapter 4 - Privacy Protection Framework for Genetic Databases

When a statistical database contains a large percentage of categorical attributes, as with genetic databases, the application of traditional SDC methods becomes less straightforward. One of the techniques that has been successfully applied to categorical values is the Post RAndomisation Method (PRAM) developed by researchers at Statistics Netherlands [54]. One of the necessary components of the technique is the development of a transition probability matrix which is then used to perturb the microdata file for general release. In Chapter 4 we provide a Privacy Protection Framework for genetic databases, which proposes the application a clustering technique on the original data set to allow us to decide similarity between categorical attribute values.

Chapter 5 - Similarity Measure for Categorical Values

By their very definition categorical attributes do not exhibit a natural ordering, which makes the task of perturbing these values more complex than for numerical data. The main contributions of Chapter 5 is our similarity measure for categorical attributes. By constructing a graph from the values in the original data set, by looking at which values in an attribute are neighbours, we measure the relative similarity between values. We term this similarity *S-Prime* similarity. One of the unique qualities of our measure is that we are able to capture a notion of transitive similarity between values,

which is called *S-Secundum* similarity. We not only look at neighbours in the graph, but also 'neighbours of neighbours'. We can then balance the two types of similarity by applying a weighting to the S-Prime and S-Secundum similarity to provide a total similarity value for each pair of attribute values.

One application of our similarity measure is that we are able to verify the 'numeric' behaviour of numerical attributes by seeing if values that appear similar according to our measure are close numerically. Another potential application of our similarity measure is that it may provide us with the ability to order categorical attribute according to their relative similarity values.

Chapter 6 - *VICUS* - A Noise Addition Technique for Categorical Values

In Chapter 6 we present our noise addition technique for categorical values - *VICUS* - from the Latin for neighbourhood. The first step of our technique is to partition our data set according to their similarity values. We next construct the transition probability matrices for each attribute in the data set. By applying the transition probability matrices we perturb the original file. We extensively evaluate the quality of our noise addition technique both in terms of disclosure risk and data quality. We provide the ability to balance these competing needs via several parameters when constructing the transition probability matrices.

Chapter 7 - Conclusion

We conclude the thesis with a comprehensive evaluation of the contribution of the thesis and present several avenues of investigation for future work.

Chapter 2

Trust in Genetic Databases

Trust knits society together, and makes it possible for people to get on with their everyday lives. Without it, society would become impossible.

–Kieron O'Hara

As we have seen from the previous chapter, there is potential for much benefit to be gained from the analysis of genetic information stored in large research databases. One way in which useful information can be extracted from such large volumes of data is via a series of statistical queries, in the hope of revealing underlying patterns and trends amongst the population in question. However, this type of analysis has the potential to expose private details about an individual, thereby constituting a breach of their privacy. Since an individual's privacy is at stake, a relationship of trust needs to exist between all parties involved for such a system to operate successfully. In this chapter we investigate the importance of trust in the context of a statistical warehouse system. We model trust and its impact on the decision to collaborate. The issue of trust in the context of statistical databases is not limited only to genetic information. Hence in the remainder of this chapter we will examine trust in the context of broader applications of statistical databases, such as statistical warehouses maintained by businesses, not only those containing genetic information.

2.1 Introduction

The past few decades have witnessed ever increasing amounts of personal data being collected and stored by both industry and governments alike. The potential benefits that arise from the analysis of these data range from improved market analysis, via strategic planning, to scientific research and development. Examples of organisational benefits achieved through the use of such large data stores include the analysis of customer buying habits, integration of heterogeneous databases to allow for more cohesive access to multiple data stores, and managing customer relationships and asset costs [60]. Governments can benefit by being able to conduct more accurate statistical analysis in order to better plan for future resource allocation and infrastructure placement.

These huge amounts of data are typically stored in so-called data warehouses. Han and Kamber define a data warehouse as a centrally accessible repository of information whose purpose is to support an organisation in strategic decision-making by allowing for fast access to and detailed analysis of data [60]. These services can be achieved more effectively than in traditional database systems, due to four key features [60, 64]. Firstly, data warehouses tend to be focused on particular subjects, such as customers, products or patients. Secondly, data is gathered and integrated from various source data repositories to ensure consistent and semantically correct content. Thirdly, a data warehouse contains data which is time-variant, providing a historical perspective to data stored. Finally, due to its physical separation from operational repositories, a data warehouse is more stable and does not suffer from many of the problems faced by traditional operational systems.

One of the important advantages of data warehouses is that they allow for statistical analysis to be performed efficiently on very large data sets. On-Line Analytical Processing (OLAP) [21, 60], introduced in the early 1990's, is one of the statistical analysis tools used extensively with data warehouses. Due to the typically multidimensional and hierarchical nature of data warehouses, OLAP operations allow for the analysis of data at varying levels of aggregation. OLAP operations, such as pivot, also allow

for a re-orientation of the multidimensional data view [21]. In this chapter we focus on the statistical analysis of the data in a warehouse at its lowest level of aggregation. This view of the data is equivalent to microdata in Statistical Database literature [125]. In this context users can only retrieve aggregate statistics such as SUM, COUNT, MEAN and AVERAGE. While some data warehouses are set up in such a way that their sole purpose is to provide statistical aggregates, more often they serve several purposes, with statistical analysis being only one of them. The aim of Statistical Disclosure Control (SDC) is to provide the highest possible quality statistics while also preventing the disclosure of values from individual records. The first step in any SDC method is to disallow access to directly identifying information, such as name and address. However, as we will see in Section 2.3, this measure alone is not sufficient to prevent confidential individual values from being inferred.

We now present several examples of data warehouses that can be used for statistical analysis. Customer loyalty schemes have become popular with both consumers and businesses alike. For a business, these schemes provide a way of extracting information about the shopping patterns of various consumer demographic groups. Customers are offered various rewards for shopping at particular stores, provided that they present their loyalty card at the point of sale. In signing up for such a scheme a customer provides a certain amount of private information (such as date of birth, address, and ages of household members) and then agrees to have this information stored along with details of their shopping transactions.

As another example consider electoral databases, which have become increasingly important to both Australian major political parties for targeted campaigning [118]. These data warehouses consist of a combination of Australian Electoral Commission rolls, telephone directories and additional data added by the party offices. Such data provide a powerful assistance to political parties wanting to decide how best to utilise limited campaign funds to their advantage, and thereby improve the party's election outcomes.

Increasingly, both hospitals and general practitioners are storing patients' medical histories electronically. Medical data warehouses formed from such data are providing an important tool in medical research. Moreover,

the potential uses also include more diverse areas such as health administration, adverse drug reactions and drug safety [88]. As an example we consider Australia's controversial Health*Connect* initiative [90]. The aim of the scheme is to implement electronic health records that can be linked and shared across various organisations. This national proposal is intended to provide patients, health professionals and third parties with fast and accurate access to comprehensive patient records. However, as such data warehouses will contain personal or otherwise sensitive information about patients, there is a potential for the invasion of the individual's privacy. Perhaps a more worrying aspect of this system is that it has not been clearly defined which private sector organisations may be granted access to aggregate patient records [90]. It is not difficult to imagine a scenario in which an employer discriminates against a chronically ill person, denying them employment, based on information they obtained while privy to sensitive medical records. This would certainly severely erode a patient's confidence that their privacy will be ensured.

In general, whenever private and sensitive information is collected about an individual, there is a potential to breach that person's privacy. A large proportion of the community has concerns about how their information is to be stored and for what purposes it will be used. In 2003, the Ponemon Institute and the CIO Institute of Carnegie Mellon University conducted a survey on privacy and trust. Both health care providers and banks rated highly, with around 80% of respondents trusting that their private information would be protected [65], while grocery stores and the U.S. Department of Homeland Security obtained the lowest confidence scores of 26% and 36% respectively [65].

So what are the consequences of low level trust between an individual and an end user of their data? When an individual feels that they can no longer trust the organisation to keep their sensitive information confidential, there are several potentially damaging outcomes. Firstly, where possible, the individual may decide to withhold information, making it challenging for the organisation to collect new data. Secondly, and potentially more damaging, the individuals may choose to falsify data they are providing in order to protect their privacy. We argue that in order to elevate the level of

trust, security measures must be employed to facilitate collection of accurate personal information.

Early work in the area of statistical database security focused on attempts to obtain 'perfect' security, or no disclosure. It was quickly seen that due to the competing needs of data quality and privacy, this goal was unattainable [30]. Recent work in the area of computer security has seen a shift from so-called hard security methods, such as authentication and access control, to soft security or social control mechanisms [74]. In this current environment there is a movement towards the analysis of trust and risk in the field of Trust Management [68]. The main objective of this chapter is to model trust in the context of SDC so as to better understand the impact it has on delegation decisions. Importantly, the model we propose is quite general and could also be applied to other contexts.

The structure of the remainder of the chapter is as follows. In the next section we define trust and discuss its various types, as well as the difference between trust and distrust. In the following section we briefly introduce the statistical security problem and demonstrate the importance of trust to the collection and management of data. More specifically, we discuss the trust relationships that exist in this context. In the subsequent section we present a quantifiable model of a secure statistical database system with trust as an essential component. We then discuss some practical aspects of quantifying components of the model and present a privacy protection framework based on this model. We finish with a discussion of future research directions and concluding remarks.

2.2 Trust

Trust is an intrinsic part of the human experience. Yet, for a concept so fundamental to our very existence, it is a mysterious beast, both challenging to pin down and awkward to model. To demonstrate the inherent complexity conveyed by the word trust, we look to its numerous dictionary definitions. McKnight and Chervany [89] scoured three popular unabridged dictionaries for a comparison in the number of definitions between trust and other sim-

ilarly vague concepts. They found that not only did trust have more than three times as many definitions as on average, but also it had close to as many definitions as 'love' and 'like'. This analysis conveys the richness of meaning and intricate nature of trust.

Past work on trust has evolved from such diverse disciplines as psychology, economics, sociology, and computer science. As pointed out by McKnight and Chervany [89], trust is an elusive and difficult to define concept. There is little consensus within the literature on a robust definition of trust, which is varyingly described as a facilitator of goals [100], and a commodity vital for cooperative transactions [27]. It is even argued that without a willingness to make oneself vulnerable by depending on others, without trust, we would find ourselves unable to face the complexities of the world [81].

In order to gain a better understanding of trust we first establish the context and the stakeholders within it. For the purpose of this paper we define a trust relationship as an asymmetric relationship occurring between two parties, the *trustor* and *trustee*. The trustor is the trusting party, while the trustee is the trusted party. In effect, the trustor is placing their trust in the trustee. It is worth noting that the trustee is not necessarily a person, but can be a more abstract entity such as a software program [68]. In our secure statistical data warehouse system we have three stakeholders, namely the *Data Source*, *Data Manager*, and *Data User*.

The Data Source is the person, or system, providing their information to the Data Manager. The Data Manager is responsible for the collection of data and the creation and management of the data warehouse. We term them 'manager' rather than 'owner' since the question of data ownership is not always a straightforward matter. At what point when providing their information does a Data Source relinquish ownership of their data? We choose to err on the side of caution and not make broad assumptions about data ownership. In Section 2.4.1, we acknowledge that a Data Source may receive some form of payment for having provided their data to the Data Manager. However, this may not always be the case, and a Data Source may in fact choose to retain some form of ownership of their information. Similarly, we choose the term Data Source instead of 'donor' used by some

authors in similar contexts [75]. The Data User relies on the Data Manager to provide high quality data on which they then perform various statistical queries. We discuss the complex trust relationships between these various stakeholders in more detail in Section 2.3 and Section 2.4.

To aid our discussion of the various properties and causal factors related to trust, we give the definition of trust proposed in [68]. Analysing this definition will provide insight into the qualities that separate trust from other similar concepts, such as cooperation.

> **Definition 1 (Trust)** *Trust is the extent to which a given party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible.*

Just like several other papers on trust [47, 81, 104], we first discuss what trust is not before considering what actually constitutes trust.

2.2.1 Related Concepts

An obvious beneficial outcome of a trusting relationship is that it encourages cooperation between the parties involved. This symbiosis can make it difficult to separate the concepts of trust and cooperation, and indeed in some early literature, the two terms were often confused [104]. An important distinction between trust and cooperation is that although trust can lead to more cooperation between parties, cooperation can still occur in the absence of trust [47]. As Mayer et al. [104, p.713] state, "You can cooperate with someone who you don't really trust". Indeed, cooperation can be fostered via other forces, such as coercion, contracts and external control mechanisms [47, 104]. An example of coercion to encourage compliance and cooperation would be a dictator state, whereby the powerful government threaten citizens into submission. However, Gambetta is quick to point out that coercion cannot be seen as an alternative to trust; on the contrary, in extreme cases it can in fact reduce the level of mutual trust and give rise to resentment [47]. Contracts and other forms of external control mechanisms

can force cooperation via the threat of punishment if the trustee defaults in some way [104]. Nonetheless, it is widely accepted that cooperation is achieved more efficiently through trust than by other means [47].

Like cooperation, confidence is often confused for trust, and the distinction between the two is not always clearly defined. For instance, when we leave the house each morning without taking a weapon with us, is this trust, or simply confidence [81]? Luhmann suggests that if we do not consider the alternatives, it is a situation of confidence. However, if we weigh up the consequent risk of not carrying a gun and still choose to leave home unarmed, then we are in a situation of trust. Both situations can lead to disappointment when our expectations are not met, but with trust we have considered the alternatives and still perceive the probability of a positive outcome to outweigh that of a negative one. Both alternatives can become routine in our everyday lives, but the distinction "depends on perception and attribution" [81, p.97]. If we are mugged after leaving the house unarmed, in a situation of trust, we will regret the decision. But if the same event occurs under confidence, we will be disappointed, but not have the same feeling of remorse [86]. Marsh equates absolute or 'blind' trust to confidence, pointing out that when no thought or consideration is involved then we are not talking about trust at all [86].

2.2.2 Trust Considered

Having examined what trust is not, we now examine some of the concepts that help us to define trust. The notion of dependence is at the heart of Definition 1, and is reflected in the "willing to depend" component. The need to delegate a task to another person is a necessary condition for trust [68, 29]. By trusting another, we are delegating an important task which we would otherwise be unable to complete ourselves, or at least not as easily. Closely related to the notion of dependence is reciprocity as stated by Mui, Mohtashemi and Halberstadt [96]. Reciprocity is defined as a mutual exchange of deeds, both positive and negative in outcome. Mui et al. argue that reciprocative actions help a person to acquire a reputation [96]. Reciprocity is connected to reputation and trust via a cyclic reinforcing relationship;

the more I trust you, the more likely I am to reciprocate positively to your actions, which leads to an increase in your reputation, and subsequently, the higher your reputation, the more likely I am to trust you [96].

Reputation and credentials are also closely linked to trust, particularly in relation to open distributed systems such as the Internet [1, 68]. These concepts embody the "feeling of relative security" component of Definition 1. A reputation allows us to infer something about future behaviour based on an informed observation of past actions [1]. Reputation systems allow for a reputation to be accrued and assessed on a global scale without the usual face to face personal interactions [68]. Reputation information can be gathered from other people who have had interactions with the trustee, and the difference between reputation and trust is well illustrated by the following two statements from [68, p.104]:

1. "I trust you because of your good reputation"
2. "I trust you despite your bad reputation."

The first statement shows that a positive reputation can lead to trust, while the second statement reflects how personal relationships and experiences can have a larger bearing on trust than reputation alone. However, when reputation includes personal experience as proposed in [1], the distinction between trust and reputation becomes blurred.

Perhaps the most discussed component of any trust definition is the notion of risk, embodied in our trust definition as the possibility of "negative consequences". Risk is pivotal to our earlier discussion of the difference between trust and confidence and indeed, as Luhmann points out, trust "presupposes a situation of risk" [81, p.97] and trust only becomes relevant and necessary when there is some level of risk involved. In their model of trust Mayer et al. [104] make a clear distinction between trust, risk and the outcome of trust, which they refer to as *risk taking in relationship* (RTR). They distinguish between the intent, or willingness to assume risk, and the trusting behaviour of actually assuming the risk, clearly separating trust from its outcomes [104]. RTR is a function of both trust and the level of perceived risk, which deals with risks outside of considerations of

the direct relationship with the trustee. RTR can alternatively be seen as "the behavioural manifestation of trust" [104, p.726]. Marsh [86] includes the perceived risk in a given situation in his calculation of a cooperation threshold, which is the amount that the trust in the given situation has to exceed for cooperation to occur. So again, while risk is not directly part of any trust definition, it is strongly related to the likelihood of cooperation occurring between the trusting parties.

We note that context is very important to any definition of trust, as shown by the excerpt "in a given situation" from Definition 1. In his formalism of trust for artificial agent systems, Marsh discusses the importance of context to trust in a given situation, and particularly the utility and importance of the outcomes of the trusting relationship [86]. His definition of utility is based on economic rationalism and refers to a measure of the benefit to be gained for the trustor in a particular context. Importantly, this value can be negative, and must consider all possible outcomes for the situation, providing an accepted and objective measure of the utility for that particular situation. The importance of the situation is a subjective and trustor centric measure of the positive benefits to be gained in the current context [86].

Another quality of trust relationships is that they are fluid or dynamic, not only depending on context, but also changing over time. Marsh incorporates a finite memory of past experiences into his trusting agents, allowing them to influence the current level of trust [86]. Trust dynamics are examined in more detail by Falcone and Castelfranchi [41, 42]. Several ways in which trust can influence future outcomes are presented, and it is argued that trust can act both as a self-fulfilling prophesy or self-defeating strategy, thus modifying the outcome in both cases. Trust creates reciprocal trust, and distrust elicits distrust. Diffuse trust spreads trust, that is, creates a trusting atmosphere.

Before looking at different types of trust, we briefly outline work done on defining trustee characteristics. McKnight and Chervany [89] have compared cross-disciplinary definitions of trust and extracted four high-level trustee or trust referent characteristics of the trustee: *benevolence*, *integrity*, *competence*, and *predictability*. A benevolent party acts in a caring way in order

to achieve the greater good, rather than acting opportunistically. Integrity refers to acting truthfully, and acting in good faith. Competence is the ability or power to achieve what is needed. Predictability refers to a consistency of trustee actions in a given situation, be they of good or bad consequence. Mayer et al. [104] defined a similar set of trustee characteristics which also includes benevolence and integrity; however, they replace competence with the similarly defined *ability* property. The interrelationship of these trustee characteristics creates a function by which the trust for a trustee can be determined, and importantly a perceived problem with any of the factors can lead to an undermining of trust [104].

Castelfranchi and Falcone [20] define seven basic trust beliefs that a trustor must have towards the trustee in order for them to delegate a task they are trying to complete. *Competence* belief equates to a positive evaluation of the trustee. *Disposition* belief relates to the trustee's willingness and predictability. *Dependence* belief relies on the trustor's dependence on the trustee on the task being performed for them. *Fulfilment* belief relates to trust in the goal itself being performed. *Willingness* belief models the intent of the trustee to perform the task for the trustor. *Persistence* belief relates to the stability of these intentions, while *Self-confidence* belief models the trustee's own self-belief in their ability to perform the delegated task. As the authors state, "It is difficult to trust someone who does not trust himself" [20, p.51].

Having discussed some of the aspects that contribute to what constitutes trust; we now focus on some of the different types of trust that can exist in a trusting relationship between a trustor and trustee.

2.2.3 Types of Trust

In his influential work on a formalisation for trust, Marsh distinguishes between three different types of trust: *basic*, *general* and *situational* [86]. *Basic* trust is an agent's general trusting disposition, not related to any specific person or situation, which can change over time depending on the outcomes of trusting encounters. In general, good experiences will increase our basic trust, and bad experiences will decrease it. *General* trust is trust in a

specific other, irrespective of the situation. Finally, as the name suggests, *situational* trust is trust in a specific other, in a given situation.

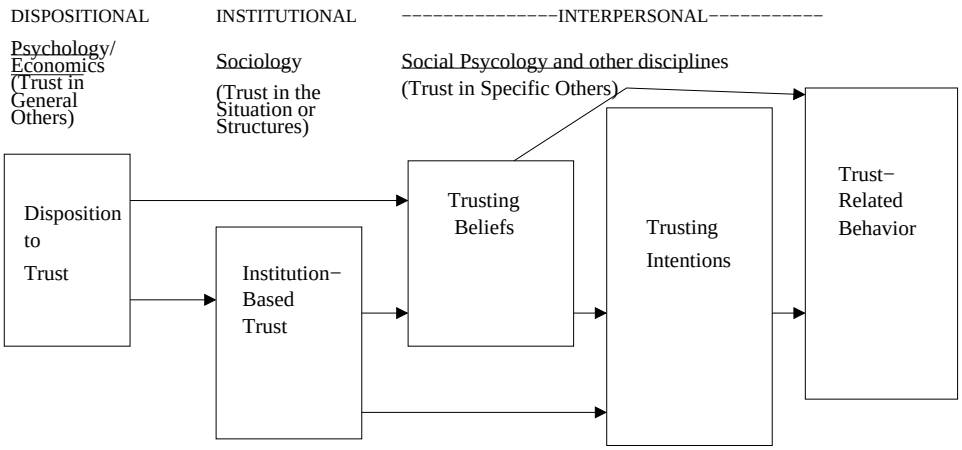

Figure 2.1: Interdisciplinary model of trust constructs (courtesy of McKnight and Chervany [89]).

In developing a cross-disciplinary trust typology, McKnight and Chervany [89] created a trust model in which trust constructs are grouped into three high-level trust types: *Dispositional*, *Institutional* and *Interpersonal*. Dispositional trust can be seen as an inclination to trust others in general, rather than specific individuals as in interpersonal trust. On the other hand, institutional trust is based on structures and situations rather than individuals [89]. Interpersonal trust is further broken down into *Trusting Beliefs*, *Trusting Intentions* and *Trust-Related Behaviour*. The relationships between these constructs are shown in Figure 2.1. Trusting beliefs form the basis for both trusting intentions and the resulting trust-related behaviour, and all three are person rather than context specific.

Having discussed various important aspects of trust and its importance in forming collaborative relationships, we now briefly analyse the fundamental differences between trust and distrust. While in the past the latter was often just dismissed as the opposite end of trust, there is a growing belief that trust and distrust are not only different constructs, but that they can indeed both exist at the same time [76].

2.2.4 Trust and Distrust

Just as trust is fundamental to the formation of cooperative endeavours, distrust can just as quickly break down relationships and stall collaboration. Distrust has often been described as the opposite end to trust along a continuum, where trust values fall in the range [-1,1) with total distrust being -1, no trust being 0 and close to 1 being high trust [86]. Note that the range does not include +1, as it is assumed that total trust is not possible. This definition precludes the idea that we can have both a high level of trust and distrust co-existing. However, it is easy to find a counter-example, for instance we can imagine allies in war simultaneously trusting and distrusting each other [89].

Marsh and Dibben [85] perform a detailed analysis of various aspects of trust and distrust and come up with four separate terms: *Trust*, *Untrust*, *Distrust* and *Mistrust*. They describe trust in a similar way to Definition 1, namely as positive expectations about the trustee providing a positive outcome for the trustor. By comparison, distrust is a negative measure of how much the trustee is actively working against the trustor to prevent the completion of the delegated task [85]. Distrust increases the complexity of a situation due to the requirement for verification or evidence of how the trustee is performing [85]. Trust is placed in the trustee only when the level of trust is higher than some cooperation threshold [86]. This implies that there is a gap between trust and distrust, termed untrust, which describes how little the trustee is trusted [85]. That is the level of trust which is below the cooperation threshold and above zero. A distinction between misplaced trust and distrust is also conjectured. What Marsh and Dibben term mistrust results from a default of trust in a given situation [85]. An example of mistrust could be where a Data Source trusts a Data Manager to properly manage their data and a series of unforseen events results in some data loss. In this situation the trust was misplaced, but this does not necessarily lead to a change in the level of trust between the two parties.

The more accepted view is that trust and distrust are opposites of one another, yet also separate constructs [76, 89]. Luhmann describes distrust as a functional equivalent of trust, yet not its equal [80]. Although both

are a means for reducing complexity in society, distrust can only do so via other means, such as placing controls over the trustee. Lewicki et al. [76] have outlined a theory of trust whereby high trust and distrust can coexist in complex relationships. However, as pointed out by McKnight and Chervany, the simultaneous existence of high trust and high distrust only makes sense when the construct is non-situation specific [89]. If dealing with a specific context, it no longer makes sense to talk about concurrently high trust and high distrust. The same applies for the opposite end of the spectrum, when we have coexisting low trust and low distrust [89].

2.2.5 Discussion

Now armed with some knowledge of the complex role that trust plays in cooperative relationships, we are ready to develop a model of how these relationships are formed in the context of statistical data warehouse systems. One problem faced when developing such a model is how to handle the conflicting needs of the system stakeholders. For instance, a consumer providing their data to a Data Manager wants to ensure that their sensitive information is kept private. On the other hand, an end user has a goal of obtaining the highest quality statistics from their statistical analysis. It is important to note that these two goals may sometimes be in conflict. When a security control measure is applied to the system, the end user's level of trust in the Data Manager's ability to preserve the quality of statistics will be generally reduced, while conversely, the consumer's trust that their privacy will be protected will increase.

Thus, Statistical Disclosure Control (SDC) techniques contribute to the trust that the Data Source has in the Data Manager. This will help to ensure that individuals are still willing to participate and provide their information. Perhaps more importantly, it will encourage consumers to provide correct information. In the next section we discuss the complexities faced by Data Managers when collecting data. We also introduce some of the security techniques that are currently used in statistical data warehouse systems. Additionally, we examine the various relationships of trust that exist between the system stakeholders.

2.3 Statistical Disclosure Control

In this section we introduce the Statistical Disclosure Control (SDC) problem as it relates to statistical data warehouse systems. This dichotomy between protecting the privacy of individuals and ensuring the highest quality statistics has been traditionally approached via two main methods, namely query restriction and noise addition. We examine both of these methods and explain some of the security threats and information loss measures. We discuss how data is collected in a data warehouse and illustrate some of the complexities and difficulties faced by Data Managers attempting to obtain quality data. The purpose of this section is not to provide a comprehensive literature review of SDC, but rather to provide the reader with a concise introduction to the basic concepts so as to clearly articulate the context in which we are examining trust relationship. Chapter 3 will give a more formal overview of the relevant concepts and techniques in the area of Statistical Database Security.

2.3.1 Data Collection and Management

The way in which data in the data warehouse is collected can impact greatly on the accuracy and completeness of the information stored; without quality data it is impossible to extract quality statistics. Methods of data collection include surveys conducted by census bureaus, medical records collected by medical practitioners, surveys conducted by National Statistics Agencies, and market analysis data accumulated through sales information. In terms of potential loss of data quality, the method most worthy of discussion is the collection of surveys. We argue that when a person has a low level of trust in an organisation they are likely to provide false information. In fact, Australian survey results indicate that only 17% of respondents believe that businesses selling over the Internet are trustworthy [57, p.18]. On the other hand, when individuals stand to gain nothing from participating and face no possible punishment, they are likely not to provide their data at all. Indeed, the Australian Community Attitudes to Privacy surveys conducted in 2001, 2004 and 2007 indicate that the number of individuals leaving information off forms is increasing [57, p.26]. By contrast, with the collection of medical

information people are much less likely to lie, since they have a higher level of trust towards medical practitioners (91% of respondents in [57, p.18]), and since it is in their best interests to provide accurate data.

So what are the potential consequences for Data Managers when their Data Source has a low level of trust or little to gain from providing their data for future statistical analysis? Having a reduced number of people willing to participate and provide information is a problem, yet a potentially more damaging outcome for Data Managers is when individuals provide false or misleading information to data collectors, which is likely to occur when they do not have the option to withhold their information. If they feel that their privacy cannot be guaranteed by the Data Manager, then they in a sense perform their own data perturbation (see Section 2.3.2) on the information prior to collection. The fallout of such behaviour is that the correctness of collected statistics cannot be ensured. Clearly, it is beneficial for Data Managers to ensure that correct privacy protection measures are employed so as to not receive bad publicity and in turn increase the willingness of the public to participate in future data collection [124].

There are further reasons why Data Managers should consider using SDC techniques to help protect their data warehouses. Firstly, there may be a legal requirement to collect data so as to adhere to the current laws of the country in which the data is being collected and/or distributed [124]. This generally involves employing SDC techniques to ensure the privacy of individuals, and failure to comply could result in legal actions. Another valid, yet perhaps less tangible requirement for Data Managers to consider is their moral obligation to ensure the individuals' privacy. In some cases, such as with statistical agencies, this requirement is even incorporated into their professional codes of conduct [124].

From this analysis it would seem clear that the SDC problem needs to be addressed by the Data Manager. In the statistical data warehouse system they sit between the Data Source and Data User, and are faced with the conflicting needs of both stakeholders. We now define this problem in more detail and provide examples of how a solution to it is traditionally approached.

2.3.2 SDC Problem

The type of system we consider is a data warehouse that only allows for statistical queries to be performed on the data. There are two key conflicting goals in such a system. Firstly, the Data Manager wants to ensure that sensitive information relating to individual records in the data warehouse is not disclosed by answering queries. Secondly, the Data Manager aims to achieve that the highest accuracy of released statistics are provided to the Data User. These goals are by their very nature in conflict, a higher level of security (privacy) implies a lower quality/amount of released statistics and vice versa. The real problem faced by the Data Manager is how to find the best balance between these conflicting goals. This is known in literature as the Statistical Security problem or the Statistical Disclosure Control (SDC) problem. Comprehensive overviews of this topic can be found in Chapter 3, or alternatively [2, 13, 9, 124, 125].

Staff No.	*Name*	*Date of Birth*	*Sex*	*Position*	*Start Date*	*Salary*
103749	James	21/06/71	F	Team Leader	23/06/99	64500
853445	Brown	23/05/64	F	Branch Manager	12/04/03	105000
332480	Brown	6/12/59	M	Admin. Assist.	9/01/79	49249
142313	Black	30/11/84	M	Programmer	12/03/05	43060
578345	Jones	28/02/65	M	Team Leader	30/01/03	70500
126950	Chen	16/07/64	F	Admin. Assist.	24/7/01	48750
476633	Smith	10/11/72	M	Technical Officer	9/01/03	50240
125342	Zhong	25/04/83	F	Database Admin.	26/11/03	49900
342523	McDonnall	5/05/79	F	Executive Officer	12/09/01	60900
451413	Liang	21/10/65	M	Programmer	5/06/99	47000

Table 2.1: An abstract model of a data warehouse

In order to explain these concepts in more detail, we present a model of a statistical data warehouse in Table 2.1. In this model each row represents an employee's employment history file and each column describes an attribute (a property). For example, the first column represents a unique employee identifier or staff number ('*Staff No*'), and the second column is the employee's surname ('*Name*'). This is followed by the employee's date of birth ('*Date of Birth*') and gender ('*Sex*'). We then have several attributes related to the employee's work history and employment conditions, the first of which is their current job description ('*Position*'), the date that they started working for the company ('*Start Date*'), and finally, the employee's

current annual salary ('*Salary*'). It is worth noting that this microdata file is only designed as an abstract example, and does not represent a real employee database. In reality, such data warehouse systems would have a much larger number of attributes and many thousands, even millions, of individual records.

The attributes in a statistical data warehouse can be further categorized into two main groups, confidential (or sensitive) and non-confidential (or identifiers). In our sample database in Table 2.1 examples of non-confidential attributes would be *Name*, *Sex* and arguably *Date of Birth*. An example of a sensitive attribute would be *Salary*, since historically people have always been reticent to disclose their income. The 2007 Community Attitudes Towards Privacy survey [57, p.23] has income details second in the list of information people are most reluctant to provide, after financial details. As many countries, including Australia, move away from standardised salary awards to a system of negotiated income packages it can be argued that salary data is becoming even more sensitive, since it may upset the relationship among work colleagues.

To ensure the integrity of the data warehouse system, it should be impossible for users to infer confidential values from any sequence of aggregate values. A situation where a user is able to determine the individual values is termed a database *compromise* or *statistical disclosure*. A user who, either deliberately or accidentally, is able to disclose information about an individual record is called an intruder, or *snooper*. An obvious first step in protecting the statistical data warehouse from such a snooper would be to remove all direct identifiers from the database. However, this alone is not enough to anonymise the data [125], hence the statistical security problem is typically dealt with in one of the following two ways: restricting queries that users can pose to the system or adding noise to the data. Query restriction methods prevent compromise from occurring, while when noise addition techniques are applied compromise is still possible, although the intruder has a degree of uncertainty about the exact values. In either case it is important to find the right balance between security and usability of the database where the latter is measured by the number and quality of released statistics.

2.3.3 Trust Relationships in the SDC Context

Before presenting our trust model, we first examine the trust relationships that exist between the three system stakeholders of our statistical data warehouse system: Data Source, Data Manager and Data User. Recall that the Data Source is the entity (individual, organisation, etc.) that is providing information to the system. The Data Manager is responsible for the collection, management and distribution of the data and is assigned the task of ensuring their accuracy and security. The Data User is a researcher or developer who performs statistical queries on the data in order to gain valuable information. It is important to note that a Data User may have hidden malicious intention to learn sensitive information about particular individuals in the data warehouse.

Data Source and Data Manager

The relationship between the Data Source and Data Manager is perhaps the most public and yet most complex of all the trust relationships. The Data Source trusts that the Data Manager will not misuse their data, that is, they will only use it for the purposes previously agreed upon and seek permission before using it for anything else. They also trust that the Data Manager will not on-sell their information to a third party without their explicit consent. In a 2007 survey conducted for the Australian Federal Privacy Commissioner [57, p.35], 94% of participants believed that a business using their information for another purpose was a misuse of their information. From the perspective of the Data Source, this invasion of privacy could easily lead to an erosion of trust in the Data Manager. The following example illustrates a breach of trust between a Data Source and Data Manager.

> **Example 1.** *In late 2003 the American airline JetBlue admitted to handing over the travel records of five million customers to Torch Concepts, a private US Department of Defence contractor. The information was then combined with additional passenger demographic data obtained from another company, Acxiom, and used to develop passenger profiles in order to detect terror-*

ist suspects. JetBlue was in direct violation of their own online privacy policy by selling customer data to a third party [4].

In addition to trusting that the Data Manager will only use data for previously agreed upon purposes, the Data Source also trusts that the Data Manager will properly manage their data and keep sensitive information private. The recent spate of high profile security breaches has increased public awareness of this issue as shown in Example 2.

Example 2. *TJX Companies Incorporated, a large US fashion retailer ranked at 133rd in the fortune 500 list [113], admitted to having its computer systems periodically hacked between July 2005 and January 2007. In the company's Security Exchange Commission (SEC) filing, it admitted that 45.7 million credit and debit cards were effected, as well as a further 455,000 merchandise return records. These contain, among other details, the driver's licence and social security numbers of customers [17]. There are now several lawsuits pending against the company from various financial institutions wanting to recoup losses resulting from the frauds that followed [17].*

In the reverse trust relationship between the Data Source and Data Manager, the Data Manager trusts that the Data Source will provide them with correct and accurate data.

Data Manager and Data User

We now examine the interaction between the Data Manager and Data User. The Data Manager trusts that the Data User will not misuse the data provided to them. One obvious way in which this type of trust could be eroded is where the Data User agrees with the Data Manager to only use the provided data for specific purposes and then uses it in another way. Example 1 above illustrates this type of breach. The Data User is also trusted by the Data Manager to keep private information confidential. Since the Data

Manager should have taken measures to protect the privacy of individuals in the data warehouse, they are in effect trusting that the Data User will not try to subvert these measures. For instance if a National Statistics Agency were to provide a researcher with data, they trust that the researcher will not make an attempt to identify sensitive information from any individual records.

In the converse relationship, the Data User trusts that the Data Manager will provide them with high quality data. This is of particular relevance when dealing with statistical data warehouses since many of the protection mechanisms can have an adverse effect on data quality. As was discussed in Section 2.3.2, restriction methods reduce the amount of statistics released, while noise addition methods reduce the accuracy of the statistics. If a high level of trust is maintained between the Data Manager and Data User, this would reduce the reliance on SDC methods and consequently preserve the data quality.

Data Source and Data User

The final grouping of trust relationships is between the Data Source and Data User. Although they do not have a direct interaction, we still argue that there exists a relationship of trust between these two parties. The Data User trusts that the Data Source will provide accurate data, albeit not directly. Conversely, the Data Source trusts that the Data User will not misuse their data, nor compromise the privacy of any individual's record in the data warehouse. An example of where this type of trust can occur is when a cancer patient has direct trust in a medical researcher attempting to find a cancer gene for their type of cancer. The nature of the delegation task itself impacts on the willingness of the Data Source to participate and place their trust in the Data User.

It is important to point out that although we talk about the Data Manager and Data User as separate entities, both of these roles can in fact be filled by the same entity. In the next section we formalise these trust relationships and introduce our trust model for statistical data warehouse systems.

2.4 Model of Trust for SDC

To gain insight into the interactions between the stakeholders we first model the trust relationships using a software engineering tool designed for early stage system development. This model illustrates the importance of trust in a well managed statistical data warehouse. The question we consider is how a Data Manager decides when the system is operating effectively. We provide a trust model designed to assist Data Managers in evaluating the influence of trust in the system, so as to better recognise potential problems. We advocate that the Data Manager then use a privacy protection framework to ensure that the needs of all stakeholders are adequately managed.

2.4.1 Modelling Trust Relationships

We model the trust relationships of the various system stakeholders using components of the so-called ***i**** framework, which was developed as a requirements engineering tool for early stage system development [132]. It was designed to provide a higher level of modelling than previous techniques such as object-oriented analysis or data flow diagramming [131]. The framework allows for qualitative reasoning about opportunities and vulnerabilities of system stakeholders. To date it has mostly been used in the context of requirements engineering, business processing re-engineering and software processes [131]. However, it has also been applied to the modelling of the role trust plays in system design, highlighting areas where an erosion of trust may develop [131]. The framework visualises the dependencies and specific trust relationships that exist between the system stakeholders. This provides the system developers with a snapshot of how the system should ideally function, and equips them with an understanding of the effects of a breakdown in trust.

In this chapter we use a subset of the ***i**** framework as presented by Yu and Liu [131]. This subset is sufficient to model the intentional dependencies within a network of system stakeholders (actors), via a *Strategic Dependency* (SD) model. Figure 2.2 shows a Strategic Dependency model for a statistical data warehouse system, with the three system stakeholders modelled as

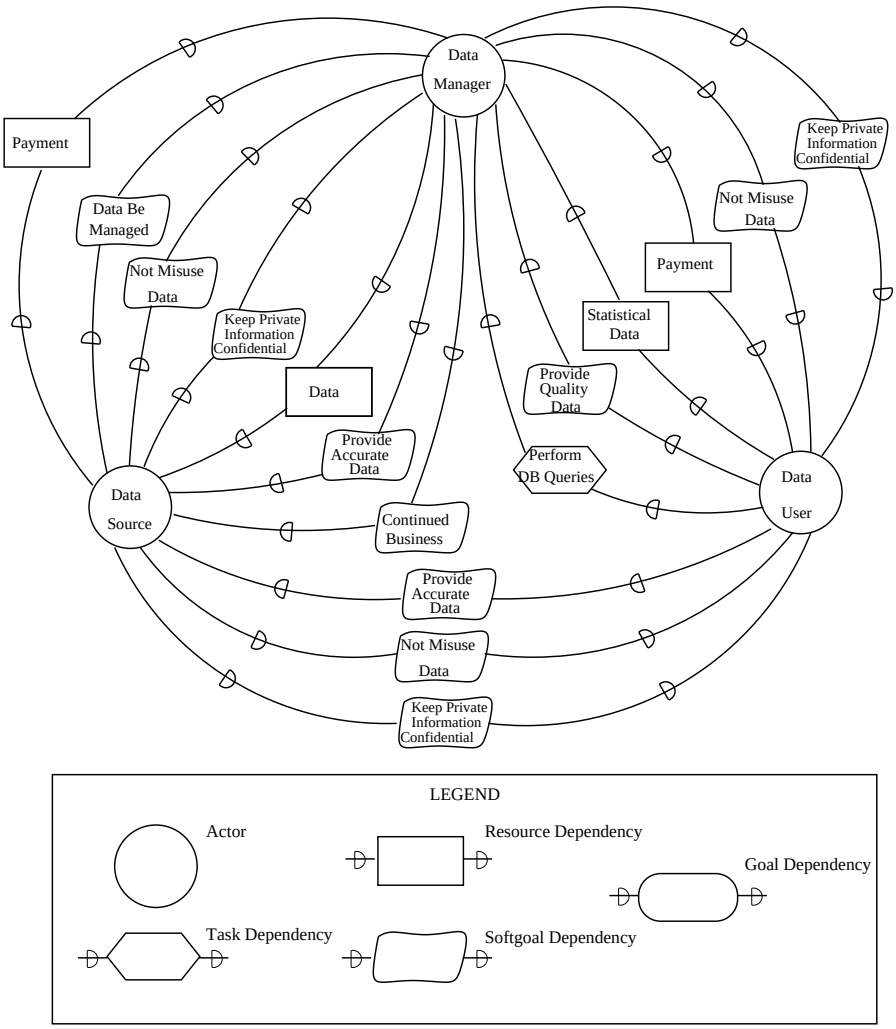

Figure 2.2: Strategic Dependency (SD) Model of a Statistical Data Warehouse System

actors. As discussed in Subsection 2.2.2, dependence and delegation are at the heart of the definition of trust, and with the SD model we are able to capture four separate types of dependencies: *task dependency*, *resource dependency*, *goal dependency* and *softgoal dependency*.

A *task dependency* describes the situation where one actor depends on another to perform an activity, independent of their motivation for having the task performed [131]. In Figure 2.2 we can see that the Data User depends on the Data Manager to perform a series of database queries. This

would occur in a situation where the Data User is not able to have access to the data warehouse directly, but must rely on the Data Manager to perform specific queries and return the results.

In a *resource dependency*, one actor is depending on another to provide them with a particular resource in an unproblematic way [131]. We can see in Figure 2.2 that the Data Source relies on the Data Manager for payment, while the Data Manager relies on the Data Source for data. For example, a shopper using a store loyalty card receives some remuneration for allowing their shopping transactions to be collated. The store relies on the shopper to provide raw data for the data warehouse system. Additionally, the Data Manager obtains payment for providing the Data User with statistical data, an example of which would be the store providing data to a market researcher.

A *goal dependency* allows us to model one actor depending on another to achieve some goal [131]. The dependee in this scenario is free to choose how this goal is met, and it is decided in advance how the completion of the goal will be verified. Note that we use the term "*dependee*" to denote the person being depended on and "*dependor*" for the person who is depending on the dependee, similarly to trustee and trustor. Further freedom is allowed with a *softgoal dependency* since there is no specific *a priori* principle for what constitutes meeting the goal, but rather the dependor and dependee must decide on an individual basis if the goal has been sufficiently accomplished [131]. We use this type of dependency to model trust relationships because it embodies the notion of risk, cooperation and dependence. For instance, in Figure 2.2 the Data Source trusts the Data Manager to keep their sensitive information private, while there is no clear-cut standard for how this confidentiality will be achieved. Modelling trust as an ***i**** softgoal was first presented by [131].

Figure 2.2 incorporates all of the trust relationships that were discussed in Section 2.3.3. We can use this figure to reason about how low levels of trust greatly impact on the management of the secure statistical data warehouse system. For instance, if a Data Source were to lose trust in the Data Manager, they may then decide to falsify their data in any future dealings with them. The Data Manager may not become immediately aware

of the change to data quality; however the Data User may notice a drop in quality of the new data they receive. This would lead to a reduction in the level of trust between the Data User and Data Manager. It is not difficult to imagine the cyclic (feedback) affect of such drops in trust levels between the system stakeholders. The ultimate outcome could easily be that the Data User no longer relies on the Data Manager to provide them with data, causing loss of business to the Data Manager.

It is clear from this discussion that the various trust relationships that exist between the three system stakeholders are vital to the correct operation of any statistical data warehouse system. When there has been a breakdown of trust between two parties it is essential that some mechanism be employed to ensure that goals are still achieved and the system runs smoothly. We now present a trust model for a statistical data warehouse system that incorporates all of the issues we have discussed thus far, after a brief discussion of existing trust models.

2.4.2 Previous Trust Models

In this section we briefly mention two important trust models that have appeared in the literature in the last fifteen years, firstly a quantitative formalism devised by Marsh [86] and secondly the more qualitative model proposed by Mayer, David and Schoorman [104]. We consider these two models in combination and build upon them to capture issues appearing in our statistical data warehouse system. Additionally, our model is applicable to other business scenarios.

Marsh 1994

Marsh [86] builds upon his various types of trust (basic, general and situational) as well as his view on trust versus distrust (see Section 2.2) to develop a formula for a cooperation threshold, a value that must be reached in order for a trust relationship to proceed. Situational trust is the trustor's level of trust towards the trustee in a particular situation, based on their general trust in the trustee and the importance and utility of the situa-

tion. Additionally, the cooperation threshold is a function of the perceived risk and importance of the situation divided by the competence and general trust of the trustor towards the trustee. When the level of situational trust is above the cooperation threshold, it is deemed that the trust delegation will proceed.

The functions for calculating these measures are now outlined as they are in Marsh [86]. Trust by one agent in another for a particular situation is calculated as follows:

$$T_x(y,\alpha) = U_x(\alpha) \times I_x(\alpha) \times \widehat{T_x(y)}$$

where

x - the trustor

y - the trustee

α - the situation

$U_x(\alpha)$ - the utility that x gains from the situation α, in the range [-1,+1]

$I_x(\alpha)$ - the importance of the situation α to trustor x, in the range [0,+1]

$\widehat{T_x(y)}$ - an estimate of the general trust by trustor x towards trustee y, in the range [-1,+1).

For cooperation to occur, the following relationship must exist

$$T_x(y,\alpha) > CT_x(\alpha) \Rightarrow WC(x,y,\alpha)$$

where

$T_x(y,\alpha)$ - trust of x in y for situation α, in the range [-1,+1)

$CT_x(\alpha)$ - cooperation threshold (defined below)

$WC(x,y,\alpha)$ - cooperation will occur between x and y in situation α.

The cooperation threshold is defined as follows:

$$CT_x(\alpha) = \frac{PR_x(\alpha) \times I_x(\alpha)^i}{PC_x(y,\alpha) + \widehat{T_x(y)}}$$

where

$\widehat{T_x(y)}$ - estimate of the general trust by trustor x towards trustee y, in the range [-1,+1)

$PR_x(\alpha)$ - perceived risk by the trustor x in the given situation α

$PC_x(y,\alpha)$ - perceived competence by trustor x in trustee y in situation

α

$I_x(\alpha)$ - the importance of the situation α to trustor x, in the range [0,+1]

i - indicates whether the cooperation threshold increases or decreases with importance and can assume a value of 1 or -1.

Such a quantifiable measure is of a great benefit in the analysis of trust relationships within a statistical data warehouse system. However, there are some shortcomings of this measure, as outlined by Marsh himself [86], regarding the inclusion of trust itself in the cooperation threshold formula, which allows the threshold to take on unreasonably high values, including infinity.

Mayer, Davis & Schoorman 1995

Mayer et al. [104] model trust as a function of the trustor's propensity to trust and the trustee's perceived trustworthiness, which is in itself made up of their ability, benevolence and integrity. Another feature of the model is what the authors term '*Risk Taking in Relationship*' (RTR), whereby they distinguish between the willingness to assume risk and the actual outcome of the trusting relationship. If the level of trust in the trustee is above the level of perceived risk in the situation the trustor will engage in the RTR. The outcomes of this RTR produce a feedback loop to update the trustee's perceived trustworthiness. A drawback of this model is that there is no clearly defined quantitative measure of any of the elements of the model.

2.4.3 Trust Relationships Re-examined

To better understand how the various elements of a trust relationship influence the delegation decision, we first examine a motivating example of a fictitious Data Source who must choose whether or not to provide her information to a Data Manager.

> **Motivating Example.** *Ann receives a letter requesting her participation in a periodic survey conducted by a National Statistics Agency. Ann knows that her curious neighbour, Bob, works for*

this agency and feels uncomfortable since she suspects that her neighbour may get access to any information she provides. Ann has a mild disability which she would not normally disclose to people outside of her close family. Feeling concerned about her privacy and not trusting in the agency's ability to manage her data properly, Ann decides to ignore the request. Several weeks later she receives another letter from the agency. This time they threaten large fines for non-compliance to their request. Against her better judgement, Ann finally decides to cooperate with the survey since she is not willing to risk the negative consequences of not doing so.

The above example illustrates how even when a relatively low level of trust exists, other factors can contribute towards delegation occurring. In particular, we perceive that a trustor must consider the possible positive and negative consequences of collaborating and not collaborating. The following scenarios will help to further exemplify the considerations faced by the stakeholders in a statistical data warehouse system, as outlined in Figure 2.2.

Data Source and Data Manager

The Data Source receives some form of payment for providing their data to the Data Manager, which can be seen as a positive consequence of their collaboration. If they decide not to collaborate, they can also benefit by having their privacy preserved at its current level. However, there are also potential pitfalls when the Data Source chooses not to collaborate with the Data Manager. One example is being unable to access certain services because they do not provide all of the details required. They could also face sanctions or even fines when they refuse to cooperate. Negative consequences can also arise from collaboration. This happens for instance, if the Data Manager is not able to correctly manage the Data Source's data, uses it for another purpose, or is not able to adequately protect their privacy. In these circumstances, the Data Source may suffer not only a loss of privacy, but also faces having incorrect information about them stored.

Data Manager and Data Source

In the reverse relationship between the Data Manager and Data Source a successful collaboration allows the Data Manager to obtain quality up-to-date data. A negative outcome of the same collaboration could be that they receive poor quality data which may affect their long-term relation with business partners (Data Users). If the Data Manager decides not to collaborate with a particular Data Source they are able to ensure that the quality of their data remains static. The downside of non-collaboration would be that they are unable to obtain new data and risk their data becoming worthless because it is no longer current.

Data Manager and Data User

The Data Manager could expect to receive some form of payment in the event of a successful collaboration with a Data User. Yet they also risk the data being misused by the Data User, for instance by on selling the data to a third party or using the data for another purpose. In the event that the Data Manager does not collaborate, they can still benefit by being able to ensure the privacy of those whose data is stored in the warehouse, and could also benefit through an increased reputation since they have been discerning in their choice of customers. A clear negative outcome from not collaborating is a loss of income which could endanger their business.

Data User and Data Manager

When a Data User collaborates with a Data Manager they might expect to receive high quality statistical data. On the other hand there is the potential to obtain poor quality statistics from an unsuccessful collaboration. Clearly if the collaboration does not occur, the Data User is unable to obtain new statistics and further their work, which would reduce future outcomes obtained through statistical analysis. In this scenario there is no obvious benefit for the Data User if they choose not to delegate to the Data Manager.

Data Source and Data User

Since there is no direct collaboration between the Data Source and Data User, we do not consider the benefits and risks of such cooperation. We note however that a relationship of trust still exists between these parties.

We are now ready to introduce our model of trust in the SDC context. The model quantifies the key collaboration elements including trust, as well as positive and negative consequences of delegation and non-delegation. All of these elements are used to calculate the Cooperation Function, which is then compared to a user defined Cooperation Threshold. The delegation occurs only if the Cooperation Function reaches or exceeds the threshold.

2.4.4 Our Model

Our trust model in Figure 2.3 incorporates the spirit of Trust Definition 1 and builds upon several existing models of trust in the literature, namely that of Mayer et al. [104], Marsh [86] mentioned in Section 2.4.2 and McKnight and Chervany [89] discussed in Section 2.2.3.

Figure 2.3 represents the *Trust in a Given Situation* as a function of the *Trustee's Reputation*, the *Trustor's Propensity to Trust*, the *Trustor's Propensity to Distrust* and the *Context in a Given Situation*. This resultant trust is measured against the *Perceived Risks & Benefits* via a *Cooperation Function* (F) to decide if the trust delegation will occur. The *Outcome* of this collaboration, either negative or positive, is then used to update the constructs on the left-hand side of the model. We now examine each individual component of the model in more detail.

Trust in a Given Situation

Trust in a Given Situation is based on four elements, namely the Trustee's Reputation, the Trustor's Propensity to Trust and Distrust and the Context of a Given Situation (see Figure 2.3).

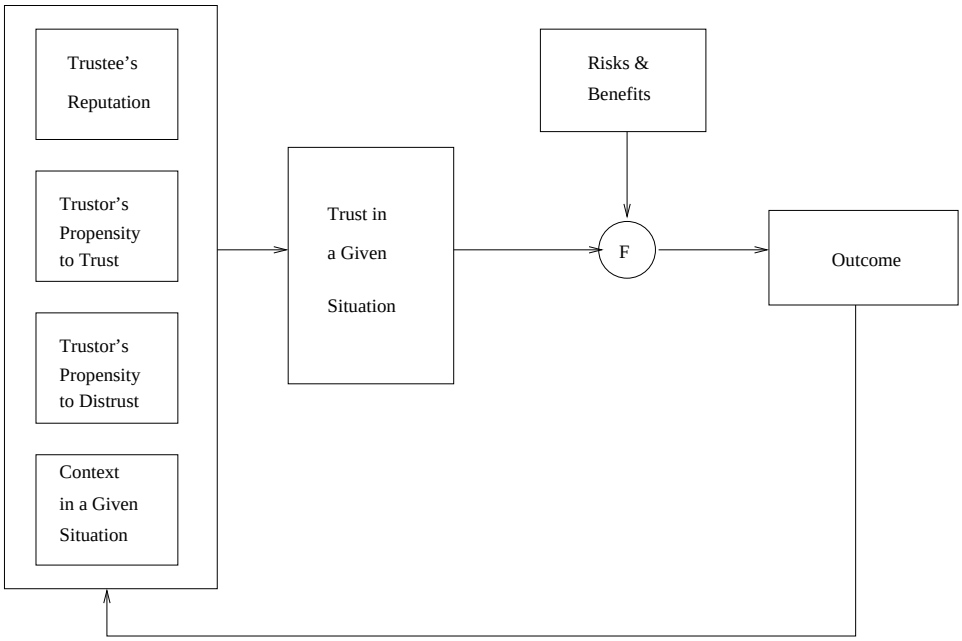

Figure 2.3: Trust model for statistical data warehouse system.

The *Trustee's Reputation* is based on the four key trustee characteristics presented in McKnight and Chervany [89], which are competence, integrity, benevolence and predictability. The perceived levels of each of these contributing factors will be constantly updated via the feedback loop shown in Figure 2.3. It is important to note that a negative delegation outcome may not necessarily lead to a reduction in the trustee's reputation. For instance, we examine the scenario where the trustee was unable to achieve the delegated goal, despite their best efforts, because of environmental circumstances. The trustor may indeed perceive the Trustee's Reputation to be higher than it was before the failure, because of the ways in which they attempted to overcome the obstacles.

The *Trustor's Propensity to Trust* is based on McKnight and Chervany's 'Disposition to Trust' [89] or Marsh's 'basic' trust [86]. This refers to a person's general trusting disposition, or how they trust in general, regardless of the situation or the person being trusted. For example, some people are naturally more trusting in general than others, leading to a higher disposition to trust. As with any type of trust, the level of trusting disposition can change over time, depending on the outcomes of trusting relationships.

Similarly, the *Trustor's Propensity to Distrust* relates to a general tendency to not be willing to depend on general others [89]. At first glance it may appear that there is no difference between these constructs, but as discussed in Section 2.2.4 we prescribe to the view that trust and distrust are opposite yet separate constructs [76, 89]. By monitoring both the levels of trust and distrust in an individual, we are better equipped to model the idiosyncrasies of human nature. We model both the propensity to trust and propensity to distrust by a probability density function specific to each individual trustor (see Figure 2.4). We illustrate this function for four different general personality types, namely *Optimist*, *Pessimist*, *Realist* and *Romantic*. We note that Marsh [86] discussed the first three of these types in terms of the way they decide the level of trust from the experiences they had. An Optimist will always select the best experience they had, the Pessimist the worst and the Realist the average. Our classification is different as we consider person's total trust record over their living memory. Thus the Optimist will have had a high level of trust in the majority of their experiences, while the Pessimist will have had the opposite. The Realist will be more moderate in their trust levels while the Romantic will be prone to extremes, exhibiting either highly trusting or highly distrusting tendencies, thus having a black-and-white approach to trust. Therefore, in the context of propensity to trust and distrust, it is possible for high level of trust and distrust to coexist.

The fourth factor that contributes to trust in a given situation is the *Context of the Given Situation* itself. The situation a person finds themselves in will alter the level of trust that they feel towards another. This can in a way be thought of as the '*Reputation of the Situation*'. The history of a person's past trusting encounters in similar situations will clearly influence how willing they will be to trust in the current situation. For example, some people might be more inclined to trust doctors than car salesman. We note again that this is independent of a particular doctor or a particular car salesman, but rather refers to the context of the situation. An Australian survey [57, p.18] shows that on average the most trustworthy organisations in regards to handling personal information are health service providers (91%) followed by government departments (73%) and financial institutions (58%), while the least trusted are real estate agencies (24%) and business selling over the internet (17%). Another important factor within the context of the

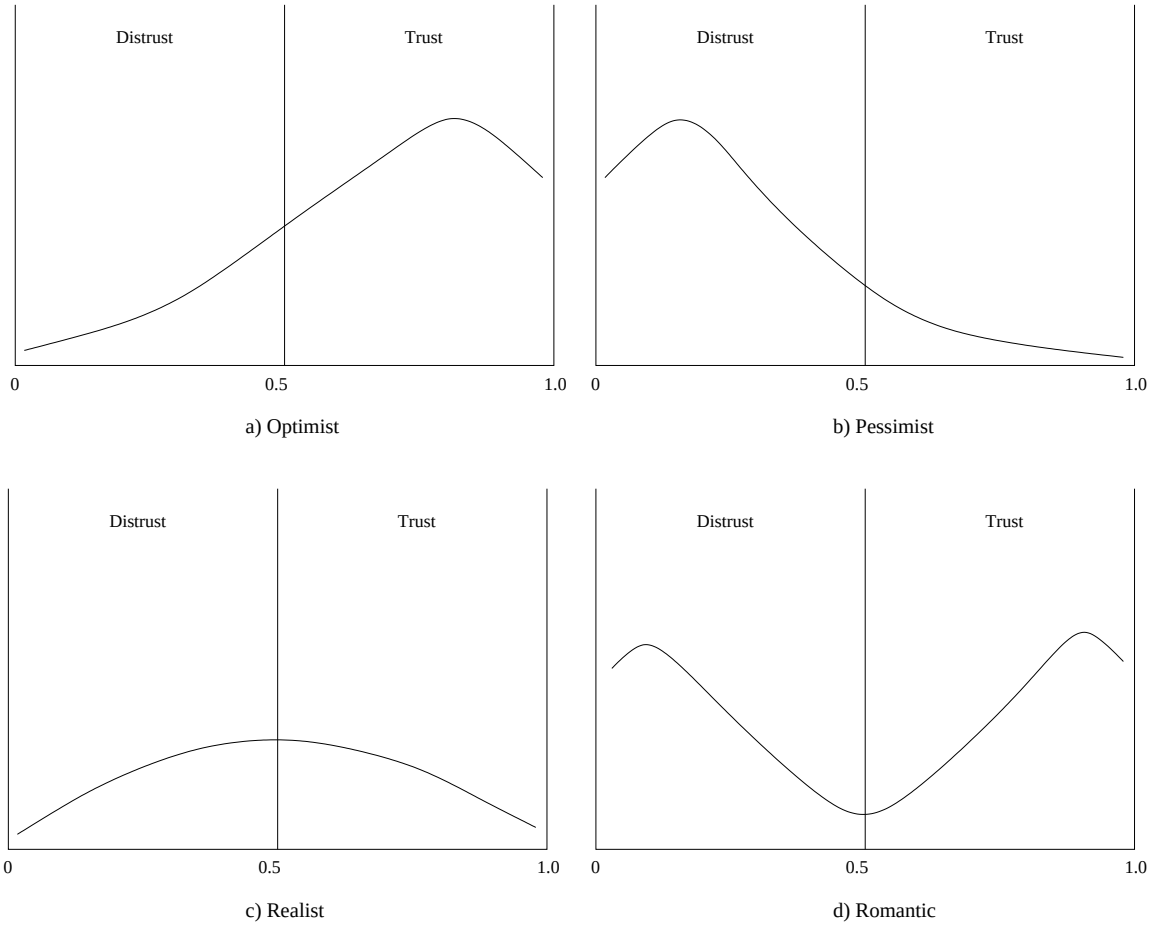

Figure 2.4: Propensity to Trust and Distrust Density Functions.

situation is the importance of the situation itself to the trustor. Clearly a trustor will be more compelled to trust someone to perform a delegation task which is very important to them. While the importance of the situation will be also considered within the risk/benefit analysis (discussed below), we still need to acknowledge its influence on the trust itself. For example, a patient deciding whether or not to proceed with an operation that could save their life will not only take into consideration the lack of other alternatives but would arguably also feel genuinely more trusting than the objective situation warrants.

Risk/Benefit Analysis

When the level of trust in the given situation has been established, it needs to be compared to the potential benefits and risks associated with the decision to delegate. The resulting function is then used to decide if cooperation will occur. This measure differs from the so-called 'cooperation threshold' proposed by Marsh [86] in that in our model the Cooperation Threshold will be a user defined value. When the Cooperation Function is higher than the Cooperation Threshold, then the trustor will delegate to, or place their trust in, the trustee. Conversely, when the function is below the Cooperation Threshold, then they will choose not to cooperate. For example, in the case of the Data Source being our trustee, their choosing not to cooperate would result in them withholding or altering their information.

There are four possible situations which need to be considered in the risk/benefit analysis. The first is the benefit to be gained from a successful collaboration, which we term *Net Benefit of Successful Collaboration* (NB_{SC}) with a range of [-1,+1]. The next is the *Net Benefit of Unsuccessful Collaboration* (NB_{UC}), in the range [-1,+1], which is the risk associated with the decision to delegate to the trustee. Finally, we consider the potential effect of non-collaboration, which we term *Net Benefit of Non-collaboration* (NB_{NC}), also in the range [-1,+1]. For all of these possible outcomes, we can estimate a monetary value and then normalise all of the values to their associated ranges appropriately.

Cooperation Function

Marsh [86] defines a cooperation threshold as the level situational trust must reach for cooperation to occur. He incorporates the level of perceived risk, the trustee's competence, the trustor's general trust towards the trustee, and the importance of the situation in his calculation of the cooperation threshold. Our Cooperation Function is somewhat simpler than this because we have incorporated most of these elements into the calculation of Trust in a Given Situation. The trustee's competence makes up part of the Trustee's Reputation, while the importance of the delegation task is considered when

we examine the context. Finally, the trustor's general trust towards the trustee is partly incorporated into the Trustor's Propensity to Trust. Now we only need consider the Trust in a Given Situation to the level of Perceived Risk and Benefits to determine if cooperation will occur. When the level of the function is higher than the selected Cooperation Threshold, the trustor will delegate to the trustee, and when it is lower, they will choose not to delegate.

We quantify the Cooperation Function as follows.

$$F = T_x(y, \alpha) \times NB_{SC} + (1 - T_x(y, \alpha)) \times NB_{UC} - NB_{NC}$$

where

F - function we evaluate against the Cooperation Threshold (CT).

$T_x(y, \alpha)$ - how much x trusts y in the given situation, in the range (0, +1); neither 0 nor 1 are included in the trust range, as these values would indicate complete certainty in the outcome of the cooperation.

NB_{SC} - the benefit obtained from a successful collaboration, in the range [-1, +1].
$NB_{SC} = P_{SC} + N_{SC}$, where P_{SC} refers to positive consequences of successful collaboration and N_{SC} refers to negative consequences. P_{SC} is in the range [0,+1], and N_{SC} is in the range [-1,0].

NB_{UC} - the negative consequences of an unsuccessful collaboration, in the range [-1, 0].
$NB_{UC} = P_{UC} + N_{UC}$, where P_{UC} refers to positive consequences of unsuccessful collaboration and N_{UC} refers to negative consequences. P_{UC} is in the range [0,+1], and N_{UC} is in the range [-1,0].

NB_{NC} - the benefit obtained from choosing not to collaborate, in the range [0, +1].
$NB_{NC} = P_{NC} + N_{NC}$, where P_{NC} refers to positive consequences of non-collaboration and N_{NC} refers to negative consequences. P_{NC} is in the range [0,+1], and N_{NC} is in the range [-1,0].

The delegation occurs when

$$F - CT \geq 0.$$

A negative value of the Cooperation Function indicates a predicted negative outcome of the collaboration. However, if the Cooperation Threshold is itself negative, collaboration may still occur.

Outcomes

The Outcomes of the cooperation between the trusting parties are used in a feedback loop to incorporate the dynamic nature of trust over time. If a trustee is able to achieve their delegated task, then this could conceivably lead to an increase in their trustworthiness and hence an increased level of trust in future interactions. For instance, when a Data Manager succeeds in protecting the privacy of the Data Source, this could lead to an increase in their trust towards the Data Manager. The inverse result of this outcome is also easy to envisage, when the trustee has failed in their delegated task, resulting in a decrease in their perceived trustworthiness.

Perhaps a less obvious affect of a negative outcome would be when the trustee's perceived reputation increases despite them being unable to complete their delegation task. An example of this would be when the situation leads to the failure of the delegation, rather than any fault on the part of the trustee. Indeed, if the trustor is happy with the trustee's handling of a difficult situation this may even increase their perceived trustworthiness.

We now re-examine the earlier motivating example in a little more detail to show how we can quantify the trust transactions discussed above.

> **Motivating Example Quantified.** *Ann has a high disposition to distrust and relatively low disposition to trust. She is fairly convinced that her neighbour, Bob, will snoop into her personal records given the opportunity. In that case Ann imagines that the whole neighbourhood will be privy to the information she wouldn't be happy to share. She feels pretty upset about it as she has a mild*

disability, which she does not want to become public knowledge. Furthermore, Ann can perceive no possible benefit to participating in the survey. In the event that Ann does not participate, she would maintain her privacy and peace of mind; however, as she later learns, she could face a stiff monetary penalty of up to \$1,000.

We now quantify all the elements of the Cooperation Function.

1. *Ann is almost certain that Bob would snoop into her record, so we evaluate the likelihood of this happening to* 0.9*. Ann does not really know whether Bob would have such an opportunity as she is not familiar with the workings of the National Statistics Agency. However, knowing that Bob is high up in the hierarchy of the organisation she imagines that such an opportunity would be quite realistic. Thus we evaluate the likelihood of such an opportunity to* 0.5*. Subsequently, the likelihood that Bob will snoop into Ann's records is* $0.9 \times 0.5 = 0.45$*. Finally, we evaluate* $T_A(NSA, S)$ *to* $1.0 - 0.45 = 0.55$*, that is, the trust that Ann has in the National Statistics Agency in the context of the survey. Here A stands for Ann, NSA for National Statistics Agency and S for survey.*
2. *In this example we consider successful collaboration to be the case where Bob does not snoop into Ann's records, and unsuccessful collaboration where he does. We evaluate the positive consequences of collaboration, both successful and unsuccessful, that is,* P_{SC} *as well as* P_{UC} *to* 0 *as Ann can not perceive any benefit of participating in the survey. However, the negative consequences of an unsuccessful collaboration include loss of her privacy and exposure of her personal details to her community. Ann appears to be quite upset by such a possibility and thus the negative consequences can subjectively be quite high. It is not straightforward to quantify these consequences but we could refer to previous privacy invasion cases dealt with by the courts and the corre-*

sponding damages awarded. For the sake of this example, let us assume that the damages would be worth $3,000. Thus we quantify N_{UC} as -$3,000. Even in the case in which collaboration is successful, that is, where Bob does not compromise the privacy of Ann's personal records, there are still negative consequences as Ann does not like the idea of her personal information being held by the NSA as it makes a future potential privacy invasion possible. Thus we evaluate N_{SC} to -$500.

3. *If Ann chooses not to participate in the survey she will not gain any direct benefit (apart from preserving her privacy), so P_{NC} is* 0*. However, in that case she might be fined, thus N_{NC} is -$2,000.*
4. *Ann is a logical and rational person who does not normally take risks. She would normally not perform an action when she predicts a negative outcome. Consequently we evaluate her Cooperation Threshold to* 0*.*

Putting it all together,

$$
\begin{aligned}
&T_A(NSA, S) = 0.55\\
&NB_{SC} = P_{SC} + N_{SC} = 0 - \$500 = -\$500\\
&NB_{UC} = P_{UC} + N_{UC} = 0 - \$3,000 = -\$3,000\\
&NB_{NC} = P_{NC} + N_{NC} = 0 - \$2,000 = -\$2,000
\end{aligned}
$$

Before we can evaluate the formula, we need to normalise NB_{SC}, NB_{UC}, NB_{NC}, as they all need to be in their prescribed ranges. We normalise by dividing all of these values by the maximum absolute value, which in this case is $3,000. Thus we have,

$$
\begin{aligned}
&T_A(NSA, S) = 0.55\\
&NB_{SC} = -0.17\\
&NB_{UC} = -1.0\\
&NB_{NC} = -0.67
\end{aligned}
$$

$$
\begin{aligned}
F = & \\
& T_A(NSA,S)\times NB_{SC}+(1-T_A(NSA,S))\times NB_{UC}-NB_{NC} \\
& = 0.55 \times (-0.17) + (1 - 0.55) \times (-1.0) - (-0.67) \\
& = -0.09 - 0.45 + 0.67 \\
& = 0.13 \\
F - 0 & = 0.13
\end{aligned}
$$

Therefore our model indicates that Ann would proceed with the survey. However, before Ann become aware of the financial penalty for not complying, the $N_{NC} = 0$ and the overall cooperation function evaluated to -0.54 so Ann initially decided not to cooperate. This reflects Ann's unusually high privacy concern due to her concealed disability.

This motivating example provides some small insight into the challenges of quantifying the various components of our trust model. Clearly there is a highly subjective nature to many of the elements of this model, and we now discuss some of the complexities faced by a Data Manager wanting to quantify trust in his/her own statistical data warehouse system.

2.5 Challenges in Quantifying Trust

Several components of the above trust model need to be evaluated on a subjective basis by the trustor, namely the trustee's perceived reputation and the perceived risks and benefits for the situation. We now discuss some of the difficulties in evaluating these constructs in the context of a statistical data warehouse system.

2.5.1 Evaluating Trustee Reputation

We firstly look at how to evaluate the perceived reputation of the Data Source, that is, the person providing information, some of which is of a sensitive nature, to the Data Manager. One of the difficulties in assessing the reputation of the Data Source is the lack of feedback that can be obtained.

In general, the person providing the data will not be known to the Data Manager. In fact where data has been anonymised, there may be no way of ever knowing who provided which data. This means that the application of any traditional reputation system, such as those presented by Abdul-Rahman and Hailes [1], would be hard to achieve and leads us to require a different approach to assessing reputation.

One way in which it may be possible to gauge a general level of trustworthiness for a Data Source, is to ask them directly if they have been honest in providing their information. This can be done by surveying a representative sample of Data Sources from various demographics. However, there are several shortcomings of this approach. Firstly, there would be an additional cost associated with obtaining such information. Secondly, there may be a response bias if the participant feels as though the question is too revealing [123]. This is likely to occur when the Data Source has a low level of trust in the Data Manager, or the end receiver of their data, the Data User.

The problem discussed above can in part be solved through the use of a randomised response technique, such as Warner's [123]. Here the respondent is asked to select a question from a list of questions one sensitive and a number of non-sensitive questions, with a prescribed probability, and to answer the question accurately. The surveyor who knows the distribution for all non-sensitive questions is then able to estimate the responses to the sensitive question via probability analysis methods. However tempting this approach may seem, it realises a large cost to the Data Manager.

A less costly and perhaps more rounded approach to dealing with the difficulty of false data could in fact be to change the focus back onto the Data Manager themselves. We argue that in general a Data Source provides false data in the case where they do not feel their privacy is adequately protected. That is, they do not hold the Data Manager's reputation in high regard. However, that may not be the only reason for providing false data. It may also be that the individual has a generally low disposition to trust. There are other possible reasons for a Data Source not to provide useful data. It could be that their own competence is lacking, and they are in fact unable to accurately provide the data, even if willing to do so. While the latter scenario is beyond the scope of trust, in general we shall assume that

in the majority of cases, the Data Source is falsifying data due to a lack of trust in the Data Manager.

Evaluating the trustworthiness of the Data Manager from the point of view of the Data User and Data Source can be difficult. One reason for this is that often there is no direct contact between the parties, particularly when talking about the Data Source. One way in which reputation information about the Data Manager can be maintained is to employ the use of a trusted third party to record and update reputation information about the Data Manager.

When it comes to evaluating the trustworthiness of the Data User, their reputation will in part be dependant on who the Data User is. That is, some occupations are naturally more trustworthy than others. For example medical practitioners elicit a much higher level of trust than say market researchers [57, p.17]. It is also important to know for what purpose the data will be used. For example, a recognised research project attached to a well respected University is likely to involve more trust than a market research survey.

2.5.2 Evaluating Risk and Benefits

When deciding whether or not to cooperate, evaluating the level of risk and benefits in a situation is just as important as evaluating the trust. One of the potential negative consequences that a Data Manager has to consider when dealing with a Data Source is whether they will withhold their information, or potentially provide false information. However, this risk can be outweighed by the benefit the Data Manager receives due to an increase in their business by obtaining new data from the Data Source. When the Data Manager is dealing with the Data User, they need to consider the risk that the Data User may misuse the information. By the same token, the Data User will be more likely to cooperate when they perceive some benefit from the collaboration, such as some form of remuneration.

The risks involved for the Data Source include the risk that the Data Manager will not properly protect their privacy, or will provide their infor-

mation to third parties not previously agreed on. The Data Source also risks their sensitive information being revealed to a malicious Data User. For the Data User the main risks involve the perceived quality of the information they receive from the Data Manager. The risks include not only the Data Source withholding, or providing false information, but also the Data Manager's incorrect collection and management of the data before passing it on to the Data User.

2.5.3 Privacy Protection Framework

We now provide a general Privacy Protection Framework that can be applied to any statistical data warehouse system. The purpose of this framework is to assist a Data Manager in evaluating the likelihood that a Data Source and a Data User will proceed with collaboration. If this likelihood is low, the Data Manager may choose to invest resources towards promoting delegation. This can be done in one of the following ways.

1. The Data Manager may work towards increasing their reputation. One way to do this would be to have harsher privacy policies that the Data Manager and its partners, including the Data User, must abide by. The policy can be enforced by legally binding contracts, or by technical measures that make privacy difficult or near impossible. We briefly mentioned such techniques in Section 2.3. In either case, for the Data Manager's reputation to increase the applied measures need to be advertised to the partners. No change to reputation can occur until both the Data Source and Data User are made aware of the means that the Data Manager has undertaken in order to ensure privacy.

2. The Data Manager might have the ability to impose methods of coercion for non collaboration, for example, fines imposed on a Data Source for refusing to provide information.

3. Another method available to the Data Manager is offering incentives for collaboration as is often applied as a marketing tool. This could include such things as payments to the Data Source for providing

their data, acknowledgement of their contribution, free samples and discounts to the Data User, special offer and incentives.

Currently, it appears that the most common ways to encourage collaboration are limited to coercion, collaboration incentives and more recently to increasing reputation via privacy policies. In our Privacy Protection Framework we focus on the remaining method, namely technical security measures, as a way to encourage collaboration via an increase in the levels of trust of the parties involved. We next analyse how increased security influences the Collaboration Function and for which level of security the function reaches the Cooperation Threshold.

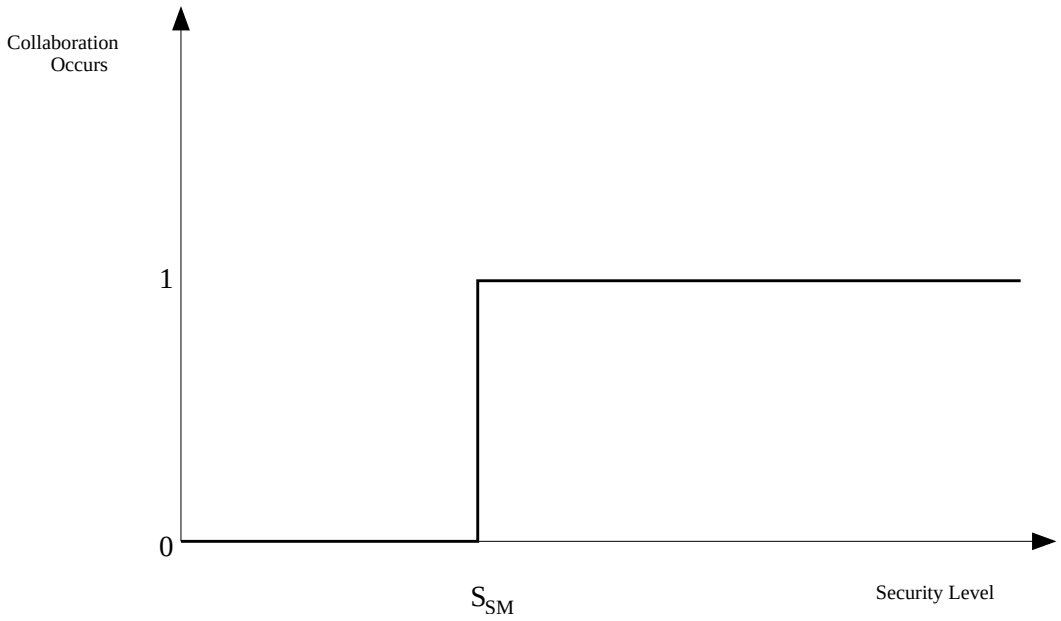

Figure 2.5: Security Threshold for Data Source.

1. The higher the security, the higher the reputation the Data Manager has in the eyes of the Data Source. We use S_{SM} to denote the security level at which the Cooperation Function reaches the Cooperation Threshold, that is, the minimum security level for the Data Source to collaborate providing that all other condition remain unchanged. Figure 2.5 shows a step function where the value 0 denotes non-collaboration and 1 indicates that collaboration will occur.

2. The higher the security, the higher the trust of the Data Manager that the Data Source will provide accurate data. We denote the lowest security level at which the Data Manager will proceed with collaboration by S_{MS}.

3. The higher the security, the higher the trust that the Data Manager that the Data User will not misuse the data. We denote the lowest security level at which the Data Manager will proceed with collaboration by S_{MU}.

4. The higher the security, the lower the confidence the Data User has in the quality of the data. As previously discussed, security measures can have a significant impact on data quality. We use S_{UM} to denote the security level at which the Cooperation Function reaches the Cooperation Threshold, that is, the maximum security level for the Data User to still collaborate providing that all other condition remain unchanged. Figure 2.6 shows a step function for the Data User.

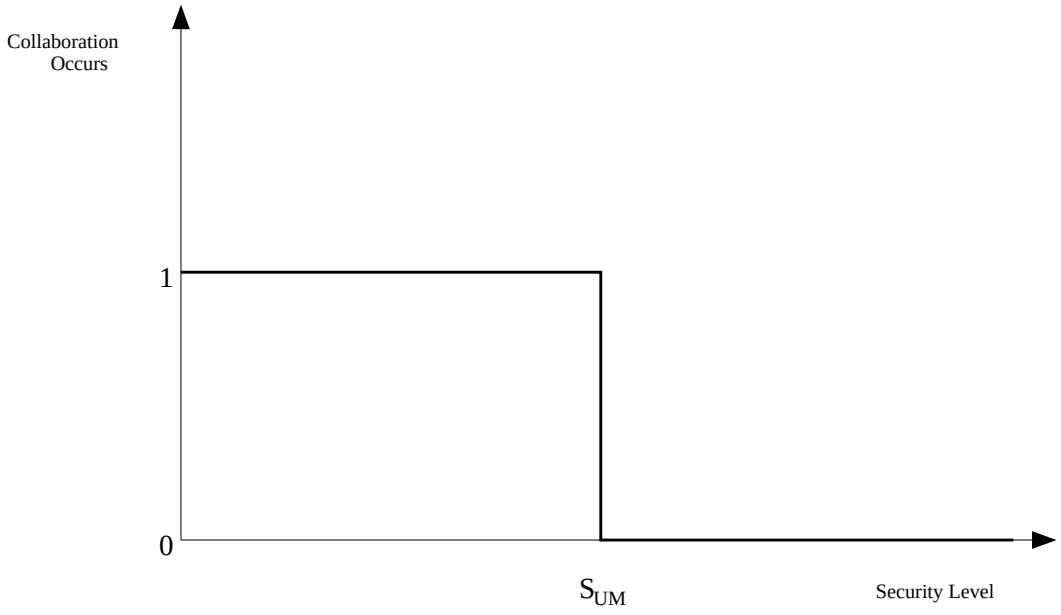

Figure 2.6: Security Threshold for Data User.

We now present the Privacy Protection Framework and give an overview of currently available technical measures.

The steps of the framework are as follows.

1. **Evaluating Security Thresholds.** The first step of our generalised framework is to evaluate the security threshold levels required by all the relevant parties, that is, S_{SM}, S_{MS}, S_{MU}, S_{UM}. It is important to note that in general not all instances of the Data Source will exhibit the same level of trust in the Data Manager and Data User and thus they may require different levels of security. Similarly, not all instances of the Data User require the same data quality, and thus their security thresholds may be different as well.

2. **Evaluating Security Range.** The Data Manager needs to determine the maximum security threshold among those required by the Data Source, as well as Data Manager in relation to both the Data Source and Data User. The security level S that will enable collaboration should be between this value and the security threshold, i.e. $S \in [max(S_{SM}, S_{MS}S_{MU}), S_{UM}]$.

3. **Selecting the Security Technique.** The final step is to decide on the most appropriate security measure to be employed in the given context, based on the information gathered in Step 1 and Step 2. We briefly discuss the main technical measures later in this section.

The first step of our generalised framework is to decide on the relative levels of security required by the Data Source and statistical quality required by the Data User. The relative levels of trust between these parties will be a major influencing factor in this step. The higher the trust of the Data Source in Data Manager and Data User, the less coercion necessary to ensure participation. It is important to note that in general not all instances of the Data Source will exhibit the same level of trust in the Data Manager and Data User and thus they may require different levels of security.

It might be possible for the Data Manager to provide higher levels of security to only those Data Source instances that exhibit low levels of trust as illustrated in Figure 2.7. Similarly, the Data Manager may choose to evaluate the trustworthiness of Data Users and to select the granularity of the released data accordingly (see Figure 2.7).

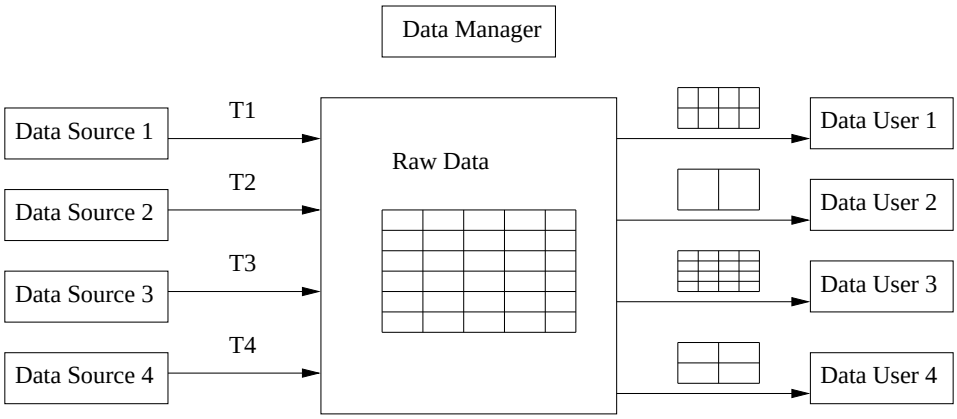

Figure 2.7: Security Framework for Statistical Data Warehouse system.

The Data Source preferences could be established through the preferences gathered from the Data Sources at the time of data collection. It appears that such granulated security would significantly improve the quality of the data released to the end user. A recent work by Williams and Barker [126] proposes a self protecting system where the Data Source selects a level of granularity they feel comfortable with and provides information at that level only. An important advantage of such a system is the handing back of control to the Data Source. A drawback is the increased burden placed on the Data Source who may not have the competence to accomplish this task successfully. We next briefly describe some of the security measures available to the Data Manager for privacy protection, as outlined in Step 3 of the Privacy Protection Framework. For each measure we comment on the security level and data quality achieved by the measure.

Query restriction mechanisms reject queries that could lead to a database compromise, and provide exact answers to other queries. Here the quality of released statistics is unaffected, but the amount of available statistics is typically overly restricted or a technique is easily subverted [2]. A well known example of a query restriction method is *cell suppression* [26], commonly used by census bureaus when releasing data in tabular form. In this method, cells that may lead to compromise are suppressed and replaced by a missing value indicator. Several query restriction methods for online databases have been developed, including *query set size control* [45], *query set overlap control* [33], and *partitioning* [22].

Query set size control releases a statistic to the user only if the query set size meets a particular condition set by the database administrator. This method was shown very early on to be easily subverted [30]. A more sophisticated approach is to consider the overlap between successive queries. Although slightly more successful, the method suffers from both a high overhead cost and is also susceptible to cooperation between multiple users [30]. Partitioning divides the individual records at the physical level into disjoint subgroups, called *atomic populations*. Queries are then answered when they only involve whole atomic populations [22]. As long as atomic populations do not contain less than the prescribed minimum number of records, a high level of security and precision can be achieved. The method does however have the drawback of being overly restrictive, greatly reducing the number of queries that can be answered and also reducing the usability of the data [2].

Unfortunately, none of these techniques can either guarantee full protection against compromise nor can they ensure that the amount of released statistics is maximised. The only exception is the so-called auditing of all previously answered queries, which allows for rejecting those and only those queries that would compromise the database [23]. The theoretical bounds for the amount of statistics that can be released without compromise have been established for general queries [12, 56], range queries [62, 10] as well as for the case when higher levels of security are needed [14, 16].

Noise addition techniques prevent a database compromise by introducing an error to results of queries or to the data itself. Then an intruder has a degree of uncertainty about the exact values even if individual values are disclosed [2]. The drawback of these techniques is the decreased quality of released statistics, where the statistical quality is measured by bias, precision and consistency of the modified data [2]. Noise addition is more challenging in databases with categorical attributes which do not exhibit a natural ordering as it is difficult to measure the added noise. A technique using decision trees was developed in [66] and another using Markov matrices in [28], which was applied to genetic databases in [53].

2.6 Conclusion

In this chapter we have presented a new perspective on the traditional view of statistical databases, one which incorporates the trust relationships that exist between the key stakeholders in the system. The direction of research in security is increasingly moving away from so-called 'hard security' systems to a softer approach, more able to cope with legitimate users providing false or misleading information [68]. By incorporating social control mechanisms, such as trust and reputation, into security models we are more able to cope with real life scenarios and ultimately can achieve a higher level of privacy and integrity, as well as usability in statistical data warehouse systems.

We have extensively examined the role that trust plays in a secure statistical data warehouse system, providing both researchers and database managers with insight into possible impacts of low levels of trust or high levels of risk in this type of system. To our knowledge this is the first time trust has been incorporated in this particular context. Additionally, we have provided a quantifiable trust model to better aid data warehouse managers in their decision making process. This should provide them with a better understanding of whether or not their system is operating effectively, and more guidance as to when they need to apply traditional statistical disclosure control mechanisms.

One direction for future research would be to investigate the trust attitudes of Data Sources, Managers and Users to examine several case studies and to empirically validate our model. Another important area requiring further attention is the evaluation of the situational trust. We intend to make a clearer distinction between subjective and objective components of trust, thus simplifying this highly complex construct. This is already partly captured within the concept of propensity to trust and distrust as a subjective component of trust. However, the subjective/objective distinction is not so clear in respect to the reputation of the trustee and the context of the situation and this will be the subject of our future work in this area.

Chapter 3

Comparative Study of Relevant Techniques

There are three kinds of lies: lies, damned lies, and statistics

–Benjamin Disraeli

Statistical database security focuses on the protection of confidential individual values stored in so-called "statistical databases" and used for statistical purposes. Examples include individual patient records used by medical researchers and detailed phones call records, statistically analysed by phone companies in order to improve their services. This problem became apparent in the seventies and has escalated in recent years due to the massive data collection and growing social awareness of individual privacy.

The techniques used for preventing statistical database compromise fall into two categories: noise addition, where all data and/or statistics are available but are only approximate rather than exact, and restriction, where the system only provides those statistics and/or data that are considered safe. In either case, a technique is evaluated by measuring both the information loss and the achieved level of privacy. The goal of statistical data protection is to maximize the security while minimizing the information loss. In order to evaluate a particular technique it is important to establish a theoretical lower bound on the information loss necessary to achieve a given level of

privacy. In this chapter, we present an overview of the problem and the most important results in the area.

3.1 Introduction

Statistical database security, also referred to as Statistical Disclosure Control, is concerned with protecting privacy of individuals whose confidential data is collected through surveys or other means and used to facilitate statistical research. In this context "individuals" can refer to persons, households, companies or other entities.

The earliest example of statistical databases is undoubtedly census data whose collection, storage and analysis went through a great transformation in the last 6,000 years. The first recorded census was taken in the Babylonian empire in 3800BC, for taxation purposes, and was then conducted regularly every 6 to 7 years. In ancient Egypt census started around 2500BC and was used to assist in planning the construction of the Pyramids [43]. The first modern census in Great Britain was taken in 1801 and was initiated out of concern that food supplies might fail to satisfy the needs of the country's growing population. The census asked only 5 questions of approximately 10 million people in 2 million households. As a contrast, 200 years later, UK census counted 60 million people in 24 million households and asked 40 questions [43].

Nowadays census is conducted regularly in virtually every corner of the world, and is used to facilitate planning by governments and various health and other authorities, for the benefit of the local population. In recent years, due to rapidly growing storage and processing capabilities offered by modern computers, data has become one of the most valuable commodities in both public and private sector of the society, as it supports both day to day activities and strategic planning. In addition to census, National Statistical Offices (NSO) in various countries also collect many other kinds of data, typically through surveys, and then process and disseminate the data to numerous other organizations and bodies. Moreover, many other entities have started collecting their own data, including hospitals, retail companies,

and a range of other service providers, either for their own research, strategic planning and/or marketing, or with the intention to sell it to other interested parties.

Not surprisingly, this massive collection and exchange of data has added to already growing public concern about misuse and unauthorised disclosure of confidential individual information. Data collectors and managers are currently facing a very challenging task of obtaining and providing rich data and unrestricted statistical access to users while at the same time ensuring that dissemination is done in such a way so as to make it impossible for the users to identify particular individuals. Unfortunately, these two requirements are typically mutually exclusive, and thus the most data managers can hope to achieve is preserving sufficiently high quality, while simultaneously making identification and disclosure as difficult as possible. This task is most commonly referred to as Statistical Disclosure Control (SDC), or statistical database security.

There are various measures one can apply in order to implement SDC. They generally fall into three groups: legal, administrative and technical. It appears that a simultaneous application of all three kinds of measures is necessary in order to ensure a satisfactory level of protection and to win public trust [115]. In this chapter we focus our attention on technical measures to ensure privacy.

An important but still not fully explored issue refers to the information that an intruder has about statistical database. This information is usually referred to as "supplementary knowledge" (SK). An intruder with extensive SK is in a position to disclose more confidential information from the database, and will need less effort to do that than a user without or with little SK. Thus, the so called "intruder modeling" is an important step in designing an adequate SDC measure, but unfortunately more work is needed in thus direction [36]. In Section 3.2.3 we shed some more light on this important issue.

There are a few different ways for dissemination of statistical databases to occur. A dissemination method has an impact on the level of security that can be achieved, and also dictates what SDC techniques can be applied.

Traditionally, NSOs have been disseminating statistical databases in the form of summary tables, usually two dimensional. Summary tables contain aggregate data and thus are less exposed to the risk of statistical disclosure. However, the level of detail in summary tables does not allow for some more complex analysis of data that has now been required by users.

Consequently, NSOs have recently started releasing anonymised microdata files, often referred to as Public Use Microdata Sample (PUMS), which can be public use or licensed files [115]. Both types contain very detailed (raw) data but they differ in the level of anonymisation. Public use files are generally available without licensing and require a high level of anonymisation, such that identification of records is not very likely. On the other hand, licensed files require the signing of legal undertaking by all the users of the file. Identification of individual records is in general more likely in licensed than in public use files.

Finally, Remote Access Facilities (RAF) and Data Laboratories (DL) provide users not with microdata files but rather with an access channel through which they can submit statistical queries and receive responses [115].

We conclude the Introduction by considering briefly the two main groups of SDC techniques that can be deployed to protect the confidentiality, namely restriction techniques and noise addition techniques. *Restriction* techniques restrict the information that is available to a user either directly or through response to their queries. However, all the information that remains available is exact. One the other hand, *noise addition* techniques preserve the availability, but not the exactness of the data. In other words, all the data is available but it is only approximate as it went through a perturbation process before being released to users. Both groups of techniques have their advantages and disadvantages and it may be necessary to apply both simultaneously in order to provide a required level of security.

The organization of the remainder of this chapter is as follows. In the next section we take a closer look at the abstract model of a statistical database and illustrate some important concepts. In Section 3.3 we discuss restriction techniques and in Section 3.4 noise addition. Section 3.5 is devoted to studying information loss and disclosure risk, and Section 3.6 to

software packages. We give concluding remarks in Section 3.7.

3.2 A Closer Look

In this section we take a closer look at the abstract model of statistical databases, introduce some important concepts from statistical database theory and illustrate them on our working example.

3.2.1 Abstract Model

Table 3.1 represents what could be a part of a census. This is, of course, just a toy example to help us exemplify some concepts. The real census in most countries typically contains millions of records and tens of variables.

In its abstract model a statistical database is a two dimensional table where each row describes an individual, whether that is a person, business or some other entity. In our Census Database example, each row corresponds to an individual household. Each column describes one property of the individual. Following the database terminology, we refer to these properties as *attributes*. In the Census Database, *HOH* stands for "Head of the Household", *NoA* (*NoC*) stands for "Number of Adults (Children)", *Dw_o* for "Dwelling Ownership" and *Dw_rep* for "The Need for Dwelling Repairs".

Each attribute has a domain associated with it, that is, a set of legal values that attribute can have. For example, in our Census Database the domain of the attribute *NoC* is the set of non-negative integers (possibly with a prescribed maximum value), while the domain of *Dw_rep* is the set {no, minor, major}. The full list of domains for the attributes in the Census Database is shown in Table 3.2.

3.2.2 Attribute Types

Attributes in a statistical database can be either confidential or nonconfidential, sometimes also referred to as identifiers and sensitive attributes,

	Address	*HOH name*	*HOH gen*	*HOH income*	*HOH age*	*NoA*	*NoC*	Total *income*	*Dw_o*	*Dw_rep*
1	12 First St	M. Smith	F	70	34	1	1	70	Y	No
2	37 Grey Ave	J. White	M	99	39	2	2	99	Y	Major
3	100 Main St	F. Brown	F	33	21	1	0	33	Y	No
4	4/18 Hunter Rd	M. Doe	M	21	21	1	0	21	Y	Major
5	30 Second St	J. Black	M	21	27	2	1	40	Y	Minor
6	15 Main St	H. Jones	F	55	38	3	2	110	N	No
7	67 River Rd	J. Smith	F	84	51	1	1	84	Y	Minor
8	92 Third Ave	A. Chang	F	67	35	2	3	100	N	No
9	2 Kerry Ave	J. Black	M	23	44	2	2	50	Y	Major
10	35 Smith St	B. Ross	M	34	28	2	3	34	N	Major
11	200 King St	K. James	M	45	47	2	1	45	N	No
12	7 Nice Rd	J. Reed	M	12	60	1	0	12	Y	Minor
13	82 Michael St	C. Doe	F	56	33	2	2	70	Y	Minor
14	26 William St	M. Chen	F	23	31	2	3	45	Y	Major

Table 3.1: Census Database for Town X

respectively. In the Census Database, arguably, nonconfidential attributes would be *Address*, *HOH name*, *HOH gen*, *NoA* and *NoC*. The remaining attributes are treated as confidential.

Nonconfidential attributes are public knowledge and likely to be known to an intruder. These attributes may be used to identify individual records. Some attributes can identify individual records directly, and they are referred to as *direct identifiers*. In the Census Database, *Address* and *HOH name* act as direct identifiers. Other can only identify the records in combination with other attributes and they are called *indirect identifiers*. A subset of indirect identifiers that can be used together to identify records is referred to as a key. Note that there is an important difference between a key in the database theory and our key here: in the database sense, a key is a combination of attributes that uniquely identifies every record in the database. In other words, there are no two records with the same values in all the attributes of the key. In our context, some values of a key may uniquely identify a record, while other values may not. For example, in the Census Database, (*HOH gen*, *NoA*, *NoC*) act together as a key. The record 13 is uniquely identified by the key value (F, 2, 2). However, the key value (M, 2, 2) matches both record 2 and record 9 and thus (*HOH gen*, *NoA*, *NoC*) would not qualify as a key in a database sense.

The first logical step in protecting the confidentiality in a statistical

Name	Description	Domain
Address	address of the household	a set of strings of characters with the prescribed maximum length
HOH name	name of the head of the household (HOH)	a set of string of letters with the prescribed maximum length
HOH gen	gender of the head of the household	the domain is the set {F,M}
HOH income	income of the head of the household in thousands of dollars	the set of non-negative integers
NoA	number of adults in the household	set of non-negative integers, with the prescribed maximum value
NoC	number of children in the household	set of non-negative integers, with the prescribed maximum value
Total income	total income of all the members of the household in thousands of dollars	the set of non-negative integers
Dw_o	dwelling ownership	the set {yes, no}
Dw_rep	the need for dwelling repairs	the set {no, minor, major}

Table 3.2: Domains of attributes for Census Database in Table 3.1

database would be to remove all direct identifiers, which is typically done in practice before data is disseminated. However, it would be wrong to assume that this step alone is enough to truly anonymise the data. A big percentage of records, especially in smaller databases are still identifiable using keys comprising indirect identifiers. For example, about 25% of Australian households are uniquely identifiable based only on age and the size and sex structure of the household [115].

Note that if the Census Database were released as a licensed anonymised microdata file, then it would probably be enough to remove direct identifiers, i.e., attributes *Address* and *HOH name*. However, if the Census Database were to be released as a public use file, then removing direct identifiers would not suffice as some records can be identified from keys containing only indirect identifiers. In that case, one of the techniques described in Section 3.3 or Section 3.4 should also be applied. If the data is not released in the form of a microdata file but rather accessed by users either through RAFs or DLs no identifiers need to be removed, however some protection technique would have to be applied in order to ensure privacy. In addition to microdata files, RAFs and DLs, the Census Database can also be released in the form of summary tables. Table 3.3 is an example of such a 2-dimensional table.

	Number of Children					
		0	1	2	3	Total
Head Of	M	33	85	149	34	334
Household	F	33	154	180	145	479
Gender	Total	66	239	329	179	813

Table 3.3: *Total income* summary table of Census Database for *HOH gen* and *NoC*

All users of a statistical database must have the so-called *working knowledge* which refers to the user's familiarity with attributes contained in the database and their domains. If the data is released through RAFs and DLs, then the working knowledge is absolutely essential, otherwise the user would not be able for formulate a statistical query. The knowledge of attribute domains is also important in the case when data is released in the form of summary tables or anonymised microdata files.

3.2.3 Supplementary Knowledge

In addition to working knowledge, a user of a statistical database may also have the so-called *supplementary knowledge* (SK). Miller [91] distinguishes between SK of Type I, II and III. SKI refers to knowledge of the value of a key, which consists either of a direct identifier or a combination of indirect identifiers. SKII refers to knowledge of a value of a confidential attribute, while SKIII includes any SK that is not of SKI or SKII.

A user with SKI could be able to identify one or more records in the database. Statistical database compromise (disclosure) occurs if such a user can then disclose the values of confidential attributes for those particular records. This is also known as *exact compromise* or *1-compromise* to stress the fact that an exact single confidential value had been disclosed. However, if a user has SKII, preventing only exact compromise may not provide adequate security. If, for a given confidential attribute, a user learns the sum of values for k records, and as a part of his/her SKII he/she already knows $k-1$ of them, he/she can easily deduce the remaining confidential value and compromise the database. We define a *k-compromise* to be a disclosure of

a statistic based on k or less records. It is often possible for a user to conclude that a particular record does not have a certain value in a confidential attribute. This is referred to as a *negative compromise*.

Approximate compromise occurs when a user can learn that for a particular record the value of a confidential attribute lies in a range r with some probability p. This often happens when data is released in the form of summary tables and it is expressed as "n-respondent, k%-dominance" rule, where n individuals contribute with k% or more of the value of a particular cell in the table. For example, if only one individual contributes with 99% of the total value, then it is easy to estimate that particular value with an error of 0.5%. This rule has been traditionally used by the NSOs for a long time. Finally, a *relative compromise* occurs when a user can learn the relative order of magnitude of two confidential values in the database [94]. For example, in the Census Database an intruder may be able to disclose that the total income of household 3 is greater than the total income of household 4.

3.3 Restriction

Techniques that restrict statistics can generally be divided into three broad categories: *global recoding*, *suppression* and *query restriction*. The purpose of global recoding and suppression is to eliminate rare combination of values in attributes of a key, that is, combinations that appear either in a single record or in a small number of records. Typically, global recoding is applied first to eliminate a majority of the rare combinations. Suppression is applied next, to eliminate the remaining ones. It is important to note that both techniques introduce some *information loss* to data. They both can be expressed as optimisation problems where information loss needs to be minimised. For a very good overview of these two techniques an interested reader is referred to [125].

3.3.1 Global Recoding

Global recoding (GR) transforms the domain of an attribute. If the attribute is categorical, GR implies collapsing a few categories into one. For numerical attributes GR defines ranges of values and then replaces each single value with its corresponding range. For example, to eliminate rare combinations in values of indirect identifiers in the Census Database, we could replace the domains of the NoA and NoC by ranges "0 or 1", and "2 or more".

GR can be combined with query restriction techniques in such a way so as to (suboptimally) minimise the number of collapsed categories and maximise the percentage of queries that can be answered without compromise [8]. GR can also be applied to data released in the form of summary tables, in which case it is referred to as "table redesign" or "collapsing rows or columns" [124]. For example, in summary Table 3.3, the cell (1,4) describing the total income of all the households with 3 children and male head of the household is sensitive as it contains a single household (record 10). Similarly, the cell (2,1) is sensitive as it also contains a single household (record 3). In order to eliminate the sensitive cells, the table can be redesigned by collapsing NoC values 0 and 1 into a single category "0 or 1" and values 2 and 3 into a single category "2 or more". The new summary table is presented in Table 3.4.

		Number of Children		
		0 or 1	2 or more	Total
Head Of Household Gender	M	118	183	334
	F	187	325	479
	Total	305	508	813

Table 3.4: Redesigned Table 3.3 after global recoding.

3.3.2 Suppression

Suppression replaces the value of an attribute in one or more records by a missing value. When applied to microdata, suppression is called *local suppression*, and when applied to summary tables it is called *cell suppression*.

It is important to note that in the case of summary tables it is generally not sufficient to suppress sensitive cells. For example, if in Table 3.3 we suppressed the two sensitive cells (1,4) and (2,1), as in Table 3.5, an intruder would still be able to deduce their values. They would just need to subtract the values of all the other cells in the corresponding row (column) from the marginal total for that row (column). Thus we need to suppress at least 2 cells in each row or column that is affected by suppression. Table 3.6 shows an example with minimum number of suppressions that we need to perform - in this case four. These additional suppressions are referred to as *secondary suppressions*. When choosing cells for secondary suppression, the following three requirements should be satisfied [124]. Firstly, no empty cells should be suppressed. Table redesign can be applied first, in order to eliminate or minimize empty and sensitive cells. Secondly, in order to minimise the information loss, the total number of suppressed cells should be as small as possible. Finally, after the secondary suppression, an intruder will still be able to determine a feasibility range for each suppressed cell. For example, from Table 3.6 one can conclude that the value for cell (1,4) lies in the range [1,67]. The third requirement that secondary suppression needs to satisfy is that the feasibility ranges are not too narrow. Secondary cell suppression is in general a challenging problem and can be formulated and (sub-optimally) solved as a linear or mixed integer programming problem [125].

	Number of Children					
		0	1	2	3	Total
Head Of	M	33	85	149	X	334
Household	F	X	154	180	145	479
Gender	Total	66	239	329	179	813

Table 3.5: Table 3.3 after primary cell suppression.

	Number of Children					
		0	1	2	3	Total
Head Of	M	X	85	149	X	334
Household	F	X	154	180	X	479
Gender	Total	66	239	329	179	813

Table 3.6: Table 3.3 after secondary cell suppression.

3.3.3 Query Restriction

The third type of restricting techniques is the so-called *query restriction*, specifically tailored towards RAFs and DLs dissemination techniques, where users are not provided with microdata files but can rather with a channel through which they can interactively ask queries. Since users will never actually see the data, it is not necessary to remove direct and indirect identifiers. The user-posed queries are either answered exactly, or are rejected, and the decision as to which queries to answer is made based on one of the following techniques [30].

The early techniques include *Query Set Size*, *Query Set Overlap* and *Maximum Order* control, which accept or reject queries based on their size, overlap with those previously answered or the total number of attributes involved in the query, respectively. All of these techniques were shown to be easily subvertible; additionally, Maximum Order unnecessarily restricts queries that do not lead to a compromise, and Query Set Overlap is computationally expensive as it requires storage and manipulation of all previously answered queries.

Partitioning groups records at the physical level into disjoint subgroups (called *atomic populations*), each containing an even number of records [22]. A query is answered only if it involves whole atomic populations. Partitioning provides superior security but it tends to be overly restrictive.

Threat monitoring and *auditing* involve keeping logs of all the answered queries, either for each user separately, or collectively for all users [23]. A new query is only answered if together with all previously answered queries it does not lead to a compromise. The superiority of auditing lies in the fact that it is the only technique that can actually guarantee prevention of compromise without being overly restrictive. Recently there has been a renewed interest in this technique and many enhancements have been proposed [15, 7, 82, 83, 69, 77, 72]. The main drawback of auditing is its excessive time and storage requirements [2]; however, for special types of queries such as additive queries, these requirements can be significantly reduced.

3.4 Noise Addition

The basic premise behind any *noise addition* technique is to mask the true values of the sensitive information by adding some level of error to the data. This is done in a controlled way so as to best balance the competing needs of security and data utility. The introduction of noise to the released statistics makes the task of ensuring the quality of statistical analyses a challenging one. Yet there are benefits for using noise addition methods, one being their relative ease of implementation and low running costs.

Noise addition techniques can be categorised in several ways. One way is by the type of attribute they can be applied to. Generally techniques that work well for numerical data do not perform well on categorical data and vise versa. Here by "numerical" we mean values that have a natural ordering, regardless of whether or not the values are actually numbers. Techniques can also be classed by which stage the perturbation is added to the data. It can be added prior to release of the statistics, in which case the original database is generally replaced by a perturbed database on which the statistical queries are performed. This type of method is generally known as the *data perturbation* approach. For *output perturbation* techniques, the perturbation is performed on the results of queries on the original data set. We now examine some of the classes of noise addition techniques in more detail.

3.4.1 Additive Data Perturbation

Additive noise methods for data perturbation were first introduced in the late eighties and early nineties by Kim [70] and Tendick [112], and subsequently in more detail [46, 71, 130]. The Kim and Tendick method, also known as *Correlated-Noise Additive Data Perturbation* (CADP) [97] uses correlated noise to perturb a microdata file. The perturbed attribute Y is obtained by adding a noise term ε to the confidential attribute X, that is, $Y = X + \varepsilon$, where ε has a multivariate normal distribution with zero mean and a covariance matrix equal to the covariance matrix of the confidential attribute X, multiplied by the level of perturbation [97]. An interested reader

is referred to [97] for a good summary of such early noise addition techniques, which can be seen as special cases of the *General Additive Data Perturbation* (GADP), for numerical attributes described by the multivariate normal distribution. GADP perturbs both confidential and non-confidential attributes, maintaining the correlations between attributes. For large data sets GADP performs well in both the data utility and disclosure prevention stakes, but like many methods does not perform well on small databases [99]. The so-called *Enhanced General Additive Data Perturbation* (EGADP) can be effectively used on both large and small data sets [98].

3.4.2 Probability Distribution

Data distortion by probability distribution involves the building of an accurate statistical model M of the original data x. The perturbed data set y is then created by randomly drawing records from the model [127]. The technique was first introduced by Liew et al. for use with confidential numerical attributes [78]. There are three main steps to the technique. Firstly, for each confidential attribute the underlying density function must be identified and associated parameters estimated. The next step is to use the estimated density function in the generation of a distorted series of data for each sensitive attribute. The final step of mapping and replacement of this distorted data in place of the original confidential attributes is required when the masked confidential attributes are to be analyzed alongside the non-confidential attributes [78].

Burridge [19] uses such a model based method of data perturbation for his *Information Preserving Statistical Obfuscation* (IPSO). The attributes are grouped into two distinct sets, namely public data (y) and specific survey data (x). For a subset of records, a model for the conditional distribution $y|x$ is created. Then a sufficient statistic T is generated based on the information contained in y. Then the perturbed dataset (y', x) , generated from the conditional distribution of $Y|(T, x)$, is released to the researcher. The advantage of this method is that it preserves the values of statistics in the sample for both large and small microdata files, unlike the GADP class of methods [19].

3.4.3 Matrix Masking

Duncan and Pearson showed that many perturbative methods are a specialisation of *matrix masking*, which can be described as follows [39]. Given a microdata file X, the data user is given the masked version of the file $M = AXB + C$, where A is a record-transforming matrix, B is a variable transforming matrix, and C the noise or displacing mask [39]. *Random Orthogonal Matrix Masking* (ROMM) is a matrix masking technique for continuous microdata [114]. A random orthogonal matrix is drawn from a distribution G and applied to the original data to obtain the perturbed microdata. This microdata is then released along with the exact distribution G and the knowledge of how the microdata has been obtained [114]. The method preserves sample means and sample covariances and also controls the amount of perturbation.

3.4.4 Categorial Techniques

We define a categorical attribute to be one that has no inherent ordering of the categories. This property makes it particularly difficult to sensibly add noise to such attributes. One of the earliest techniques specifically designed for categorical attributes is inspired by Warner's random sample method [123]. One of the more promising disclosure protection methods for categorical data, proposed by a group of researchers at Statistics Netherlands, is known as *Post RAndomization Method* (PRAM) [28]. PRAM can be applied to one or more attributes simultaneously. The method is similar to the randomized response technique, in that it misclassifies categories with known transition probabilities, allowing for unbiased estimates of certain underlying statistics to be obtained[28]. This noise addition technique will be discussed in more detail in Chapter 4 when we investigate its use in a privacy protection framework for genetic databases.

One novel technique to arise from the area of data compression is *Lossy Compression*, based on the well known JPEG algorithm [35, 87]. The basic premise behind the method is to convert the numerical data file to pixels, compress the resulting file, which is then regarded as a masked data

file. Scaling of the original data will generally be required to achieve pixel grayscale values [35].

Du et al. [38] have applied a Bootstrap method to additive fixed data perturbation techniques to evaluate the security of such methods. An detailed analysis of binary random data perturbation has been conducted in [119].

3.5 Information Loss and Disclosure Risk

A good SDC technique finds a balance between minimising *information loss* on one hand and a *disclosure risk* on the other hand. This is a challenging task and can be expressed as a *multiple objectives decision problem* [116].

In principle, for currently used noise addition techniques, a user can estimate the distribution of original data, which can sometimes lead to disclosure of individual values, especially when the number of attributes is large [37]. *Re-identification disclosure* occurs when a malicious user re-identifies a record and then learns the value of a sensitive attribute for that record; *prediction disclosure* occurs when a user estimates the value of a sensitive attribute without necessarily re-identifying the record [36]. *Re-identification risk* itself is very difficult to estimate and can be expressed as a risk per record or an overall risk [125]. One of the proposed measures for re-identification risk is the probability that a unique match between a microdata record and a population unit is correct [108].

In the case of a noise addition technique, the information loss is measured by deterioration in data quality, in terms of *bias* (the difference between unperturbed statistics and the expected value of its perturbed estimate), *precision* (variance of an estimator) and *consistency* (absence of contradictions and paradoxes) [2]. More generally, entropy can be used to measure information loss for any technique that modifies the original data before releasing it to users. Such techniques include global recoding, suppression, and noise addition techniques. In other words, these are all the techniques except the query restriction ones where the information loss [%] is measured

as $100\% - U$, where U is the usability, that is, the percentage of queries that are accepted under the given technique.

The idea behind the entropy based measures is to evaluate the uncertainty that the user still has about the original data if he/she has been provided with the modified data. If the uncertainty is zero, there is no information loss. Formally, information loss can be expressed as $H(Original|Modified)$, where $H(Original|Modified)$ is the equivocation, or the conditional entropy of the original data given the modified data. For example, in the case of local suppression $H(Original|Modified) = H(Original)$, which means that all the information has been lost. The main drawbacks of using the entropy to evaluate an information loss are that it is a formal measure that is not always easy to calculate, and that it does not allow for the data owner's preferences regarding, for example, the importance of particular attributes. Other information loss measures include "subjective" measures that are based on weights indicating which attributes are more important than others and thus should be modified as little as possible [125]. Yet another measure evaluates how "different" the modified and the original data sets are, in terms of mean square and absolute error, and mean variation of the original and perturbed data sets and their covariance and correlation matrices [34].

In order to evaluate and compare different SDC techniques, it is important to determine the minimum information loss necessary to achieve a given security level. For example, it was shown that, for binary data protected by noise, a clever user who has access to perturbed subset sums can in fact reconstruct most if not all of the original values, unless the added noise is of the magnitude $O(\sqrt{n})$, where n is the number of records in the database [32]. If the data has been protected by a query restriction technique, we would like to determine the maximum usability for a given security level, that is, the maximum percentage of queries that can be answered without database compromise. Determining maximum usability is a challenging problem, but some progress has been made for additive queries. For example, for a security level that requires prevention of exact compromise (or 1-compromise) in a database of n records where only additive queries are allowed, maximum usability is of order $\Theta \sum(n^{-\frac{1}{2}})$ [92].

This means that in a database with 100 records only 10% of the queries is answerable, and for 10000 records only 1% of queries is answerable. This is of course unacceptably low, which indicates that this level of security cannot be reasonably achieved by query restriction techniques alone. If the prevention of k-compromise is required, the maximum usability is $O(n^{-1-\frac{k}{2}})$ [55], and if a relative compromise is to be avoided then the maximum usability is $\Theta(n^{-\frac{3}{2}})$ [93, 55]. Thus in order to avoid a k-compromise, we can only answer a very small portion of additive queries. However, situation is very different for *range* queries in multidimensional databases (OLAP cubes). For large m-dimensional databases that contain at least one record in each cell the maximum usability is at least $(2^m - 1)/2^m$ [62, 11]. Thus, most queries are answerable without causing a compromise. General OLAP cubes have been further studied in [121, 122, 120]. If prevention of k-compromise is required then the maximum usability in a 1-dimensional databases is $\Theta(k^{-2})$ [14, 16].

3.6 Software Packages

In the past, most of the SDC techniques and software was produced by the NSOs for use within their own organisations. In 1995 Statistics Netherlands developed a prototype version of a software package, ARGUS, to protect microdata files against statistical disclosure. This prototype served as a starting point for the development of μ-ARGUS, a software package for the SDC of microdata. The project also saw the development of τ-ARGUS, software devoted to protecting tabular data [44]. The SDC methods that can be used in μ-ARGUS include global recoding, local suppression, microaggregation and PRAM to name but a few. The sister package τ-ARGUS uses a combination of sensitive cell recognition and cell suppression to protect tabular data. The GHMITER hypercube heuristic software developed by the Statistical office of Northrhine-Westphalia/Germany, has now also been incorporated into τ-ARGUS. Other commercially available software packages include the cell suppression packages ACS, which builds on an earlier software, CONFID, developed at Statistics Canada, and Datafly which was developed specifically for the anonymisation of medical data [110].

3.7 Conclusion

Statistical Database Security has undergone a big transformation in the last few decades. What started off as disconnected efforts within NSOs and academia is now developing into a joint international venture. The importance of unifying policies as well as control and dissemination methods across national borders has been repeatedly stressed at conferences and other international gatherings of statisticians [115].

In this chapter we have provided a comprehensive overview of the complexities involved in protecting statistical databases from a potential intruder. Existing methods of protection have been presented, as have respective advantages and disadvantages of those techniques. Now that we have some idea of the difficult task faced by a data manager attempting to balance the competing needs of disclosure risk and quality assurance, we will next provide a comprehensive framework for the protection of genetic databases.

Chapter 4

Privacy Protection Framework for Genetic Databases

The most certain test by which we judge whether a country is really free is the amount of security enjoyed by minorities.

–Lord Acton

Genetic databases generally contain a combination of personal information, medical history as well as some form of genetic testing and/or sequencing results. Much of this supplementary data associated with the genetic information can be of a sensitive nature, and the individual about whom this information pertains may be concerned about the privacy of this information. Additionally, the often categorical nature of the data can pose special challenges for a data manager wanting to adequately protect the individual's privacy. As we saw in Chapter 3, it can be a much less straightforward task to balance the competing needs of privacy and data quality when dealing with categorical data than when attributes have a natural ordering.

In this chapter we will outline how the application of a SDC method specifically designed for categorical attributes, the so-called PRAM method

[54], can be applied with other techniques to provide a privacy protection framework. In developing the framework we will take into account the functional dependencies that can exist between different attributes in the database, and also consider how we might best implement a measure of similarity between categorical values within an attribute.

4.1 Introduction

Many countries around the world are currently putting a lot of effort and resources into genetic technology. Australia is no exception. This initiative is seen as particularly important as it can not only have general health benefits for the population, but also "has enormous potential to create wealth and knowledge-based jobs for Australia" [61, p.25]. The expectations from genetic research are immense and wide ranging. They include improvement in diagnosis of diseases, detection of genetic predisposition to diseases, developments in gene therapy and the design of drugs tailored to an individual's genetic make-up [63]. Therefore, genetic research is currently receiving vast amounts of funding.

At the same time, surveys conducted around the globe show that there is a growing mistrust in genetic technology. Particularly illustrative is the fact that only 41% of Europeans surveyed in 1999 [40] believed that biotechnology would improve the way of life in the next 20 years, down from 46% in 1996. The results of this survey indicate that the only technology that enjoys less confidence than biotechnology is nuclear power. It seems that most of the public uneasiness is caused by genetically modified food and cloning, and the fear that human genome research will lead to discrimination by employers and insurance companies.

While at present in Australia there is no legislation specifically for use of genetic information by employers, in the US there is an Executive Order (2000) protecting federal employees from genetic testing as a hiring or benefits requirement [24]. However, this order allows for the disclosure of the genetic information of employees for the purpose of providing occupational and health researchers with data. Moreover, no other employees apart from

federal are covered by this legislation. This absence of appropriate legislation increases the danger that individuals will not agree with genetic testing. Even when testing is used for treatment or research purposes, individuals may fear that the genetic information will be passed on to the employers and insurance companies.

Many believe that genetic data is more sensitive than other medical or personal data, as it carries more information about a person's health. Additionally, it can provide insight into the potential health risks to the individual, years into the future. A concern is that possibly imprecise genetic information could be used by insurance companies, employers or public authorities in the decision-making process in relation to the individual.

There are numerous examples of genetic research databases used around the world. The best known example is the Icelandic Health Sector Database which was proposed to consist of three separate yet connected databases, the Health Database, the Genotype Database and the Genealogical Database [6]. The databases would contain information about the majority of the Icelandic population, and they would be the first centralised databases of this kind in Iceland.

We argue that such databases do not have to represent a threat to individual privacy. It is not necessary for such databases to contain 100% accurate data. We propose perturbing some of this data, so as to make identification of individuals impossible and at the same time preserve the usefulness of the database for medical research. To date, this issue has not been successfully solved in practice. For example, the Icelandic Health Sector Database specified the use of query set size control restrictions as a statistical inference countermeasure, with the minimum query set size being set to ten [6]. This method, although very simple to implement, is known to be very weak [31]. The security function would also employ noise addition techniques, however, the actual technique and parameters were not specified [6]. The original specifications did not include any statistical protection whatsoever, and the above measures were only added after harsh criticism by Ross Anderson [3], an expert in computer security. He pointed out that originally the only measure proposed for de-identification of medical data was the removal of obvious identifiers such as name, with encrypted social

security numbers used as personal identifiers for records [6]. However, it is important to note that these measures are not sufficient to provide complete anonymity as any combination of attributes that uniquely identifies an individual can be used to infer confidential individual values [3]. This clearly indicates that more effort was required to provide an adequate level of security in the Icelandic Health Sector Database.

The framework proposed in this chapter will solve this problem by carefully adding noise to the personal and health data so as not to jeopardize statistical usefulness of the database but rather to protect the privacy of the individuals.

The organisation of the remainder of this chapter is as follows. In the next section we give a brief reminder of the existing techniques for the protection from statistical disclosure and we concentrate on one method in particular, the so-called PRAM in Section 4.3. In Section 4.4 we investigate the suitability of this method for genetic databases. In Section 4.5 we propose the use of techniques for clustering categorical attributes in order to construct PRAM matrices. In Section 4.6 we outline the framework for protecting confidentiality in genetic databases and we give concluding remarks in Section 4.7.

4.2 Privacy Protection Techniques

As discussed in Chapter 3, the Statistical Disclosure Control (SDC) problem is difficult to solve and is typically dealt with in one of the following two ways: restricting queries that users can pose to the system and adding noise to the data. When noise is added to the data, an intruder has a degree of uncertainty about the exact value of the data, even if an individual value is disclosed [2]. In either case it is important to find the right balance between the security, and usability of the database and/or the statistical quality of released queries. We shall now briefly reiterate each of these two strategies, and refer the reader to Section 3.3 and Section 3.4 for a more detailed review.

Query restriction mechanisms reject queries that could lead to a database compromise, and provide exact answers to the other queries. Here the quality of released statistics is unaffected, but the amount of available statistics is typically overly restricted or a technique is easily subverted [2]. An example of the later is the query size restriction control proposed for use in the protection of the Icelandic Health Sector Database.

Noise addition techniques prevent a database compromise by introducing an error to results of queries or to the data itself. The drawback of these techniques is the decreased quality of released statistics, where the statistical quality is measured by bias, precision and consistency [2].

However, the vast majority of these techniques only deal with numerical attributes and as such are not suitable for genetic databases which typically contain categorical attributes (those whose values do not exhibit a natural ordering). One exception is an early paper by Warner [123]. Recently more papers dealing with noise addition to categorical attributes have appeared [28]. The first paper in this series was published by researchers at Statistics Netherlands and it introduces the Post RAndomisation Method (PRAM), a technique for disclosure protection of categorical variables in microdata files [28]. PRAM is designed to provide protection to confidential attributes and at the same time preserve the underlying statistical quality of the perturbed data file. Using a predefined probability distribution, the values of one or more attributes in a record are perturbed when applying PRAM. We shall now describe exactly how this is achieved, using the notation from [28, 54, 51].

4.3 Post RAndomisation Method (PRAM)

A microdata file, that loosely corresponds to a relation in a relational database, consists of rows which represents an individual record, and columns which represent the attributes. A sample of such a microdata file is shown in Table 4.1, with the records representing patients who have been tested for a genetic disorder. The first column of the table shows the record number and the following column ("*Name*")gives the surname of the patient. The

patient's date of birth ("*DOB*") and gender is also provided, along with the name of their family doctor. The following attributes relate to the patient's recent medical history, including their last hospital admission, most recent diagnosis, the number of past pregnancies and finally a flag indicating if the patient has been previously diagnosed with a genetic disorder.

	Name	*DOB*	*Gender*	*Family Doctor*	*Last Hospital*	*Current Diagnosis*	*Number Preg.*	*Genetic Disorder*
1	Brown	23/05/64	f	Chen	12/07/08	ABS234	> 1	Y
2	Brown	6/12/59	m	James	13/07/08	ABS234	0	Y
3	Black	16/03/45	f	Ross	13/07/08	BRC102	1	N
4	Brown	30/11/84	m	Smith	15/07/08	HAV529	0	N
5	Jones	28/02/25	m	Wang	20/07/08	HAV529	0	N
6	Smith	16/07/64	f	Smith	21/07/08	BRC102	0	Y
7	Black	10/11/72	m	Chen	-	ABS234	0	N

Table 4.1: Sample Genetic Database.

We now formalise the notation used in the application of PRAM on a microdata file such as that shown in Table 4.1 We denote a single categorical attribute in the original file with the term ξ, and let X signify the same attribute after the application of PRAM. We assume that ξ, and hence X, has a domain of K categories with a labeling 1,2,...,K. The probability, $p_{kl} = P(X = l \mid \xi = k)$, that a value, $\xi = k$, from the original file has been changed into another value, $X = l$, in the perturbed file, where $k, l = 1, 2, \ldots, K$ is termed the 'transition probability' [28, 51]. The sum of probabilities over all K categories in an attribute equals one, that is

$$\sum_{j=1}^{K} p_{ij} = 1.0$$

where $1 \leq i \leq K$. Hence, the *transition probability* matrix P for the attribute ξ is given by the $K \times K$ Markov matrix shown in Equation 4.1. Any entry p_{kl} give the probability that the value $\xi = k$ in the original file has been changed to $X = l$ in the perturbed microdata file. Note that the PRAM method is fully described by P for each attribute in the data set, with the

rows and columns of P representing an individual categorical value.

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1K} \\ p_{21} & p_{22} & \cdots & p_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ p_{K1} & p_{K2} & \cdots & p_{KK} \end{bmatrix}. \tag{4.2}$$

For each record r in the original microdata file we apply PRAM independently as follows. Let $\xi^{(r)}$ and $X^{(r)}$ denote the value of ξ and X respectively, of the r-th record in the corresponding data files. To apply PRAM, given the record r with value $\xi^{(r)} = k$ for a particular attribute, the value of $X^{(r)}$ is drawn from the probability distribution $p_{k1}, p_{k2}, \ldots, p_{kK}$, where $p_{kl} = P(X = l \mid \xi = k)$ is the probability that the original value k will become l in the perturbed file.

Example. *Consider the Sample Genetic Database in Table 4.1. We take the sixth attribute, 'Current Diagnosis', as our unperturbed attribute ξ, with possible values of 'ABS234', 'BRC102' and 'HAV529'. We can represent these categories numerically by assigning ABS234 = 1, BRC102 = 2 and HAV529 = 3. The Markov matrix P (shown below) indicates that the probability of a value remaining unchanged in the perturbed file is 80%.*

$$P = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

Note that the first row and column correspond to the category 'ABS234', while the second row and column correspond to 'BRC102', and finally the third row and column correspond to the category 'HAV529'. Now suppose that there are 300 records in the unperturbed file, where 100 of the records have the current diagnosis as 'ABS234', for another 100 records the current diagnosis is 'BRC102', and for the remaining 100 the diagnosis is 'HAV529'. After applying PRAM to this file using the matrix P, the expected value of each diagnosis is still 100. However, we would expect

that only 80 of the records with a current diagnosis of, for example, 'BRC102' in the original file would remain unchanged in the perturbed file.

Consider two categorical attributes denoted ξ_1 and ξ_2, with K_1 and K_2 categories respectively, and let X_1 and X_2 represent the same attributes in the perturbed file [54, 51]. Let $p_{(k_1,k_2),(l_1,l_2)}$ be the probability that the original value $\xi_1 = k_1$ is changed into value $X_1 = l_1$ while the original value $\xi_2 = k_2$ is changed into $X_2 = l_2$. That is,

$$p_{(k_1,k_2),(l_1,l_2)} =$$

$$= P(X_1 = l_1; \quad X_2 = l_2 \mid \xi_1 = k_1; \xi_2 = k_2)$$

for $k_1, l_1 = 1, \ldots, K_1$ and $k_2, l_2 = 1, \ldots, K_2$. By applying PRAM simultaneously to ξ_1 and ξ_2 we are able to compound the attributes into the attribute ξ, which has K_1K_2 categories [54]. For an arbitrary sth attribute, we define the transition probabilities matrix as $P^{(s)} = \{p^{(s)}_{k,l}\}$. We can compound our Markov matrices, $P^{(1)}$ and $P^{(2)}$ for the attributes ξ_1 and ξ_2 according to the following equation

$$P = \{p_{(k_1,k_2),(l_1,l_2)}\} = P^{(2)} \otimes P^{(1)}.$$

where $\otimes$ is the *Kronecker product* [51]. The Kronecker product (or direct product) $C = A \otimes B$ is an $(mp) \times (nq)$ matrix with elements

$$c_{\alpha\beta} = a_{ij} b_{kl}$$

where

$\alpha = p(i-1) + k$,
$\beta = q(j-1) + l$,
A is an $m \times n$ matrix, and
B is an $p \times q$ matrix

Thus if PRAM is applied independently to the attributes ξ_1 and ξ_2, we can equivalently apply PRAM defined by $P = P^{(2)} \otimes P^{(1)}$ to the compound

attribute. Clearly it is computationally more efficient to apply PRAM to each attribute separately rather than to the compound attribute. However, sometimes it is only possible to apply PRAM to the compound attribute in order to avoid inconsistencies in the database [28]. An example of such an inconsistency would be a perturbed database that contains a record describing a pregnant male individual. We discuss this problem in relation to genetic databases in the following section. To alleviate such risks and preserve the consistency of the database it is desirable to apply PRAM to more than one attribute simultaneously [54]. We note that in this case we are using the compound Markov matrix P, and that in general it is not possible to construct matrices $P^{(1)}$ and $P^{(2)}$.

"Let $T_\xi = (T_\xi(1), \ldots, T_\xi(K))^t$ be the K-vector of frequencies of the K categories of the attribute ξ in the original file. Similarly, T_X is the vector of frequencies in the perturbed data file" [51, p.42]. It is important that the user is able to obtain an unbiased estimate of original frequency vector T_ξ from the perturbed vector T_X. In order to achieve this it is necessary for the Markov matrix P to be invertible or non-singular. This inverse can be used to obtain unbiased estimates of the original file. An unbiased estimator of T_ξ can be obtained as

$$\hat{T_\xi} = (P^{-1})^t T_X.$$

We note that in this case we assumed that the Markov matrix itself will be released to the user together with the perturbed data file. If we want to avoid releasing the Markov matrix and if we want to simplify the analysis of the perturbed data file, we can impose a special selection criteria for P. On the application of invariant PRAM it is desirable to have the frequencies for the perturbed file be as close as possible to those of the original file. [54, 51]. To achieve this P is chosen in such a way as to satisfy

$$P^t T_\xi = T_\xi$$

where P^t is the the transpose of P. The K-vector of frequencies T_X is then then able to be used as an unbiased estimator for T_ξ, and we no longer need to multiply it by the inverse of P [51]. "However, when ξ has a large

number of categories with many containing only a few observations, it can be very difficult, or even impossible, to construct invariant PRAM so as to preserve the simultaneous distribution of all attributes in the perturbed file [28]" [51, p.42]. The security of PRAM is measured by the probability that a value l in the perturbed compound attribute X corresponds to the same value in the original attribute ξ [28]. This probability can be estimated by an expectation ratio $ER(l)$ as follows.

$$ER(l) = \frac{p_{ll}T_\xi(l)}{\sum_{k \neq l} p_{kl}T_\xi(k)}$$

We note that in order to estimate the expectation ratio a snooper needs the knowledge of the Markov matrix which is available to him in the case of general PRAM. Thus in the case of invariant PRAM when the Markov matrix is not known to the intruder, they will not be able to estimate the expectation ratio and the data file can be deemed more secure.

We now investigate how the above techniques could potentially be applied in the context of genetic databases.

4.4 Applying PRAM to Genetic Databases

There a several ways in which PRAM can be applied to help ensure the privacy of individuals in the database [51]. In the first instance, it can make the intruder's task of inferring the true values of confidential attributes for a particular record more difficult by perturbing these attributes. For example, if we perturb the attribute 'Genetic Disorder' and an intruder is able to determine the value of this confidential attribute for a particular record in the perturbed file, they will have a degree on uncertainty as to whether or not the inferred value is the true value from the original file [51].

The second way in which we can apply PRAM, is to not only perturb the confidential attributes, but also perturb non-confidential attributes. This will make it more difficult for the intruder to apply their supplementary knowledge about an individual to help them uniquely identify that indi-

vidual's record in the perturbed microdata file [51]. Because the nature of genetic disorders is not yet fully understood, we feel that protecting the privacy of an individual by perturbing the supplementary information, rather than genetic information, is more sensible [51]

As discussed in Section 4.3, the application of PRAM to individual attributes rather than compound attributes can cause inconsistencies when their is some form of functional dependency between attributes. This is very much the case in the context of genetic databases which have numerous integrity constraints. This problem is illustrated in the following example, which refers to our Sample Genetic Database from Table 4.1.

Example. *By applying PRAM independently to the attributes 'Gender', ξ_1, and the 'Current Diagnosis', ξ_2, we might get inconsistencies in the perturbed data file. We denote the categorical value of 'BRC102' to represent the diagnosis of breast cancer on a female patient, noting that males are also diagnosed with breast cancer, although in much lower numbers. Therefore, if we were to end up with records in the perturbed file with male 'Gender' and 'Current Diagnosis' as 'BRC102' this would provide the intruder with the certainty that at least one of values has been modified in the perturbed file. For simplicity, let us represent the two categories in the attribute 'Gender', male and female, as 1 and 2. Similarly, let the category 'BRC102' in the attribute 'Current Diagnosis' be denoted as 1. "To apply PRAM to these attributes independently and exclude the unwanted result, it is necessary to impose a probability of zero to $p_{21}^{(1)}$ and $p_{k1}^{(2)}$, where $k = 2, 3, \ldots, K_2$" [51, p.43]. An unwanted side-effect of applying PRAM in this restrictive way is that we have ensured that a male in the original microdata file is also male in the perturbed file, and the same applies to the diagnosis of breast cancer. However, by compounding the two variables we can overcome these restrictions imposed by applying PRAM to the attributes independently [51]. We just need to make sure that for the set of transition*

probabilities $\{p_{(k_1,k_2),(l_1,l_2)}\}$ *we have*

$$p_{(k_1,k_2),(l_1,l_2)} = 0$$

whenever $l_1 = l_2 = 1$.

One of the potential drawbacks of using compounded Markov matrices is that they can become much larger than the corresponding individual matrices, so it is important to only apply this strategy when absolutely necessary. PRAM can be applied to any combination of independent or compound variables for the m attributes $\xi_1, \ldots, \xi_m$, and this flexibility is clearly a strength of PRAM. However, the difficult task here is how best to choose the Markov matrices in order to adequately ensure the statistical quality of the data. By compounding any strongly related attributes, we have the best chance of preserving the underlying statistics of our original data in the perturbed file. Although, the best way to decide which attributes should be compounded is still an open problem [28, 51].

Once the compound attributes are identified the corresponding Markov matrices have to be constructed. The authors of [28] suggest that the set of categories for each attribute be partitioned into groups such that a category can only be replaced by a category in the same group. They further suggest that the categories within the same group are "in some sense similar". We note that the notion of similarity is not straightforward for categorical values. We suggest utilising results from the emerging topic of clustering categorical attributes, which we present in the following Section.

4.5 Clustering Categorical Attributes

There has been a lot of work done on clustering numerical data [60]. Only a few recent papers propose techniques for clustering categorical data [48, 50, 58]. We next describe some of this work.

In the paper by Gibson, Kleinberg and Raghavan [50], clustering is performed using techniques from non-linear dynamical systems motivated by

spectral graph partitioning. The algorithm presented in this paper is named *STIRR*. The data file is represented by a hypergraph whose vertices are values from the "actual domains" of the attributes. The *actual domain* of an attribute is the set of values of that attribute that actually appear in the data file. Hyperedges in the hypergraph represent tuples in the data file. We illustrate this using an example similar to the one in [50]. Consider the data file shown in Figure 4.1, containing three attributes, a, b and c. The actual domain of a is {a1,a2,a3}, the actual domain of b is {b1,b2,b3,b4} and for c it is {c1,c2,c3}. Each of the values in the actual domains represents a vertex in the hypergraph and each tuple represents a hyperedge, as shown in Figure 4.1.

a	b	c
a1	b1	c1
a1	b2	c1
a2	b1	c2
a2	b4	c2
a3	b3	c3
a3	b4	c3
a3	b2	c2

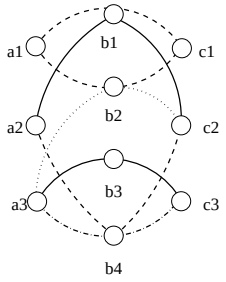

Figure 4.1: A Sample data file and the corresponding hypergraph.

Each vertex v has a weight w_v associated with it. Each weight is then propagated across all hyperedges containing the vertex. This weight propagation process is applied iteratively until a *basin* is reached, that is, until weights remain constant under repeated application or they go through a finite cycle. In the post-processing step clustering of vertices is performed on the basis of the weights, where large positive or negative weights represent dense regions with lots of hyperedges within and few between the regions. In other words, categories in each attribute are partitioned on the basis of their co-occurrence with the same categories of other attributes. For example, if two diagnoses, say BRC102 and ABS234, often appear with gender female and age group over 40, they may be considered similar.

An approach taken by Gantri, Gehrke and Ramakrishnan [48] describes clusters for categorical attributes as generalised clusters for numerical attributes. The algorithm that discovers the clusters, known as CACTUS, is

very efficient and doesn't require any post-processing. The clusters can be found across all attributes or across a specified subset of the attributes. This feature looks particularly interesting for application in constructing PRAM Markov matrices as it can be used to identify clusters for compound attributes. A drawback of this method in relation to PRAM matrices is that it discovers overlapping clusters and it appears to be non-trivial to extract the best non-overlapping clusters.

The so-called "*ROCK*" technique proposed by Guha, Rastogi and Shim [58] clusters tuples based on the number of *links* between them, where a link corresponds to a common neighbour of the tuples. Two tuples are neighbours if the similarity between them is greater than some given threshold for a given similarity function.

The value of identifying clusters for constructing Markov matrices lies in the similarities between categories that clusters imply. In the context of noise addition we are really interested in finding a similarity measure between categories in order to construct the Markov matrices. The probability of a category changing into another category should decrease as the similarity between these two categories increases.

One possible approach to solving the problem of deciding similarity between categorical values, is to break down a categorical attribute into sub-attributes. This idea originated in [7], and we illustrate it with the following example presented in [51].

> "Suppose that an attribute 'city' has the following categories: Sydney, Newcastle, Dubbo, Perth, Cobar, Alice Springs and Adelaide. It is very hard to assign an ordering to this attribute that would apply equally to various contexts. However, we can break down this attribute into the following sub-attributes: 'population', 'geographical longitude and latitude', 'pollution index' and 'is coastal'. Which of these attributes we use will depend on the application. For example, in genetic databases pollution index would probably be a relevant attribute." [51, p.44]

By breaking down the attributes into sub-attributes we are effectively

reducing the task of finding the similarities between the categories of the original attribute to the task of finding similarities between categories of sub-attributes. In the above example, each of the obtained sub-attributes can be naturally ordered which makes the task of finding similarities straightforward.

We propose combining the clustering approach with the attribute division approach by adding all the relevant sub-attributes to the data file schema. So, for example, if we originally had the attribute 'city' in the data file, we may now also have 'population' and 'pollution index'. It is quite possible that some of the identified sub-attributes would already exist in the data file and we add the others as required. We would next apply a clustering algorithm, for example STIRR [50], to the expanded data file. We note that this will have an effect only when for two categories the values of the same sub-attribute are identical. Only then will there be new connections between hyperedges. To ensure that the similar values of a sub-attribute also create connections we need to group categories in the sub-attributes and to replace individual categories with group representatives. Our future work will include experiments in this direction, but the specific application of clustering to sub-attributes is beyond the scope of this thesis.

4.6 Framework for a Genetic Database Security System

We propose a framework for the provision of security on genetic databases. This framework involves four steps as outlined below.

Step 1: Integrity rules
Step 2: Compound attributes and Markov matrices identification
Step 3: Category clustering
Step 4: Markov matrices.

In the first step we identify the integrity rules that exist in the data file. Such rules could include that people under 15 cannot be married, or in

full-time employment. This step requires a very thorough analysis since any unidentified rule carries a risk of introducing inconsistencies in the perturbed data file.

In the second step we partition the set of all attributes into compound attributes some of which may be individual, and we associate a Markov matrix to each compound attribute. We propose focusing on integrity rules and whenever two attributes are involved in the same integrity rule, they appear in the same compound attribute. For example, the two above mentioned integrity rules will imply that 'age', 'marital status' and 'employment' should be in the same compound attribute.

As for the clustering step, we consider the STIRR algorithm [50] particularly suitable for constructing Markov matrices. The reason for this lies in the weight propagation method that in some way incorporates the correlation between various attributes. Recall that the categories are considered similar if they appear with the same categories in other attributes. For example, consider the current diagnosis 'BRC102' which typically appears in patients over 40 and another diagnosis, say measles, which typically appears in children. It is unlikely that these two diagnoses will belong to the same cluster unless they strongly exhibit some other common properties. Our future work will include experimental evaluation of correlation preservation via the STIRR clustering method.

In the last step we construct Markov matrices based on identified clusters of categories. One approach is to choose the transition probabilities so that a perturbed category is always in the same cluster as the original category. Another approach is to allow perturbation that takes a category to another cluster but with a much smaller probability than when keeping it in the same cluster. Whenever possible the resulting Markov matrices should be non-singular.

4.7 Conclusion

In this chapter we proposed a framework for protecting confidentiality in genetic databases that is also applicable to other databases with predominantly categorical attributes. The framework is based on the PRAM noise addition technique proposed in [54]. However, the original technique does not specify how the Markov matrices should be constructed. The authors suggest that the compound attributes should be constructed in such a way that the inconsistencies are minimised and that the correlations between attributes are preserved [28]. Rather than suggesting how the categories should be partitioned into groups, the authors indicate that this is up to the data protector and that one possible way is to group similar categories together [54]. The main contribution of this chapter is in the construction of Markov matrices based on clusters of the categories in the compound attributes. We expect this approach to improve the statistical quality of the perturbed data file, while at the same time ensuring the security of sensitive attributes. This approach will be further examined in Chapter 6. In the next chapter we will investigate a new similarity measure specifically designed for use with data sets containing categorical attributes.

Chapter 5

Similarity Measure for Categorical Values

Learning is not attained by chance, it must be sought for with ardor and attended to with diligence.

–Abigail Adams

In the previous chapter we outlined a privacy protection framework designed for application to categorical attributes. An important step in this framework is the clustering of categorical values into partitions so that we can then apply a noise addition technique based on the partitioning. An important component of any partitioning technique is the notion of similarity between the attribute values being grouped. We seek to maximise the similarity between nodes in the same partition, while also minimising the similarity between nodes from different partitions. Hence, before we can perform a partitioning of the values within our data set, we must first define what makes two vertices in our graph similar.

In this chapter we outline the similarity measure that will be employed in our noise addition technique in Chapter 6. We first outline the motivation for the similarity measure, before formally defining the measure. We then present experimental results on several different data sets, which will highlight the effectiveness of our measure. We finish with a discussion of

several novel applications of the security measure which became apparent during the analysis of experimental results.

5.1 Motivating Example

The following example is designed to illustrate the relationships that exist in the microdata file, and how our technique attempts to capture these relationships. Table 5.1 shows a sample 'toy' microdata file for lecturers teaching courses within several different disciplines at a fictional university.

Lecturer	*Program*	*Course*	*Tutor*
Prof. T. Green	Medicine	Intr. Medicine	Dr D. Smith
Dr D. Blue	Medicine	Bioinformatics	Dr D. Smith
Dr M. Brown	Mathematics	Mathematics 1	R. Jones
Dr H. Pink	Mathematics	Mathematics 2	W. Wong
Prof. K. White	Computer Science	Mathematics 2	W. Wong
Dr J. Black	Computer Science	Mathematics 1	W. Wong
Dr J. Black	Computer Science	Bioinformatics	M. James

Table 5.1: Lecturer Microdata File - Sample

The following is a description of records belonging to lecturers as shown in the above table.

- Professor T. Green, who is an academic in the medicine faculty and teaches an introductory medical course tutored by Dr D. Smith.
- Associate Lecturer Dr D. Blue also works in the medicine faculty and teaches a course in bioinformatics also tutored by Dr D. Smith.
- Lecturer Dr M. Brown is a mathematician teaching a first year calculus course, his tutor in this course is R. Jones.
- Dr H. Pink lectures for the mathematics faculty and teaches a second year mathematics course tutored by W. Wong.
- Professor K. White is a computer scientist who teaches a first year calculus course to computer science students. This subject is tutored by W. Wong.

- Dr J. Black is a Senior Lecturer from the computer science faculty who teaches both bioinformatics and a second year mathematics course. He has a different tutor in each course, with W. Wong tutoring mathematics and M. James tutoring the bioinformatics course.

On reading the above scenarios we can clearly see a connection between Prof. Green and Dr Blue, who both teach in the medical faculty and use the same tutor for their courses. However, it may not be so obvious that there is a connection between Dr Brown and Prof. White as they have no attribute values in common; both teach different courses, in different programs and use different tutors. However, further investigation shows that they both share attribute values in common with Dr Pink and Dr Black, so there should be some notion of similarity between these two lecturers.

To better understand these connections between the lecturers we can represent the database shown in Table 5.1 as a graph. This is done by assigning values that appear in the database to vertices. An edge appears between two vertices when the corresponding two values appear together in a record. Figure 5.1 shows the graph generated from the lecturer micro-data file, with the vertex labels assigned to the values shown in Table 5.2. Each record forms a clique in the graph. The red circled subgraph in Figure 5.1 represent record 7 in the database, that is, Dr Black who teaches Bioinformatics in the computer science faculty using M. James as a tutor.

Note that we will be evaluating similarity only between vertices corresponding to the values of the same attribute in the data set. Thus similarity will be evaluated between vertices 1-6, 7-9, 10-13 and 14-17. In the sample database Dr Black (vertex 6) has direct similarity with every other lecturer except for Prof. Green. This direct similarity is indicated by one or more common neighbours of the corresponding vertices (or, equivalently, by a path of length two between the vertices). We note that there are no edges between vertices corresponding to the values of the same attribute.

Figure 5.1 shows that there are no common neighbours of vertex 3 (Dr Brown) and vertex 5 (Prof. White). This effectively means that the records pertaining to Prof. White and Dr Brown will have no values in common. So any method only looking at common values, that is, common neighbours,

Value	*Vertex Number*	*Attribute Number*
Prof. T. Green	1	1
Dr D. Blue	2	1
Dr M. Brown	3	1
Dr H. Pink	4	1
Prof. K. White	5	1
Dr J. Black	6	1
Medicine	7	2
Mathematics	8	2
Computer Science	9	2
Intr. Medicine	10	3
Bioinformatics	11	3
Mathematics 1	12	3
Mathematics 2	13	3
Dr D. Smith	14	4
R. Jones	15	4
W. Wong	16	4
M. James	17	4

Table 5.2: Vertex labeling for Table 5.1 in Figure 5.1

would not find these two values at all similar. However, looking at the data set it is clear that there is some transitive similarity between Dr Brown and Prof. White, as they both teach mathematics although at different levels. Nevertheless, in most circumstances these two courses would typically be considered similar in the university context. In database terms this similarity is captured by the fact that both Mathematics 1 and Mathematics 2 are always taught to the same student cohorts (computer science and mathematics). Moreover, subjects taught by Prof. White and Dr Brown are tutored by different support staff, R. Jones and W. Wong. However, these two staff are considered similar because they both teach Mathematics 1. Thus, although Prof. White and Dr Brown do not teach the same course, and are not assisted by the same tutor, they do indeed teach similar courses and are supported by tutors of similar expertise. Consequently, we may still wish to consider Prof. White and Dr Brown as similar. Our method aims to capture this kind of similarity by looking not just at common neighbours of two vertices, but also at common neighbours of their neighbours. We now outline how this type of similarity can be measured.

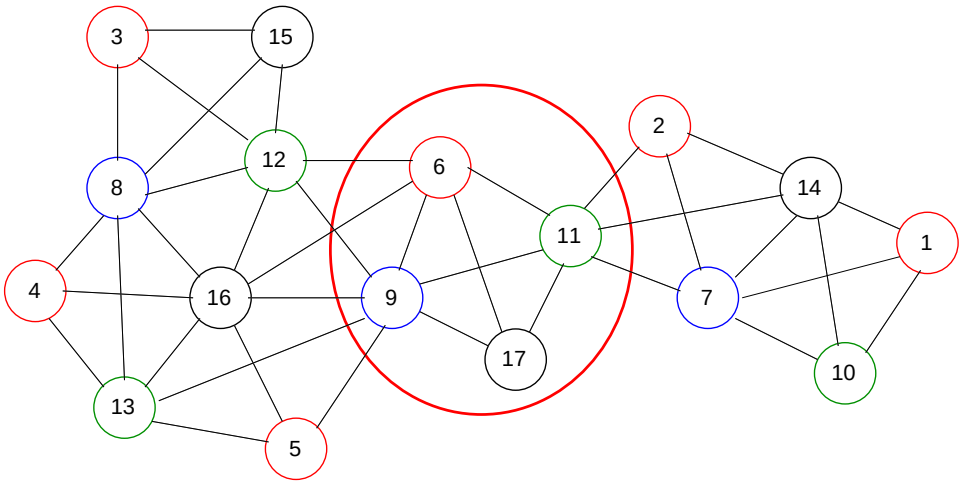

Figure 5.1: Motivating example database represented as a graph.

5.2 Evaluating Similarity

The first step in calculating similarity between attribute values is to create a graph of the original microdata file represented via an adjacency matrix. Recall that each value occurring in the file becomes a vertex in the graph, where there is an edge between vertices when two corresponding values appear together in a record. Hence, for each record in the microdata file, we form a clique in the resulting graph. Note that we have considered both the simple graph and multigraph created from the data set. In the simple graph we create an edge between two attribute values if they co-occur in any record. In the multigraph form we count the number of co-occurrences of the two values and consider this as the number of edges between the two corresponding vertices.

The first type of similarity we consider is based on the values co-occurring in records. For example, in our Motivating Example from Section 5.1, we would consider that Prof. Green and Dr Blue are similar since they both teach in the medicine faculty and use the same tutor for their courses. This type of similarity, which we term S' or *S-Prime* similarity, is measured by the number of common neighbours of these two vertices in the graph. Looking at Figure 5.1 we see that vertex 1 (*Prof. Green*) and vertex 2 (*Dr Blue*) are both adjacent to vertex 7 (*Medicine*) and vertex 14 (*Dr D. Smith*). We

consider this as a high similarity since these two values only have mostly common neighbours in the graph. In other words, most of the values that appear in the same record with one of these values, also appear with the other one.

The second type of similarity we consider is that of 'neighbours of neighbours'. We denote this type of similarity as S'' or *S-Secundum* similarity, and measure it by first considering the S-Prime similarity of 'neighbours'. An example of this type of similarity as discussed in Section 5.1 is between Dr Brown and Prof. White, who although do not share any attribute values in common, do share 'similar' values. For instance, the *Computer Science* and *Mathematics* programs would be considered similar via an S-Prime calculation. Similarly, the *Mathematics 1* course is similar to the *Mathematics 2* course, and tutor *R. Jones* is similar to *W. Wong*. This means that all of the values that Dr Brown and Prof. White appear with in the data set are considered similar via S-Prime similarity, and hence these two values would have a high S-Secundum similarity.

The Total Similarity S for two attribute values is taken to be composed of both the S-Prime and S-Secundum similarity for the values. We now provide a formal definition of our similarity measure S.

5.2.1 Similarity Measure - S_{ij}

We define a simple graph $G = (V, E)$ on n vertices and m edges, where $v \in V$ represents an attribute value in the data set. An edge $\{i, j\} \in E$ exists between two vertices $i, j \in V$ when the values i and j both appear together in one or more records in the data set. The adjacency matrix, A, for graph G will contain a 1 in position a_{ij} if an edge $\{ij\}$ appears between the vertices i and j, and 0 otherwise.

We also consider a multigraph representation of the data set. We define a multigraph $H = (V, E)$ on n vertices and m edges, where $v \in V$ represents an attribute value in the data set. An edge $\{i, j\} \in E$ exists between two vertices $i, j \in V$ for each record that contains both the values i and j. We do not allow self-loops in this graph. In the adjacency matrix A for graph

H, a_{ij} is the number of edges appearing between the vertices i and j in H.

The S'_{ij} or *S-Prime* similarity between two attribute values is given by the following formula

$$S'_{ij} = \frac{\sum_{k=1}^{n} \sqrt{a_{ik} \times a_{kj}}}{\sqrt{d(i) \times d(j)}} \tag{5.2}$$

where the sum is over all vertices in the graph G (or H), a_{lm} is the adjacency matrix entry for vertices l and m ($1 \leq l, m \leq n$) and $d(l)$ is the degree of vertex l. Note that S'_{ij} has a maximum value of 1 when the two vertices have all their ($d(i) = d(j)$) neighbours in common, and a minimum value when two vertices have no neighbours in common ($S'_{ij} = 0$). S-Prime values are only calculated within an attribute and not across attributes.

The S''_{ij} or S-Secundum similarity attempts to capture a notion of transitive similarity for attribute values that are not necessarily directly connected to a common neighbour but are connected to similar values, that is, values which have a S'_{ij} value greater than the user defined Threshold T. The algorithm for calculating S''_{ij} is shown in Algorithm 5.1.

The actual algorithm implemented in the program, as shown in Appendix A in Section A.1, is in effect significantly more efficient than the algorithm shown in Algorithm 5.1.

We calculate the total similarity S_{ij} as a weighted sum of the S-Prime (S'_{ij}) and S-Secundum (S''_{ij}) similarities.

$$S_{ij} = c_1 \times S'_{ij} + c_2 \times S''_{ij} \tag{5.3}$$

where $c_1 + c_2 = 1$. Typical values might be $c_1 = 0.65$ and $c_2 = 0.35$. In the next section we experiment with different values for c_1 and c_2.

Input: Graph G, Threshold T
Output: S'' values for G

initialise S'' matrix to 0;
for *each attribute* $x \in G$ **do**
  get the list of attribute values val_x;
  `/* Loop over all pairs of values for the attribute` x `*/`
  for *each value* $i \in val_x$ **do**
    for *each value* $j \in val_x$ **do**
      initialise *mergedGraph* **to** G;
      `/* Loop over all attributes in` G`, excluding` x `*/`
      for *each attribute* $y \in G \setminus x$ **do**
        get the list of attribute values val_y;
        `/* Loop over all pairs of values in` y `*/`
        for *each value* $c \in val_y$ **do**
          for *each value* $d \in val_y$ **do**
            if *(there are egdes* $(\{c,i\}$ *and* $\{d,j\}) \vee (\{c,j\}$ *and* $\{d,i\})$ *in* $G) \wedge$ *(*c *and* d *not already in the same vertex in mergedGraph)* **then**
              if $S'_{cd} >$ *Threshold* T **then**
                merge vertex c and d in *mergedGraph*;
                `/* Note: if one vertex has already been merged with another, merge all together */`
              end
            end
          end
        end
      end
      $S''_{ij} = S'_{ij}$ calculated on *mergedGraph*
    end
  end
end
return S'' matrix;

Algorithm 5.1: Calculating S'' values for graph G

5.3 Experiments - Similarity

In this section we present the results of experiments conducted on several data sets, both synthetic and real life, to observe the effectiveness of our similarity measure.

5.3.1 Data Sets

A variety of data sets have been selected to best demonstrate various qualities and characteristics of our similarity measure S_{ij}. The main properties of these data sets are given in Table 5.3, and more detailed descriptions of each are provided below. Note that Table 5.3 shows information for the data sets after we perform some preprocessing, with the number of records, number of attributes, and number of distinct values in the data set noted.

Data Set	*No. Records*	*No. Attr.*	*No. Values*	*Notes*
Motivating Example	7	4	17	Example from Section 5.1
Mushroom	5,644	22	99	Fully categorical
Contr. Meth. Choice	1,473	10	74	Mix of num. and cat.
Wisc. Breast Cancer	683	10	91	9 num. and 1 class attr.
ACS PUMS	20,000	14	664	Mix of num. and cat.

Table 5.3: Data Set Summary.

- **Motivating Example.** This is the same data set presented in Section 5.1, and is used to illustrate advantages of our technique over other similarity measures.
- **Mushroom.** This data set was selected as it contains only categorical values, and has been previously studied in the context of classification [58]. It is obtained from the UCI Machine Learning Repository [5]. The original data set contains 8124 instances on 23 attributes (including the class attribute). The following preprocessing of the data was undertaken. Attribute 17 (veil-type) was removed as it only had one value in the data set and including it would falsify similarity across all values, this is due to the fact that records with missing values were not considered. This reduces the size of the data set from 8124 to 5644

records. It also reduces the number of attribute values, from 118 to 99. Although it is not ideal to have this reduction in the number of values considered, it is beyond the scope of this work to consider how best to deal with missing values in data. Table A.4 in Appendix A shows the full list of attribute values and corresponding vertex labels for this graph.

- **Contraceptive Method Choice.** This data set is also sourced from the UCI Machine Learning Repository [5, 79], and is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The subjects of the survey were married women from different demographic and socio-economic backgrounds, who were not currently aware of being pregnant. The data set contains 1473 records over 10 attributes, five of which are numerical, or have a natural ordering, and the remainder are categorical or binary. The attributes are, *Wife's Age*, *Wife's Education*, *Husband's Education*, *Number of Children Ever Born*, *Wife's Religion*, *Wife's Now Working*, *Husband's Occupation*, *Standard-of-Living Index*, *Media Exposure* and *Contraceptive Method Used*. For more details about the attribute types and attribute values please see Section A.4 in Appendix A.

- **Wisconsin Breast Cancer.** This data set is sourced from the UCI Machine Learning Repository [5] and was originally obtained from the University of Wisconsin Hospitals, Madison, from Dr. William H. Wolberg [84]. The data pertains to measurements performed on tissue samples taken from suspected breast cancer patients. It consists of 9 numerical attributes and one class attribute, namely *Clump Thickness*, *Uniformity of Cell Size*, *Uniformity of Cell Shape*, *Marginal Adhesion*, *Single Epithelial*, *Bare Nuclei Cell Size*, *Bland Chromatin*, *Normal Nucleoli*, *Mitoses* and *Class*. It classifies records as either benign or malignant based on the values of the other 10 attributes whose values have all been normalised to discrete numbers in the range [1,10]. There were 16 records in the original data set with one or more missing values, which have been removed, leaving 683 records and 91 distinct values in the dataset.

- **ACS PUMS.** The American Community Survey (ACS) is an ongoing survey conducted annually by the United States Census Bureau

and was designed to provide a snapshot of the community. We took a random sample of 20,000 records from the 2006 Housing Records Public Use Microdata Sample (PUMS)[1] for the whole of the US. The sub-sample was chosen on only 14 attributes of the available 239, and any records with missing values on these attributes was not considered. The attributes selected were *State Code* (ST), *Age* (AGEP), *Citizenship status* (CIT), *Class of worker* (COW), *Means of transportation to work* (JWTR), *Marital status* (MAR), *Educational attainment*(SCHL), *Sex* (SEX), *Wages or salary income past 12 months* (WAGP), *Usual hours worked per week past 12 months* (WKHP), *Recoded detailed ancestry* (ANC1P), *Total person's income* (PINCP), *Recoded detailed race code* (RACE1P) and *World area of birth* (WAOB). For more details on how the data set was extracted please see Section A.6 in Appendix A.

5.3.2 Parameter Selection

There is a certain amount of flexibility in the calculation of the similarity measure S which must be considered when running experiments on the selected data sets. First, there is a choice for the value of the S-Secundum Threshold T, which is in the range [0,1]. One observation on the selection of this threshold, is that for smaller/sparser graphs the threshold generally needs to be set at a lower value than it does for larger/denser graphs.

The second parameter that needs to be selected is the weighting values c_1 and c_2 in Equation 5.2. The restriction for these values is that $c_1 + c_2 = 1$, and a typical value choice for these parameters would be $c_1 = 0.6$ and $c_2 = 0.4$. This gives a slightly higher weighting to S-Prime than to S-Secundum.

The final parameter to be selected relates to the graph being generated from the data set. We have the choice of making this graph a simple graph or a multigraph, that is, a graph with multiple edges.

When showing results for the similarity measure we will generally present

[1] http://factfinder.census.gov/home/en/acs_pums_2006.html

a range of parameters for comparison. However, in general we would not want to set the values of the parameters neither too high, nor too low. For instance, setting the value of c_1 to a value of 1.0 will have mean that the total similarity be equivalent to S-Prime, since none of the S-Secundum has been taken into consideration. Similarly, if we set the S-Secundum Threshold T to a very low value, we will be considering values with only a very small number of neighbours in common to contribute to the S-Secundum similarity.

5.3.3 Results

We now present the results of experiments conducted on the data sets outlined in Section 5.3.1. The results presented are only a subset of the overall experiments conducted in order to test our similarity measure. We present a range of results to illustrate the effectiveness of our measure.

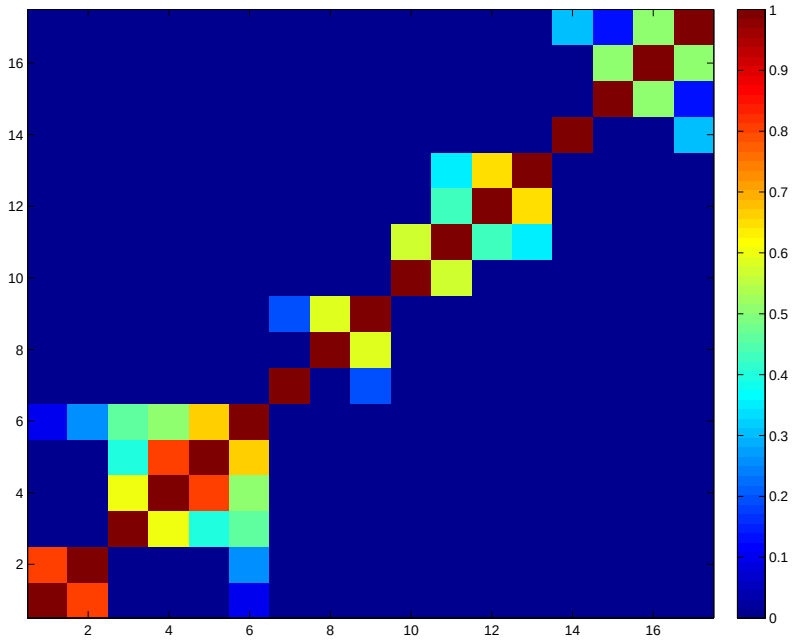

Figure 5.2: Similarity color map for Motivating Example.

One way in which we present the similarity values is via a colour map,

as shown in Figure 5.2. The colour map assigns different colours to different values as per the colour bar on the right hand side of the diagram. Dark red maps to 1.0 and and dark blue to 0.0. This figure shows S_{ij} values for the Motivating Example graph on all 17 values over the 4 attributes. The parameters are as follows, the S-secundum Threshold T has a value of 0.4, while $c_1 = 0.6$ and $c_2 = 0.4$ are the percentage weightings for Equation 5.2. Value $S_{1,1}$ is in the bottom left hand corner of Figure 5.2, and value $S_{17,17}$ is in the right hand top corner. If you look at the diagonal between these two values, you will see that all values along the diagonal are 1.0, since each value has maximum similarity with itself. Areas outside of an attribute are dark blue since we do not consider the similarity between values from different attributes. Note that the diagram is symmetric since the similarity of S_{ij} is equivalent to S_{ji} in our measure.

As we discussed at the beginning of Section 5.3 when first proposing our similarity measure, in the Motivating Example vertex 3 (*Dr Brown*) and vertex 5 (*Prof. White*) are somewhat similar in our graph (Figure 5.1) via an S-Secundum similarity, even though their S-Prime similarity is 0. This is reflected in Figure 5.2 by the pale blue colour for entry $S_{3,5}$, which is a value of 0.4. Note that this value would change for a different choice of c_1 and c_2.

Motivating Example

Here we present the similarity values for the Motivating example presented in Section 5.1. Figure 5.3 shows the similarity value comparison for different S'' Threshold T values across all attributes in the data set. From this diagram we can see that when T is very low, for instance 0.1, many of the values within attributes appear very similar to on another, indicated by dark red in the diagram. As T progresses from 0.1 to 0.6 red gradually gives way to orange and eventually green and blue and we can see that for such a small data set there is little difference between the S'' and S' similarities after T is set above 0.7

Examining the similarity values we can compare the S'_{ij} and S''_{ij} values to the scenarios discussed in Section 5.1. Table 5.4 give the S'_{ij} similarity

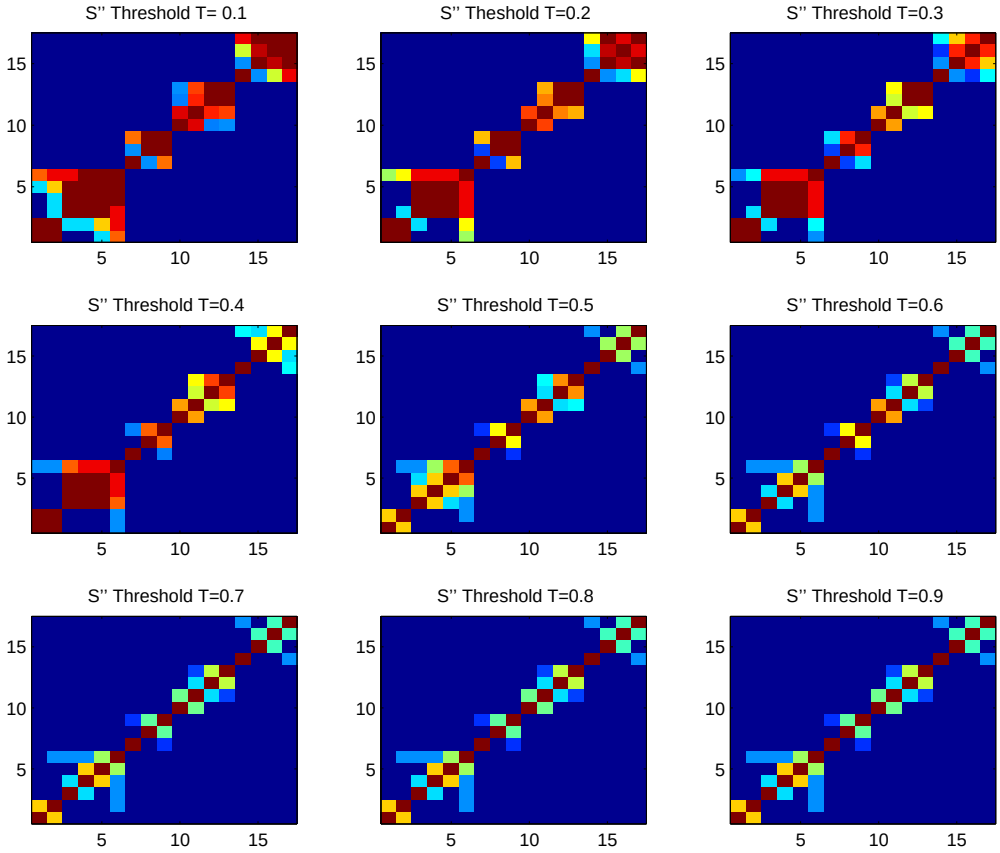

Figure 5.3: S'' Threshold T comparison for motivating example.

values for the first attribute in our motivating example, that is, *Lecturer*. The vertex numbers in the table correspond to the attribute values as follows, Vertex 1 (Prof. Green), Vertex 2 (Dr Blue), Vertex 3 (Dr Brown), Vertex 4 (Dr Pink), Vertex 5 (Prof. White) and Vertex 6 (Dr Black). Table 5.4 shows that Vertex 5 (Prof. White) and Vertex 3 (Dr Brown) have no S-Prime (S'_{ij}) similarity since they have no values in common in the data set. However, Table 5.5 shows that when the S-Secundum (S''_{ij}) Threshold (T) is equal to 0.4, these two values have a S''_{ij} similarity of 1.0. This supports the notion that although these two lecturers do not have any direct similarity in the data set, they do have a transitive similarity which should be considered in any subsequent clustering of these values. Now by choosing the values for c_1

S'_{ij}	1	2	3	4	5	6
1	1.000	0.667	0.000	0.000	0.000	0.000
2	0.667	1.000	0.000	0.000	0.000	0.258
3	0.000	0.000	1.000	0.333	0.000	0.258
4	0.000	0.000	0.333	1.000	0.667	0.258
5	0.000	0.000	**0.000**	0.667	1.000	0.516
6	0.000	0.258	0.258	0.258	0.515	1.000

Table 5.4: S'_{ij} values for *Lecturer* attribute in Motivating Example.

and c_2 we can give the desired weight to this indirect similarity represented by S''_{ij}. In Table 5.6 we can see the situation for $c_1 = 0.6$ and $c_2 = 0.4$. Note that the similarity results for the remaining attributes in the Motivating Example are provided in Section A.2 of Appendix A.

S''_{ij}	1	2	3	4	5	6
1	1.000	1.000	0.000	0.000	0.000	0.258
2	1.000	1.000	0.000	0.000	0.000	0.258
3	0.000	0.000	1.000	1.000	1.000	0.775
4	0.000	0.000	1.000	1.000	1.000	0.882
5	0.000	0.000	**1.000**	1.000	1.000	0.882
6	0.258	0.258	0.775	0.882	0.882	1.000

Table 5.5: S''_{ij} values for *Lecturer* attribute in Motivating Example with T=0.4.

S_{ij}	1	2	3	4	5	6
1	1.000	0.800	0.000	0.000	0.000	0.103
2	0.800	1.000	0.000	0.000	0.000	0.258
3	0.000	0.000	1.000	0.600	0.400	0.465
4	0.000	0.000	0.600	1.000	0.800	0.508
5	0.000	0.000	**0.400**	0.800	1.000	0.663
6	0.103	0.258	0.465	0.508	0.663	1.000

Table 5.6: S_{ij} values for *Lecturer* attribute in Motivating Example with $T = 0.4$, $c_1 = 0.6$ and $c_2 = 0.4$.

In general, for the Motivating Example data set we show the effect of the choice of c_1 and c_2 on the total similarity. The colour map in the top left corner of Figure 5.4 shows the S-Prime values, and the position directly to its right shows the S-Secundum values when $T = 0.4$. The remaining

frames provide us with an example of how a change in the relative values of c_1 and c_2 impacts on the resulting total similarity. For instance the image at the bottom right hand corner of Figure 5.4 is barely distinguishable from the original S-Prime image in the top left corner. This is due to the weight of c_1 being set to 0.9, meaning that 90% of the similarity is drawn from the S-Prime similarity.

Mushroom Data Set

This data set is fully categorical and represents various properties of mushrooms and further classifies the mushrooms as either edible or poisonous. The attributes and their associated ordering is shown in Table 5.7. This data set has many attributes with a small number of values per attribute. The largest number of values for any one attribute is for the attribute *Gill Colour* which has 9 values. For a full list of attribute values and their associated vertex numbering, please refer to Table A.4 in Appendix A.

As Figure 5.5 shows, since this data set is categorical, there is a lot of variation both in the similarity between values within an attribute, and across attributes. For example, attribute 12 - *Stalk Root* - shows that the attribute values all have a reasonably high degree of similarity with one another. However, attribute 19 - *Ring Type* - exhibits more numeric quality with some values being very similar to one other, while very dissimilar to others.

Attribute Number	*Attribute Name*	*Attribute Number*	*Attribute Name*
1	Class	12	Stalk Root
2	Cap Shape	13	Stalk Surface Above Root
3	Cap Surface	14	Stalk Surface Below Ring
4	Cap colour	15	Stalk Colour Above Ring
5	Bruises	16	Stalk Colour Below Ring
6	Odour	17	Veil Colour
7	Gill Attachment	18	Ring Number
8	Gill Spacing	19	Ring Type
9	Gill Size	20	Spore Print Colour
10	Gill Colour	21	Population
11	Stalk Shape	22	Habitat

Table 5.7: Attribute names and ordering for Mushroom data set.

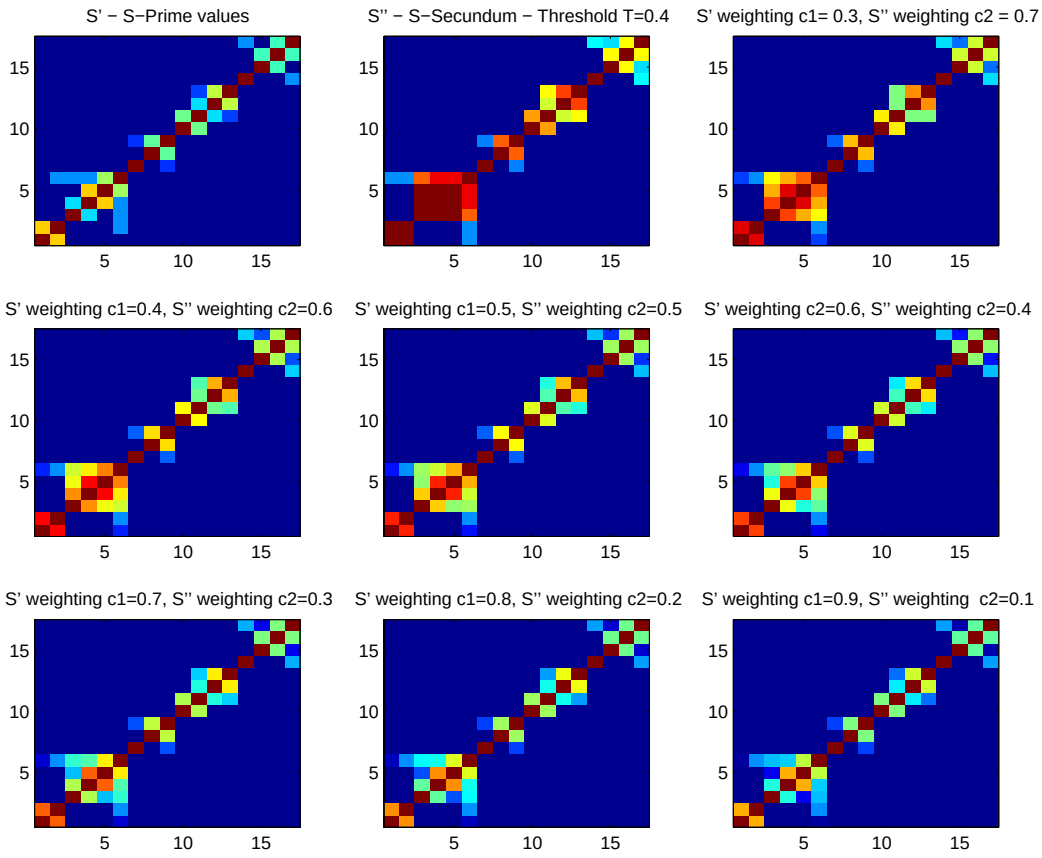

Figure 5.4: $S^{'}$ and $S^{''}$ weighting comparison for Motivating Example.

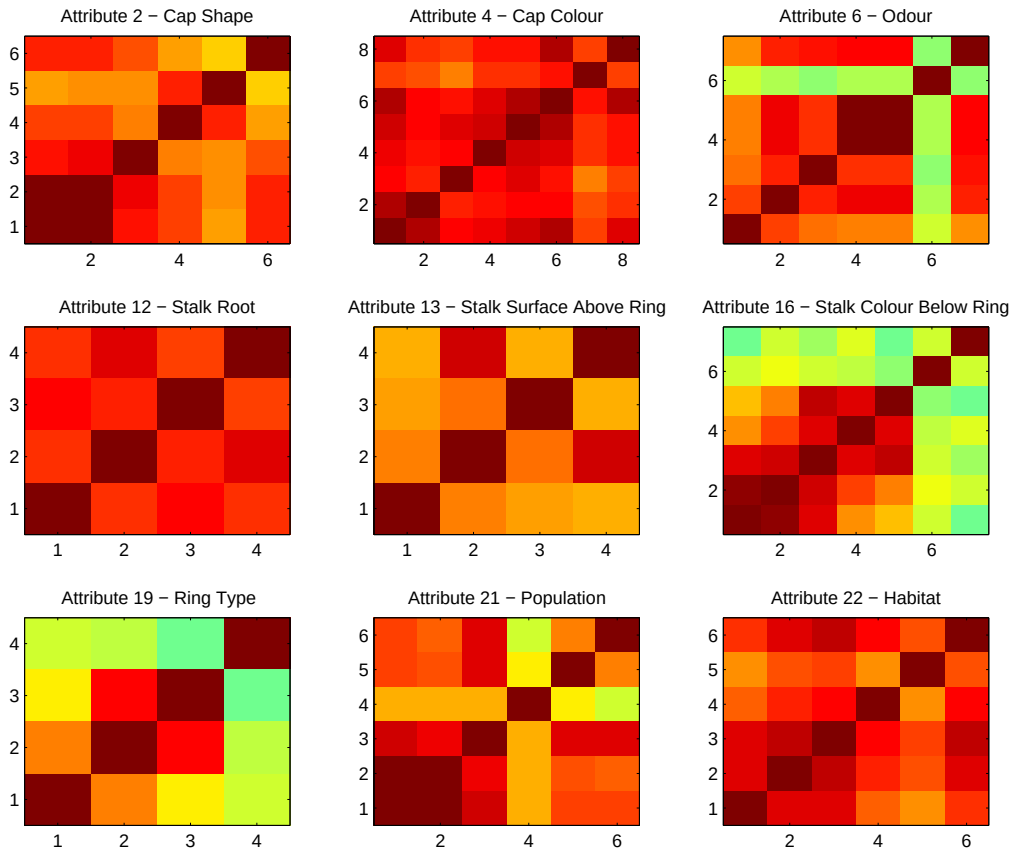

Figure 5.5: A close look at S_{ij} values for selected attributes in Mushroom. ($T = 0.6$, $c_1 = 0.6$, $c_2 = 0.4$).

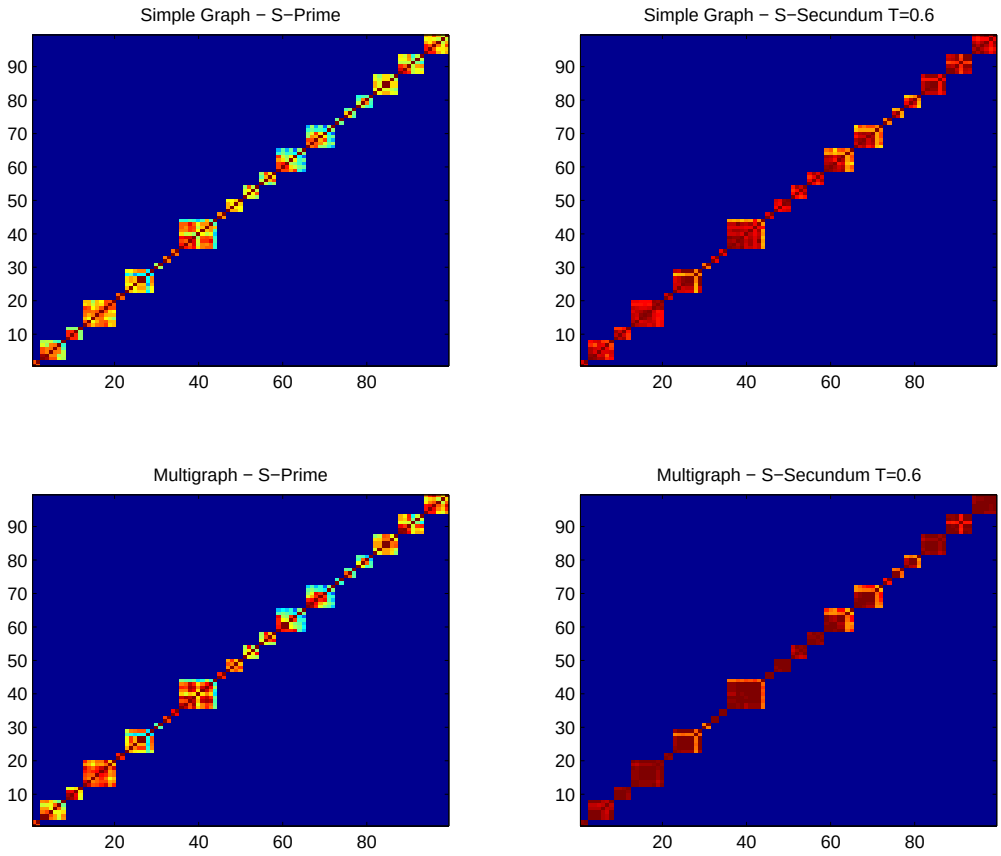

Figure 5.6: Comparing the simple and multigraph representation of the Mushroom data set, all attributes.

As mentioned previously in Section 5.3.2, one of the parameters that can be used in measuring similarity is the choice of either a simple or multigraph representation of the data set. Figure 5.6 shows a comparison between these two representations with all attribute values shown. The left top corner of the figure shows the S-Prime values for the simple graph version, while the bottom left corner gives the S-Prime values for the multigraph. Similarly, a comparison is also shown between the simple and mulitgraph S-Secundum similarities with a threshold of $T = 0.6$. In general the S-Prime, S-Secundum and resultant total similarity are higher for the multigraph representation than for the simple graph version. In the multigraph version there can also be a more pronounced difference in similarity values when there are outlier values, that is, attribute values that do not co-appear with many other values in the data set.

By examining Figure 5.7 we can better understand the differences between a Simple and Mulitgraph representation of the data. Figure 5.7 shows a close up view of the S-Secundum similarities for attribute 10 (*Gill Colour*) and attribute 16 (*Stalk Colour Below Ring*) with a Threshold $T = 0.6$. The Simple Graph similarities are shown to the left of the figure, and the Multigraph similarities to the right. For attribute 10, there is one attribute value that is clearly most dissimilar to the other values, this is the value representing *Gill Colour* yellow (position 9 in Figure 5.7). Although this attribute value has a low support in the original data set, with only 22 out 5644 records containing the value, it is not the only value for this attribute with a low number of occurrences. The value *Gill Colour* green (position 5 in Figure 5.7) also has a low number of occurrences, with only 24 out of 5644 records containing this value. However, this value is shown to be very similar to the other attribute values, excluding yellow, in Figure 5.7. A similar observation can be made with attribute 16, whereby the value which is least similar to all other values, that is, *Stalk Colour Below Ring* cinnamon (position 6 in Figure 5.7) has support of 36 out of 5644 records, while value *Stalk Colour Below Ring* yellow (position 7) has support of only 8 out 5644 records yet has higher similarity to several of the other attribute values.

These results would seem to indicate that although values occurring in a relatively small number of records are more likely to have low similarity

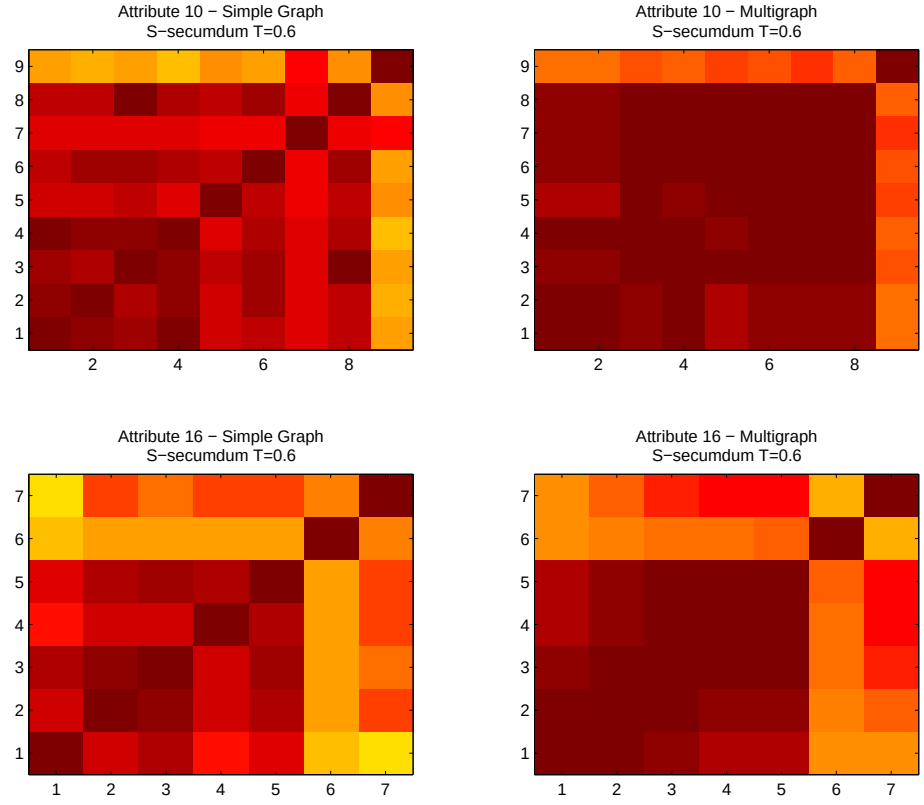

Figure 5.7: Comparing the simple and multigraph representation of the Mushroom data set (attributes 10 and 16 S-Secundum $T = 0.6$).

to other values, there is not a direct correlation between how many times a value appears in the data set to its similarity value.

Contraceptive Method Choice Data Set

The Contraceptive Method Choice data set is of particular interest since it exhibits relatively high similarity amongst many of the attribute values, even when the value of the S-Secundum threshold T is set very high. This indicates that there is a high degree of connectedness amongst the values in the data set. Or in other words, most pairs of values share a high number of neighbours in the graph. For example, if we look at Figure 5.8, we can see

that for values of T even as high as 0.9 there are very few values that are not very similar to one another. Only when $T = 1.0$ do we see any S-Secundum values fall below 0.5, that is, into the green colour range. As a result, the choice of the Secundum Threshold will have an important impact on any resulting partitioning of the values since if all S-Secundum values are equally high then the only effect they will have on the total similarity is to make it higher for all attribute values. If there is little variation in S-Secundum values across all attributes, then we will not gain any benefit by adding the S-Secundum component to the total similarity. Note that for the results shown we have used the Simple Graph version of the data set.

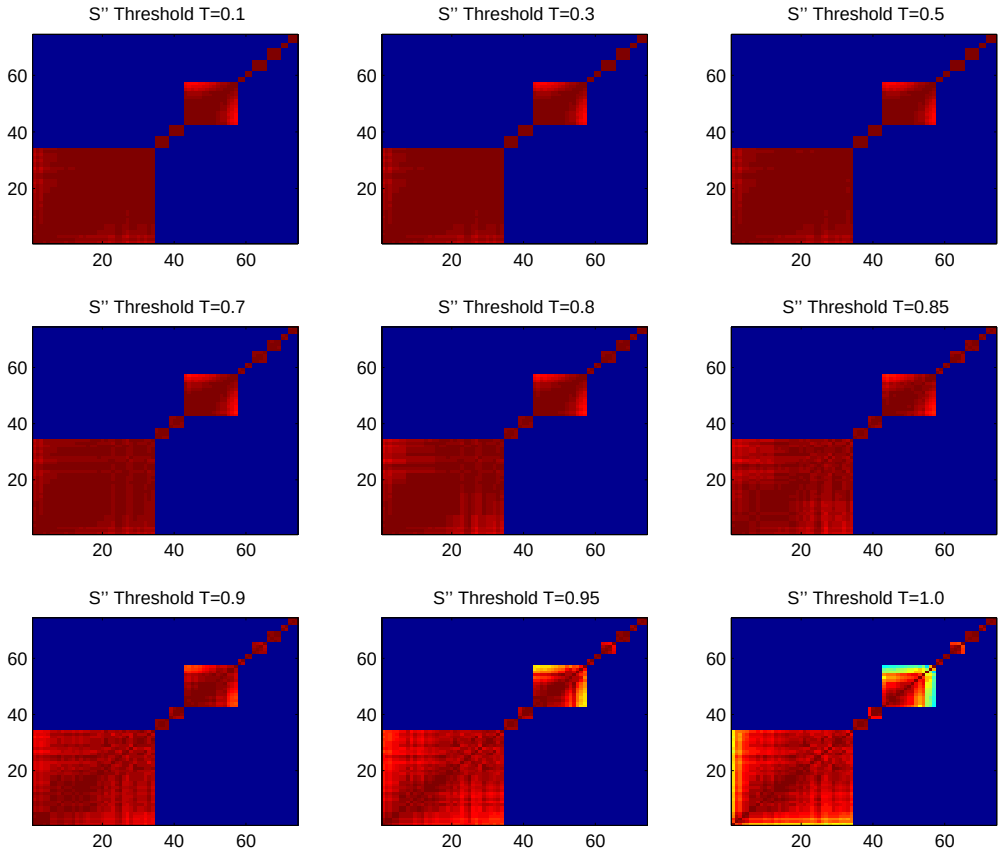

Figure 5.8: S'' Threshold T comparison for Contraceptive Method Choice data set.

One way in which we can test the effectiveness of our similarity measure is to observe how it behaves on numerical attributes. In Figure 5.9 there are two numerical attributes that are easily observed, the first being attribute 1 (*Wife's age*) in the very bottom left corner of the images, and the second being attribute 4 (*Number of children ever born*) which is the second largest square just above centre in the images. Note that for both attributes, the vertex numbering has been applied according to the numerical ordering of the values. Looking first to attribute 1 (*Wife's age*) we see that although there is a relatively high level of similarity between all values, the values that are further away from each other numerically are less similar than those that are close numerically. This effect is seen more obviously for attribute 4 (*Number of children ever born*) where in the top left and bottom right hand corners of the attribute's similarity matrix we see a marked change in the colour from red to yellow (and even blue for the S-Prime image).

We also note that for the S-Prime similarity (top left hand corner image in Figure 5.9) the last few values for attribute 4 appear to have little similarity with the other values in the attribute. These values correspond to the *Number of children ever born* being above 10, with the highest value being 16. These values appear in very few records in the original data set, indeed the value 16 only appears in only one record. Therefore these values have a low S-Prime similarity as they will have a relatively low number of neighbours in the graph. However, when the S-Secundum similarity is calculated there is shown to be a transitive similarity between these values, and we achieve a more numerical-like similarity pattern when we combine the S-Prime and S-Secundum similarity, as for example when $c_1 = 0.5$ and $c_2 = 0.5$, shown as the very middle image in Figure 5.9.

Wisconsin Breast Cancer

The Wisconsin Breast Cancer data set is a numerical data set, with the attribute ordering and attribute domains provided in Table 5.8. When we observe the total similarity values based on the parameters $T = 0.5, c_1 = 0.6$ and $c_2 = 0.4$ (Figure 5.10), we see that the attributes do not appear to exhibit a strong numerical ordering as shown for attributes in the Contra-

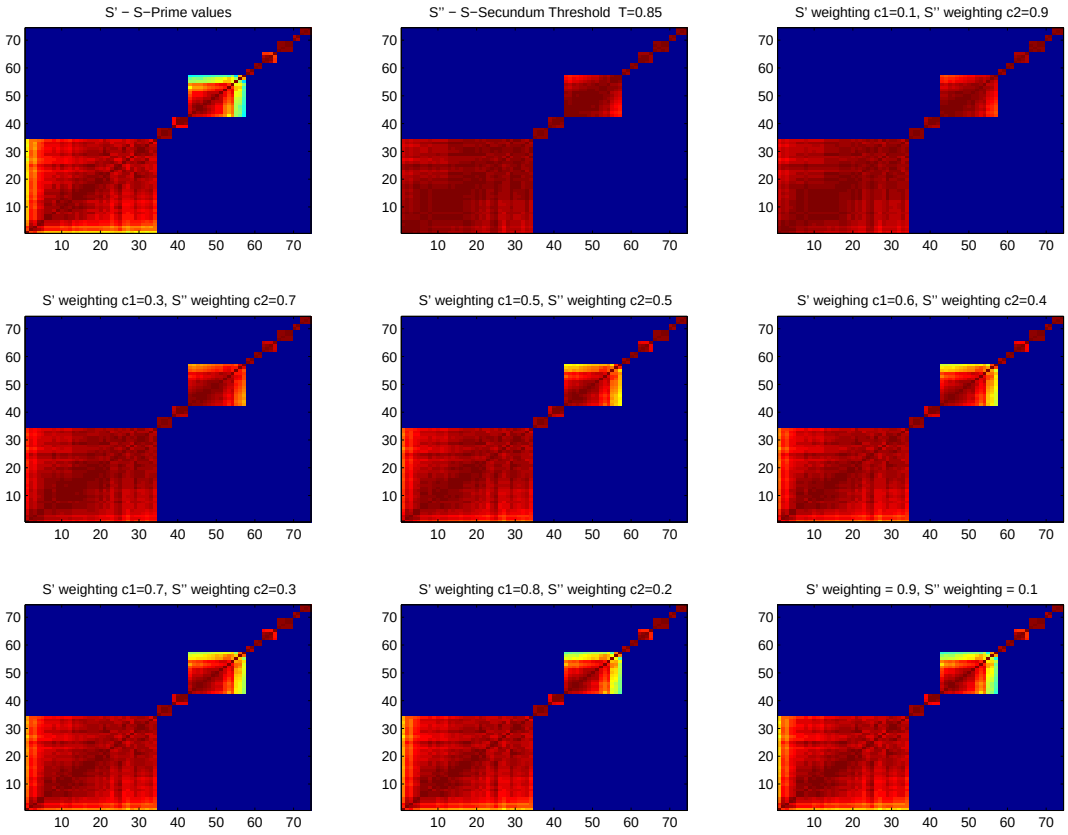

Figure 5.9: $S^{'}$ and $S^{''}$ weighting comparison for Contraceptive Method Choice data set.

ceptive Method Choice data set. For several of the attributes, (e.g attribute 6 and 9) there are values that appear towards the middle of the attribute in numerical ordering, yet they are not very similar to other values around them. For other attributes, (e.g attribute 8), all of the values appear very similar, with there being only a small colour change distinguishing any of the values.

Attribute Number	*Attribute Name*	*Domain*
1	Clump Thickness	{1,2,3,4,5,6,7,8,9,10}
2	Uniformity of Cell Size	{1,2,3,4,5,6,7,8,9,10}
3	Uniformity of Cell Shape	{1,2,3,4,5,6,7,8,9,10}
4	Marginal Adhesion	{1,2,3,4,5,6,7,8,9,10}
5	Single Epithelial Cell Size	{1,2,3,4,5,6,7,8,9,10}
6	Bare Nuclei	{1,2,3,4,5,6,7,8,9,10}
7	Bland Chromatin	{1,2,3,4,5,6,7,8,9,10}
8	Normal Nucleoli	{1,2,3,4,5,6,7,8,9,10}
9	Mitoses	{1,2,3,4,5,6,7,8,10}
10	Class	{benign,malignant}

Table 5.8: Attribute names and ordering for Wisconsin Breast Cancer data set.

The implications of these observations is that traditional noise addition techniques for application to numerical attributes may not be the best choice for some numerical attributes, such as those in the WBC data set. For this reason, in Chapter 6, we will further examine this data set by applying our noise addition technique for categorical attributes to it. We will treat the discrete numerical values as categories rather than numbers, and apply noise to the data set as if it were a fully categorical data set.

ACS PUMS

A sample subset of the The American Community Survey (ACS), this data set is good mixture of categorical and numerical attributes of varying sizes. A sample of total similarity results for parameters $T = 0.75, c_1 = 0.6$ and $c_2 = 0.4$ are shown in Figure 5.11.

We remind the reader of the attribute names corresponding to the attribute codes shown in Figure 5.11. Ordered from top left to bottom right, they are *Wages or salary income past 12 months* (WAGP), *Total person's*

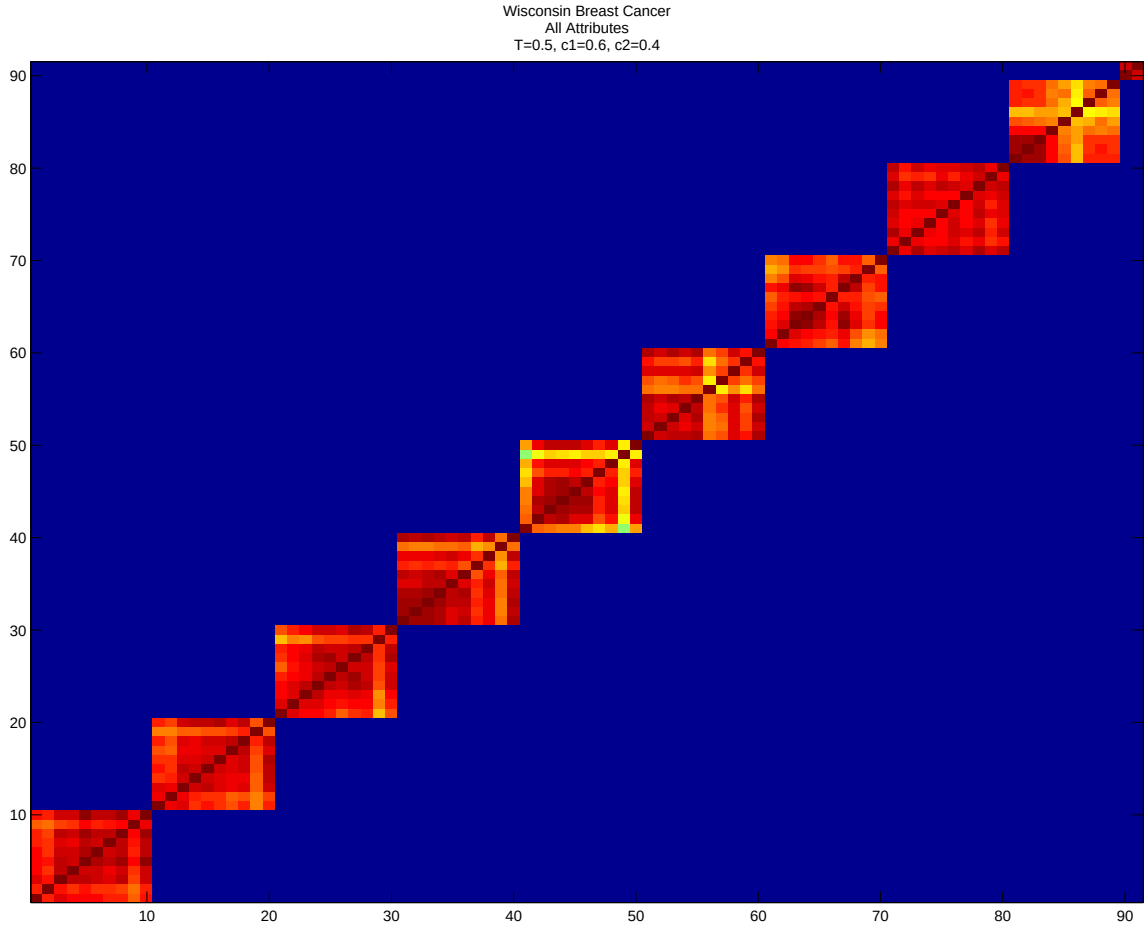

Figure 5.10: Wisconsin Breast Cancer data set, total similarity for all attributes ($T = 0.50, c_1 = 0.6, c_2 = 0.4$).

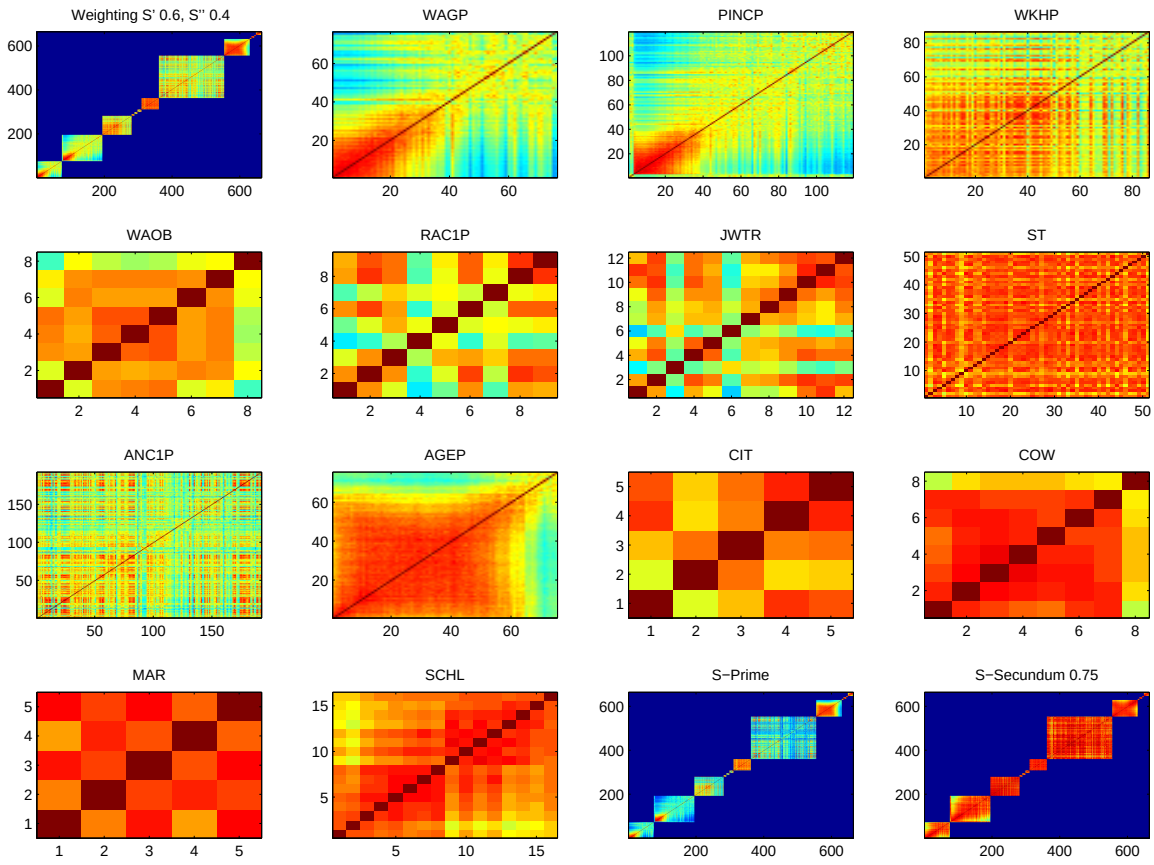

Figure 5.11: Sample results for Census PUMS data set ($T = 0.75, c_1 = 0.6, c_2 = 0.4$).

income (PINCP), *Usual hours worked per week past 12 months* (WKHP), *World area of birth* (WAOB), *Recoded detailed race code* (RACE1P), *Means of transportation to work* (JWTR), *State Code* (ST), *Recoded detailed ancestry* (ANC1P), *Age* (AGEP), *Citizenship status* (CIT), *Class of worker* (COW), *Marital status* (MAR) and *Educational attainment* (SCHL). The only attribute not shown in Figure 5.11 is *Sex* (SEX), since it contains only two values.

The total similarity across all attributes is shown in the image to the top left hand corner of Figure 5.11, while the S-Prime and S-Secundum values are shown in the bottom right hand corner. There are several numerical values worth mentioning here, the first being the two attributes related to income, WAGP (*Wages or income in the past 12 months*) and PINCP (*Person's total income*). Both of these attributes exhibit very numerical tendencies in that values that are close together numerically tend to be more similar than those that are further apart numerically. However, there is a noted exception to this rule for the attribute PINCP, since when the income is below zero these values have very low similarity to values just above zero, yet appear more similar to high incomes.

Another attribute worth noting is *Educational attainment* (SCHL), in the bottom row of Figure 5.11. The values in this attribute appear to be partitioned into two distinct groups that have a high level of similarity within a partition, and lower similarity outside of it. The two values at the boundary of these two groups are values 8 and 9, which correspond to '*Grade 12 no diploma*' and '*High school graduate*' respectively. This result indicates that, based on the subset of attributes in the data set, there is a strong relationship between having attained a level of education above that of high school graduate, and also between the levels of education that fall below this benchmark.

An example of a numerical attribute from the ACS PUMS data set which does not exhibit a numerical ordering is that of '*Usual hours worked per week last 12 months*' (WKHP), shown in the top right hand corner of Figure 5.11. Although there are quite a few of the values which are numerically close that also have a high level of similarity, there are also many values which do not follow this convention. This would seem to indicate that the application of a

traditional numerical noise addition technique on this attribute could result in reduced data quality in the perturbed data set. We will now discuss how best to handle this phenomenon when applying statistical disclosure control methods in practice.

5.3.4 Numerical Attribute Phenomenon

One of the novel applications of our similarity measure is that we can test whether numerical attributes really behave in a 'numeric' way for the purpose of noise addition. One way in which we can verify the quality of our similarity measure is to see if numerical attributes behave in such a way that values that are close together numerically are similar. In our experiments on several data sets, we observed that many numerical attributes exhibited this property. However, we also observed that for some numerical attributes, there did not appear to be such a clear relationship between value closeness and similarity.

The implications of this observation are that for some numerical attributes on particular data sets, it would not be advisable to add noise in a traditional numerical way, such as via a normal distribution. In this situation it may be more advantageous to apply a noise addition technique more commonly associated with categorical attributes, such as our *VICUS* technique presented in Section 6.4.

5.3.5 Ordering Categorical Values

The opposite end of the spectrum to the numerical attribute phenomenon is the application of an ordering to categorical values. By their very definition, categorical values have no natural ordering, yet by rearranging the categorical values according to their similarity value we may be able to assign an ordering to them as well as determining the spacing between categories (e.g. equal). It would then be possible to apply traditional numerical noise addition techniques to the data set. The advantage being that these techniques have been highly studied and are generally easier to apply and computationally less expensive than categorical techniques.

An example of an attribute that has had its values reordered according to the relative similarity values is shown in Figure 5.12. The square to the bottom left of the figure shows the attribute in its original ordering, while the top right square shows the attribute values ordered according to similarities. Clearly this attribute does exhibit some sort of numerical-like qualities in that values that are situated next to one another appear more similar than those spaced further away. This application of our similarity measure will be the subject of future work.

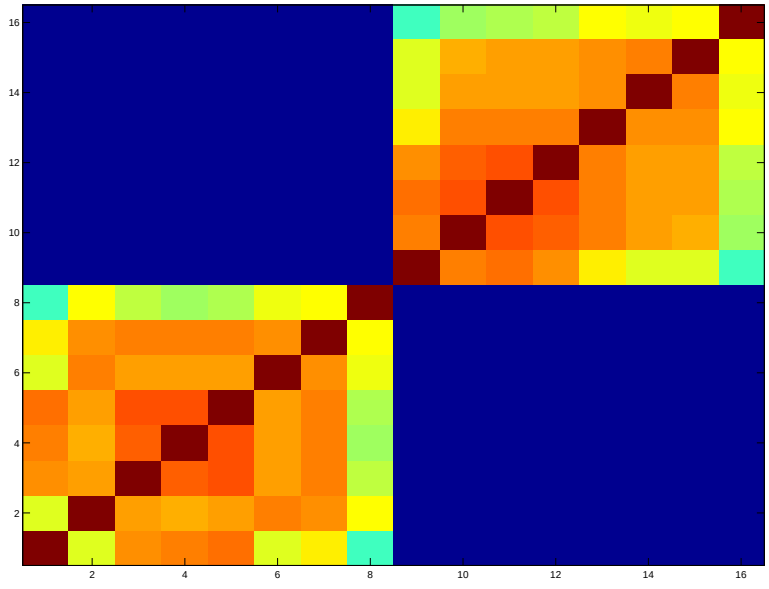

Figure 5.12: ACS PUMS Attribute *World area of birth* original and reordered.

5.4 Conclusion

In this chapter we have presented a new similarity measure designed specifically for use on categorical attribute values. The similarity measure aims to capture the notion of transitive similarity between values of an attribute, the so called S-Secundum similarity. As the results of experimental analysis show, our similarity measure is effective in capturing the similarities

that occur in the database when values co-appear in different records. In the next chapter we will incorporate our similarity measure into a noise addition technique for categorical values.

Chapter 6

VICUS - A Noise Addition Technique for Categorical Values

All truths are easy to understand once they are discovered; the point is to discover them.

–Galileo Galilei

Noise addition techniques have traditionally been applied to numerical values with great success as a means of Statistical Disclosure Control. However, when the data set contains categorical values the application of these techniques tends to be much less straightforward [125]. In Chapter 4 we proposed a framework that can be applied to categorical attributes to best meet the competing needs of security and data quality. A key component of this framework is the clustering of categorical values as a way to assign transition probabilities to the attribute values. In this technique values will be changed to other values in the same cluster with a much higher probability than they will change to a value in a different cluster.

In this chapter we propose a noise addition technique for categorical values which incorporates our similarity measure from Chapter 5 and assigns

transition probabilities based on the discovered clusters of attribute values. We also provide an analysis of experimental results to see how well our technique performs in the conflicting areas of security and data quality.

6.1 Motivating Example

Recall our Motivating Example lecturers data set from Section 5.1 in Chapter 5. Having evaluated the similarity values for the attributes in this data set, we are now faced with the problem of how best to partition the values so as to maximise similarity within a partition, and minimise similarity across partitions. Although it is not difficult to define a maximisation function that will indicate the quality of a selected partitioning of the graph, it is more difficult to decide how best to arrive at an optimal solution.

6.2 *VICUS* - Noise Addition Technique

We have named our noise addition technique *VICUS*, after the Latin word which loosely translates as village [1]. This is to reflect the underlying motivation for our similarity measure presented in Chapter 5.

Noise is added to a data set by applying the following three steps.

- **Step 1:** We partition the graph using the similarity measure for values within an attribute. We use a genetic algorithm to explore the solution space and arrive at a close to optimal partitioning of the graph.
- **Step 2:** Using the partitioning of the graph obtained from Step 1, we generate a transition probability matrix for all attribute values. The transition matrix gives the probabilities of each attribute value changing to every other value within the attribute.
- **Step 3:** We perturb each individual in the original data file by applying the transition probabilities to determine which value the original

[1] http://en.wiktionary.org/wiki/vicus

value becomes in the perturbed file. Noting that the value will generally have a relatively high probability of remaining the same in the perturbed file.

We next describe each of the steps in more detail.

6.2.1 Graph Partitioning

We now define the graph partitioning problem as presented in Bui and Moon [18]. Given a graph $G = (V, E)$ on n vertices and e edges, we define a partition $\mathcal{P}$ to consist of disjoint subsets of vertices of G. The *cut-size* of a partition is defined to be the number of edges whose end-points are in different subsets of the partition. A balanced *k-way* partition is the partitioning of the vertex set V into k disjoint subsets where the difference of cardinalities between the largest subset and the smallest one is at most one. The k-way partitioning problem is the problem of finding a k-way partition with the minimum cut-size [18]. We relax the condition of difference of partition sizes being at most 1, and we impose a lower bound on the minimum size of the partition $minS$. The k-way partitioning problem has been well studied and has been shown to be NP-complete in both the balanced and unbalanced form [49, 18]. Hence, we will apply a heuristic, namely a genetic algorithm, to solve the problem of moving from one solution to the next. Please note that in what follows we use "partitioning" to mean a collection of disjoint subsets whose union gives the original set, and we use "partition" to denote each such "subset".

Genetic Algorithm

A genetic algorithm starts with a set of initial solutions (*chromosomes*), known as a *population*. A series of functions are then performed over typically many iterations (*generations*) in an attempt to advance the quality of the candidate solutions. Once finished the best solution from the population is chosen as the final solution [18].

A single iteration proceeds as follows. Two members of the population

(*parents*) are chosen based on some fitness criteria. Then they are combined using a crossover function to produce a new solution (*offspring*). With some low probability this offspring is then modified using a mutation operator, potentially allowing the solution to climb out of a local maximum and provide a better chance of reaching the global maximum. The offspring is then tested to see if it is suitable to be added to the population and a *replacement* method is used to select the member of the population to be replaced. We then start a new iteration and continue until some stopping criteria is met. This type of genetic algorithm, which produces only one new solution per generation, is known as a steady-state genetic algorithm [18]. The pseudo code for this algorithm, as stated in Bui and Moon [18], is as follows

```
create initial population of fixed size;
do {
    choose parent1 and parent2 from population;
    offspring = crossover(parent1,parent2);
    mutation(offspring);
    if suited(offspring) then replace(population, offspring);
} until(stopping condition); report the best answer;
```

When setting the initial population before the first iteration of the algorithm for our graph partitioning problem, we generate a user defined number of partitions. Care is taken to ensure that the partitions within the solution all have at least the minimum number, $minS$, of values per partition. The minimum partition size, $minS$, is a user defined parameter.

At the *parent selection step*, a fitness function is used to select the two parent chromosomes used to create the next generation. The fitness of a chromosome is given by

$$F_c = \sum_{k=1}^{Atts} F_i \tag{6.2}$$

where $Atts$ is the number of attributes in the data set, and

$$F_i = \sum_{i}^{numVals} \sum_{j}^{numVals} (S_{ij}^{Same} - S_{ij}^{Diff}) \tag{6.3}$$

where

$numVals$ - the number of values in attribute k,
S_{ij}^{Same} - the sum of total similarities S_{ij} within the same partition,
S_{ij}^{Diff} - the sum of total similarities S_{ij} across the partition,
S_{ij} is the similarity between value i and j, which has been centred around 0 by subtracting the median total similarity value from all similarities.

The proportional selection fitness F_{ps} of the chromosome is given by

$$F_{ps} = (F_w - F_c) + (F_w - F_b)/3 \tag{6.4}$$

Where

F_w is the fitness of the worst solution in the population,
F_b is the fitness of the best solution in the population,
F_c is the fitness of chromosome c.

Each chromosome is selected as a parent with a probability that is proportional to its fitness value. Hence, the probability that the best chromosome is chosen is four times as high as the probability that the worst chromosome is chosen. This type of selection scheme is called *proportional selection* [18].

A *crossover operator* creates a new offspring by combining parts of the two parent chromosomes.

The *mutation operator* is applied as follows, m positions on the chromosome are selected at random and their values are randomly changed to another partition, where m is a uniform random integer variable on the interval $[0, \lfloor n/100 \rfloor]$. Once this mutation is applied we have to ensure that the partitions do not become too unbalanced, that is, that no partition has a size less than $minS$. To do this select a random point on the chromosome and change the required number of values starting at that point and moving to the right, wrapping if needed. This process will also produce some additional mutation [18].

A balance between population diversity and reasonable running time

needs to be found. First an attempt is made to replace the parent with the fitness below that of the child, if the parents both have a higher fitness than the child, replace the chromosome with the worst fitness.

The stopping criterion used is to stop when either the number of iterations reaches a user defined level, or the overall fitness of the population has reached a status quo.

6.2.2 Transition Probability Matrix

When deciding how much noise to add when we perturb a data set, we must decide on how best to distribute the transition probabilities amongst the possible choices. Note that we define two separate methods for defining the transition probabilities, the first being our *VICUS* method, and the second being a method we term *Random*, which is used to evaluate the effectiveness of the *VICUS* method. We next describe both of them in more detail.

VICUS Method

Given a partitioning $\mathcal{P}$ of the original data set which divides all possible values into k disjoint sets, we calculate the transition probabilities for each attribute individually. We define the notation as follows

A - the set of values of attribute A.
$\mathcal{P}_A \subset \mathcal{P}$- a partitioning - a collection of disjoint subsets (some of which may be empty) of A such that $\bigcup_{i=1}^{k} = A$.
$\mathcal{P}_A = \{A_1, A_2, \cdots, A_k\}$
$\mathcal{S}$ - the partition from $\mathcal{P}_A$ containing an attribute value, $a_i \in A$.
$\mathcal{S}^{'} = \mathcal{P}_A \setminus \mathcal{S}$ - the relative compliment set containing all other partitions.
$\mathcal{P}_A = \mathcal{S} \cup \mathcal{S}^{'}$

The define the transition probabilities for an attribute value, a_i, as follows

$$P_s + P_{sp} + P_{dp} = 1 \tag{6.5}$$

where P_s is the probability of an attribute value remaining unchanged,

$$P_{sp} = (|\mathcal{S}| - 1) \times p_{sp} \tag{6.6}$$

where p_{sp} is the probability that an attribute value is changed into a different attribute value from the same partition, and $|\mathcal{S}|$ is the number of attribute values in the partition containing the value a_i.

$$P_{dp} = |\mathcal{S}^{'}| \times p_{dp} \tag{6.7}$$

where p_{dp} is the probability that the value a_i changes to a value from a different partition and $|\mathcal{S}^{'}|$ is the number of attribute values that are in a different partition to a_i.

We now introduce two parameters that allow the data manager to adjust the amount of noise to be added to the microdata file. The first parameter, k_1, is defined such that an attribute value a_i is k_1 times more likely to stay the same than to change to another value in the same partition. The second parameter, k_2, tells us how many times more likely a value a_i is to change to another in the same partition than one in a different partition. Hence, we can reformulate our probabilities as

$$P_s = k_1 \times p_{sp} = k_1 \times k_2 \times p_{dp} \tag{6.8}$$

and Equation 6.4 becomes

$$P_s + (|\mathcal{S}| - 1) \times \frac{P_s}{k_1} + |\mathcal{S}^{'}| \times \frac{P_s}{k_1 \times k_2} = 1$$

From the above, the probability that a value remains the same becomes

$$P_s = \frac{k_1 \times k_2}{k_1 \times k_2 + k_2 \times (|\mathcal{S}| - 1) + |\mathcal{S}'|} \tag{6.9}$$

The probability of a value changing to another in a different partition becomes

$$p_{dp} = \frac{1}{k_1 \times k_2 + k_2 \times (|\mathcal{S}| - 1) + |\mathcal{S}'|} \tag{6.10}$$

The probability that a value changes to another in the same partition

becomes

$$p_{sp} = \frac{k_2}{k_1 \times k_2 + k_2 \times (|\mathcal{S}| - 1) + |\mathcal{S}'|} \tag{6.11}$$

***Random* Method**

We also define a set of transition probabilities for the method we term *Random*. This method does not assign probabilities for P_{sp} and P_{dp}, but rather introduces the probability of a value changing to any other value in the attribute, which is denoted P_c. However, it still uses Equation 6.8 to calculate the probability of a value remaining unchanged in the perturbed data set. We define the probability of a value changing to any other value in the attribute as follows

$$P_c = \frac{1 - P_s}{|\mathcal{S}| + |\mathcal{S}'| - 1} \tag{6.12}$$

The resulting method will perform better than a truly random method, as it is imparting some of the information from our partitioning of the values when calculating the value for P_s. However, to evaluate the quality of our method we need to perturb the 'random' in such a way as to be able to compare the results of our security measure and data quality tests.

6.2.3 Perturbing Microdata File

Once the transition probability matrix has been generated for each attribute, the next step is to simply perturb the original microdata file according to the transition probabilities assuming that a random value is drawn to decide if the value changes to another value in the same partition, one from a different partition, or remains unchanged.

6.3 Evaluation Methods

We now need to define the evaluation of how well our noise addition technique meets the conflicting goals of security and data quality. In evaluating the security of a perturbed data set we assume that the intruder is aware

of the exact perturbation technique and we apply an information theory entropy [107] measure to estimate the amount of uncertainty the intruder has about the identity of a record as well as the value of a confidential attribute. We apply two techniques to gauge how well our noise addition technique preserves the underlying data quality, the first is a decision tree classifier, and the second is the chi-square statistic test.

6.3.1 Security Measure

One way in which we can measure the security of a released microdata file is by estimating how sure an intruder is that they have identified a record, and more importantly the correct confidential value for that record. To gauge the amount of uncertainty an intruder has about having identified a particular record in the perturbed microdata file, we calculate the entropy for this record. Similarly, by calculating the entropy of a confidential value we can estimate the amount of uncertainty the intruder has that the value was the same value in the original microdata file. We assume that there is only one confidential or sensitive attribute in the microdata file; it is straightforward to generalise the following method to a case where there is more than one confidential attribute. Let us assume that an intruder knows how noise has been added in order to perturb the released microdata file; the intruder also knows one or more attribute values from a particular record that they are interested in and they wish to learn the confidential value for this record.

We assume that an intruder only has access to the perturbed and not the original microdata file. We further assume that the intruder is trying to compromise one particular record in the database and that they know the original values of some or all non-confidential attributes for that record. We calculate the entropy of a record $H(P)$ according to Algorithm 6.1, where the following notation is used:

A_x is the perturbed value of attribute A in record x;
$O(A_x)$ is the original value of attribute A in record x;
V_A is the original value of attribute A known to the intruder;
$p(V_A = O(A_x))$ is the probability that V_A is the perturbed value of attribute A known to the intruder;

p_x is the probability that record x is the record that contained V_A in the original file.

Input: Transition Probability matrix M, Perturbed microdata file P, Attribute value $V_A \in A$ for each attribute A known to the intruder
Output: $H(P)$ entropy

initialise sum_p to 0;
for *each record* x *in* P **do**
  initialise p_x to 1;
  for *each attribute* A *for which the intruder knows* V_A **do**
    /* Product of the probability that V_A in O became A_x in P, taken over all attributes with values known to the intruder */
    $p_x = p_x \times p(V_A = O(A_x))$;
  end
  $sum_p = sum_p + p_x$;
end
for *each record* x *in* P **do**
  /* normalise the value of p_x to between 0 and 1 */
  $p_x = p_x / sum_p$;
end
/* Now calculate the entropy of the record */
$H(P) = \sum_1^{|P|} p_x log_2 \frac{1}{p_x}$;
return $H(P)$;

Algorithm 6.1: Calculating entropy of a record in perturbed microdata file.

We calculate the entropy $H(D)$ for the confidential value V_C according the Algorithm 6.2. The entropy will show how much uncertainty the intruder has about the confidential value V_C for record x that they are trying to compromise. We use the following notation:

C_x is the perturbed value of the confidential attribute C in record x;
$p(c_i = O(C_x))$ is the probability that confidential value C_x in the perturbed file originated from the value c_i in the original file.

Input: Transition Probability matrix M, Perturbed microdata file P, Probabilities p_x for all records (from Algorithm 6.1)
Output: $H(D)$ entropy

for *each value* $c_i \in C$ **do**
  initialise probability D_i to 0;
end
for *each record* x *in* P *with* C_x *for* C **do**
  for *each confidential value in* $c_i \in C$ **do**
    /* Sum the probability that C_x in P originated from c_i */
    /* in O and multiply by the probability that */
    /* record x is the record the intruder 'knows' */
    $D_i += p(c_i = O(C_x)) \times p_x$;
  end
end
/* Now calculate the entropy of the confidential value V_c */
$H(D) = \sum_1^{|C|} D_i log_2 \frac{1}{D_i}$;
return $H(D)$;

Algorithm 6.2: Calculating entropy of confidential attribute in perturbed microdata file.

When we run experiments on our test data sets, we will compare the entropies for when the user knows 1, 2 and 3 attribute values for a record in the original file. We also compare these results to the 'worst case scenario' of when the intruder knows all attribute values for a record excluding the confidential value.

6.3.2 Data Quality

As discussed in Section 3.5, information loss is an important consideration when evaluating the quality of a perturbation technique [116]. Ideally the goal of the data manager is to minimise the reduction in data quality while at the same time maximise the security of released data. In order to evaluate how our method performs in terms of information loss we present two differ-

ent methods of evaluating the quality of the perturbed data files. The first method is to apply a decision tree builder on both the original and perturbed data sets and compare the classification errors. The second method is to apply a chi-square statistical test to both the original and perturbed data sets to ascertain how successfully *VICUS* preserves the underlying statistics from the original data set. We now describe each technique in more detail.

Decision Tree Classification

Several of the data sets that we have selected for experimental analysis were originally designated as classification data sets. In data mining, classification is a process of assigning a record to a class based on the values of the one or more attributes in the record [60]. One example of a class attribute in the data sets we introduced in Section 5.3.1 is the classification of mushrooms into either poisonous or edible. This indicates that a good way to test if our method preserves the underlying data quality would be to compare the classification results for the original data sets with those of the perturbed data sets.

According to Rokach, a decision tree is "a classifier expressed as a recursive partition of the instance space" [105]. The partitioning occurs at nodes in the tree, whereby a discrete function of the input attribute values causes the tree to branch into two or more sub-nodes [105]. The tree is directed, with the root node being the node with no edges entering it. In the top down approach to decision tree building the algorithm generally starts with all training instances at the root node, and then recursively partitions based on selected attribute values, or ranges in the case of numerical values [105, 60]. The selection of the test attribute is generally based on some heuristic or a statistical measure such as information gain, as with Quinlan's C4.5 algorithm [103] which is implemented in WEKA as the J48 algorithm [128].

We use the WEKA J48 decision tree builder [128, 129] on the original and perturbed microdata files and compare the percentage of incorrectly classified instances to assess the quality of our noise addition technique. When evaluating the quality of the decision trees for the perturbed data, we use the

perturbed file as the 'training' data set, and the original microdata file as the 'testing' set. To evaluate the quality of the decision tree produced from the original data sets, we use the "*k-fold cross-validation*" method [59], and we also evaluate the misclassification when the decision tree is both trained and tested on the original data set. The k-fold cross-validation method first partitions the original data set into k disjoint subsets, ($S = \{S_1, S_2, \ldots, S_k\}$), also know as 'folds' [59]. The method then performs k iterations to iteratively build the classification tree, where in the ith iteration, the subset S_i is kept aside as the training set, and the tree is built on the the remaining $S \setminus S_i$ records [59]. The overall classification error is given by dividing the total number of misclassified records for all iterations, by k.

Chi-square Test

The *chi-square test* is a commonly applied statistical measure for determining the statistical significance of an association between two categorical attributes [67]. We follow the five step approach to determining statistical significance as outlined by Utts and Heckard [67, p.184].

- **Step 1:** Determine the null and alternative hypotheses.
- **Step 2:** Summarise the test statistic after verifying any necessary conditions (such as data set size).
- **Step 3:** Determine the p-value on the assumption that the null hypothesis is true.
- **Step 4:** Determine any statistical significance according to the p-value.
- **Step 5:** Report the conclusion.

Note that our aim here is not to determine if there is any statistical significance of the attribute associations studied, rather we aim to determine that any such significance is undisturbed by our perturbation method. However, we will approach the analysis in a manner consistent with the above outlined steps.

		Program			
		Computer Science	Mathematics	Medicine	**Total**
Lecturer	Prof. K. White	1	0	0	**1**
	Dr J. Black	2	0	0	**2**
	Dr H. Pink	0	1	0	**1**
	Dr M. Brown	0	1	0	**1**
	Prof. T. Green	0	0	1	**1**
	Dr D. Blue	0	0	1	**1**
	Total	**3**	**2**	**2**	**7**

Table 6.1: Lecturer Microdata File - Sample

We will analyse our data set in the form of two-way contingency tables, which count the co-occurrence of a categorical value from one attribute with a value in another attribute, for all combinations of values. For instance, if we consider our Motivating Example in Section 5.1, Table 6.1 shows the contingency table for Attribute 1 and 2. We consider the null and alternative hypotheses about categorical data presented in a two-way contingency table as

$$H_0 : \text{The two variables are not related}$$
$$H_a : \text{The two variables are related}$$

where H_0 is the null hypothesis, and H_a is the alternative hypothesis [67, p.528].

Since we are dealing with categorical data specifically, and all of our experimental data sets have been perturbed under this assumption, a natural choice for a test statistic, as outlined in Step 2, is the chi-square statistic. The chi-square statistic measures the difference between the observed counts in the contingency table and the so-called expected counts, which are those that would occur if the there was no relationship between the two categorical variables. The general form of the chi-square statistic (χ^2) (also referred to as Pearson's Chi-Square) is given in Equation 6.12, the notation is the that used in [67].

$$\chi^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{6.13}$$

where

O_{ij} - the observed count for the i, jth cell of the contingency table

E_{ij} - the expected count for the i, jth cell of the contingency table if there were no relationship between the two variables

R_i - row sum for ith row of contingency table

C_j - column sum for jth column of contingency table

n - total cell count

r - number of rows

c - number of columns

$E_{ij} = \frac{R_i C_j}{n}$ - expected count for cell i, j

An alternative statistic is the Likelihood Chi-Square statistic shown in Equation 6.13

$$\chi^2_{LR} = -2 \sum_{i,j} O_{ij} ln \frac{E_{ij}}{O_{ij}} \tag{6.14}$$

For large data sets the values of χ^2 and χ^2_{LR} should be comparable. The same method is used to compare these statistics to the p-value to evaluate the correctness of the null hypothesis.

An important necessary requirement for the application of the chi-square test on a data set is that the data is large enough to be able to give a reasonable approximation. According to Utts and Heckard [67], a commonly applied guideline which satisfies this co-called "large" sample requirement is for all expected counts to be larger than 1, and for at least 80% of cells in the contingency table to have an expected count larger than 5.

Once we have calculated the chi-square statistic for a pair of attributes, we need to apply Step three of our statistical method, that is, we need to determine the p-value under the assumption that the null hypothesis, H_0, is true. The p-value is "the probability that the chi-square statistic could have been as large or larger if the null hypothesis were true" [67, p.533]. This value equates to the area under the chi-square distribution with df degrees of freedom. The degrees of freedom (df) for both χ^2 and χ^2_{LR} is $(r-1)(c-1)$. Once the p-value has been determined for the calculated chi-square statistic, we need only compare this value to the *critical value*

at the set degrees of freedom. This is the value that the p-value must be below for there to be a reasonable significance level α. A generally accepted significance level would be $\alpha = 0.05$. This signifies that there is less than a 5% chance that the relationship between the two variables occurred by chance. When the p-value is less than α we say that the null hypothesis can be rejected, and there is a relationship between the two variables.

6.4 Experiments - Noise Addition

From the data sets examined in Section 5.3, where we looked at experimental results for our similarity measure, we have selected two to run further experiments to illustrate the quality of our *VICUS* noise addition technique.

Mushroom Data Set

The Mushroom data set is a fully categorical data set, so is very suitable for the analysis of our technique. In addition, the data set is a classification data set, so is particularly relevant for the testing of data quality via the techniques we have selected.

In preparing our experiments we generated a multigraph from the original data set. The genetic algorithm and similarity measure parameters used in the running our partitioning technique are as follows

- Initial population size: 50
- Number of partitions k: 6
- Minimum partition size ($minS$): 6
- Number of cut-points: 5
- Maximum number of iterations: 100,000
- S-Secundum threshold T: 0.65
- S-Prime weighting c_1: 0.6
- S-Secundum weighting c_2: 0.4

The genetic algorithm was run 30 times and the partitioning with the largest fitness function was selected as the solution partitioning. Recall that we use the term "partitioning" (PG) to denote a collection of disjoint subsets of the vertex set V, such that the union of all the subsets gives the set V. We use the term "partition" to refer to such a subset. Thus we have :

$$PG = \{P_1, P_2, \ldots, P_k\}$$

$$P_i \cup P_j = \emptyset$$

for all $1 \leq i, j \leq k$ and

$$\bigcup_{i=1}^{k} P_i = V$$

The solution partitioning is shown below. The top row of each partition shows the vertex labelling, and the bottom row the attribute number that this value belongs to in the data set.

```
Partition 1 labels: [ 11 19 21 23 24 25 26 27 29 30 36 37 38 39 41
            attrib: [  3  4  5  6  6  6  6  6  6  7 10 10 10 10 10
                      42 43 46 48 50 55 74 77 78 91 ]
                      10 10 11 12 12 14 17 18 19 21 ]

Partition 2 labels: [ 8 12 20 40 52 53 54 60 61 66 67 68 69 70 81 93 ]
            attrib: [ 2  3  4 10 13 13 13 15 15 16 16 16 16 16 19 21 ]

Partition 3 labels: [ 2 7 15 33 35 65 75 76 82 97 ]
            attrib: [ 1 2  4  8  9 15 18 18 20 22 ]

Partition 4 labels: [ 6 31 32 45 64 92 ]
            attrib: [ 2  7  8 11 15 21 ]

Partition 5 labels: [ 9 10 16 17 18 44 71 73 79 80 83 84 85 86 87 98 ]
            attrib: [ 3  3  4  4  4 10 16 17 19 19 20 20 20 20 20 22 ]

Partition 6 labels: [ 1 3 4 5 13 14 22 28 34 47 49 51 56 57 58 59 62 63
            attrib: [ 1 2 2 2  4  4  5  6  9 12 12 13 14 14 14 15 15 15
```

```
72 88 89 90 94 95 96 99 ]
16 21 21 21 22 22 22 22 ]
```

We next selected five different combinations of parameters for the transition probability generation, which are shown in Table 6.2. For each parameter selection we perturbed 30 files according to the generated transition probabilities.

Perturbation	k_1	k_2
Mush1	2	20
Mush2	5	20
Mush3	10	10
Mush4	10	20
Mush5	10	50

Table 6.2: Probability parameters for perturbations on Mushroom data set

Wisconsin Breast Cancer data set

The Wisconsin Breast Cancer (WBC) data set has been selected because it is a classification data set, which will be useful when we want to evaluate how our perturbation technique impacts on quality of the released data set. It also has a manageably small number of attributes (10 including the class attribute) and distinct attribute values (91). Note that although this data set consists of discrete numerical values, for the purpose of these experiments we will treat the data set as if it were fully categorical. That is, every distinct discrete numerical value will be converted into a categorical value for the purposes of noise addition. However, once perturbed, we will again be treating the data set as numerical when running a decision tree builder on the data sets to evaluate the quality of our *VICUS* noise addition technique.

In preparing our experiments we generated a simple graph from the original data set. The genetic algorithm and similarity measure parameters used in the running our partitioning technique are as follows

- Initial population size: 50

- Number of partitions k: 10
- Minimum partition size ($minS$): 4
- Number of cut-points: 5
- Maximum number of iterations: 100,000
- S-Secundum threshold T: 0.5
- S-Prime weighting c_1: 0.6
- S-Secundum weighting c_2: 0.4

The genetic algorithm was run 30 times and the partitioning with the largest fitness function was selected as the solution partitioning. This partitioning is shown below. The top row of each partition shows the vertex labelling, and the bottom row the attribute number that this value belongs to in the data set.

```
Partition 1 labels: [ 13 14 15 16 17 18 20 22 23 24 25 26 27 28 30 39 42
            attrib: [  2  2  2  2  2  2  2  3  3  3  3  3  3  3  3  4  5
                      43 44 45 46 47 48 50 59 62 63 64 65 66 67 68 70 71
                       5  5  5  5  5  5  5  6  7  7  7  7  7  7  7  7  8
                      73 78 80 88 91 ]
                       8  8  8  9 10 ]

Partition 2 labels: [ 51 53 55 60 79 ]
            attrib: [  6  6  6  6  8 ]

Partition 3 labels: [ 19 37 56 75 85 ]
            attrib: [  2  4  6  8  9 ]

Partition 4 labels: [ 1 3 4 5 6 7 8 10 54 69 ]
            attrib: [ 1 1 1 1 1 1 1  1  6  7 ]
```

```
Partition 5 labels: [ 35 58 76 81 82 83 84 ]
            attrib: [  4  6  8  9  9  9  9 ]

Partition 6 labels: [ 9 21 72 87 ]
            attrib: [ 1  3  8  9 ]

Partition 7 labels: [ 36 41 52 74 ]
            attrib: [  4  5  6  8 ]

Partition 8 labels: [ 29 31 32 33 34 40 77 86 ]
            attrib: [  3  4  4  4  4  4  8  9 ]

Partition 9 labels: [ 11 12 38 49 89 ]
            attrib: [  2  2  4  5  9 ]

Partition 10 labels: [ 2 57 61 90 ]
             attrib: [ 1  6  7 10 ]
```

We next selected five different combinations of parameters for the transition probability generation, which are shown in Table 6.3. For each parameter selection we perturbed the original microdata file 30 times to create 30 perturbed files according to the generated transition probabilities.

Perturbation	k_1	k_2
WBC1	2	2
WBC2	2	10
WBC3	2	50
WBC4	5	20
WBC5	10	10

Table 6.3: Probability parameters for perturbations on Wisconsin Breast Cancer data set

6.4.1 Security

As outlined in Section 6.3.1 we evaluate the security of our noise addition technique using entropy. For each data set we chose an attribute to be the confidential attribute C. We calculated the entropies for the situation when the user knows one, two and three attributes, and also when the user knows all attributes excluding the confidential one, to give a comparison 'worst case scenario' result.

For each of the 30 perturbed files, we averaged the entropies over all records and all files for when the intruder knows one attribute value.

Mushroom Data Set

The results for the record entropies for the case when the intruder knows 1 attribute value for a record are shown in Table 6.4, Entropies for the corresponding confidential attribute are shown in Table 6.5.

Perturbation	k_1	k_2	VICUS	Random
Mush1	2	20	11.9720	12.0074
Mush2	5	20	11.7502	11.7316
Mush3	10	10	11.6141	11.5772
Mush4	10	20	11.5826	11.5487
Mush5	10	50	11.5567	11.5284

Table 6.4: Average record entropy for Mushroom perturbations, intruder knows 1 attribute.

Perturbation	k_1	k_2	VICUS	Random
Mush1	2	20	2.0743	2.4496
Mush2	5	20	1.8778	2.1962
Mush3	10	10	1.8280	2.0036
Mush4	10	20	1.7220	1.9450
Mush5	10	50	1.6409	1.9045

Table 6.5: Average confidential attribute entropy for Mushroom perturbations, intruder knows 1 attribute.

In Figure 6.1 we can see how the entropy drops when the user learns more

attribute values for a particular record. We have selected for comparison the two perturbations that gave the highest and lowest entropy results on average, that is *Mush1* ($k_1 = 2, k_2 = 20$) and *Mush5* ($k_1 = 10, k_2 = 50$). Note that the dotted lines shown in this figure are only designed as a guide as to indicate the position of the next value, and there has been no form of extrapolation performed on this figure or others of a similar vein. Figure 6.1 shows that when the user knows only a few attributes, the difference in security between the *VICUS* and *Random* methods is minimal. When the user knows all 21 non-confidential attribute values for one record, the record entropy drops significantly and the *VICUS* method outperforms the *Random* method.

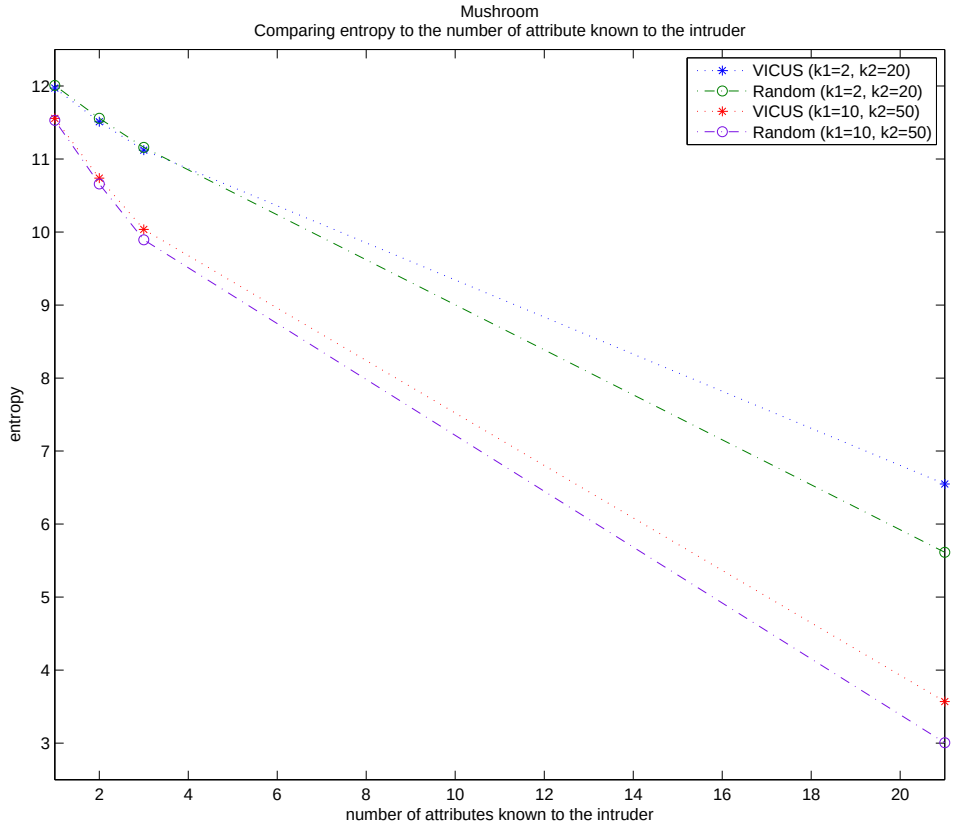

Figure 6.1: Comparing record entropy to the number of attributes known by intruder, Mushroom data set.

When analysing the confidential entropies in terms of the number of at-

tribute values known to the intruder, as shown in Figure 6.2, there is little change in the entropy for the *Mush1* perturbation, while for *Mush5* the *Random* method entropy drops away more quickly than for the *VICUS* method as the intruder's knowledge of all 21 non-confidential attribute values grows.

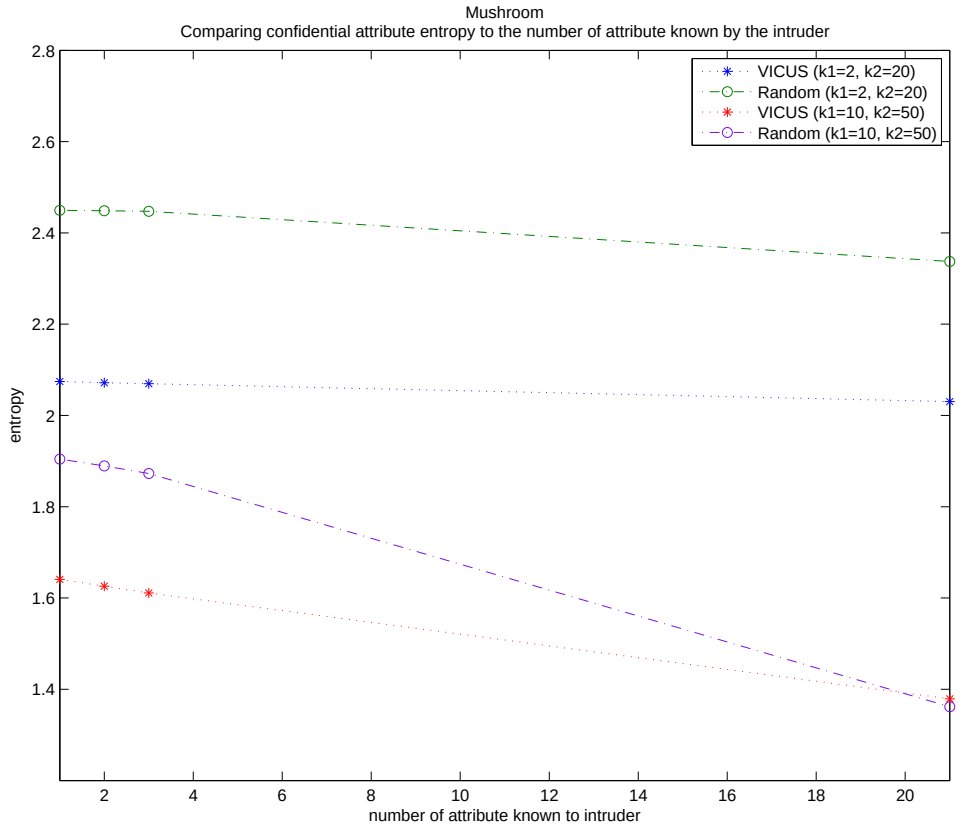

Figure 6.2: Comparing confidential attribute entropy to the number of attributes known by intruder, Mushroom data set.

We are also interested to know if certain attributes are more or less revealing than others, that is, if they yield a lower or higher entropy than average. Figure 6.3 shows the entropies for when an intruder knows 1 attribute value, and compares the average entropy for different attributes. Note that attribute 2 is not shown as it is the confidential attribute. There is relatively small variation in the confidential entropy regardless of which attribute the user knows a value for. While for the record entropy, we can clearly see that some attributes are more sensitive than others. The most

revealing attribute is attribute 3 (*Cap Surface*), while the least revealing is attribute 17 (*Veil Colour*). Figure 6.4 provides a more close up view of the entropies for each individual attribute, and also compares the entropies for the *VICUS* and *Random* methods. We show the maximal entropy value, that is, the value that the entropy would be if all attribute values were equally likely in the perturbation ($log_2 5644$). For attribute 7 and 17 the average values are extremely close to this indicator. This is due to the fact that both of these attributes contain only two values which fall into different partitions in the graph partitioning $\mathcal{P}$, so when perturbing these values there is no probability that a value will change to another value in the same partition.

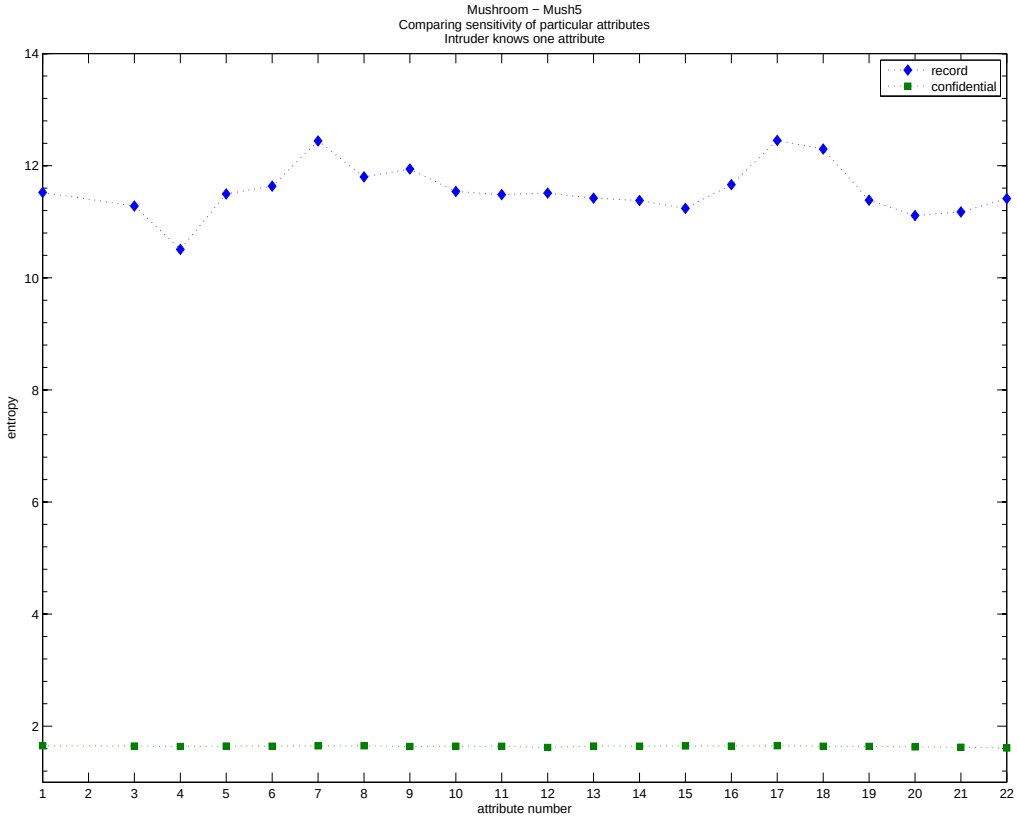

Figure 6.3: Entropy sensitivity for when the user knows 1 particular attribute, Mushroom data set.

For the confidential value entropy, it is not so obvious from Figure 6.5

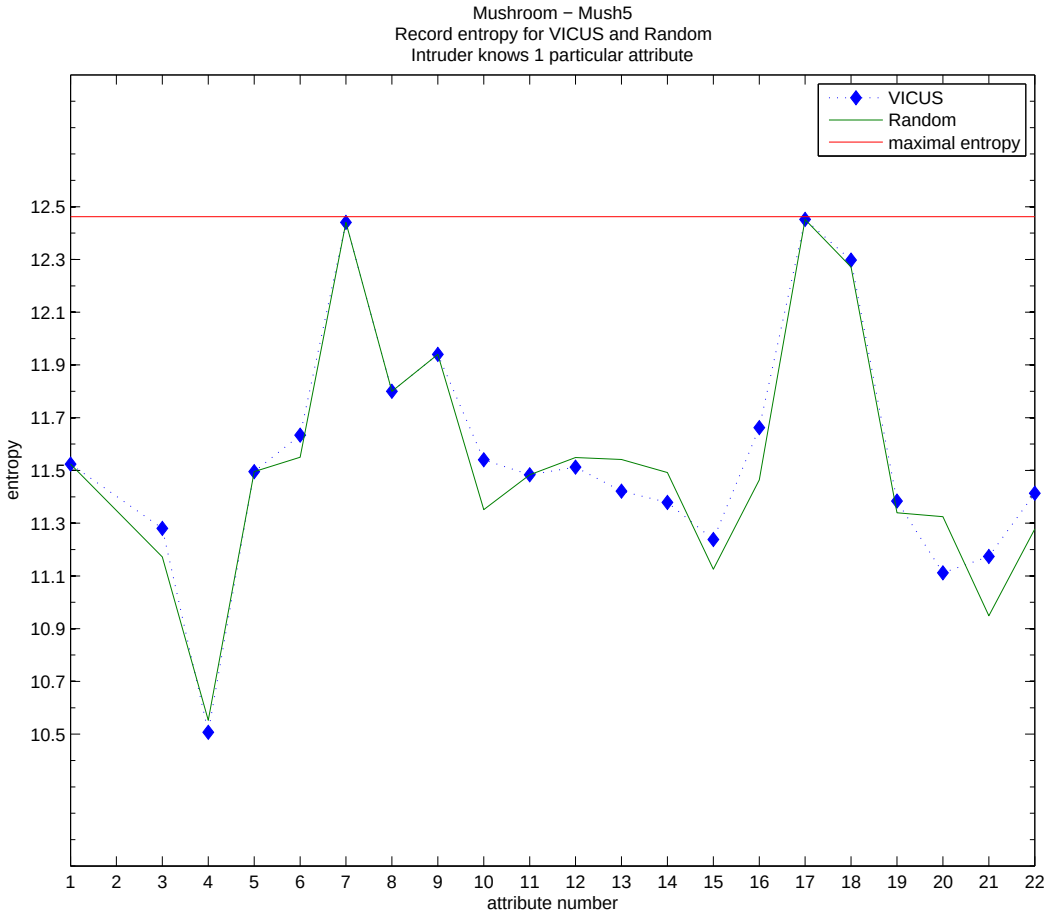

Figure 6.4: Record entropy sensitivity for when the user knows 1 particular attribute, Mushroom data set.

which attributes are the most and least revealing. However, attribute 1 (*Class*) is the least revealing, with the highest average entropy for the *VICUS* method, and attribute 21 (*Habitat*) is the most revealing with the lowest average entropy. With an overall average entropy of 1.6409 for the *VICUS* method, the perturbation *Mush5* is not ideal as there is little uncertainty for the intruder about the true value of the confidential attribute.

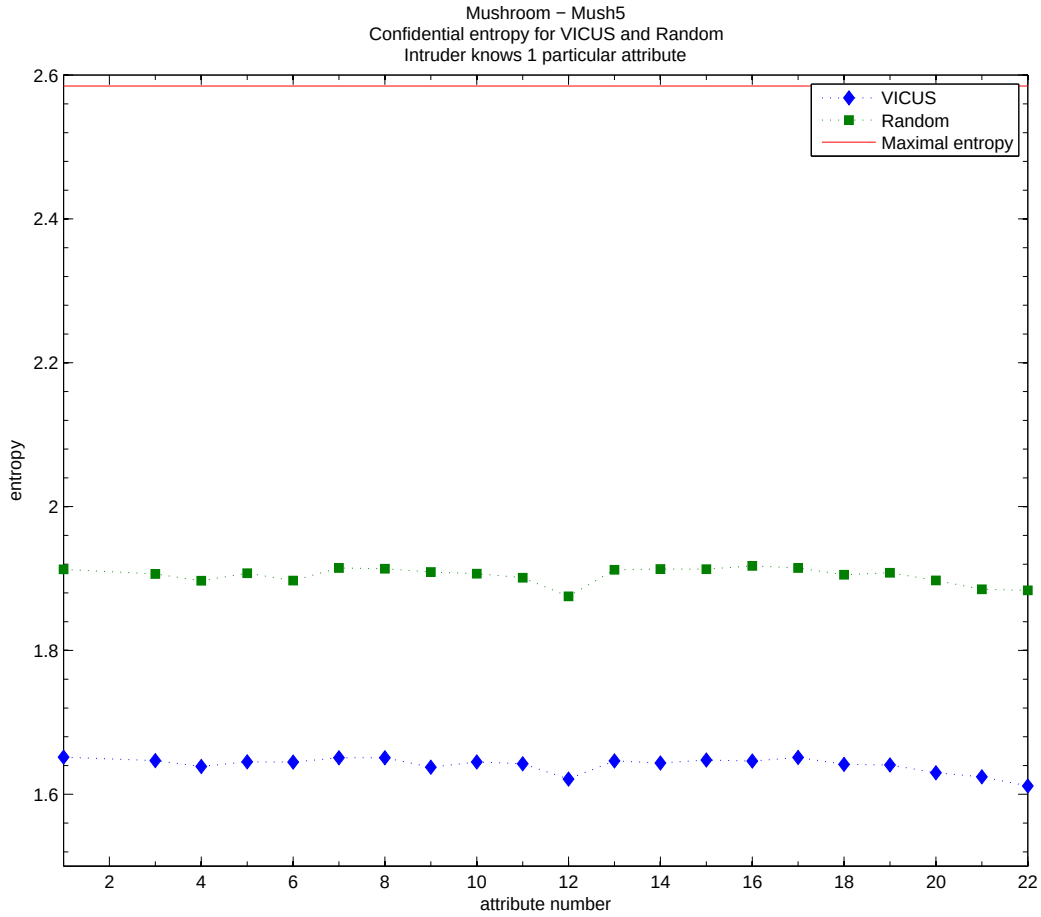

Figure 6.5: Confidential entropy sensitivity for when the user knows 1 particular attribute, Mushroom data set.

We next want to demonstrate that there is very little variation between the average entropies when the user knows a particular number of attribute values for files perturbed via the *VICUS* method. Note that we do not show a similar comparison for the confidential attributes since there is even

less variation for those entropy results. Figure 6.6 shows the distribution of the results across all 30 *Mush5* files perturbed via the *VICUS* method, for when the user knows three attribute values in a record. Note that due to the large number of combinations when we are choosing 3 attributes from 21, we randomly selected 100 attribute combinations for the 3 attribute entropy analysis on the Mushroom data files. Similarly, Figure 6.7 gives the distribution when the user knows 2 attribute values for a particular record.

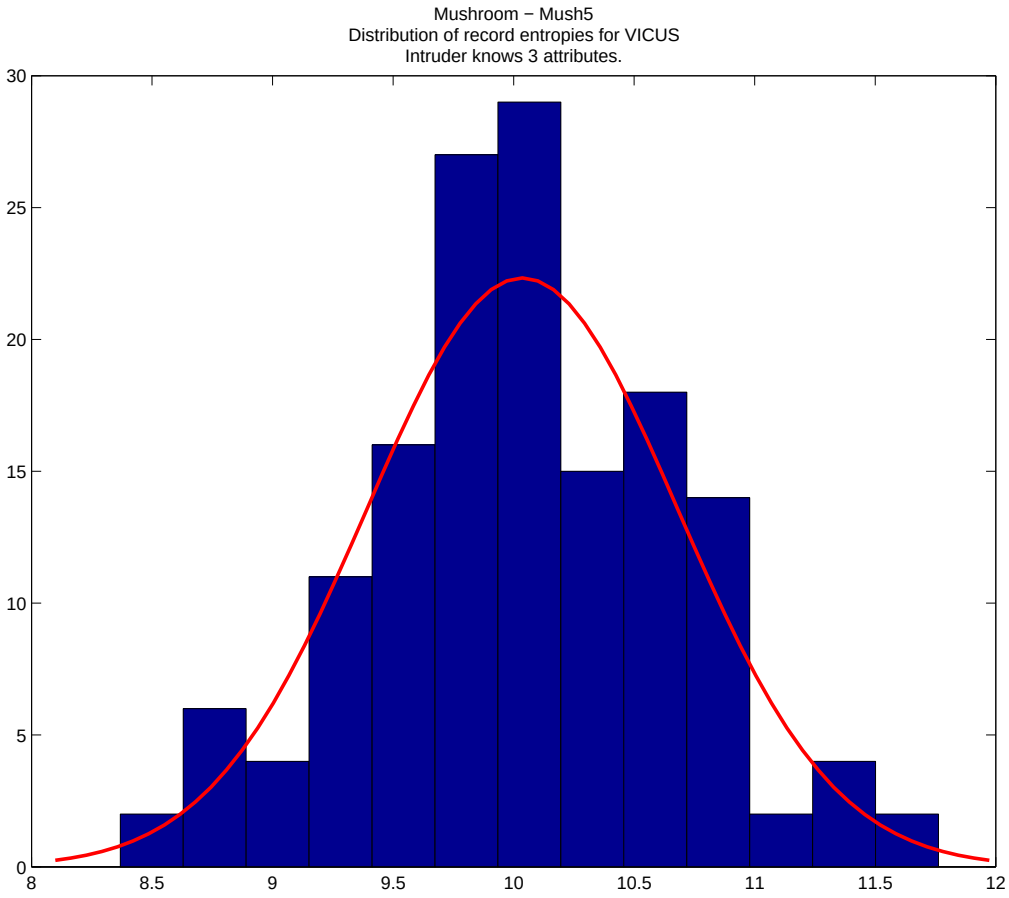

Figure 6.6: Distribution of record entropies when user knows 3 attributes for each individual attribute combination, Mushroom data set.

We also examine the distribution of the record entropies over the 30 perturbed files rather than according to which attribute value the intruder knows. In Figure 6.8 we can see that there is only a very small variation

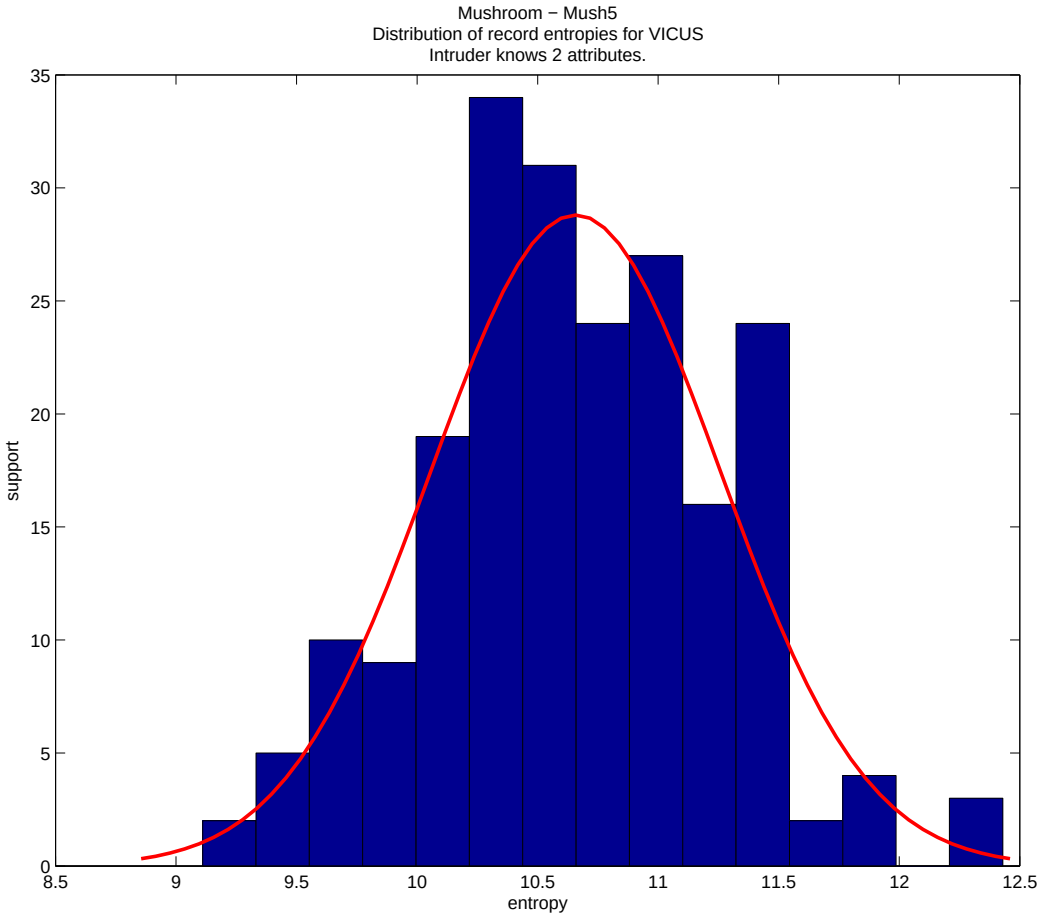

Figure 6.7: Distribution of record entropies when user knows 2 attributes for each individual attribute combination, Mushroom data set.

between any of the 30 perturbed files when the intruder knows one attribute value for a particular record. This indicates that there is only a small variation in the relative security of any of the files perturbed via the *VICUS* method. To gain a better understanding of the overall trend in distributions based on the number of attribute values known to the intruder refer to Figure 6.9, which comapres the distributions for 1, 2 and 3 attributes values. Note that for when the user knows only one attribute value the distribution falls into a smaller range of values.

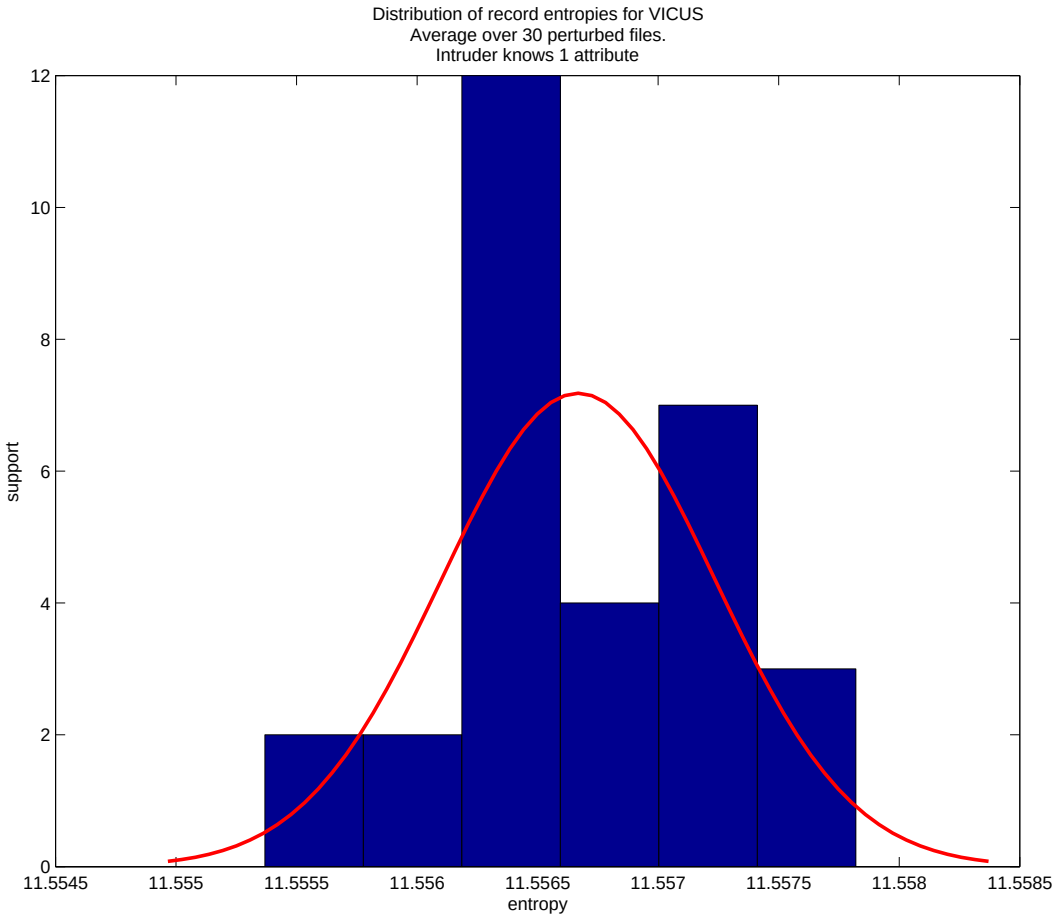

Figure 6.8: Distribution of record entropies when user knows 1 attribute, averaged over the 30 perturbed files, Mushroom data set.

The final comparison we make for the Mushroom data files is to examine how the *VICUS* method performs against the *Random* method for different

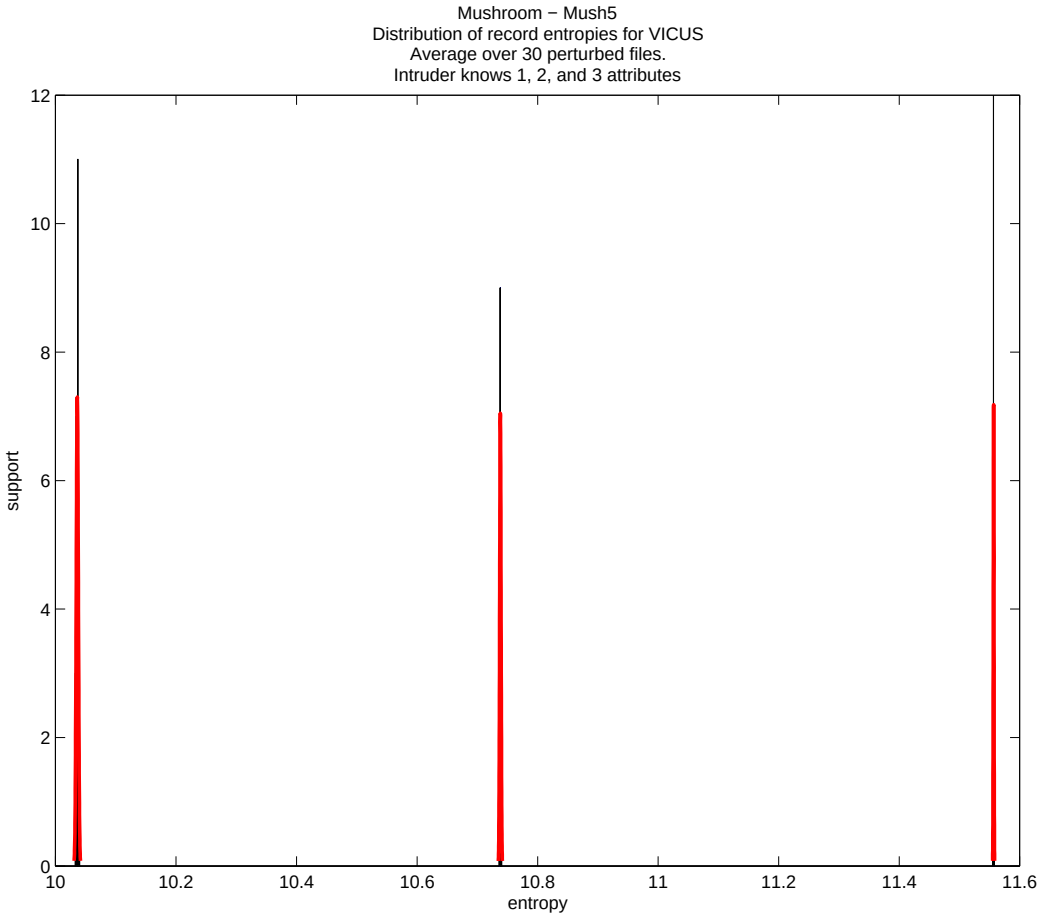

Figure 6.9: Distribution of record entropies when user knows 1, 2 and 3 attributes, averaged over the 30 perturbed files, Mushroom data set.

choices of the k_1 and k_2 transition probability parameters. Recall that k_1 is how many times more likely a value is to remain unchanged than to change to another value in the same partition, and a value is k_2 times more likely to change to a value in the same partition than one in a different partition. The bottom left hand image in Figure 6.10 fixes the values of the k_2 parameter, and shows how a variation in the k_1 parameter impacts on record entropies. Note that when the value of k_1 increases we see a slight drop in the entropy, and hence a slight drop in the security of the perturbed microdata file since the intruder has less uncertainty about which record in the perturbed file is the one containing the value he knows. We also note that for all three values of k_1 there is little difference between the average entropy for the files perturbed via the *VICUS* method and those perturbed via the *Random* method.

We now examine the bottom right hand image of Figure 6.10 to see how changing the value of k_2 while fixing the value of k_1 effects the record entropy. By having a relatively high value of k_1 we see little change in the entropy when we increase the value of k_2 since the amount of uncertainty is already quite low. Setting a higher value for k_1 will result in a higher probability of a value staying the same in the perturbed file, leading to less uncertainty about what the attribute value was in the original file.

Wisconsin Breast Cancer Data Set

For the Wisconsin Breast Cancer data set we have set attribute 1 (Clump Thickness) as the confidential attribute that the intruder is trying to uncover the value of in a particular record that they have supplementary knowledge for. Table 6.6 gives a summary of the average record entropies for the different k_1 and k_2 parameter choices, and Table 6.7 has the average confidential entropies for the same parameters. The top two images in Figure 6.10 also show how a change in one of the transition probability parameters effects the record entropy in the perturbed file. It is interesting to note that when comparing the product on k_1 and k_2, in the top right hand image in Figure 6.10, having a small value for k_1 and a large value for k_2 provides the highest level of security on average.

We provide a similar set of diagrams for the Wisconsin Breast Cancer

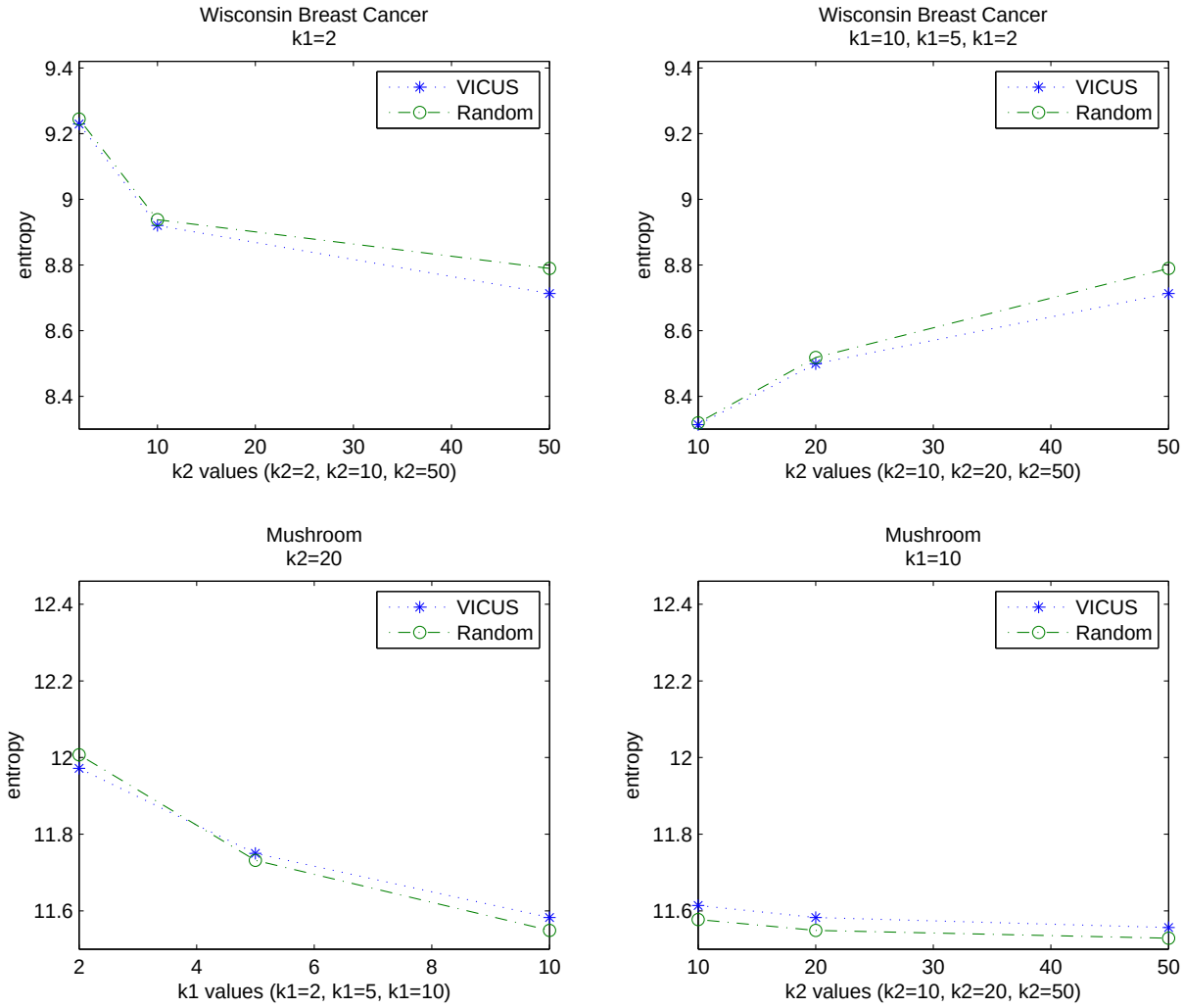

Figure 6.10: Record entropy comparison for WBC and Mushroom data sets.

Perturbation	k_1	k_2	VICUS	Random
WBC1	2	2	9.2294	9.2447
WBC2	2	10	8.9207	8.9379
WBC3	2	50	8.7131	8.7897
WBC4	5	20	8.4995	8.5186
WBC5	10	10	8.3135	8.3195

Table 6.6: Average record entropy for WBC perturbations, intruder knows 1 attribute..

Perturbation	k_1	k_2	VICUS	Random
WBC1	2	2	3.3212	3.3216
WBC2	2	10	3.2964	3.3166
WBC3	2	50	3.2481	3.3052
WBC4	5	20	3.2434	3.2884
WBC5	10	10	3.2184	3.2483

Table 6.7: Average confidential attribute entropy for WBC perturbations, intruder knows 1 attribute.

data set as for the Muchroom data set in Section A.5.1 of Appendix A. Having already discussed the results for the Mushroom data set, there is little to be gained by examining these figures in detail since they show very similar results.

6.4.2 Data Quality

We now present the experimental results for the data quality tests which were outlined in Section 6.3.2, that is, decision tree classification and Chi-square statistics testing.

Decision Tree Results

We use the WEKA software package [128] to test how well our perturbed data sets maintain their underlying data model when compared to the original data sets. The particular decision tree builder that we used was J48

decision tree builder, which is WEKA's implementation on the C4.5 decision tree builder [128]. For an explanation of the algorithm employed, we refer the reader to Section 6.3.2. The parameters that we used for the decision tree builder on all data sets tested are shown in the following list.

- To gauge the accuracy of a classification model built from the original data sets, we used the k-fold cross validation method and noted the predictive accuracy of the model. We also trained a decision tree on the original data set, and then tested the same data set against the model to evaluate how accurately it classified the instances. This second method also allows us to visualise the decision tree trained on the whole of the original data set.
- When testing the relative quality of the decision trees trained on the perturbed microdata files, we use the original data set for testing purposes. We note the percentage of incorrectly classified instances, and then compare this result to that obtained for decision tree trained on the original data set. As an additional comparison, the decision tree trained on the perturbed file, the same perturbed data file was also used as the testing data set to evaluate how accurately it classifies the instances from the same data set.
- In the parameter selection for the J48 decision tree builder in WEKA [129], the binary split option was selected, so that each inner node of the decision tree has exactly two children (e.g. val and $!val$). For categorical attributes, this results in each branch originating from an inner node containing either an attribute value or its negation. For numerical values it results in a range of values on each edge (e.g. ≤ 2 and > 2).
- The minimum number of instances classified to each leaf node is set to 20, so as to discourage the tree from becoming too large.
- The confidence factor for pruning is set to 25%, which is a the default value in WEKA [129].

In general, the quality of the decision trees produced will be evaluated using the percentage of incorrectly classified instances. For the perturbed

data sets, we will compare these percentages to the values obtained from the decision tree trained on the original data set.

Mushroom Data Set

The decision tree built from the original Mushroom data set is shown in Figure 6.11. The attribute which appears most prominently in this decision tree is *Odour* (attribute 6). The other two attributes which feature are *Spore Print Colour* (attribute 20) and *Population* (attribute 21). The confusion matrix for this decision tree tested against the whole original microdata file is given in Table 6.8, and shows that all instances in the data set were correctly classified. There are 2156 records classified as poisonous, and the remaining 3488 are classified as edible. The same predictive accuracy was obtained when the data set was trained using the k-folds cross-validation method, with the number of subsets $k = 10$. Looking at Figure 6.11, we can see that close to 75% of the records can be classified as poisonous solely on the basis of the value for the *Odour* (attribute 6) attribute being equal to *foul* (f).

	poisonous	edible	total
poisonous	2156	0	2156
edible	0	3488	3488
total	2156	3488	5644

Table 6.8: Confusion matrix for original Mushroom data set

We trained the J48 decision tree builder on all of the perturbed data sets described at the start of Section 6.4, that is, 30 files for each perturbation method (*VICUS* and *Random*), for each of the 5 different parameter choices for k_1 and k_2. The classification errors for all perturbed files is shown in Table A.7 of Appendix A, and the averages for each set of 30 perturbed files is shown in Table 6.9. A reminder that a value within an attribute is k_2 times more likely to change to another value in the same partition than one in a different partition, and k_1 times more likely to stay the same than change to a value in the same partition. Note that the percentage of incorrectly classified instances for the original Mushroom data set was 0, since all instance were correctly classified. Hence, for all perturbation the predictive accuracy is close to that for the original data set, since for most

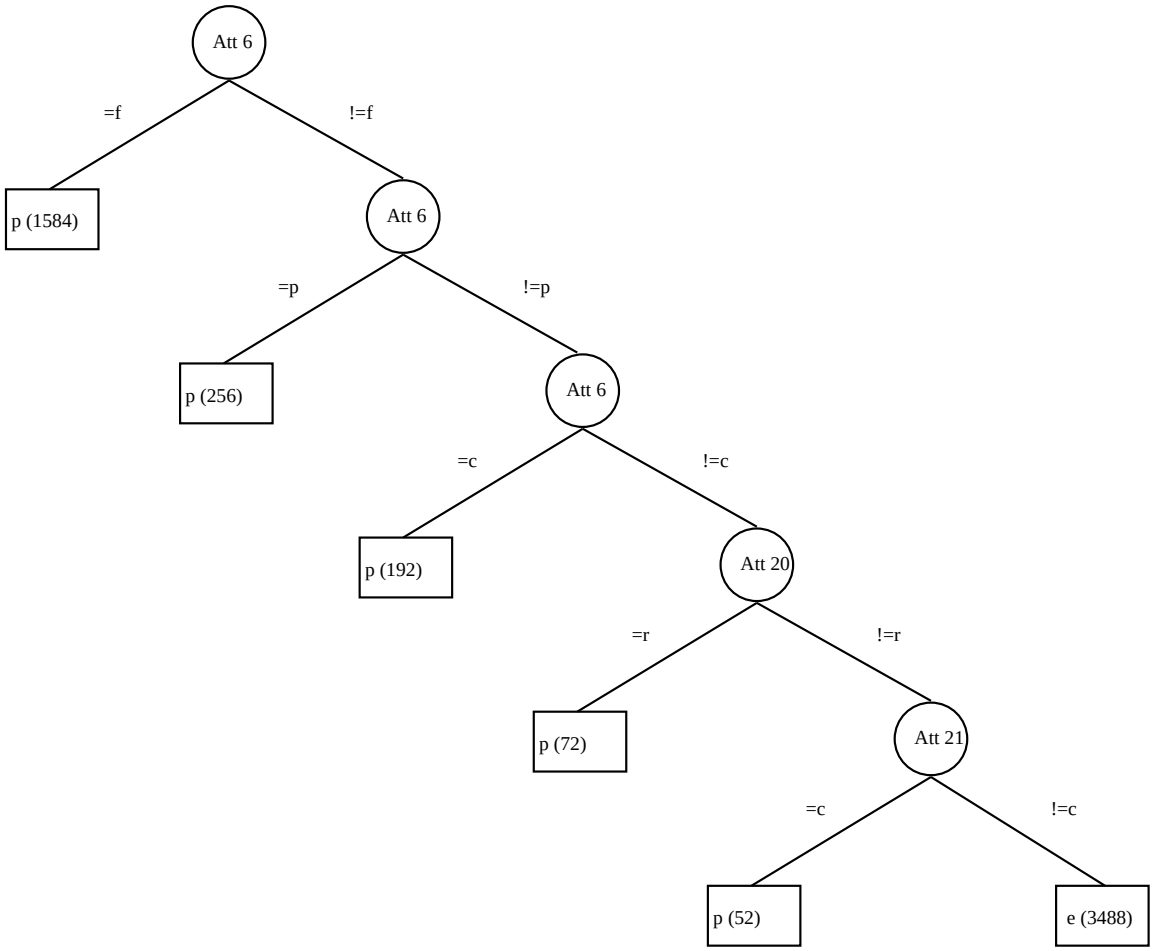

Figure 6.11: Decision tree for original Mushroom data set.

(k_1, k_2) parameter choices the average classification error is below 1.0%. Only when the highest level of noise is added, for $k_1 = 2$ and $k_2 = 20$ does the predictive accuracy fall below 99%. Figure 6.12 shows the distributions of classification errors for the *Mush5* perturbed files ($k_1 = 10, k_2 = 50$) for both methods. Note that there is very little difference between the two distributions as the classification error values only range from 0 to 0.99.

Perturbation	k_1	k_2	VICUS	Random
Mush1	2	20	1.3962	1.0091
Mush2	5	20	0.7891	0.6311
Mush3	10	10	0.6139	0.1370
Mush4	10	20	0.4252	0.1606
Mush5	10	50	0.4519	0.2268

Table 6.9: Average percentage of incorrectly classified instances for Mushroom perturbations, when tested against the original data set.

In summary, it would appear that even when a relatively large amount of noise is added in perturbing the Mushroom data set, there is still a high level of predictive accuracy achieved with both the *VICUS* and *Random* methods.

Wisconsin Breast Cancer Data Set

When testing the predictive accuracy for a classification model trained on the original WBC data set, we tested several values for k to find a good result using the k-fold cross-validation method. A summary of these relative errors for each value of k selected is shown in Table 6.10, note that the best predictive accuracy was provided by $k = 15$ with only 5.12% incorrectly classified instances.

The confusion matrix for the decision tree built on the whole original WBC data set is shown in Table 6.11. The matrix shows that 435+217 = 652 instances, or 95.46%, were correctly classified. Of the 457 instance that were classified as benign, 22 of these were incorrect, which would result in 22 patients being incorrectly given the all clear, when they in fact had a malignant cancer. Similarly, 9 records were incorrectly classified as malignant resulting in a false positive. Although these results are clearly not ideal, our focus

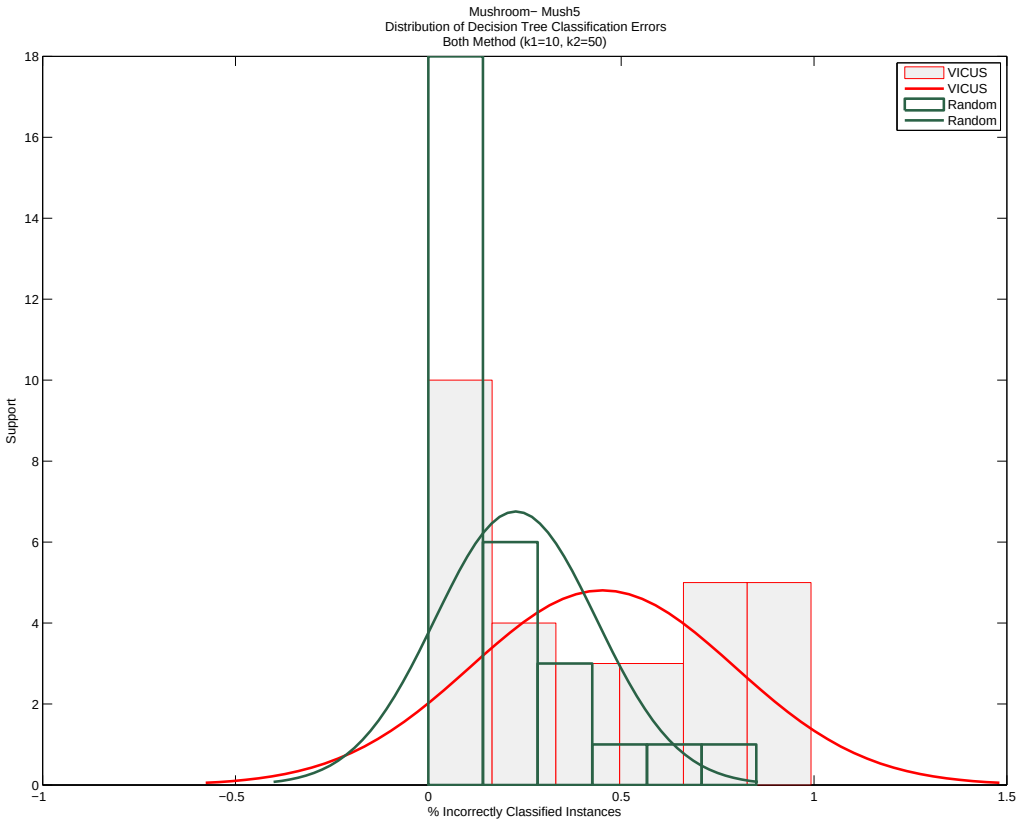

Figure 6.12: Classification error distribution comparison for Mushroom - Mush5 - 30 perturbation files.

k	% error
5	7.1742
10	6.1493
15	5.1245
20	5.8565
50	6.0029

Table 6.10: Selection of k-fold cross-validation parameter, showing k versus percentage of incorrectly classified instances over all folds.

here is not on building the most accurate decision tree, but rather comparing the trees built on the original and perturbed data sets. So when comparing the amount of instances misclassified on the perturbed data sets, we need to bear in mind that the classification error on the original data set is 4.54% when the whole data set is then again used for training, and 5.12% for a 15-fold cross validation.

	benign	malignant	total
benign	435	9	444
malignant	22	217	239
total	457	226	683

Table 6.11: Confusion matrix for original WBC data set

The decision tree built from the original Wisconsin Breast Cancer data set is shown in Figure 6.13. Attribute 2, *Uniformity of Cell Size*, is very prominent in the decision tree, with it accounting for two of the three inner nodes, and 86.8% of the classifications based simply on the value of this attribute. The other attribute which features in the decision tree is Attribute 6, *Bare Nuclei*, and it only appears in a logic rule alongside Attribute 2, never on its own.

As for the Mushroom data set, we ran the J48 decision tree builder on all of the perturbed data sets described at the beginning of Section 6.4. The percentage of incorrectly classified instances for all the decision trees trained on the files perturbed using the *VICUS* and *Random* methods, and then tested against the original WBC microdata file is shown in Table A.13 in Appendix A. A summary table of the average classification errors for

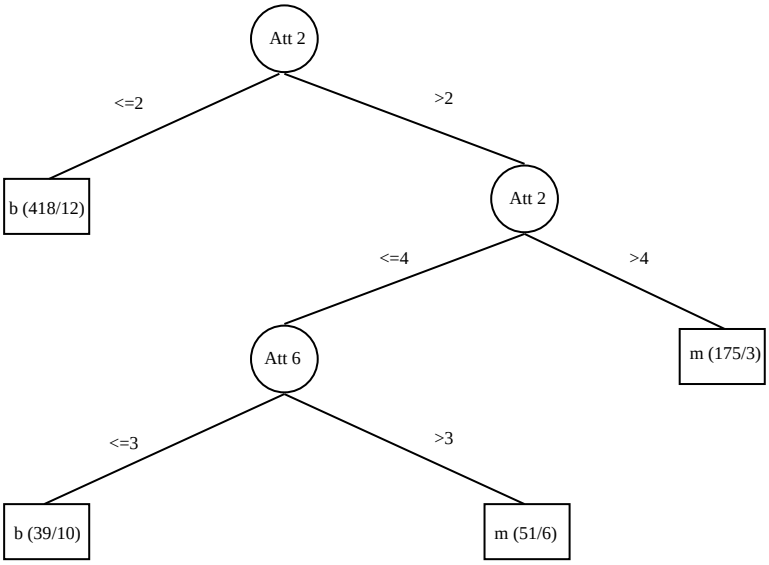

Figure 6.13: Decision tree for original Wisconsin Breast Cancer data set.

different values of the probability parameters k_1 and k_2 is shown in Table 6.12. A reminder that the percentage of incorrectly classified instances for the original WBC data set was 5.12% for the k-fold cross validation method. We can see that for relatively small amounts of noise, i.e. when ($k_1 \times k_2 = 100$), the *VICUS* and *Random* method have a predictive accuracy within approximately 1% of the result for the original WBC data set. A comparison of the relative predictive accuracy for when $k_1 \times k_2 = 100$ is shown in Figure 6.14. Note that for all three parameter choices, the difference between average classification errors is never more than 0.5%. Additionally, we note that for when $k_2 = 50$, that is, a value is 50 times more likely to change to another value in the same partition than one in a a different partition, the *VICUS* method performs better than the *Random* method. Similar results were also noted for when the decision trees were trained and tested on the same file, rather than being tested on the original data set (see Appendix A, Section A.5 for results). In combination, these results indicate that although the combined value of k_1 and k_2 is important in ensuring a high level of data quality, having a high value for k_2 is more important as it ensures that the when attribute values are perturbed they are only changed to another attribute value that they are very similar with. This is the underlying strength of our noise addition technique.

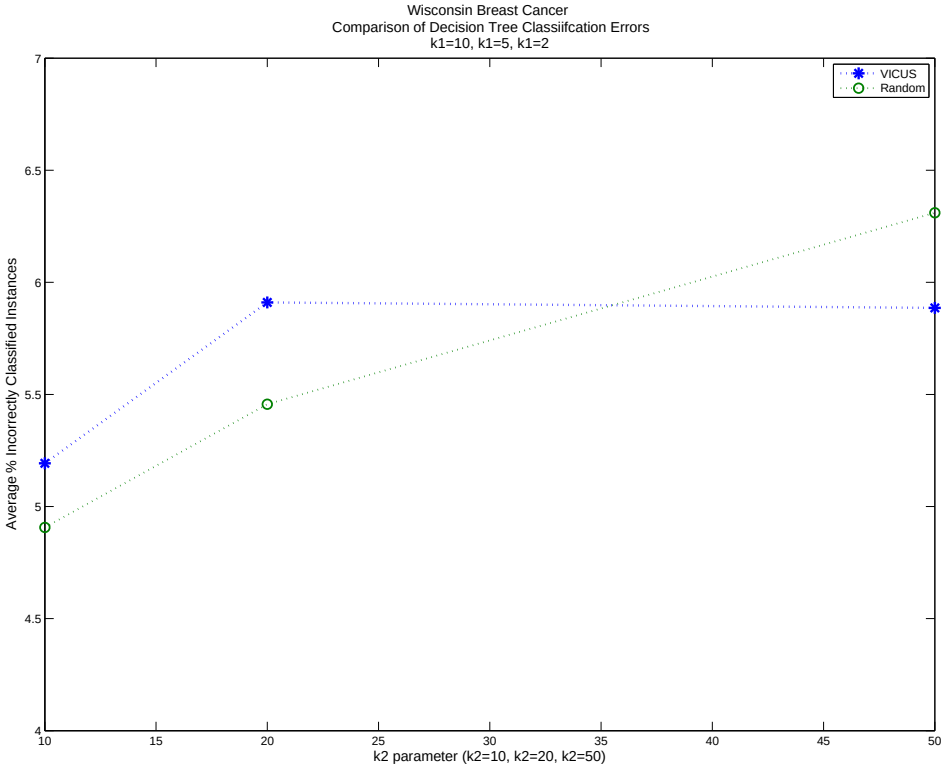

Figure 6.14: Decision tree classification error comparison for WBC perturbed files when $k_1 \times k_2 = 100$.

Perturbation	k_1	k_2	VICUS	Random
WBC1	2	2	16.0957	18.0768
WBC2	2	10	5.7447	7.9795
WBC3	2	50	5.8858	6.3104
WBC4	5	20	5.9102	5.4563
WBC5	10	10	5.1928	4.9067

Table 6.12: Average percentage of incorrectly classified instances for WBC perturbations.

We examine how the *VICUS* method compares to the *Random* when a relatively large amount of noise is added, i.e., for smaller values of k_1 and k_2. Recall that when we have a low value for k_1 a value is more likely to change to another value than stay the same. Figure 6.15 compares the average classification errors for both the *VICUS* and *Random* methods when the value of k_1 is fixed, and we vary the value of k_2. We note that the *VICUS* method performs better in terms of predictive accuracy for all three values of k_2. Figure 6.16 shows the comparative distribution of the classification errors over the 30 perturbed files for *WBC2* ($k_1 = 2, k_2 = 10$), and we can see that the distribution for the *VICUS* files is much narrower than for the *Random* method.

To this point we have focused on the predictive accuracy and not necessarily on the decision trees produced on the perturbed files. We did not conduct extensive experimental analysis to compare how the two methods performed in relation to the relative logic rules extracted from the decision trees, instead choosing to focus on the predictive accuracy. However, we will comment on one comparison set of decision trees as an illustrative example, selecting two decision trees generated from the *WBC3* ($k_1 = 2, k_2 = 50$) perturbation for comparison. Figure 6.17 shows one of the decision trees trained on a file perturbed via the *VICUS* noise addition method. Like the tree produced on the original WBC data set (see Figure 6.13), this tree also features Attribute 2 as the root element, and indeed has the same rule on the branches of this node. A similar number of records can be classified as benign, based on the value of Attribute 2 being ≤ 2, for the original decision tree and the one from Figure 6.17. In the original tree there were 418 records classified as benign based on this rule, with only 12 of those being incor-

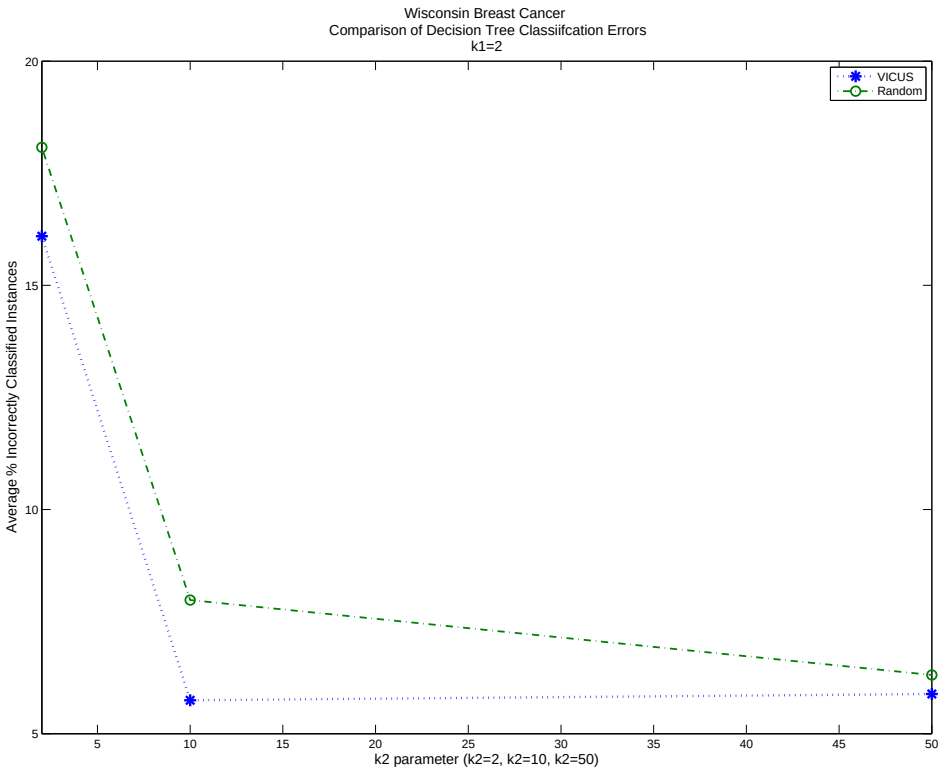

Figure 6.15: Decision tree classification error comparison for WBC perturbed files when $k_2 = 2$.

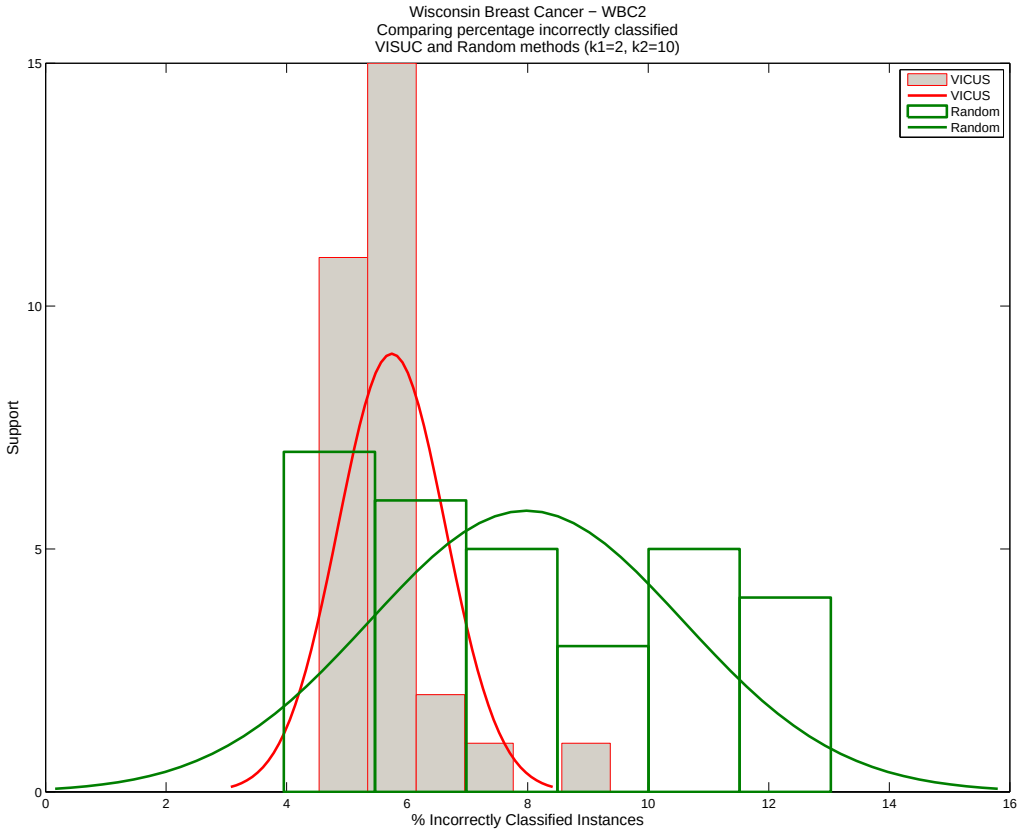

Figure 6.16: Classification error distribution comparison for WBC2 perturbation files.

rectly classified, and for the tree from the file perturbed using the *VICUS* method there are 401 records classified using the same rule, with only 14 incorrectly classified. Although the rest of the tree is quite different to the original, there is clearly some degree of similarity between these two trees.

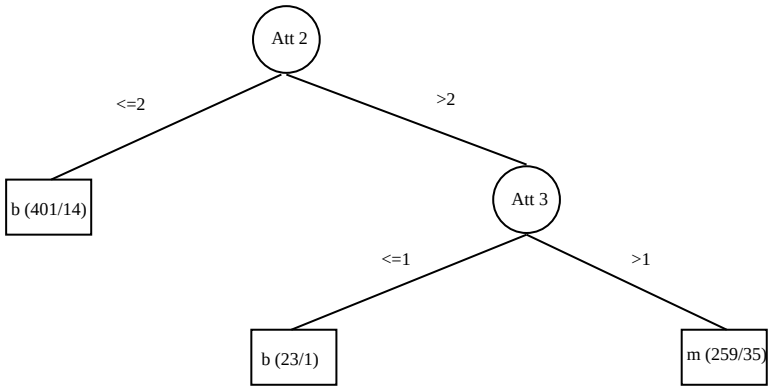

Figure 6.17: Sample decision tree for *VICUS* perturbed Wisconsin Breast Cancer data set. ($k_1 = 2$, $k_2 = 50$)

When we compare the decision tree trained from the original data set with one trained on a file perturbed using the *Random* method, as shown in Figure 6.18, there appears to be much less in common between the two trees. The second tree is also considerably more complex, with 10 rules producing 6 leaf nodes, versus 6 rules producing 4 leaf nodes in the original. Although the same rule for Attribute 2, discussed above, appears in the decision tree, it is not as prominent and no records are classified on the basis of this rule alone.

To summarise, the results in this section have shown that when a good choice of probability parameters is selected to perturb the files, our *VICUS* method performs better than the so-called *Random* method. For the Mushroom data set, the predictive accuracy for all perturbed files was over 98%, and for most parameter selections as within 0.5% of the original file. For the WBC data set when relatively low levels of noise were added, e.g. $k_1 \times k_2 = 100$, the predictive accuracy for the *VICUS* and *Random* methods were both within around 1% of the result for the original file. When larger amounts of noise were added, we noted that the *VICUS* method performed slightly better than the *Random* method. In the next subsection we will examine how well our method performs when tested using a different measure

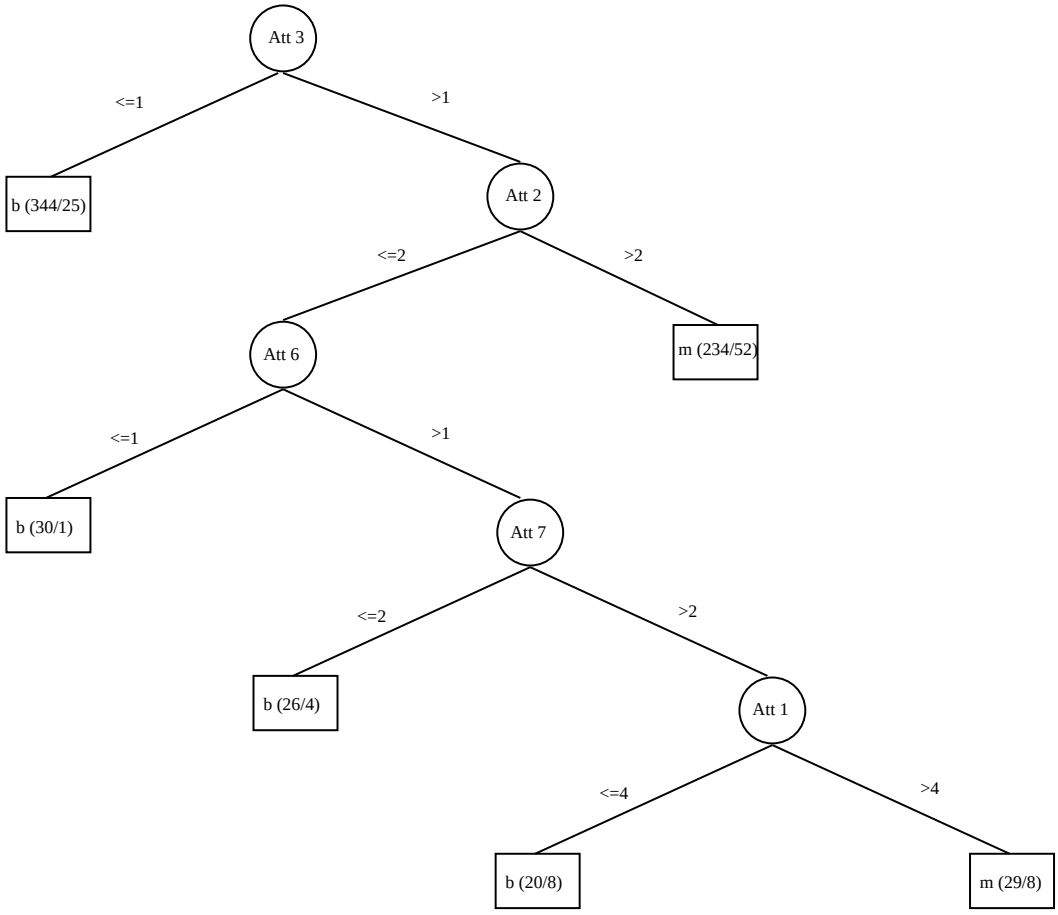

Figure 6.18: Sample decision tree for *Random* perturbed Wisconsin Breast Cancer data set. ($k_1 = 2$, $k_2 = 50$)

of data quality.

Chi-Square Test Results

We used the SPSS Statistical Software package to analyse the Chi-square statistics of our two experimental data sets. For each attribute pair we calculated the Pearson's Chi-Square statistic and Likelihood Ratio Chi-Square statistic on the original data set, 30 files perturbed using *VICUS* method and 30 file perturbed via the *Random* method. We compared both the chi-square statistic value and associated p-value for each. We first want to see how our *VICUS* method performed in terms of how far the χ^2 values were from those on the original file and the Randomly perturbed files. We next wanted to verify if there was a change in the outcome of the null hypothesis for the files perturbed with the *VICUS* method.

Mushroom Data Set

We chose to look at Attribute 4 (*Cap Colour*) against the other attributes, since this attribute showed to be the most sensitive in terms of security when we calculated the entropy for the user knowing one attribute value (refer to Section 6.4.2). We also selected only one perturbation in terms of probability parameters, selecting the values of $k_1 = 5$ and $k_2 = 20$ for these parameters. This combination was chosen since this set of parameters gave middle of the range results on both entropy and decision tree classification error.

Of the 21 attribute combinations, there were 7 attribute pairs that satisfied the large sample requirement on the original data set. That is, these attributes had over 80% of cells in the contingency table with expected counts larger than 5, and all cells had an expected count higher than 1. These attribute pairs selected for further analysis were as follows.

- *Class* (Attribute 1) and *Cap Colour* (Attribute 4)
- *Bruises* (Attribute 5) and *Cap Colour* (Attribute 4)
- *Gill Spacing* (Attribute 8) and *Cap Colour* (Attribute 4)

- *Gill Size* (Attribute 9) and *Cap Colour* (Attribute 4)
- *Stalk Shape* (Attribute 11) and *Cap Colour* (Attribute 4)
- *Stalk Root* (Attribute 12) and *Cap Colour* (Attribute 4)
- *Stalk Surface Below Ring* (Attribute 14) and *Cap Colour* (Attribute 4)

Attribute Pairing	Original		VICUS		Random		Degrees of Freedom
	χ^2	p-value	χ^2	p-value	χ^2	p-value	
1 & 4	1297.28	6.45×10^{-276}	821.13	5.11×10^{-173}	657.48	1.01×10^{-137}	7
5 & 4	982.36	7.81×10^{-208}	766.42	3.27×10^{-161}	562.44	2.97×10^{-117}	7
8 & 4	617.27	4.64×10^{-129}	426.03	6.20×106^{-88}	298.62	1.19×10^{-60}	7
9 & 4	756.27	5.06×10^{-159}	435.99	4.73×10^{-90}	286.97	3.66×10^{-58}	7
11 & 4	1832.44	0	1048.97	3.15×10^{-222}	1032.52	1.13×10^{-218}	7
12 & 4	2427.35	0	720.65	5.72×10^{-139}	633.45	1.46×10^{-120}	21
14 & 4	2711.80	0	1626.64	0	848.46	4.74×10^{-166}	21

Table 6.13: χ^2 and associated p-value summary for Mushroom data set

Figure 6.19 shows the distribution of the χ^2 statistic values for the 30 files perturbed via the *VICUS* method, the attribute pair shown here is *Cap Colour* and *Stalk Root Below Ring* (Attributes 4 and 14). Similarly, Figure 6.20 shows the distribution for 30 files perturbed via the *Random* method on the same attribute combination. Finally, Figure 6.21 compares these two distributions, and shows how far away they are from the χ^2 statistic for the original file, which had the value of 2711.8. On all attribute pairs examined, our *VICUS* method produced χ^2 values closer to the original than the those of the random method. The average Pearson's χ^2 results are summarised in Table 6.13, and the Likelihood Ratio χ^2_{LR} results are shown in Table 6.14. Note that entries that show zero for the p-value are rounded to this value by SPSS, since the probability, although extremely low, will never be zero. Assuming a significance value $\alpha = 0.05$, all of the cases (for both the original and perturbed files) would result in the null hypothesis being rejected, and hence there is a statistical relationship between the attribute pairs examined.

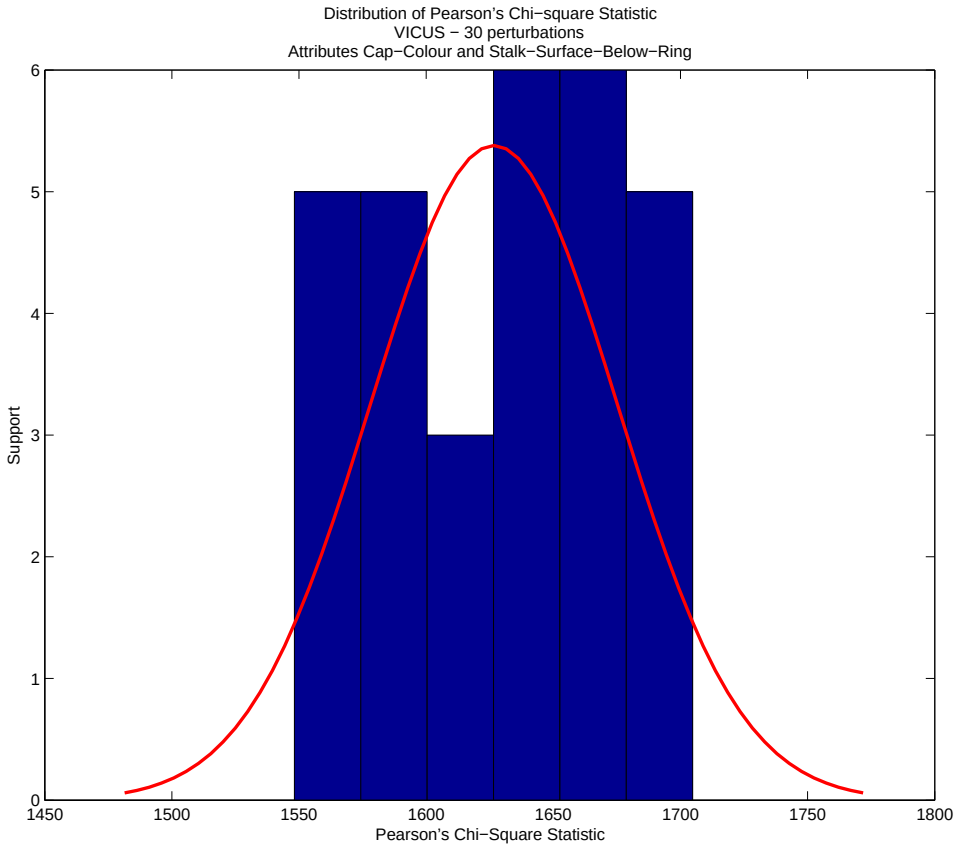

Figure 6.19: Distribution of chi-square statistic for 30 files perturbed via *VICUS* method. (Attribute 4 and 14 Mushroom data set).

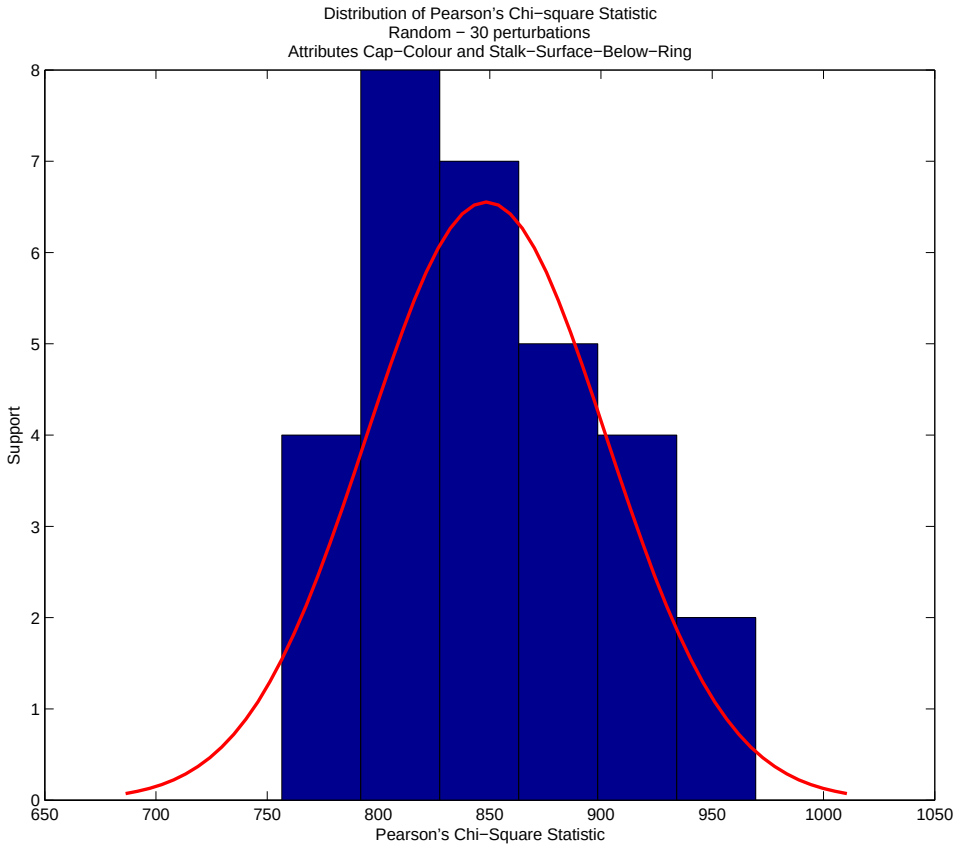

Figure 6.20: Distribution of chi-square statistic for 30 files perturbed via *Random* method. (Attribute 4 and 14 Mushroom data set).

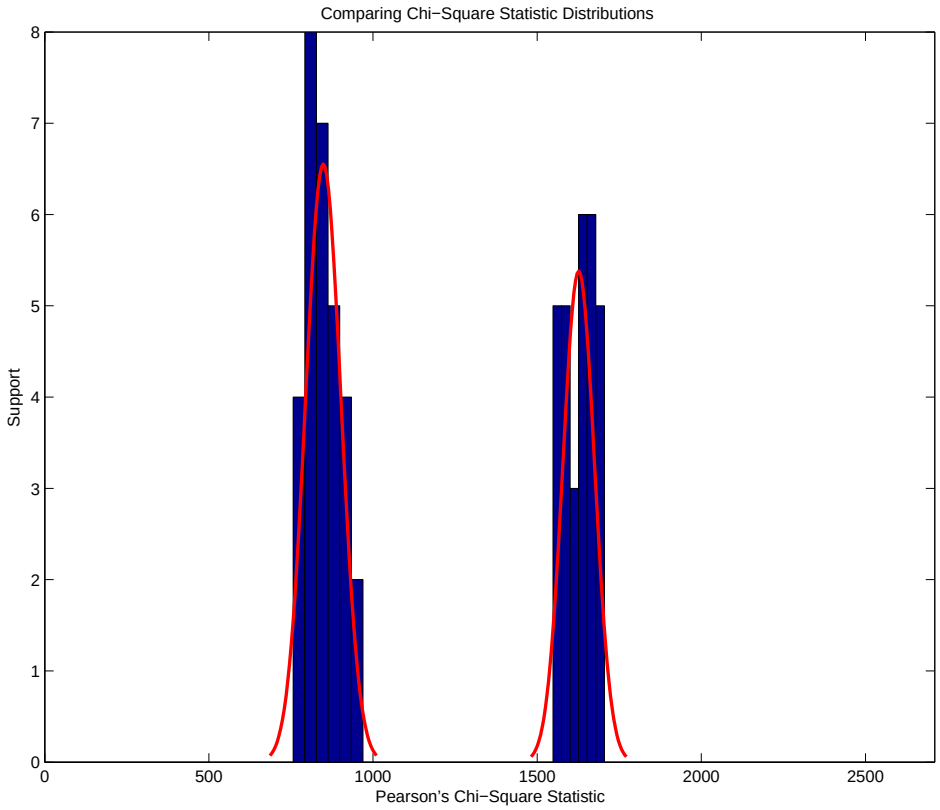

Figure 6.21: Distribution of chi-square statistic for 30 files perturbed via *VICUS* and *Random* methods. (Attribute 4 and 14 Mushroom data set).

Attribute Pairing	Original		VICUS		Random		Degrees of Freedom
	χ^2_{LR}	p-value	χ^2_{LR}	p-value	χ^2_{LR}	p-value	
1 & 4	1525.36	0	911.22	1.81×10^{-192}	709.61	5.85×10^{-149}	7
5 & 4	1177.56	5.03×10^{-250}	861.12	1.19×10^{-181}	601.34	1.25×10^{-125}	7
8 & 4	717.32	1.27×10^{-150}	463.12	6.76×10^{-96}	306.48	2.50×10^{-62}	7
9 & 4	686.62	5.28×10^{-144}	436.16	4.16×10^{-90}	274.24	1.90×10^{-55}	7
11 & 4	2253.66	0	1179.73	1.71×10^{-250}	1131.77	3.98×10^{-240}	7
12 & 4	2880.86	0	740.67	3.34×10^{-143}	633.98	1.13×10^{-120}	21
14 & 4	3304.05	0	1832.83	0	857.04	7.17×10^{-168}	21

Table 6.14: χ^2_{LR} and associated p-value summary for Mushroom data set

Wisconsin Breast Cancer Data Set

For this data set we chose to look at Attribute 3 (*Uniformity of Cell Shape*) and Attribute 5 (*Single Epithelial Cell Size*) in combination with other attributes. The reason being that these two attributes were at the extremes when it came to entropy results, with Attribute 3 having the lowest entropy and Attribute 5 having the highest, when the user knows one attribute. The probability parameters selected for the perturbations using the *VICUS* and *Random* method were $k_1 = 2$ and $k_2 = 50$ since this combination gave good results for both the entropy and decision tree classification tasks.

Of all the attribute combinations permissable with Attribute 3 and Attribute 5, there were only two combination that satisfied the large sample requirement. These two combinations were as follows.

- *Uniformity of Cell Shape* (Attribute 3) and *Class* (Attribute 10)
- *Single Epithelial Cell Size* (Attribute 5) and Class (Attribute 10)

The summary results for the Pearson's Chi-square and Likelihood Chi-square statistics are shown in Table 6.15 and Table 6.16 respectively. These are the average χ^2 results over 30 files perturbed via the *VICUS* and *Random* methods. As with the Mushroom data set, for the attribute combinations examined, the *VICUS* method performed slightly better than the *Random* method on average. Although the p-values were significantly higher on the perturbed files for attribute combination 5 and 10, they were still below the

alpha value of 0.05 indicating that the statistical relationship between these two attributes was at least in part preserved.

Attribute Pairing	Original		VICUS		Random		Degrees of Freedom
	χ^2	p-value	χ^2	p-value	χ^2	p-value	
3 & 10	523.07	6.58×10^{-107}	308.10	4.98×10^{-61}	250.30	8.63×10^{-49}	9
5 & 10	447.86	8.22×10^{-91}	35.78	4.34×10^{-05}	28.76	0.00071	9

Table 6.15: χ^2 and associated p-value summary for Wisconsin Breast Cancer data set

Attribute Pairing	Original		VICUS		Random		Degrees of Freedom
	χ^2_{LR}	p-value	χ^2_{LR}	p-value	χ^2_{LR}	p-value	
3 & 10	640.79	3.65×10^{-132}	372.67	9.23×10^{-75}	275.24	4.60×10^{-54}	9
5 & 10	506.02	2.96×10^{-103}	42.02	3.26×10^{-06}	29.91	0.00045	9

Table 6.16: χ^2_{LR} and associated p-value summary for Wisconsin Breast Cancer data set

6.5 Conclusion

We have outlined a new noise addition technique, *VICUS*, for application on categorical values. The technique can also be applied to numerical attributes by treating the discrete values as categories. Initial results indicate that our method performs well in both the areas of security and data quality. Analysis indicates that a low value for k_1 and high value for k_2 transition probability parameters leads to improved performance in terms of both data quality and security of *VICUS* over the *Random* method. Setting the product $(k_1 \times k_2)$ of these parameters to a value of 100 or higher, while also ensuring that k_1 is low and k_2 is high appears to give the best balance between the conflicting goals of security and data quality.

Chapter 7

Conclusion

Success is not a place at which one arrives but rather the spirit with which one undertakes and continues the journey.

–Alex Noble

In this chapter we conclude the thesis with a discussion of the main contributions, and present ideas for the direction of future work arising from the thesis.

7.1 Summary

There is the potential for great benefit to be gained from the analysis of genetic data warehouses. However, due to the highly sensitive nature of genetic information there is also a great risk for harm to individuals via breaches of privacy and discrimination. This thesis has provided a holistic approach to solving this problem, and provided practical tools to help ensure that a balance between research outcomes and the rights of the individual can be best achieved. As researchers, we have not only an ethical, but also a legal obligation to ensure that all reasonable effort is made to ensure the privacy of the individuals whose information we are dealing with in genetic databases.

One way in which the trust between data managers and the public can be nurtured is to ensure that the appropriate Statistical Disclosure Control (SDC) measures are employed. By eliciting a higher level of trust amongst all parties involved we may be able to reduce the dependence on so-called 'hard security' measures, and instead move more towards an environment of trust management systems. By providing a quantifiable measure of trust in the context of statistical databases, we aim to progress things further in this direction.

While it is important to investigate avenues for reducing the reliance on tradition SDC methods, they still have an important role to play in helping to solve the SDC problem. One of the challenges that genetic databases pose in the context of these methods is their relatively high percentage of categorical attributes. The application of SDC techniques to categorical values can be less straightforward than for numerical attributes since we do not have an easy way to decide similarity, or ordering for these values. One existing technique that has been successfully applied to categorical values is the Post RAndomisation Method (PRAM) developed by a group of researchers at Statistics Netherlands [54]. By incorporating this SDC method into our Privacy Protection Framework, we aim to better protect genetic databases without the need to modify the not yet fully understood genetic information.

One of the main components of our Privacy Protection Framework is the clustering of categorical values to help in the design of the transition probability matrices. We have provided a similarity measure that captures the direct and transitive similarity between values in an attribute. By incorporating our similarity measure into the *VICUS* clustering technique we make the task of assigning probabilities to the transition matrix more straightforward, and we can move a step closer to providing a complete framework for the protection of genetic data warehouses.

7.2 Contributions

Recalling our first research question from Section 1.2.

What role does trust play in the relationships between the stakeholders in a genetic statistical data warehouse system and how best can it be modelled?

We have been able to go a long way towards fully answering this question, not only in the context of genetic data warehouses but also for statistical data warehouse in general. In Chapter 2 we extensively examined the trust relationship between the three key stakeholders in a statistical data warehouse system, namely the Data Source, Data Manager and Data User. By modeling these trust relationships we were able to see that when low levels of trust exist between the system stakeholders, the overall functioning of the system is placed in jeopardy. We next developed a model of trust in the context of statistical data warehouse systems, with many of the elements of this model being quantifiable. By providing data managers with a quantifiable model of trust in the SDC context we are providing them with insight into the workings of their system, and allowing them to provide users with the appropriate level of SDC methods based on their varying levels of trust. Indeed, we have outlined a comprehensive Privacy Protection Framework that when applied by data managers will increase the likelihood of positive collaboration by all system stakeholders. The overall objective of reducing the reliance on traditional SDC security methods can become a reality with the application of this framework.

Although we have been able to provide a quantifiable measure for the Cooperation Threshold, we do acknowledge that measuring certain components of the trust in a given situation is difficult due to their subjective nature. In particular, the propensity to trust and distrust constructs have proved challenging to quantify, and this is an immediate focus of our future research in this area.

Despite the promising developments in the application of trust management to the protection of genetic statistical data warehouse systems, there is still a need in many circumstance to employ more traditional SDC methods. This is of particular relevance when low levels of trust between the system stakeholders is exhibited. This leads to the importance of our second research question from Section 1.2.

How can we successfully apply statistical disclosure control measures to a genetic statistical data warehouse systems when a large proportion of the data is categorical in nature?

In addressing this research question we have first presented a framework for the protection of genetic databases based around the clustering of categorical values to provide a measure of similarity. Importantly this framework is also applicable to any statistical database which contains categorical attributes.

Based on the PRAM [54] noise addition technique, the framework proposes the application of categorical clustering techniques to help decide on the probability transition matrices. On investigation of existing categorical clustering techniques we discovered that none were successfully able to capture the notion of transitive similarity that we observed in real data sets. With this in mind we developed a new similarity measure for categorical attribute values. We were also able to incorporate flexibility into our similarity measure via the ability to adjust the relative levels of S-Prime and S-Secundum similarity. After considerable experimental analysis we observed two important corollaries of our similarity measure exploration.

1. Not all numerical attributes appear to behave in a numerical manner, which we termed the 'numerical attribute phenomenon', This was a somewhat surprising discovery which implies that current noise addition techniques for numerical attributes are not suitable for application to the attributes that exhibit the numerical attribute phenomenon. For these attributes alternative noise addition techniques should be applied, such as our *VICUS* technique.

2. Another interesting application of our similarity measure which arose during its development is the potential to provide an ordering for categorical attributes. Importantly, this could lead to the application of numerical noise addition techniques to categorical attributes directly. This would be beneficial as these techniques have been relatively well studied and are computationally less expensive.

By incorporating our similarity measure for categorical attributes into a new clustering technique, *VICUS*, we have been able to provide a straightforward mechanism for deciding the transition probability matrices used in the application of PRAM. The technique can also be applied to numerical attributes directly by treating their discrete values as categories. After extensive experimental analysis our technique appears to provide a good balance between security and data quality. This being the overall goal of SDC techniques in general, we feel confident that we have gone a long way towards solving the difficult problem of privacy protection in genetic data warehouses. Not only are our methods also applicable for any statistical database containing categorical attributes, but we have also been able to provide the data manager with a degree of flexibility in their application through the parameters incorporated into our *VICUS* noise addition technique.

7.3 Future Work

One of the challenges we faced when developing our trust model (Chapter 2) was how to quantify situational trust. By gaining a better insight into the distinction between the subjective and objective components of situational trust, we hope to make further steps towards quantifying this elusive concept. One way in which this might be achieved is through modelling the propensity to trust and distrust density functions.

From our Privacy Protection Framework for genetic databases in Chapter 4, we would like to further investigate the use of sub-attributes in the application of a similarity measure. Another aspect of the framework which warrants further research is the incorporation of integrity rules into the decision of when to compound attributes in the PRAM SDC method.

In Chapter 5 we discussed the numerical attribute phenomenon and the impact that this could have on the application of SDC methods for numerical attributes. An area of future research in this direction would be to study how treating numerical attributes as categorical and perturbing them via our *VICUS* method effects to overall quality of the perturbed file. Another

possible application of our similarity measure which was discussed briefly in Chapter 5 is using the relative similarities in order to provide a distance measure, or relative ordering, for categorical values.

Appendix A

Experiments - Detailed

This Appendix contains details of the experiments outlined in Chapter 5 and Chapter 6.

A.1 S-Secundum Algorithms

Input: Graph G, Threshold T
Output: S'' values for G

initialise S'' matrix to 0;
for *each attribute* $x \in G$ **do**
  get the list of attribute values val_x;
  `/* Loop over all pairs of values for the attribute */`
  for *each value* $i \in val_x$ **do**
    for *each value* $j \in val_x$ **do**
      $currIndex \leftarrow 0$;
      $neighList[] \leftarrow$ -1 for all vertices in G;
      `/* Loop over all attributes in` G`, excluding` x `*/`
      for *each attribute* $y \in G \setminus x$ **do**
        get the list of attribute values val_y;
        `/* Loop over all pairs of values in` y `*/`
        for *each value* $c \in val_y$ **do**
          for *each value* $d \in val_y$ **do**
            if *there are egdes* $(\{c,i\}$ *and* $\{d,j\}) \vee (\{c,j\}$ *and* $\{d,i\})$ *in* G **then**
              $neighList \leftarrow mergeNeighbours()$;
            end
          end
        end
      end
      $S''_{ij} \leftarrow S'_{ij}$ calculated on $neighList$
    end
  end
end
return S'' matrix;

Algorithm A.1: Calculating S'' values for graph G, as implemented

```
Input: Graph G, Threshold, i, j, c, d,neighList[]
Output: neighList[]

Function mergeNeighbours();
if S'_cd > Threshold then
    if neighList[c] == −1 ∧ neighList[d] == −1 then
        neighList[c] ← currIndex;
        neighList[d] ← currIndex;
        currIndex++;
    end
    else if neighList[c]! = −1 ∧ neighList[d]! = −1 then
        if neighList[c]! = neighList[d] then
            neighList[d] ← neighList[c];
            for z ← to n do
                if neighList[z] == neighList[d] then
                    neighList[z] ← neighList[c];
                end
            end
        end
    end
    else if neighList[c]! = −1 ∧ neighList[d] == −1 then
        neighList[d] ← neighList[c];
    end
    else if neighList[c] == −1 ∧ neighList[d]! = −1 then
        neighList[c] ← neighList[d];
    end
end
```

Algorithm A.2: Calculating neighbour pairs list, used in Algorithm A.1

A.2 Motivating Example

Table A.1 shows the S'_{ij} values for attributes 2-4 in the motivating example. The similarities within an attribute are shown in bold in the table. Vertex labels 7-9 are for attribute *Program*, labels 10-13 are for attribute *Course* and labels 14-17 are for attribute *Tutor*. For exact matching between labels

and attribute values please refer to Table 5.2 in Section 5.1. Table A.2 gives the S''_{ij} values for the Motivating example graph, with a S''_{ij} Threshold of 0.4. Table A.3 shows the S_{ij} similarity values for the Motivating Example with weightings of $c_1 = 0.6$ and $c_2 = 0.4$.

S'_{ij}	7	8	9	10	11	12	13	14	15	16	17
7	**1.000**	**0.000**	**0.169**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
8	**0.000**	**1.000**	**0.463**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
9	**0.169**	**0.463**	**1.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
10	0.000	0.000	0.000	**1.000**	**0.471**	**0.000**	**0.000**	0.000	0.000	0.000	0.000
11	0.000	0.000	0.000	**0.471**	**1.000**	**0.333**	**0.183**	0.000	0.000	0.000	0.000
12	0.000	0.000	0.000	**0.000**	**0.333**	**1.000**	**0.548**	0.000	0.000	0.000	0.000
13	0.000	0.000	0.000	**0.000**	**0.183**	**0.548**	**1.000**	0.000	0.000	0.000	0.000
14	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**1.000**	**0.000**	**0.000**	**0.258**
15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.000**	**1.000**	**0.436**	**0.000**
16	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.000**	**0.436**	**1.000**	**0.436**
17	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.258**	**0.000**	**0.436**	**1.000**

Table A.1: S'_{ij} values for attributes 2-4 of Motivating Example

S''_{ij}	7	8	9	10	11	12	13	14	15	16	17
7	**1.000**	**0.000**	**0.239**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
8	**0.000**	**1.000**	**0.772**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
9	**0.239**	**0.772**	**1.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
10	0.000	0.000	0.000	**1.000**	**0.707**	**0.000**	**0.000**	0.000	0.000	0.000	0.000
11	0.000	0.000	0.000	**0.707**	**1.000**	**0.569**	**0.623**	0.000	0.000	0.000	0.000
12	0.000	0.000	0.000	**0.000**	**0.569**	**1.000**	**0.806**	0.000	0.000	0.000	0.000
13	0.000	0.000	0.000	**0.000**	**0.623**	**0.806**	**1.000**	0.000	0.000	0.000	0.000
14	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**1.000**	**0.000**	**0.000**	**0.365**
15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.000**	**1.000**	**0.617**	**0.333**
16	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.000**	**0.617**	**1.000**	**0.617**
17	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.365**	**0.333**	**0.617**	**1.000**

Table A.2: S''_{ij} values for attributes 2-4 of Motivating Example, T=0.4

A.3 Mushroom Data Set

Table A.4 provides all of the attribute values and their corresponding vertex labels for the Mushroom data set [5].

The bottom two graphs in Figure A.1 show how the relative level of error changes based on a change in either the k_1 or k_2 parameters, when

S_{ij}	7	8	9	10	11	12	13	14	15	16	17
7	**1.000**	**0.000**	**0.197**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
8	**0.000**	**1.000**	**0.586**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
9	**0.197**	**0.586**	**1.000**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
10	0.000	0.000	0.000	**1.000**	**0.566**	**0.000**	**0.000**	0.000	0.000	0.000	0.000
11	0.000	0.000	0.000	**0.566**	**1.000**	**0.428**	**0.359**	0.000	0.000	0.000	0.000
12	0.000	0.000	0.000	**0.000**	**0.428**	**1.000**	**0.651**	0.000	0.000	0.000	0.000
13	0.000	0.000	0.000	**0.000**	**0.359**	**0.651**	**1.000**	0.000	0.000	0.000	0.000
14	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**1.000**	**0.000**	**0.000**	**0.301**
15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.000**	**1.000**	**0.509**	**0.133**
16	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.000**	**0.509**	**1.000**	**0.509**
17	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.301**	**0.133**	**0.509**	**1.000**

Table A.3: S_{ij} values for attributes 2-4 of Motivating Example, T=0.4, $c_1 = 0.6$ and $c_2 = 0.4$

the same data file is used for training and testing. The bottom left graph in Figure A.1 shows how the relative level of classification error changes when we fix the k_2 parameter at 20 and change the k_1 parameter. This means that an attribute value is always 20 times more likely to stay the same than change to another value in the same partition. The three values chosen for the k_1 parameter were 2, 5 and 10. Note that due to the relatively high value for k_2, at 20, the difference in relative percentage error between the *VICUS* and *Random* methods stays at around 80% for all three choices for k_1. That is, the average classification error for the 30 files perturbed via the *VICUS* method was around 20% lower than the average error for the 30 files perturbed using the *Random* method.

The second graph for the Mushroom data set in Figure A.1, in the bottom right hand corner, shows how the relative classification errors behave when the value of k_1 is fixed and k_2 is varied. The value of k_1 has been fixed at 10, and the three values of k_2 shown are 10, 20 and 50. Interestingly, the higher the value for k_2, the relative distance between the average classification errors for the *VICUS* and *Random* methods grows. For instance, for $k_1 = 10$ and $k_2 = 10$ the average percentage of instances misclassified via the *VICUS* method is 3.627%, which is 14.5% lower than the result for the *Random* method at 4.243%. By contract, the *VICUS* error for $k_1 = 10$ and $k_2 = 50$ is 39.2% lower than that for the *Random* method. This would seem to indicate that there is an improvement in data quality when choosing

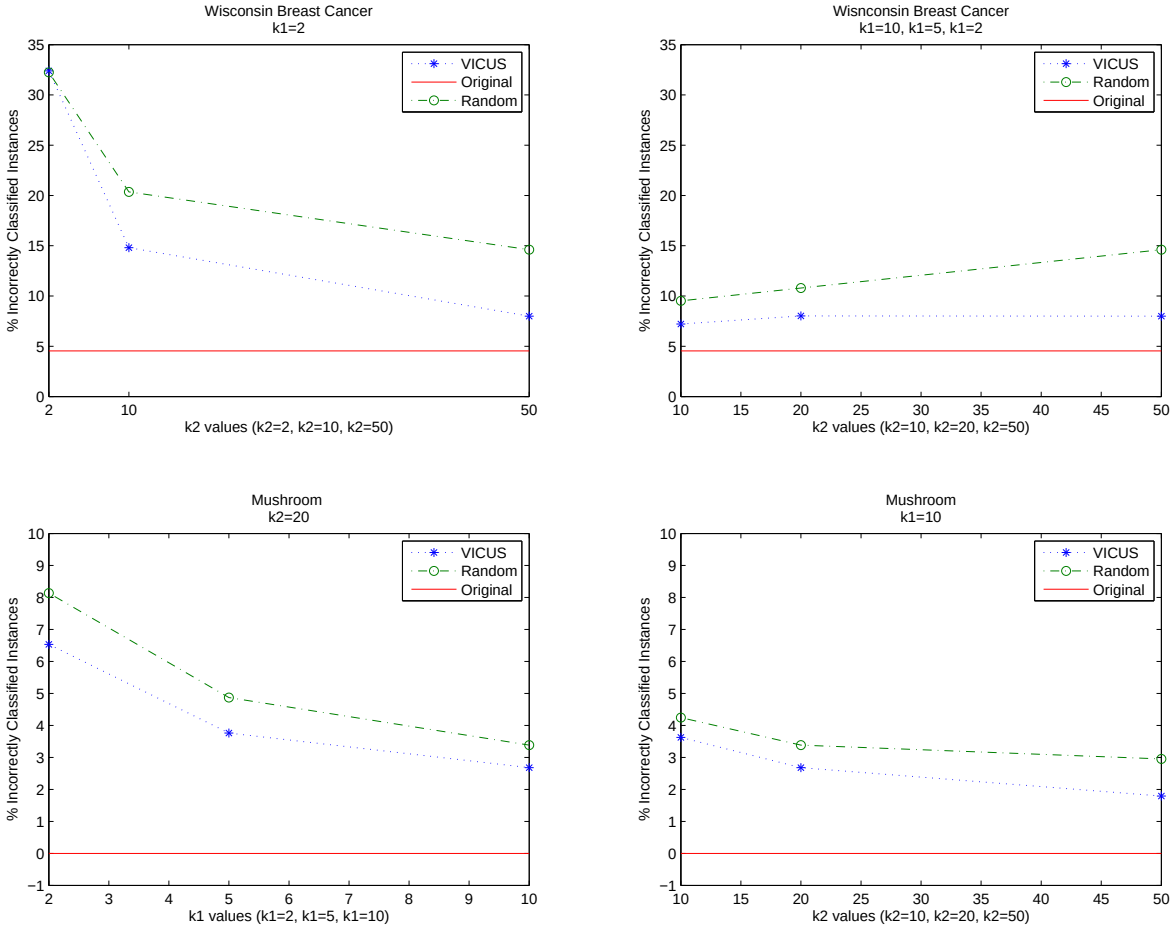

Figure A.1: Decision tree classification error comparison for WBC and Mushroom data sets.

the *VICUS* method over the *Random* method. The lower the probability that an attribute value will change to one in a different partition, that is one deemed less similar, the smaller the percentage of misclassified records. This is relative to the classification error when the probability of an attribute value changing to another value is equal for all other values.

Attribute	*Value*	*Vertex Number*	*Attribute*	*Value*	*Vertex Number*
Class Attribute	poisonous (p)	1	Stalk Surface	silky (k)	51
	edible (e)	2	Above Ring	smooth (s)	52
Cap Shape	flat (f)	3		scaly (y)	53
	convex (x)	4		fibrous (f)	54
	bell (b)	5	Stalk Surface	silky (k)	55
	knobbed (k)	6	Below Ring	smooth (s)	56
	conical (c)	7		scaly (y)	57
	sunken (s)	8		fibrous (f)	58
Cap Surface	fibrous (f)	9	Stalk Colour	pink (p)	59
	scaly (y)	10	Above Ring	brown (n)	60
	smooth (s)	11		buff (b)	61
	grooves (g)	12		white (w)	62
Cap Colour	grey (g)	13		grey (g)	63
	yellow (y)	14		cinnamon (c)	64
	buff (b)	15		yellow (y)	65
	pink (p)	16	Stalk Colour	buff (b)	66
	white (w)	17	Below Ring	brown (n)	67
	brown (n)	18		pink (p)	68
	cinnamon (c)	19		white (w)	69
	red (e)	20		grey (g)	70
Bruises	no (f)	21		cinnamon (c)	71
	bruises (t)	22		yellow (y)	72
Odour	foul (f)	23	Veil Colour	white (w)	73
	none (n)	24		yellow (y)	74
	creosote (c)	25	Ring Number	one (o)	75
	almond (a)	26		two (t)	76
	anise (l)	27		none (n)	77
	musty (m)	28	Ring Type	large (l)	78
	pungent (p)	29		pendant (p)	79
Gill Attachment	free (f)	30		evanescent (e)	80
	attached (a)	31		none (n)	81
Gill Spacing	close (c)	32	Spore Print	chocolate (h)	82
	crowded (w)	33	Colour	green (r)	83
Gill Size	broad (b)	34		brown (n)	84
	narrow (n)	35		black (k)	85
Gill Colour	grey (g)	36		white (w)	86
	chocolate (h)	37		purple (u)	87
	brown (n)	38	Population	several (v)	88
	pink (p)	39		solitary (y)	89
	green (r)	40		scattered (s)	90
	purple (u)	41		clustered (c)	91
	white (w)	42		numerous (n)	92
	black (k)	43		abundant (a)	93
	yellow (y)	44	Habitat	paths (p)	94
Stalk Shape	enlarging (e)	45		woods (d)	95
	tapering (t)	46		grasses (g)	96
Stalk Root	bulbous (b)	47		meadows (m)	97
	club (c)	48		leaves (l)	98
	equal (e)	49		urban (u)	99
	rooted (r)	50			

Table A.4: Vertex labeling for Mushroom data set

File No.	Mush1		Mush2		Mush3		Mush4		Mush5	
	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random
1	11.9715	12.0080	11.7502	11.7321	11.6130	11.5761	11.6130	11.5761	11.5578	11.5289
2	11.9716	12.0075	11.7499	11.7311	11.6124	11.5778	11.6124	11.5778	11.5571	11.5276
3	11.9721	12.0076	11.7504	11.7322	11.6137	11.5778	11.6137	11.5778	11.5567	11.5279
4	11.9720	12.0082	11.7492	11.7307	11.6132	11.5762	11.6132	11.5762	11.5566	11.5280
5	11.9714	12.0076	11.7495	11.7310	11.6146	11.5758	11.6146	11.5758	11.5571	11.5284
6	11.9724	12.0062	11.7504	11.7322	11.6139	11.5770	11.6139	11.5770	11.5577	11.5293
7	11.9726	12.0074	11.7506	11.7318	11.6144	11.5770	11.6144	11.5770	11.5571	11.5277
8	11.9719	12.0073	11.7502	11.7326	11.6141	11.5776	11.6141	11.5776	11.5565	11.5292
9	11.9718	12.0081	11.7505	11.7314	11.6142	11.5757	11.6142	11.5757	11.5560	11.5289
10	11.9721	12.0069	11.7507	11.7302	11.6146	11.5781	11.6146	11.5781	11.5557	11.5280
11	11.9716	12.0072	11.7498	11.7319	11.6128	11.5759	11.6127	11.5758	11.5565	11.5281
12	11.9727	12.0072	11.7505	11.7328	11.6146	11.5769	11.6146	11.5769	11.5564	11.5282
13	11.9713	12.0073	11.7496	11.7316	11.6143	11.5777	11.6142	11.5777	11.5559	11.5275
14	11.9725	12.0072	11.7498	11.7304	11.6139	11.5777	11.6139	11.5777	11.5554	11.5290
15	11.9707	12.0079	11.7503	11.7322	11.6148	11.5773	11.6148	11.5773	11.5573	11.5290
16	11.9723	12.0075	11.7502	11.7316	11.6140	11.5775	11.6140	11.5775	11.5562	11.5282
17	11.9720	12.0077	11.7498	11.7316	11.6141	11.5793	11.6141	11.5793	11.5570	11.5287
18	11.9718	12.0074	11.7503	11.7317	11.6145	11.5782	11.6145	11.5782	11.5572	11.5282
19	11.9726	12.0080	11.7497	11.7313	11.6147	11.5780	11.6147	11.5780	11.5567	11.5285
20	11.9716	12.0070	11.7501	11.7316	11.6136	11.5782	11.6136	11.5782	11.5571	11.5278
21	11.9721	12.0074	11.7507	11.7320	11.6142	11.5772	11.6142	11.5772	11.5564	11.5279
22	11.9714	12.0069	11.7501	11.7324	11.6140	11.5768	11.6140	11.5765	11.5564	11.5281
23	11.9720	12.0069	11.7503	11.7317	11.6135	11.5775	11.6135	11.5775	11.5565	11.5285
24	11.9719	12.0077	11.7506	11.7306	11.6148	11.5767	11.6148	11.5767	11.5564	11.5290
25	11.9718	12.0072	11.7512	11.7322	11.6138	11.5766	11.6138	11.5766	11.5563	11.5280
26	11.9715	12.0078	11.7504	11.7316	11.6136	11.5774	11.6136	11.5774	11.5566	11.5285
27	11.9723	12.0074	11.7497	11.7315	11.6139	11.5785	11.6139	11.5785	11.5569	11.5280
28	11.9722	12.0064	11.7498	11.7324	11.6158	11.5763	11.6158	11.5763	11.5563	11.5280
29	11.9725	12.0067	11.7506	11.7315	11.6140	11.5771	11.6140	11.5771	11.5575	11.5278
30	11.9725	12.0075	11.7496	11.7313	11.6144	11.5770	11.6144	11.5770	11.5566	11.5298

Table A.5: Record entropies on perturbed files for Mushroom data set, intruder knows 1 attribute

File No.	Mush1		Mush2		Mush3		Mush4		Mush5	
	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random
1	2.0591	2.4440	1.8844	2.2029	1.8411	1.9980	1.7145	1.9476	1.6362	1.8804
2	2.0885	2.4430	1.8693	2.2249	1.8274	2.0051	1.7263	1.9452	1.6412	1.9277
3	2.0793	2.4480	1.8771	2.1985	1.8287	2.0130	1.7246	1.9293	1.6443	1.9245
4	2.0808	2.4526	1.8880	2.1960	1.8183	2.0040	1.7164	1.9536	1.6326	1.9025
5	2.0729	2.4484	1.8690	2.1884	1.8294	2.0012	1.7184	1.9724	1.6417	1.9063
6	2.0828	2.4663	1.8709	2.1912	1.8347	2.0168	1.7167	1.9467	1.6500	1.9008
7	2.0680	2.4454	1.8857	2.1973	1.8284	2.0237	1.7310	1.9649	1.6496	1.8800
8	2.0720	2.4460	1.8755	2.1903	1.8365	2.0259	1.7245	1.9573	1.6367	1.8931
9	2.0780	2.4517	1.8973	2.1834	1.8172	1.9895	1.7250	1.9535	1.6462	1.9063
10	2.0847	2.4543	1.8805	2.1814	1.8256	2.0140	1.7163	1.9599	1.6526	1.9064
11	2.0787	2.4460	1.8803	2.1984	1.8291	1.9961	1.7238	1.9478	1.6460	1.9165
12	2.0631	2.4485	1.8814	2.2010	1.8359	1.9886	1.7072	1.9376	1.6449	1.9070
13	2.0693	2.4527	1.8699	2.2004	1.8289	2.0090	1.7231	1.9412	1.6529	1.9086
14	2.0638	2.4430	1.8692	2.2009	1.8333	1.9940	1.7219	1.9693	1.6337	1.8933
15	2.0804	2.4547	1.8830	2.1963	1.8262	2.0069	1.7270	1.9495	1.6364	1.8792
16	2.0628	2.4485	1.8782	2.1930	1.8172	2.0145	1.7424	1.9437	1.6340	1.8898
17	2.0714	2.4510	1.8830	2.2038	1.8204	2.0159	1.7057	1.9281	1.6421	1.9230
18	2.0846	2.4583	1.8679	2.1866	1.8229	2.0111	1.7307	1.9652	1.6382	1.9016
19	2.0571	2.4545	1.8782	2.1958	1.8336	2.0157	1.7323	1.9541	1.6338	1.9014
20	2.0821	2.4444	1.8801	2.2031	1.8500	1.9841	1.7255	1.9275	1.6445	1.8965
21	2.0812	2.4490	1.8645	2.1837	1.8296	1.9893	1.7126	1.9488	1.6401	1.9056
22	2.0777	2.4397	1.8863	2.1784	1.8237	2.0049	1.7135	1.9550	1.6347	1.9205
23	2.0763	2.4505	1.8847	2.2027	1.8192	2.0050	1.7196	1.9400	1.6481	1.8895
24	2.0604	2.4558	1.8780	2.1823	1.8139	2.0110	1.7207	1.9479	1.6353	1.9007
25	2.0764	2.4489	1.8787	2.1961	1.8286	1.9832	1.7401	1.9417	1.6366	1.9189
26	2.0719	2.4460	1.8743	2.2085	1.8349	2.0023	1.7172	1.9400	1.6450	1.9198
27	2.0874	2.4406	1.8793	2.2023	1.8111	2.0164	1.7237	1.9561	1.6410	1.9092
28	2.0688	2.4456	1.8749	2.2040	1.8220	1.9775	1.7298	1.9558	1.6349	1.9077
29	2.0794	2.4520	1.8761	2.1961	1.8472	1.9906	1.7119	1.9539	1.6319	1.9096
30	2.0682	2.4578	1.8686	2.1989	1.8244	1.9992	1.7167	1.9542	1.6413	1.9035

Table A.6: Confidential attribute entropies on perturbed files for Mushroom data set, intruder knows 1 attribute

File No.	Mush1		Mush2		Mush3		Mush4		Mush5	
	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random
1	1.3466	1.4174	0.8505	0.7087	1.0631	0	0.1417	0	0.1417	0.2835
2	1.63	2.2679	0.8505	0.4252	0.805	0.1417	0	0.1417	0.567	0.1417
3	1.5592	0.8505	0.8505	0.1417	0.567	0	0.2126	0.1417	0.1417	0.1417
4	1.4174	0.2835	0.8505	0	0.9922	0.1417	1.2757	0.1417	0.9922	0.1417
5	1.1339	0.9922	0.567	0.4252	0.1417	0.1417	0.9922	0.1417	0.567	0.1417
6	2.6931	1.2048	1.2048	0.9922	0.8505	0.1417	0.1417	0.4252	0.567	0
7	0.9922	1.2757	0	0.7087	0.1417	0	0.4252	0	0.1417	0.2835
8	0.9922	0.8505	0.8505	0.1417	0.1417	0.4252	0	0.1417	0	0
9	0.9922	1.2757	1.2757	0.9922	0	0.1417	0.1417	0.1417	0.2126	0.4252
10	1.1339	0.9217	0.7087	0.4252	0.567	0	0.9922	0	0	0.1417
11	1.8427	0.7796	0.2835	0.1417	0.1417	0.1417	0.7087	0.1417	0.4961	0.1417
12	1.9844	0.1417	0.7796	0.4252	0	0.1417	0.7087	0.1417	0.9922	0.1417
13	1.0631	1.3466	0.8505	0.1417	0.7087	0.1417	0.567	0	0.1417	0
14	2.2679	0.7087	1.0631	0.9922	1.0631	0	0.7087	0.1417	0.4252	0.567
15	1.1339	0.8505	0.567	0.1417	0.9922	0.1417	0.1417	0.2126	0.4253	0.1417
16	0.9922	0.2835	1.1339	0.9922	0	0	0.1417	0	0.7796	0
17	1.1339	1.2757	0.2835	0.567	0.7796	0.1417	0	0	0.7087	0.2835
18	0.9922	0.4252	0.8505	0.8505	0.1417	0	0.7087	0.2835	0.9922	0.7087
19	1.3466	1.2757	0.2835	0.7087	0.8859	0.1417	0.7087	0.4252	0.9922	0.2835
20	0.9922	0.4252	1.1339	1.63	0.8505	0	0.567	0.1417	0.1417	0.2835
21	0.9922	0.7796	0.567	0.567	0.9922	0.1417	0	0	0.7796	0.1417
22	1.3466	0.2835	0.9922	0.9922	0.567	0.1417	0	0.1417	0.2126	0.4252
23	1.7718	1.2757	0.8505	0.2835	0.7087	0.2835	0.7087	0.567	0.1417	0.2835
24	1.9844	1.63	0.7796	1.429	0.7087	0.4252	0.1417	0.2126	0.7087	0.1417
25	1.9135	1.9844	1.0631	1.2757	0.7087	0.4252	0.567	0	0.9922	0.4252
26	1.7009	1.2048	0.8505	0.9922	0.9922	0.1417	0.7796	0.2835	0.1417	0.8505
27	1.2048	0.7087	0.8505	0.8505	0.9922	0.1417	0.9922	0.4252	0.2126	0
28	1.2048	0.9922	0.2126	0.7087	0.7087	0.1417	0	0.1417	0.7087	0.1417
29	0.9922	1.4174	1.1339	0.1417	0.2126	0.1417	0.1417	0.1417	0.2126	0
30	1.1339	1.1339	1.1339	0.1417	0.9922	0.1417	0.1417	0.1417	0.02126	0.1417

Table A.7: Percentage of incorrectly classified instances for J48 decision tree builder on perturbed files for Mushroom data set, when tested against original microdata file

Perturbation	k_1	k_2	VICUS	Random
Mush1	2	20	6.532	8.137
Mush2	5	20	3.763	4.872
Mush3	10	10	3.627	4.243
Mush4	10	20	2.681	3.383
Mush5	10	50	1.793	2.951

Table A.8: Average percentage of incorrectly classified instances for Mushroom perturbations, when the perturbed file is used for testing.

File No.	Mush1		Mush2		Mush3		Mush4		Mush5	
	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random
1	6.6088	8.3629	4.1460	4.7130	3.6145	4.5358	2.7286	2.9943	1.6478	2.7640
2	6.6974	8.2211	3.8448	4.4295	4.0220	3.8979	2.7817	3.4904	2.0198	2.9943
3	6.8214	8.4160	3.2069	4.7307	3.7208	3.9157	2.6754	3.5082	1.7364	3.1538
4	6.6774	8.5932	4.1283	4.4826	3.9157	4.2700	2.6931	3.1184	1.7541	3.0475
5	6.5556	7.7250	3.6853	5.3685	3.6145	4.3232	3.0829	3.8625	1.6832	2.9589
6	6.6797	8.2034	3.8448	4.9610	3.5259	3.8271	2.7108	3.3664	1.7541	2.6223
7	6.1304	7.7073	3.8448	5.0850	3.5967	4.7307	2.6223	3.5082	1.4529	2.6577
8	6.6797	8.3274	3.4373	4.2326	3.6322	4.2877	2.1084	3.5613	1.7895	2.8172
9	6.0595	8.1680	3.5790	4.7661	3.3487	4.2169	2.4274	3.6676	1.7718	3.0829
10	6.5025	8.3806	3.2778	5.1028	3.5436	4.3055	2.7108	2.9412	1.9313	2.9943
11	6.5025	7.9376	3.3841	4.2535	3.2247	3.7208	3.2069	3.7030	1.7009	2.9589
12	6.8214	8.4692	3.9688	4.6421	3.3133	4.4118	2.6754	3.3664	1.9667	2.8703
13	6.4493	7.7959	3.7208	5.2445	3.7385	4.2346	2.6931	3.5082	1.4174	2.5159
14	6.0595	8.1502	4.5358	4.6775	3.8448	3.6499	2.9412	3.4550	1.9313	3.0120
15	6.2544	7.6010	3.7916	3.9334	4.4295	3.9511	2.7463	3.5613	1.6300	2.7286
16	6.4316	8.1502	4.0751	4.9965	3.2601	4.1814	2.4805	2.8880	2.0376	2.9766
17	6.6088	7.8845	4.0928	4.8370	3.5436	4.6775	2.7994	3.3664	1.9135	3.1361
18	6.9454	8.2211	3.6853	4.9610	3.6322	4.5004	2.9057	3.6322	2.0553	3.3664
19	6.6974	8.0085	3.5082	5.0850	3.5613	4.3586	2.8172	3.3310	2.1084	3.0652
20	6.6084	8.0971	4.2523	6.3430	3.7030	4.0043	2.9235	3.4196	1.8249	3.3133
21	6.6797	7.8668	3.4196	4.7484	4.0751	4.2523	2.4805	3.2601	2.0198	2.7994
22	6.3076	7.5656	3.5082	5.0319	3.8448	3.8448	2.2325	3.5436	1.5592	2.7640
23	6.4493	8.5046	3.5259	4.8193	3.1715	4.7307	2.8349	3.2424	1.6832	3.2955
24	6.3962	7.9908	4.0928	5.2268	3.6676	3.8802	2.3565	3.1006	1.6832	2.5514
25	6.7682	8.0439	4.5712	5.3685	3.4904	4.5358	2.8349	3.0829	1.8249	2.9412
26	7.1580	8.0439	3.5436	4.7307	3.7030	4.8016	2.9589	3.8979	1.5769	3.4373
27	6.7151	8.4515	3.2247	4.7838	3.7562	4.5358	2.6754	3.4018	1.7718	3.2424
28	6.2190	8.3274	3.5613	5.3685	3.2955	4.3586	2.3210	2.7640	1.9313	2.4628
29	6.0241	8.2920	3.8979	4.3409	3.4018	4.2523	2.5691	3.7562	1.8072	2.5691
30	6.4493	8.6109	3.5259	4.9079	3.6322	4.1106	2.4451	3.2069	1.8072	3.4373

Table A.9: Percentage of incorrectly classified instances for J48 decision tree builder on perturbed files for Mushroom data set

A.4 Contraceptive Method Choice

The attribute values and associated vertex numbering for the Contraceptive Method Choice data set is shown in Table A.10

Attribute	*Value*	*Vertex Number*	*Attribute*	*Value*	*Vertex Number*
Wife's Age	16	1		4 (high)	38
	17	2	Husband's	1 (low)	39
	18	3	Education	2	40
	19	4		3	41
	20	5		4 (high)	42
	21	6	Number of	0	43
	22	7	Children	1	44
	23	8	Ever Born	2	45
	24	9		3	46
	25	10		4	47
	26	11		5	48
	27	12		6	49
	28	13		7	50
	29	14		8	51
	30	15		9	52
	31	16		10	53
	32	17		11	54
	33	18		12	55
	34	19		13	56
	35	20		16	57
	36	21	Wife's	0 (Non-Islam)	58
	37	22	Religion	1 (Islam)	59
	38	23	Wife	0 (Yes)	60
	39	24	Working	1 (No)	61
	40	25	Husband's	1	62
	41	26	Occupation	2	63
	42	27		3	64
	43	28		4	65
	44	29	Standard of	1 (low)	66
	45	30	Living Index	2	67
	46	31		3	68
	47	32		4 (high)	69
	48	33	Media	0 (good)	70
	49	34	Exposure	1 (not-good)	71
Wife's	1 (low)	35	Contraceptive	1 (none)	72
Education	2	36	Method Use	2 (long-term)	73
	3	37		3 (short-term)	74

Table A.10: Vertex labeling for Contraceptive Method Choice data set

A.5 Wisconsin Breast Cancer

File No.	WBC1		WBC2		WBC3		WBC4		WBC5	
	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random
1	9.2273	9.2438	8.9324	8.9406	8.7189	8.7872	8.5026	8.5167	8.3148	8.3151
2	9.2300	9.2451	8.9133	8.9361	8.7107	8.7974	8.4974	8.5161	8.3148	8.3091
3	9.2281	9.2449	8.9217	8.9358	8.7147	8.7896	8.4953	8.5182	8.3157	8.3215
4	9.2293	9.2446	8.9114	8.9399	8.7118	8.7917	8.4993	8.5236	8.3233	8.3130
5	9.2278	9.2447	8.9240	8.9390	8.7118	8.7878	8.5012	8.5211	8.3096	8.3231
6	9.2301	9.2434	8.9235	8.9390	8.7127	8.7925	8.5025	8.5236	8.3071	8.3263
7	9.2289	9.2463	8.9181	8.9349	8.7111	8.7869	8.4990	8.5204	8.3163	8.3154
8	9.2300	9.2452	8.9237	8.9370	8.7130	8.7908	8.4952	8.5159	8.3190	8.3251
9	9.2299	9.2445	8.9246	8.9378	8.7139	8.7972	8.4989	8.5216	8.3105	8.3144
10	9.2296	9.2449	8.9224	8.9413	8.7132	8.7932	8.5047	8.5156	8.3086	8.3244
11	9.2304	9.2446	8.9181	8.9321	8.7110	8.7898	8.5016	8.5165	8.3133	8.3236
12	9.2279	9.2440	8.9195	8.9354	8.7145	8.7880	8.5017	8.5202	8.3152	8.3183
13	9.2300	9.2455	8.9133	8.9386	8.7141	8.7869	8.4988	8.5244	8.3096	8.3106
14	9.2280	9.2437	8.9175	8.9391	8.7157	8.7847	8.5025	8.5141	8.3088	8.3178
15	9.2277	9.2432	8.9228	8.9416	8.7139	8.7938	8.5011	8.5139	8.3134	8.3301
16	9.2288	9.2455	8.9195	8.9393	8.7102	8.7866	8.4973	8.5179	8.3086	8.3166
17	9.2306	9.2454	8.9171	8.9376	8.7092	8.7932	8.5027	8.5273	8.3151	8.3187
18	9.2303	9.2442	8.9181	8.9364	8.7142	8.7873	8.5018	8.5216	8.3134	8.3252
19	9.2271	9.2445	8.9155	8.9404	8.7091	8.7915	8.4992	8.5181	8.3091	8.3159
20	9.2284	9.2447	8.9243	8.9423	8.7127	8.7884	8.5061	8.5075	8.3098	8.3206
21	9.2294	9.2438	8.9223	8.9382	8.7120	8.7843	8.4990	8.5131	8.3087	8.3151
22	9.2314	9.2462	8.9203	8.9372	8.7188	8.7896	8.4965	8.5193	8.3166	8.3223
23	9.2296	9.2430	8.9185	8.9394	8.7161	8.7946	8.5001	8.5192	8.3152	8.3257
24	9.2280	9.2442	8.9188	8.9382	8.7102	8.7904	8.4944	8.5146	8.3121	8.3119
25	9.2308	9.2424	8.9260	8.9392	8.7133	8.7895	8.4963	8.5185	8.3175	8.3222
26	9.2286	9.2469	8.9223	8.9379	8.7176	8.7895	8.5009	8.5138	8.3170	8.3260
27	9.2306	9.2462	8.9230	8.9360	8.7092	8.7883	8.4991	8.5280	8.3101	8.3172
28	9.2296	9.2450	8.9222	8.9359	8.7165	8.7869	8.4976	8.5114	8.3227	8.3181
29	9.2318	9.2446	8.9222	8.9351	8.7143	8.7878	8.4955	8.5196	8.3131	8.3186
30	9.2308	9.2461	8.9238	8.9361	8.7100	8.7860	8.4969	8.5245	8.3172	8.3217

Table A.11: Record entropies on perturbed files for Wisconsin Breast Cancer data set, intruder knows 1 attribute.

We again refer to Figure A.1 to discuss some important aspects of a good choice for the probability parameters k_1 and k_2. The top left graph of the figure compares different k_2 values when the value of $k_1 = 2$. Hence the probability of an attribute remaining unchanged in the perturbed data file is only twice the probability of it changing to another value in the same partition. The three values for k_2 shown in the top left graph of Figure A.1 ar $k_2 = 2$, $k_2 = 10$ and $k_2 = 50$. Note that for $k_2 = 2$ the average classification errors for the *VICUS* and *Random* method are effectively the same. However, when k_2 is increased significantly to 50, the difference in error between the *VICUS* and *Random* methods is now significant with the *VICUS* method's average error being almost half that of the *Random*. The

File No.	WBC1		WBC2		WBC3		WBC4		WBC5	
	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random
1	3.3211	3.3216	3.3018	3.3179	3.2471	3.3016	3.2537	3.2799	3.2108	3.2494
2	3.3213	3.3215	3.2916	3.3108	3.2418	3.3070	3.2457	3.2929	3.2137	3.2426
3	3.3207	3.3216	3.2923	3.3146	3.2511	3.3064	3.2422	3.2787	3.2278	3.2506
4	3.3212	3.3215	3.2987	3.3170	3.2481	3.3041	3.2563	3.2833	3.2095	3.2531
5	3.3215	3.3214	3.3002	3.3168	3.2419	3.3022	3.2260	3.2944	3.2212	3.2567
6	3.3209	3.3212	3.3017	3.3173	3.2477	3.2918	3.2390	3.2885	3.2055	3.2522
7	3.3215	3.3215	3.2943	3.3184	3.2509	3.3088	3.2387	3.2850	3.2157	3.2393
8	3.3212	3.3218	3.2939	3.3157	3.2415	3.3067	3.2416	3.2976	3.2230	3.2525
9	3.3210	3.3216	3.3019	3.3178	3.2454	3.3054	3.2463	3.2837	3.2317	3.2563
10	3.3212	3.3217	3.2978	3.3184	3.2542	3.3002	3.2469	3.2918	3.2197	3.2334
11	3.3204	3.3215	3.3030	3.3175	3.2565	3.3105	3.2473	3.2756	3.2119	3.2358
12	3.3212	3.3217	3.2991	3.3156	3.2490	3.3070	3.2461	3.2887	3.2293	3.2393
13	3.3214	3.3216	3.2969	3.3181	3.2414	3.3053	3.2432	3.2890	3.2349	3.2530
14	3.3214	3.3214	3.2869	3.3163	3.2457	3.3129	3.2552	3.2842	3.2273	3.2641
15	3.3211	3.3215	3.2992	3.3168	3.2495	3.3042	3.2473	3.2775	3.2171	3.2511
16	3.3211	3.3217	3.2927	3.3174	3.2424	3.3086	3.2404	3.3002	3.2108	3.2429
17	3.3216	3.3217	3.2908	3.3162	3.2562	3.3033	3.2476	3.2968	3.2145	3.2513
18	3.3215	3.3215	3.2950	3.3167	3.2527	3.3069	3.2416	3.2973	3.2145	3.2605
19	3.3210	3.3215	3.3000	3.3154	3.2519	3.3077	3.2326	3.2929	3.2148	3.2448
20	3.3213	3.3217	3.2933	3.3166	3.2532	3.3040	3.2378	3.2909	3.2212	3.2438
21	3.3213	3.3217	3.2890	3.3158	3.2502	3.3104	3.2458	3.2903	3.2137	3.2452
22	3.3212	3.3217	3.3021	3.3121	3.2439	3.3048	3.2469	3.2852	3.2144	3.2375
23	3.3213	3.3216	3.3030	3.3157	3.2482	3.2980	3.2397	3.2951	3.2201	3.2643
24	3.3197	3.3216	3.2948	3.3193	3.2498	3.3109	3.2473	3.2791	3.1962	3.2292
25	3.3216	3.3216	3.2990	3.3164	3.2525	3.3041	3.2519	3.2797	3.2242	3.2588
26	3.3211	3.3211	3.2953	3.3180	3.2472	3.2989	3.2380	3.2895	3.2149	3.2542
27	3.3213	3.3217	3.2914	3.3164	3.2517	3.3022	3.2366	3.2918	3.2207	3.2563
28	3.3215	3.3218	3.2971	3.3160	3.2415	3.3111	3.2498	3.2896	3.2267	3.2543
29	3.3209	3.3216	3.2968	3.3185	3.2465	3.2996	3.2370	3.2846	3.2250	3.2403
30	3.3211	3.3216	3.2919	3.3194	3.2428	3.3105	3.2330	3.2967	3.2219	3.2376

Table A.12: Confidential attribute entropies on perturbed files for Wisconsin Breast Cancer data set, intruder knows 1 attribute

distribution of the classification errors for the 60 WBC3 perturbed files, that is with $k_1 = 2$ and $k_2 = 5$, are shown in Figure A.2. Note that for the *VICUS* files, to the left of the figure, the values fall within a much tighter range than for the *Random* method.

The final graph in the top right hand side of Figure A.1 compares how the decision tree classification errors change when for different values of k_1 and k_2 we always have $k_1 \times k_2 = 100$. The three parameter combinations shown, from left to right, are $(k_1 = 10, k_2 = 10)$, $(k_1 = 5, k_2 = 20)$ and $(k_1 = 2, k_2 = 50)$. We note that for the *VICUS* method, there is little change in the average percentage of incorrectly classified instances (7.218, 8.019 and 7.989). Yet for the *Random* method the difference in errors between this method and the *VICUS* method becomes more marked as the value for k_2

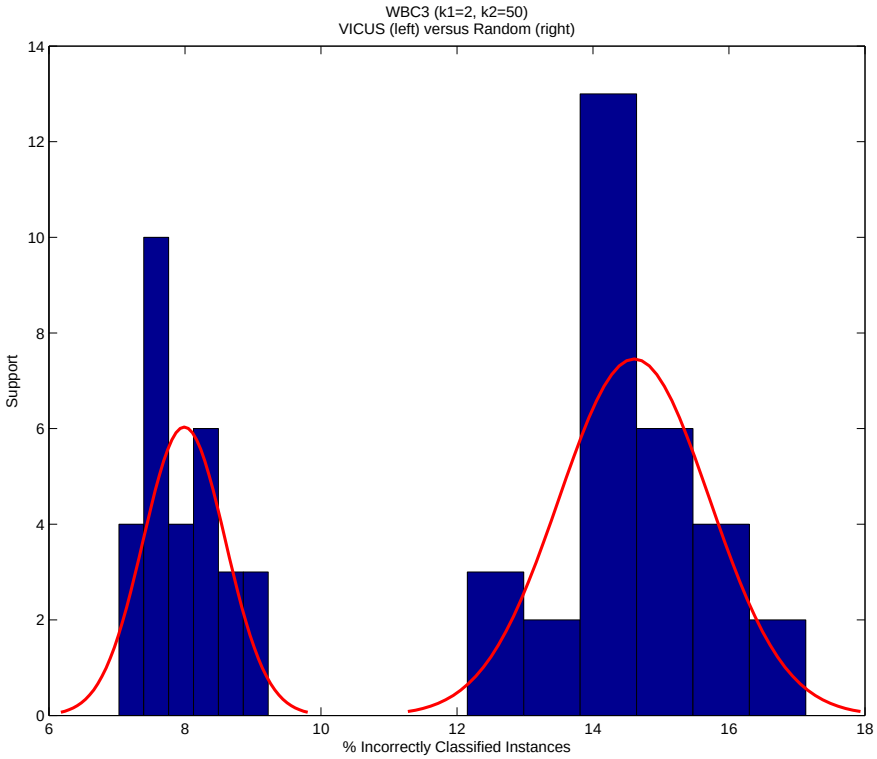

Figure A.2: Classification error distribution comparison for WBC3 perturbation files.

File No.	WBC1		WBC2		WBC3		WBC4		WBC5	
	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random
1	18.7408	19.1801	6.0029	8.1991	6.0029	5.1245	6.0029	5.2709	5.4173	5.1245
2	53.4407	16.5447	4.978	11.5666	6.0029	10.5417	4.8316	5.8565	5.1245	5.4173
3	13.47	16.3982	7.1742	5.4173	6.0029	5.8565	6.0029	5.5637	6.0029	4.246
4	9.9561	13.47	6.0029	10.1025	6.0029	4.8316	6.0029	5.7101	6.0029	6.1493
5	15.8126	25.183	5.1245	8.1991	6.0029	7.3206	6.0029	5.7101	5.1245	5.4173
6	23.1332	12.0059	5.8565	3.9531	5.1245	6.735	4.246	5.4173	4.3924	4.5388
7	18.0088	16.8375	4.5388	4.5388	6.0029	4.5388	6.0029	6.2958	4.246	5.5637
8	9.9561	23.4216	6.0029	8.0527	5.8565	5.7101	6.0029	4.978	6.0029	5.4173
9	11.4202	15.662	4.8316	5.4173	6.0029	7.0278	4.978	4.8316	5.8565	4.0996
10	6.1493	17.2767	5.8565	4.5388	6.0029	5.8565	6.0029	3.3675	4.978	3.9531
11	17.1303	11.1274	6.1493	10.3953	6.0029	6.2958	6.0029	5.7101	4.6852	5.5637
12	15.5198	27.8184	5.5637	10.2489	6.0029	10.981	6.0029	6.735	4.3924	4.6852
13	14.7877	19.6192	4.989	7.9063	4.6852	5.7101	6.1493	5.4173	5.8565	5.5637
14	10.2489	22.5476	9.3704	9.9561	6.0029	8.1991	5.1245	4.3924	4.5388	3.5139
15	18.5944	19.9122	6.2958	8.9312	6.0029	4.6852	6.0029	5.8565	4.978	3.2211
16	14.3485	12.4451	5.1245	6.1493	6.0029	4.5388	6.0029	5.4173	4.6852	3.5139
17	14.0556	14.9341	5.8565	5.1245	6.0029	4.246	6.0029	6.1493	5.4173	4.6852
18	8.9312	22.1083	6.1493	13.0307	6.0029	5.5637	6.1493	4.6852	4.6852	6.422
19	12.7379	10.8346	5.7107	11.1274	6.0029	5.4172	6.0029	5.2709	5.1245	4.987
20	32.2108	15.3734	6.4422	5.8565	6.0029	7.0278	6.0029	5.7101	5.1245	4.246
21	8.4919	14.6413	5.1245	8.6384	6.0029	5.8565	6.0029	5.8565	6.0029	5.7101
22	23.4261	34.9927	6.0029	11.8594	5.7107	5.2709	6.0029	5.4173	6.1493	5.2709
23	18.8873	15.5198	5.5637	6.735	6.0029	5.8565	6.0029	5.5637	4.978	5.4173
24	10.6881	9.6633	5.1245	6.5886	6.0029	10.1025	5.2709	5.1245	4.5388	5.5637
25	16.2518	18.448	4.978	10.1025	6.0029	7.4671	6.0029	5.5637	5.4173	3.8067
26	16.8375	12.1523	5.2709	5.8565	5.1245	5.8565	6.0029	5.2709	4.6852	3.9531
27	13.3236	30.6003	6.0029	5.7101	6.0029	6.0029	6.0029	5.4173	4.978	4.5388
28	10.5417	22.8404	5.1245	8.3455	6.0029	5.7101	6.0029	8.3455	4.6852	7.0278
29	12.8843	11.2738	5.5637	12.0059	6.0029	6.1493	8.4919	4.0996	6.0029	5.1245
30	12.8843	19.4729	5.5637	4.8316	6.0029	4.8316	6.0029	4.6852	5.7101	4.46

Table A.13: Percentage of incorrectly classified instances for J48 decision tree builder on perturbed files for Wisconsin Breast Cancer data set, when the perturbed file is tested using the original file.

grows (9.517, 10.791 and 14.607).

File No.	WBC1		WBC2		WBC3		WBC4		WBC5	
	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random	VICUS	Random
1	30.7467	33.2357	15.8162	20.6442	7.3206	14.3485	8.1991	11.2738	9.2240	10.5417
2	30.7467	28.1113	14.4949	18.8873	7.1742	14.6413	6.8814	10.1025	7.4671	9.2240
3	32.0644	28.8433	15.6662	20.2050	7.4671	15.0805	8.0527	10.6881	9.2240	8.7848
4	30.6003	27.6720	16.2518	20.4978	8.9312	13.0307	8.9312	12.1523	7.9063	9.8097
5	31.6252	29.5754	14.7877	19.6193	7.6135	14.0556	8.0527	11.4202	6.4422	9.3704
6	31.9180	32.9429	14.6413	22.1083	8.6384	14.9341	7.4671	10.1025	6.1493	8.0527
7	33.2357	29.5754	13.4700	20.6442	7.9063	15.0805	7.7599	11.2738	5.5637	8.4919
8	26.3543	35.5783	14.4949	20.4978	9.0776	14.0556	7.4671	12.1523	7.9063	8.1991
9	35.1391	31.4788	14.9341	18.7408	9.2240	12.7379	8.4919	12.4451	7.4671	11.2738
10	36.0176	28.8433	14.2020	22.5476	8.3455	15.8126	8.3455	9.3704	7.9063	10.8346
11	31.1859	29.7218	13.4628	21.5227	8.4919	14.3485	7.4671	10.8346	6.0029	9.6633
12	34.4070	37.7745	13.4700	19.4729	8.3455	14.6413	9.2240	13.1772	7.1742	8.7848
13	33.5286	36.3104	12.8843	20.9370	8.4919	14.6413	7.7599	10.8346	7.1742	12.2987
14	37.9209	32.2108	13.9092	21.8155	7.4671	15.5198	7.3206	9.8097	8.0527	9.3704
15	33.5286	33.9678	14.2020	21.8155	7.6135	15.5198	8.1991	10.1025	7.9063	9.3704
16	33.5286	31.3324	15.3734	17.5695	7.0278	14.2020	7.7599	11.4202	7.0278	8.3455
17	38.0673	33.2357	16.5447	18.5944	7.9063	15.3734	8.3455	9.8097	6.4422	8.4919
18	30.6003	33.5286	15.3734	20.6442	8.1991	16.5447	7.9063	12.8843	6.2958	9.2240
19	30.7467	34.4070	13.0307	19.0337	7.4671	14.6413	8.6384	10.3953	6.2958	9.2240
20	32.7965	29.5754	15.3734	20.3514	8.0527	13.3236	7.3206	10.3953	7.4671	7.7599
21	35.8712	33.8214	14.6413	21.8155	7.7599	15.2269	8.3455	10.2489	7.9063	10.3953
22	32.3572	40.1171	16.1054	21.5227	8.6384	15.2269	8.9312	9.9561	6.8814	9.6633
23	29.4290	32.0644	15.0805	18.7408	7.6135	13.9092	7.9063	10.5417	6.8814	11.7130
24	31.6252	35.2855	15.0805	19.7657	7.6135	14.6413	6.7350	10.1025	6.0029	11.5666
25	29.1362	28.8433	14.0556	20.3514	7.6135	12.1523	8.4919	10.2489	8.1991	8.6384
26	31.9180	31.4788	14.7877	20.9370	8.7848	12.7379	8.0527	9.3704	6.8814	9.8097
27	32.5037	33.3821	15.9590	20.7906	8.4919	14.4949	8.1991	10.6881	6.8814	9.6633
28	33.6750	34.1142	14.9341	20.0586	7.4671	15.9590	7.7599	13.0307	7.4671	9.9561
29	31.4788	29.4290	14.4949	20.0586	7.3206	17.1303	8.0527	9.8097	7.4671	8.0527
30	28.9898	30.8931	16.3982	20.3514	7.6135	14.2020	8.4919	9.0776	6.8814	8.9312

Table A.14: Percentage of incorrectly classified instances for J48 decision tree builder on perturbed files for Wisconsin Breast Cancer data set, when the perturbed file is used for both training and testing.

Perturbation	k_1	k_2	VICUS	Random
WBC1	2	2	32.391	32.245
WBC2	2	10	14.807	20.351
WBC3	2	50	7.989	14.607
WBC4	5	20	8.019	10.791
WBC5	10	10	7.218	9.517

Table A.15: Average percentage of incorrectly classified instances for WBC perturbations, when the perturbed file is used for both training and testing.

A.5.1 Security Measure Figures

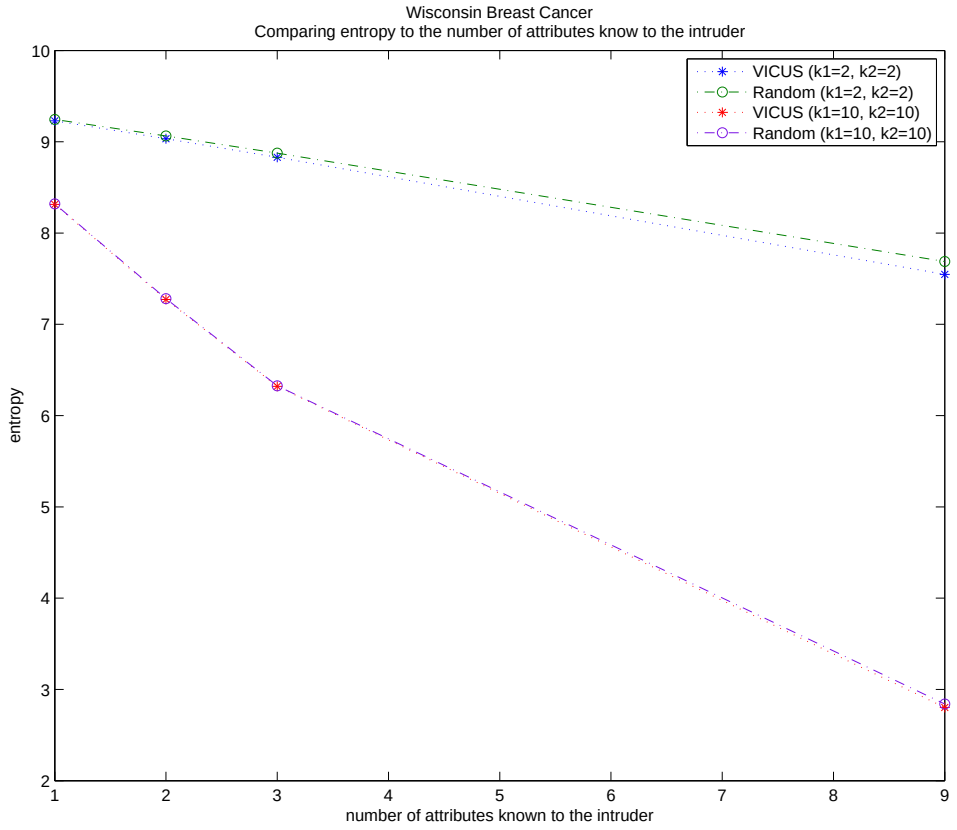

Figure A.3: Comparing record entropy to the number of attributes known by intruder, WBC data set.

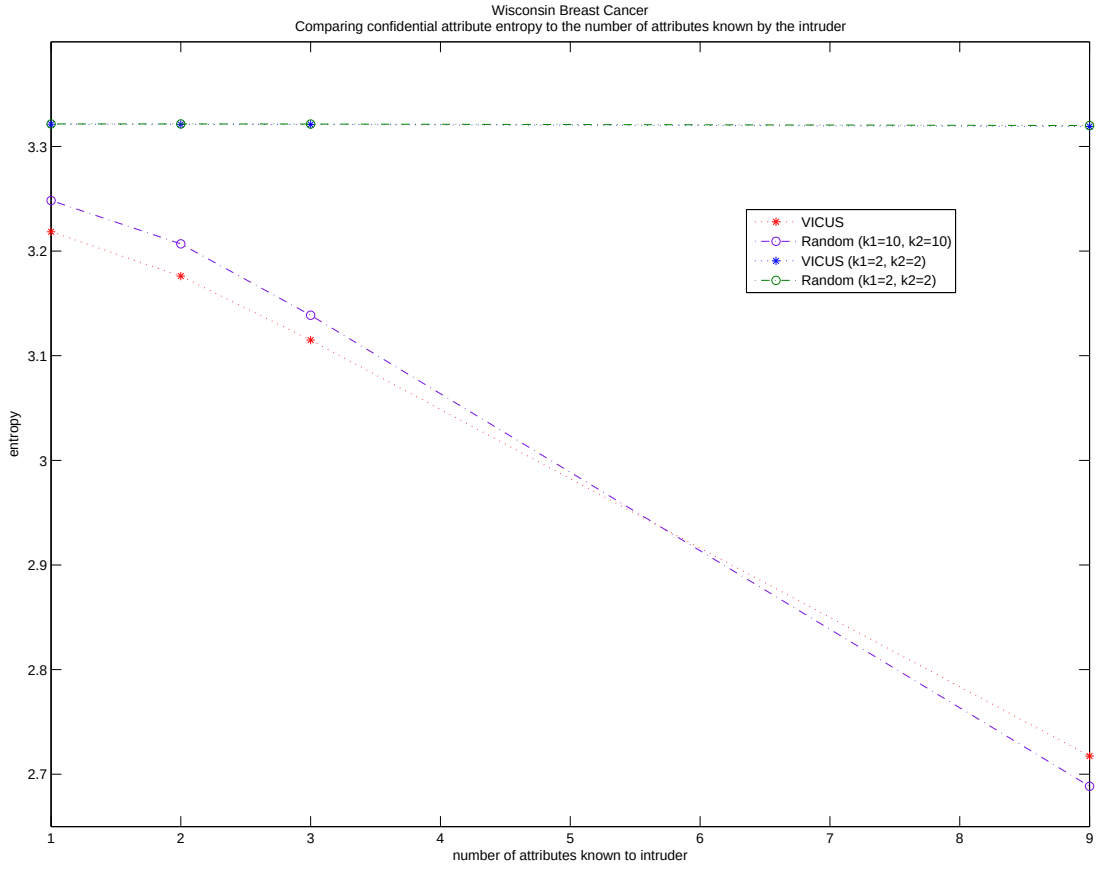

Figure A.4: Comparing confidential entropy to the number of attributes known by intruder, WBC data set.

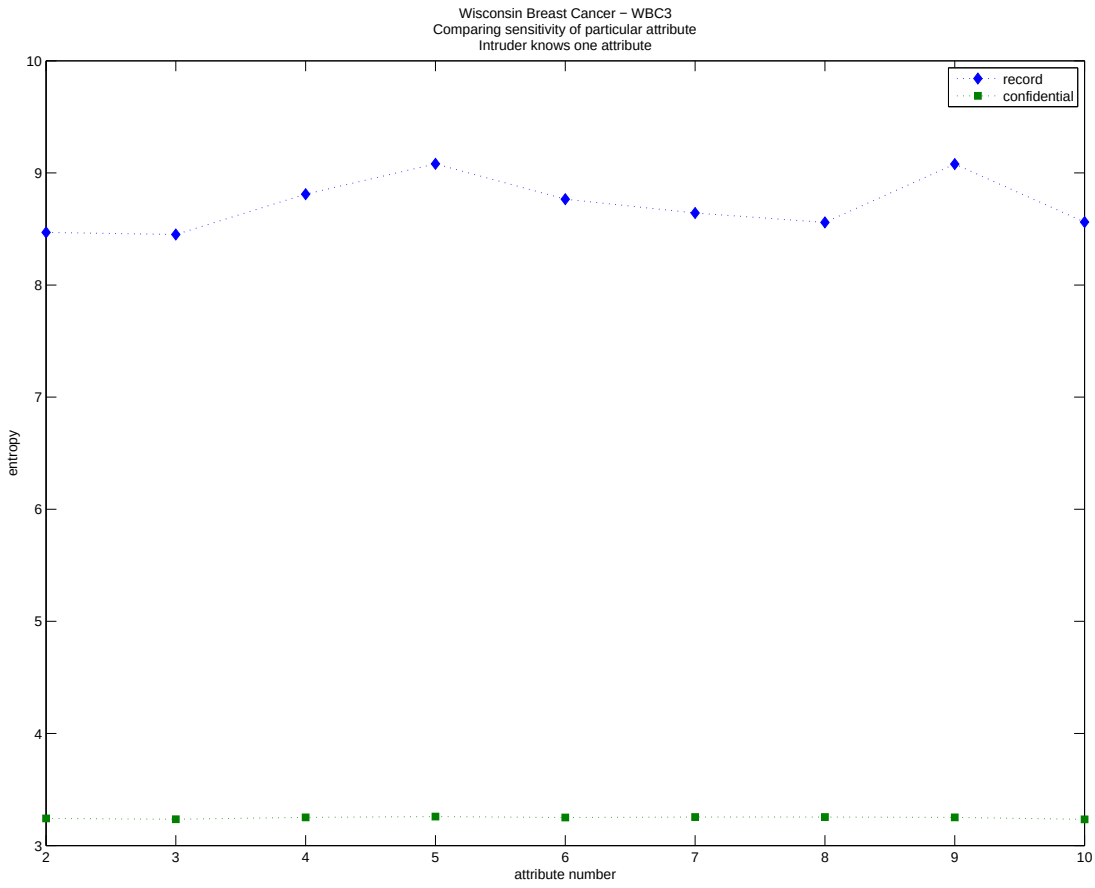

Figure A.5: Entropy sensitivity for when the user knows 1 particular attribute, WBC data set.

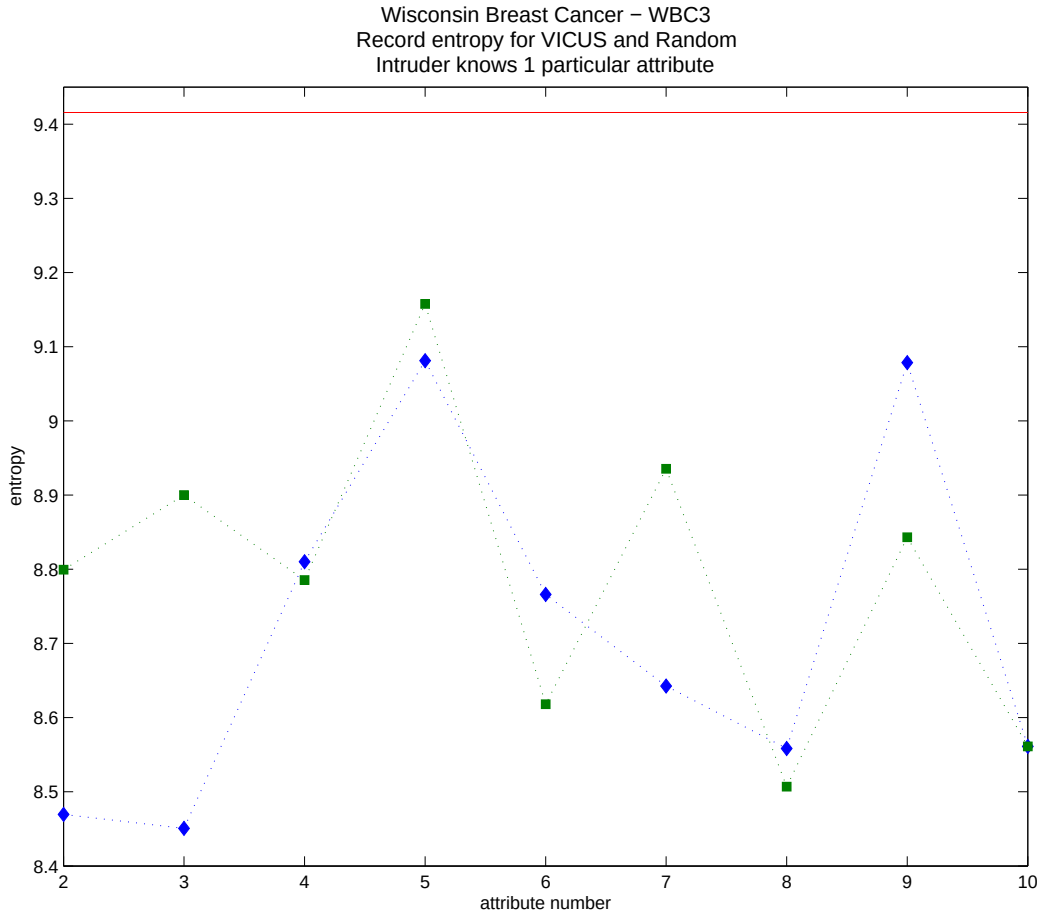

Figure A.6: Record entropy sensitivity for when the user knows 1 particular attribute, WBC data set.

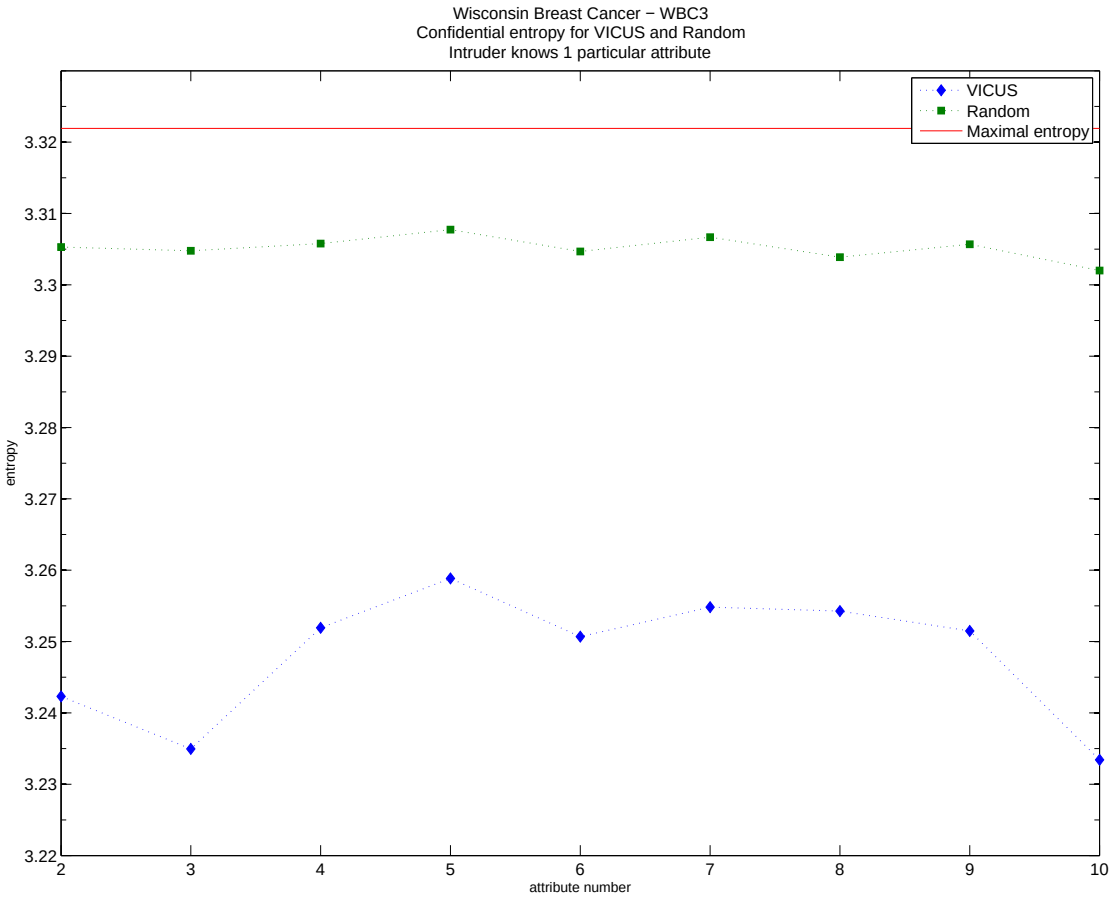

Figure A.7: Confidential entropy sensitivity for when the user knows 1 particular attribute, WBC data set.

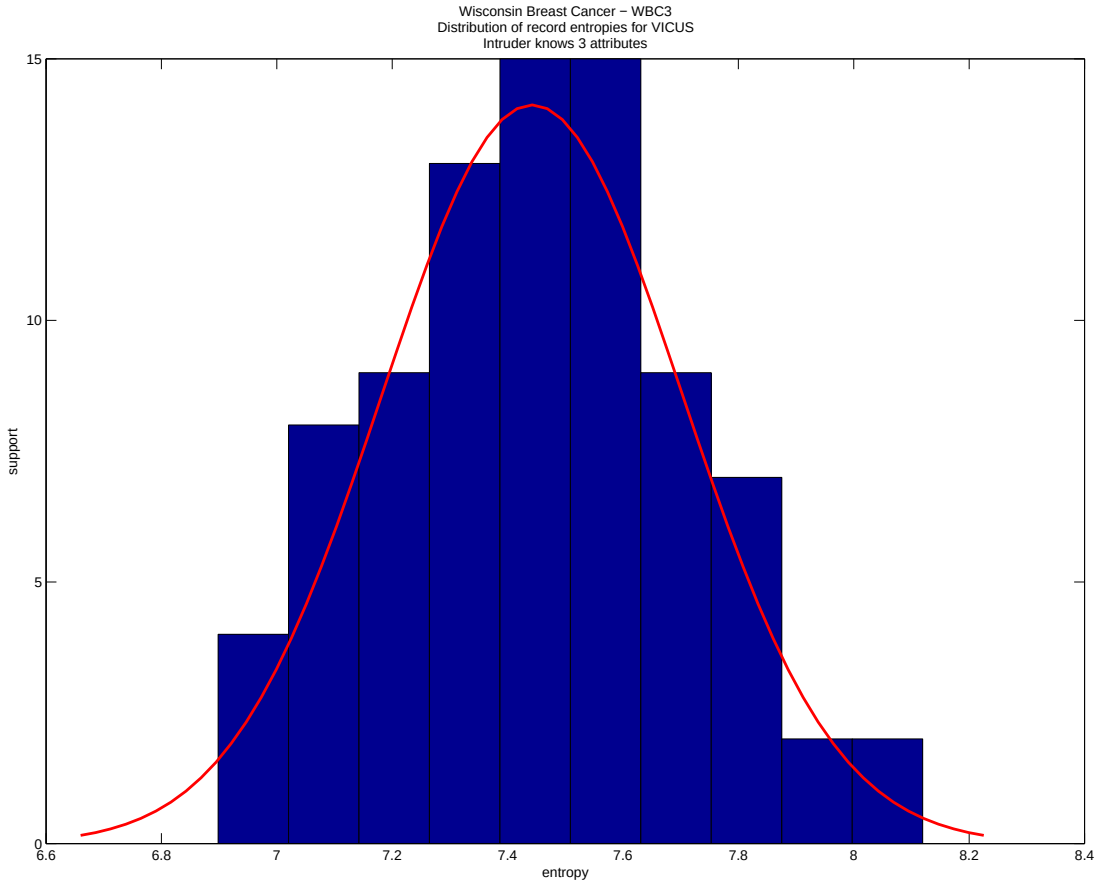

Figure A.8: Distribution of record entropies when user knows 3 attributes for each individual attribute combination, WBC data set.

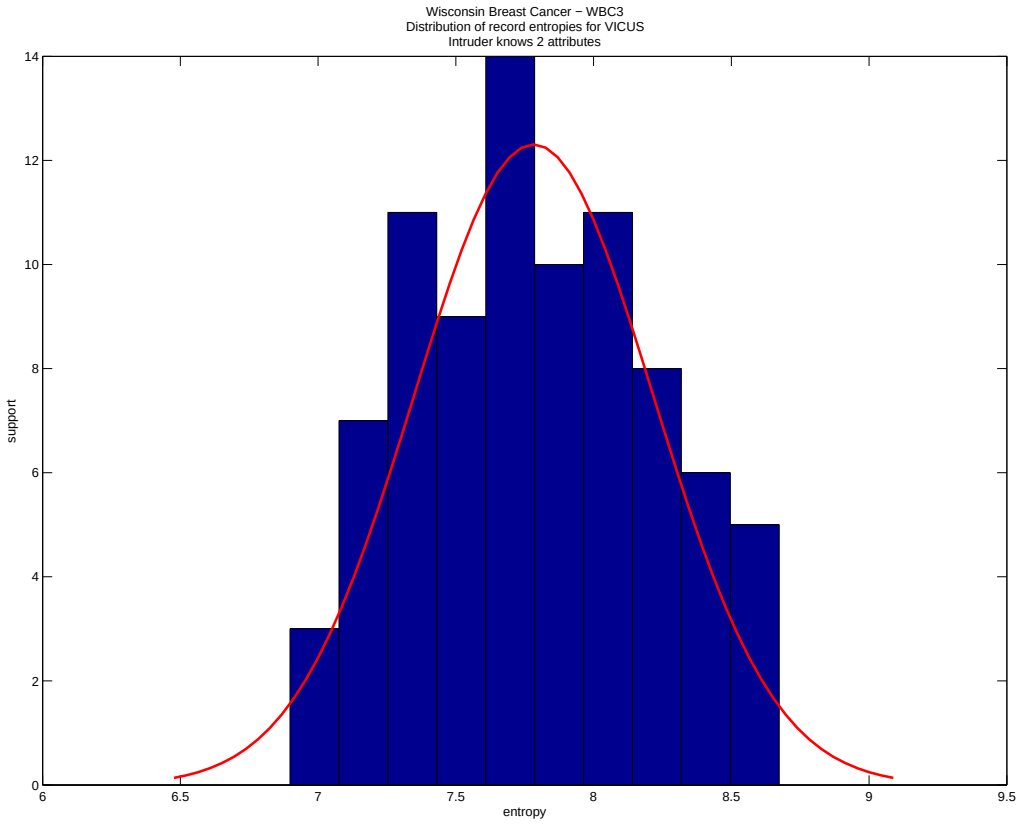

Figure A.9: Distribution of record entropies when user knows 2 attributes for each individual attribute combination, WBC data set.

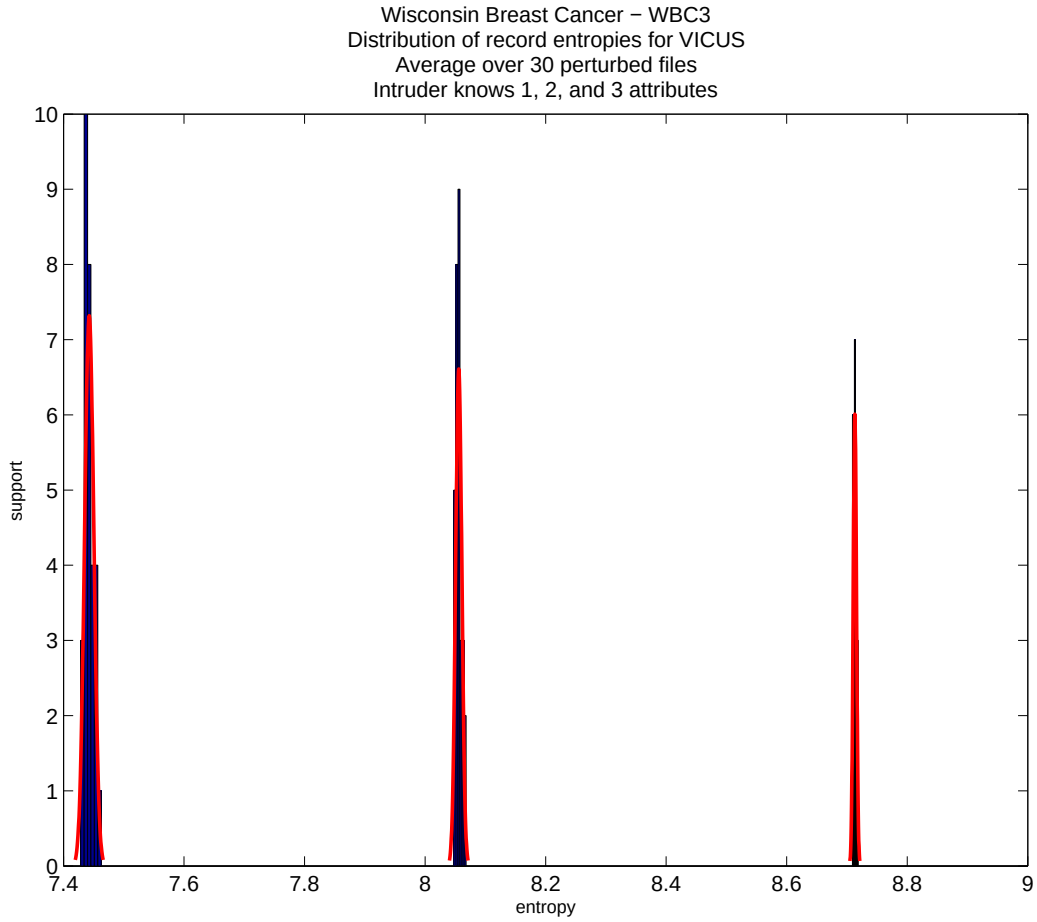

Figure A.10: Distribution of record entropies when user knows 1, 2 and 3 attributes, averaged over the 30 perturbed files, WBC data set.

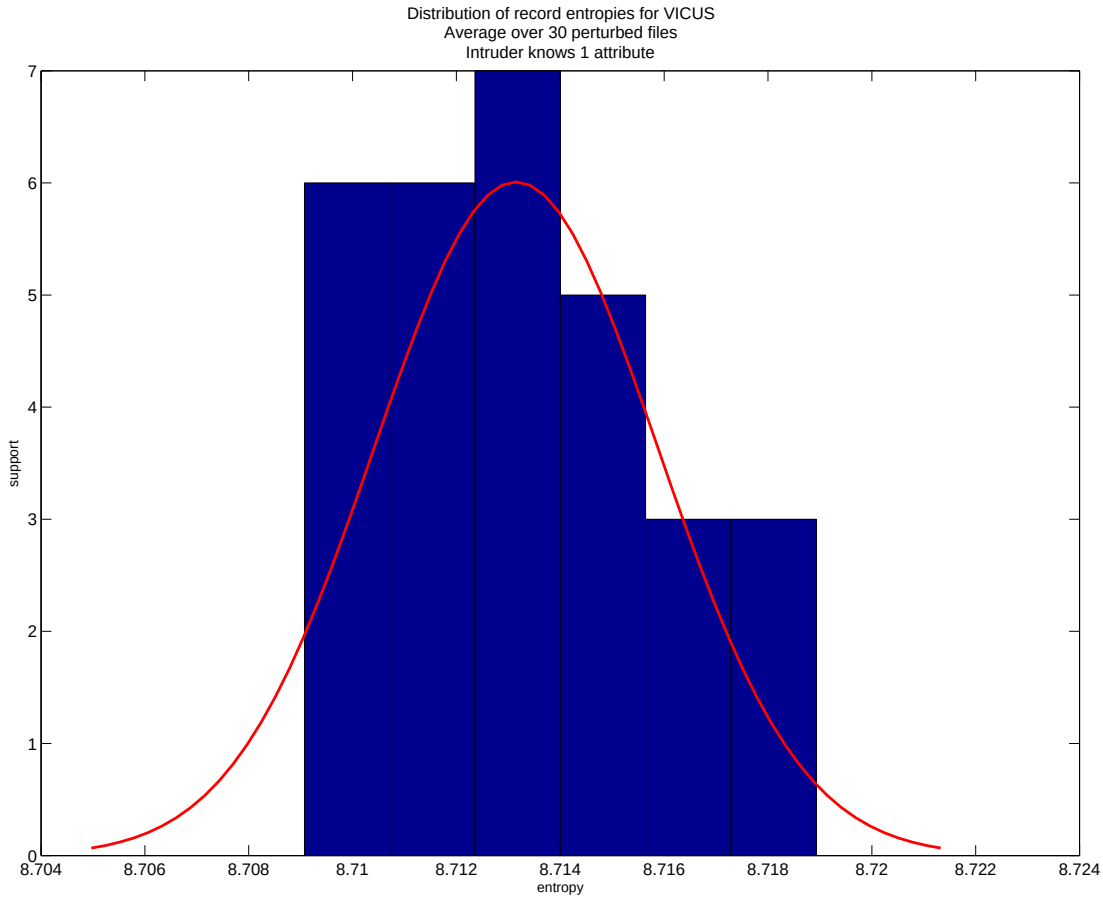

Figure A.11: Distribution of record entropies when user knows 1 attribute, averaged over the 30 perturbed files, WBC data set.

A.6 ACS PUMS

The original PUMS file contained 2,9969,741 records and by only considering those records without any missing values on the subset chosen we selected 20,000 random records from the remaining 1,348,929. For attributes 1 and 2, that is those relating to income, we rounded the values to the nearest 5,000.

The vertex numbering and corresponding attribute values are given in Tables A.16 through A.31.

Attribute	*Value*	*Vertex Number*	*Attribute*	*Value*	*Vertex Number*
WAGP	0	1	WAGP	190000	39
	5000	2		195000	40
	10000	3		200000	41
	15000	4		205000	42
	20000	5		210000	43
	25000	6		220000	44
	30000	7		225000	45
	35000	8		230000	46
	40000	9		235000	47
	45000	10		240000	48
	50000	11		245000	49
	55000	12		250000	50
	60000	13		260000	51
	65000	14		265000	52
	70000	15		270000	53
	75000	16		285000	54
	80000	17		290000	55
	85000	18		295000	56
	90000	19		300000	57
	95000	20		305000	58
	100000	21		310000	59
	105000	22		315000	60
	110000	23		320000	61
	115000	24		330000	62
	120000	25		335000	63
	125000	26		340000	64
	130000	27		350000	65
	135000	28		360000	66
	140000	29		370000	67
	145000	30		380000	68
	150000	31		400000	69
	155000	32		410000	70
	160000	33		415000	71
	165000	34		420000	72
	170000	35		495000	73
	175000	36		505000	74
	180000	37		560000	75
	185000	38		645000	76

Table A.16: Vertex labeling for PUMS data set, Attribute 1 WAGP (*Wage or salary income past 12 months*, rounded to nearest 5,000

Attribute	*Value*	*Vertex Number*	*Attribute*	*Value*	*Vertex Number*
PINCP	-15000	77	PINCP	285000	137
	-10000	78		290000	138
	-5000	79		295000	139
	0	80		300000	140
	5000	81		305000	141
	10000	82		310000	142
	15000	83		315000	143
	20000	84		320000	144
	25000	85		325000	145
	30000	86		330000	146
	35000	87		335000	147
	40000	88		340000	148
	45000	89		345000	149
	50000	90		350000	150
	55000	91		355000	151
	60000	92		360000	152
	65000	93		365000	153
	70000	94		370000	154
	75000	95		375000	155
	80000	96		380000	156
	85000	97		385000	157
	90000	98		390000	158
	95000	99		400000	159
	100000	100		405000	160
	105000	101		410000	161
	110000	102		415000	162
	115000	103		420000	163
	120000	104		425000	164
	125000	105		430000	165
	130000	106		435000	166
	135000	107		440000	167
	140000	108		445000	168
	145000	109		450000	169
	150000	110		455000	170
	155000	111		460000	171
	160000	112		465000	172
	165000	113		485000	173
	170000	114		490000	174
	175000	115		495000	175
	180000	116		500000	176
	185000	117		505000	177
	190000	118		510000	178
	195000	119		515000	179
	200000	120		525000	180
	205000	121		535000	181
	210000	122		545000	182
	215000	123		550000	183
	220000	124		555000	184
	225000	125		560000	185
	230000	126		565000	186
	235000	127		595000	187
	240000	128		635000	188
	245000	129		640000	189
	250000	130		650000	190
	255000	131		665000	191
	260000	132		785000	192
	265000	133		810000	193
	270000	134		820000	194
	275000	135		850000	195
	280000	136			

Table A.17: Vertex labeling for PUMS data set, Attribute 2 PINCP (*Total person's income*), rounded to nearest 5,000

Attribute	*Value*	*Vertex Number*	*Attribute*	*Value*	*Vertex Number*
WKHP	1	196	WKHP	44	239
	2	197		45	240
	3	198		46	241
	4	199		47	242
	5	200		48	243
	6	201		49	244
	7	202		50	245
	8	203		51	246
	9	204		52	247
	10	205		53	248
	11	206		54	249
	12	207		55	250
	13	208		56	251
	14	209		57	252
	15	210		58	253
	16	211		60	254
	17	212		61	255
	18	213		62	256
	19	214		63	257
	20	215		64	258
	21	216		65	259
	22	217		66	260
	23	218		67	261
	24	219		68	262
	25	220		69	263
	26	221		70	264
	27	222		72	265
	28	223		73	266
	29	224		75	267
	30	225		76	268
	31	226		77	269
	32	227		78	270
	33	228		80	271
	34	229		83	272
	35	230		84	273
	36	231		85	274
	37	232		86	275
	38	233		89	276
	39	234		90	277
	40	235		92	278
	41	236		95	279
	42	237		96	280
	43	238		99	281

Table A.18: Vertex labeling for PUMS data set, Attribute 3 WKHP (*Usual hours worked per week past 12 months*) top-coded at 99 hours

Attribute	*Value*	*Vertex Number*
WAOB	United States	282
	Puerto Rico and US Island Areas	283
	Latin America	284
	Asia	285
	Europe	286
	Africa	287
	Northern America	288
	Oceania and at sea	289

Table A.19: Vertex labeling for PUMS data set, Attribute 4 WAOB (*World area of birth*)

Attribute	*Value*	*Vertex Number*
RAC1P	White alone	290
	Black or African American alone	291
	American Indian alone	292
	Alaska Native alone	293
	American Indian and Alaska Native tribes specified;	294
	or American Indian or Alaska Native, not specified	
	Asian alone	295
	Native Hawaiian and other Pacific Islander alone	296
	some other race alone	297
	Two or more major race groups	298

Table A.20: Vertex labeling for PUMS data set, Attribute 5 RACE1P (*Recoded detailed race code*)

Attribute	*Value*	*Vertex Number*
JWTR	Car, truck or van	299
	Bus or trolley bus	300
	Streetcar or trolley car	301
	Subway or elevated	302
	Railroad	303
	Ferryboat	304
	Taxicab	305
	Motorcycle	306
	Bicycle	307
	Walked	308
	Worked at home	309
	Other method	310

Table A.21: Vertex labeling for PUMS data set, Attribute 6 JWTR (*Means of transportation to work*)

Attribute	*Value*	*Vertex Number*	*Attribute*	*Value*	*Vertex Number*
ST	Alabama	311	ST	Montana	337
	Alaska	312		Nebraska	338
	Arizona	313		Nevada	339
	Arkansas	314		New Hampshire	340
	California	315		New Jersey	341
	Colorado	316		New Mexico	342
	Connecticut	317		New York	343
	Delaware	318		North Carolina	344
	District of Columbia	319		North Dakota	345
	Florida	320		Ohio	346
	Georgia	321		Oklahoma	347
	Hawaii	322		Oregon	348
	Idaho	323		Pennsylvania	349
	Illinois	324		Rhode Island	350
	Indiana	325		South Carolina	351
	Iowa	326		South Dakota	352
	Kansas	327		Tennessee	353
	Kentucky	328		Texas	354
	Louisiana	329		Utah	355
	Maine	330		Vermont	356
	Maryland	331		Virginia	357
	Massachusetts	332		Washington	358
	Michigan	333		West Virginia	359
	Minnesota	334		Wisconsin	360
	Mississippi	335		Wyoming	361
	Missouri	336			

Table A.22: Vertex labeling for PUMS data set, Attribute 7 ST (*State code*)

Attribute	*Value*	*Vertex Number*	*Attribute*	*Value*	*Vertex Number*
ANC1P	Alsation	362	ANC1P	Czech	395
	Austrian	363		Bohemian	396
	Basque	364		Czechoslovakian	397
	Belgian	365		Hungarian	398
	Flemish	366		Latvian	399
	British	367		Lithuanian	400
	British Isles	368		Macedonian	401
	Danish	369		Polish	402
	Dutch	370		Romanian	403
	English	371		Russian	404
	Finnish	372		Serbian	405
	French	373		Slovak	406
	German	374		Slovene	407
	Prussian	375		Ukrainian	408
	Greek	376		Yugoslavian	409
	Irish	378		Herzegovinian	410
	Italian	379		Slavic	411
	Sicilian	380		Slavonian	412
	Luxemburger	381		Northern European	413
	Maltese	382		Western European	414
	Norwegian	383		Eastern European	415
	Portuguese	384		European	416
	Scotch Irish	385		Spaniard	417
	Scottish	386		Mexican	418
	Swedish	387		Mexican American	419
	Swiss	388		Mexicano	420
	Welsh	389		Chicano	421
	Scandinavian	390		Mexican American Indian	422
	Celtic	391		Mexican State	423
	Albanian	392		Costa Rican	424
	Bulgarian	393		Guatemalan	425
	Croatian	394		Honduran	426

Table A.23: Vertex labeling for PUMS data set, Attribute 8 ANC1P (*Recoded detailed ancestry*) Part I

Attribute	*Value*	*Vertex Number*	*Attribute*	*Value*	*Vertex Number*
ANC1P	Nicaraguan	427	ANC1P	Brazilian	459
	Panamanian	428		Guyanese	460
	Salvadoran	429		Egyptian	461
	Central American	430		Moroccan	462
	Argentinean	431		Iranian	463
	Bolivian	432		Iraqi	464
	Chilean	433		Israeli	465
	Columbian	434		Jordanian	466
	Ecuadorian	435		Lebanese	467
	Peruvian	436		Syrian	468
	Venezuelan	437		Armenian	469
	South American	438		Turkish	470
	Latin	439		Yemeni	471
	Latino	440		Palestinian	472
	Puerto Rican	441		Assyrian	473
	Cuban	442		Chaldean	474
	Dominican	443		Mideast	475
	Hispanic	444		Arab	476
	Spanish	445		Arabic	477
	Spanish American	446		Other Arab	478
	Barbadian	447		Cape Verdean	479
	Belizean	448		Ethiopian	480
	Jamaican	449		Eritrean	481
	Dutch West Indian	450		Ghanian	482
	Trinidadian Tobagonian	451		Kenyan	483
	British West Indian	452		Liberian	484
	Antigua and Barbuda	453		Nigerian	485
	Grenadian	454		Sierra Leonean	486
	Vincent-Grenadine Islander	455		Somalian	487
	West Indian	456		South African	488
	Haitian	457		Sudanese	489
	Other West Indian	458		Other Subsaharan African	490

Table A.24: Vertex labeling for PUMS data set, Attribute 8 ANC1P (*Recoded detailed ancestry*) Part II

Attribute	*Value*	*Vertex Number*	*Attribute*	*Value*	*Vertex Number*
ANC1P	Western African	491	ANC1P	Samoan	523
	African	492		Tongan	524
	Afghan	493		Guamanian	525
	Bangladeshi	494		Chamorro Islander	526
	Nepali	495		Fijian	527
	Asian Indian	496		Pacific Islander	528
	East Indian	497		Afro American	529
	Pakistani	498		African American	530
	Sri Lankan	499		Black	531
	Burmese	500		Negro	532
	Cambodian	501		Central American Indian	533
	Chinese	502		South American Indian	534
	Cantonese	503		Native American	535
	Mongolian	504		Indian	536
	Filipino	505		Cherokee	537
	Indonesian	506		American Indian	538
	Japanese	507		White	539
	Okanawan	508		Anglo	540
	Korean	509		Pennsylvania German	541
	Laotian	510		Canadian	542
	Hmong	511		French Canadian	543
	Malaysian	512		Acadian	544
	Thai	513		Cajun	545
	Taiwanese	514		American or United States	546
	Vietnamese	515		Texas	547
	Eurasian	516		North American	548
	Asian	517		Mixture	549
	Other Asian	518		Uncodable Entries	550
	Australian	519		Other Groups	551
	New Zealander	520		Other Responses	552
	Polynesian	521		Not Reported	553
	Hawaiian	522		Icelander	377

Table A.25: Vertex labeling for PUMS data set, Attribute 8 ANC1P (*Recoded detailed ancestry*) Part III

Attribute	*Value*	*Vertex Number*	*Attribute*	*Value*	*Vertex Number*
AGEP	16	554	AGEP	54	592
	17	555		55	593
	18	556		56	594
	19	557		57	595
	20	558		58	596
	21	559		59	597
	22	560		60	598
	23	561		61	599
	24	562		62	600
	25	563		63	601
	26	564		64	602
	27	565		65	603
	28	566		66	604
	29	567		67	605
	30	568		68	606
	31	569		69	607
	32	570		70	608
	33	571		71	609
	34	572		72	610
	35	573		73	611
	36	574		74	612
	37	575		75	613
	38	576		76	614
	39	577		77	615
	40	578		78	616
	41	579		79	617
	42	580		80	618
	43	581		81	619
	44	582		82	620
	45	583		83	621
	46	584		84	622
	47	585		85	623
	48	586		86	624
	49	587		87	625
	50	588		88	626
	51	589		89	627
	52	590		92	628
	53	591			

Table A.26: Vertex labeling for PUMS data set, Attribute 9 AGEP (*Age*)

Attribute	*Value*	*Vertex Number*
CIT	Born in the U.S.	629
	Born in Puerto Rico, Guam, etc.	630
	Born abroad of U.S parents	631
	U.S. citizen by naturalization	632
	Not a citizen of the U.S.	633

Table A.27: Vertex labeling for PUMS data set, Attribute 10 CIT (*Citizenship status*)

Attribute	*Value*	*Vertex Number*
COW	Employee of a private for profit company.	634
	Employee of a private not-for-profit organisation	635
	Local Government employee	636
	State Government employee	637
	Federal Government employee	638
	Self-employed in own not incorporated business	639
	Working without pay in family business of farm	640
	Unemployed	641

Table A.28: Vertex labeling for PUMS data set, Attribute 11 COW (*Class of Worker*)

Attribute	*Value*	*Vertex Number*
MAR	Married	642
	Widowed	643
	Divorced	644
	Separated	645
	Never married or under 15 years old	646

Table A.29: Vertex labeling for PUMS data set, Attribute 12 MAR (*Marital status*)

Attribute	*Value*	*Vertex Number*	*Attribute*	*Value*	*Vertex Number*
SCHL	No school completed	647	SCHL	High school graduate	655
	Nursery school to grade 4	648		Some college, less than 1 year	656
	Grade 5 or grade 6	649		One or more years of college, no degree	657
	Grade 7 or grade 8	650		Associate's degree	658
	Grade 9	651		Bachelor's degree	659
	Grade 10	652		Master's degree	660
	Grade 11	653		Professional school degree	661
	Grade 12 no diploma	654		Doctorate degree	662

Table A.30: Vertex labeling for PUMS data set, Attribute 13 SCHL (*Educational attainment*

Attribute	*Value*	*Vertex Number*
SEX	Male	663
	Female	664

Table A.31: Vertex labeling for PUMS data set, Attribute 14 SEX (*Sex*)

Bibliography

[1] Alfarez Abdul-Rahman and Stephen Hailes. Relying on trust to find reliable information. In *Proceedings of the International Symposium on Database, Web and Cooperative Systems (DWACOS '99)*, Baden-Baden, Germany, 1999.

[2] Nabil R. Adam and John C. Wortmann. Security-control methods for statistical databases: a comparative study. *ACM Comput. Surv.*, 21(4):515–556, 1989.

[3] Ross Anderson. The decode proposal for an icelandic health database. *The Icelandic Medical Journal*, 84(11):874–875, 1998.

[4] Annie I. Anton, Qingfeng He, and David L. Baumer. Inside jetblue's privacy policy violations. *IEEE Security and Privacy*, 2(6):12–18, 2004.

[5] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[6] Icelandic Data Protection Authority. Security target for an icelandic health database. Internet, 2001.

[7] Ljiljana Brankovic. *Usability of Secure Statistical Databases*. PhD thesis, The University of Newcastle, Australia, 1998.

[8] Ljiljana Brankovic and Henning Fernau. Approximability of a 0,1-matrix problem. In *Proc. of the AWOCA2005*, September 2005.

[9] Ljiljana Brankovic and Helen Giggins. *Security, Privacy and Trust in Modern Data Management*, chapter Statistical Database Security, pages 167–182. Springer, New York, 2007.

[10] Ljiljana Brankovic, Peter Horák, and Mirka Miller. An optimization problem in statistical databases. *SIAM J. Discrete Math.*, 13(3):346–353, 2000.

[11] Ljiljana Brankovic, Peter Horák, and Mirka Miller. An optimization problem in statistical databases. *SIAM Journal on Discrete Mathematics*, 13(2):346–353, 2000.

[12] Ljiljana Brankovic and Mirka Miller. An application of combinatorics to the security of statistical databases. *Australian Mathematical Society Gazette*, 22(4):173–177, 1995.

[13] Ljiljana Brankovic and Mirka Miller. Introduction to statistical database security. *Communications of the CCISA*, 9(4):1–30, 2003. In: Special Issue, Selected Topics of Cryptography and Information Security.

[14] Ljiljana Brankovic, Mirka Miller, and Jozef Širáň. Range query usability of statistical databases. *Int. J. Comp. Math.*, 79(12):1265–1271, 2002.

[15] Ljiljana Brankovic, Mirka Miller, and Jozef Širáň. Towards a practical auditing method for the prevention of statistical database compromise. In *Proceeding of Australasian Database Conference*, 1996.

[16] Ljiljana Brankovic and Jozef Širáň. 2-compromise usability in 1-dimensional statistical databases. In *Proc. 8th Int. Computing and Combinatorics Conference, COCOON2002*, 2002.

[17] Bill Brenner. Txj security breach tied to wi-fi exploits. ComputerWeekly.com: The Internet, May 2007.

[18] T. N. Bui and B. R. Moon. Genetic algorithm and graph partitioning. *IEEE Transactions on Computers*, 45(7):841–855, 1996.

[19] Jim Burridge. Information preserving statistical obfuscation. *Statistics and Computing*, 13:321–327, 2003.

[20] C. Castelfranchi and R. Falcone. The dynamics of trust: From beliefs to action. In *Proc. of Deception, Fraud and Trust in Agent Societies Workshop, Autonomous Agent 1999*, pages 49–60, Seattle, USA, May 1999.

[21] Surajit Chaudhuri and Umeshwar Dayal. An overview of data warehousing and olap technology. *ACM SIGMOD Record*, 26(1):65–74, 1997.

[22] F. Y. Chin and G. Ozsoyoglu. Security in partitioned dynamic statistical databases. In *Proceedings of the IEEE COMPSAC Conference*, pages 594–601, 1979.

[23] F. Y. Chin and G. Ozsoyoglu. Auditing and inference control in statistical databases. *IEEE Transactions on Software Engineering*, SE-8(6):574–582, 1982.

[24] W. J. Clinton. Executive order 13145. Internet, 2000.

[25] Australian Law Reform Commission. *Discussion Paper 66 - Protection of Human Genetic Information*. Commonwealth of Australia, 2001.

[26] Lawrence H. Cox. Network models for complementary cell suppression. *Journal of the American Statistical Association*, pages 1453–1462, 1995.

[27] Partha Dasgupta. *Trust: Making and Breaking Cooperative Relations*, chapter Trust as a Commodity, pages 49–72. Basil Blackwell, New York, USA, 1988.

[28] Peter-Paul de Wolf, Jose M. Gouweleeuw, Peter Kooiman, and Leon Willenborg. Reflections on pram. In *Proceedings of Statistical Data Protection '98*, 1998.

[29] Ronald S. Burt Denise M. Rousseau, Sim B. Sitkin and Colin Camerer. Not so different after all: a cross-discipline view of trust. *The Academy of Management Review*, 23(3):393–404, 1998.

[30] D. E. R. Denning. *Cryptography and Data Security*. Addison-Wesley, 1982.

[31] Dorothy E. Denning and Peter J. Denning. The tracker: a threat to statistical database security. *ACM Trans. Database Syst.*, 4(1):76–96, 1979.

[32] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS'03)*, pages 202–210, New York, NY, USA, 2003. ACM Press.

[33] David P. Dobkin, Anita K. Jones, and Richard J. Lipton. Secure databases: Protection against user influence. *ACM Transactions on Database Systems*, 4(1):97–106, 1979.

[34] J. Domingo-Ferrer, Josep Maria Mateo-Sanz, and Vicenc Torra. Comparing sdc methods for microdata on the basis of information loss and disclosure risk. In *Proceedings of NTTS and ETK*, 2001.

[35] J. Domingo-Ferrer and Vicenc Torra. Disclosure control methods and information loss for microdata. In P. Doyle, J. Lane, J. Theeuwes, and L. Zayatz, editors, *Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies*, pages 93–112. Elsevier, 2002.

[36] Josep Domingo-Ferrer and Josep Maria Mateo-Sanz. Current directions in statistical data protection. *Research in Official Statistics*, 1(2):105–112, 1998.

[37] Josep Domingo-Ferrer, Francesc Sebé, and Jordi Castellà-Roca. On the security of noise addition for privacy in statistical databases. In *Proceedings of Privacy in Statistical Databases: CASC Project International Workshop, PSD 2004*, pages 149–161, Barcelona, Spain, 2004.

[38] Timon C. Du, Fu-Kwun Wang, and Jen-Chuan Ro. the effect of the bootstrap method on addative fixed data perturbation in statistical database. *Omega*, 30:267–279, 2002.

[39] George T. Duncan and Robert W. Pearson. Enhancing access to microdata while protecting confidentiality: Prospects for the future. *Statistical Science*, 6(3):219–232, 1991.

[40] INRA (EUROPE) ECOSA. Eurobarometer 52.1 - the europeans and biotechnology. Internet, 2000.

[41] Rino Falcone and Cristiano Castelfranchi. The socio-cognitive dynamics of trust: Does trust create trust? *Lecture Notes in Computer Science*, 2246:55–72, 2001.

[42] Rino Falcone and Cristiano Castelfranchi. Trust dynamics: How trust is influenced by direct experiences and by trust itself. In *AAMAS*, pages 740–747, 2004.

[43] Office for National Statistics. *200 Years of Census*. 2001.

[44] Luisa Franconi and Silvia Polettini. Individual risk estimation in -argus: A review. In *Proceedings of Privacy in Statistical Databases: CASC Project International Workshop, PSD 2004*, pages 262–272, Barcelona, Spain, 2004.

[45] Theodore D. Friedman and Lance J. Hoffman. Towards a fail-safe approach to secure databases. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 18–21, 1980.

[46] Wayne A. Fuller. Masking procedures for microdata disclosure limitation. *Journal of Official Statistics*, 9(2):383–406, 1993.

[47] Diego Gambetta. *Trust: Making and Breaking Cooperative Relations*, chapter Can We Trust Trust?, pages 213–237. Basil Blackwell, New York, USA, 1988.

[48] Venkatesh Ganti, Johannes Gehrke, and Raghu Ramakrishnan. Cactus -clustering categorical data using summaries. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 73–83, New York, NY, USA, 1999. ACM Press.

[49] Michael R. Garey and David S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman and Company, San Francisco, 1979.

[50] David Gibson, Jon Kleinberg, and Prabhakar Raghavan. Clustering categorical data: an approach based on dynamical systems. *The VLDB Journal*, 8(3-4):222–236, 2000.

[51] Helen Giggins. Protection of sensitive genetic information. Honours Thesis, November 2002. School of Electrical Engineering and Computer Science, The University of Newcastle, Australia.

[52] Helen Giggins and Ljiljana Brankovic. Ethical and privacy issues in genetic databases. In *Proceedings of the Third Australian Instituite of Computer Ethics Conference*, Sydney, Australia, 2002.

[53] Helen Giggins and Ljiljana Brankovic. Protecting privacy in genetic databases. In R. L. May and W. F. Blyth, editors, *Proceedings of the Sixth Engineering Mathematics and Applications Conference*, pages 73–78, Sydney, Australia, 2003.

[54] J. M. Gouweleeuw, P. Kooiman, L. C. R. J. Willenborg, and P. P. de Wolf. Post randomisation for statistical disclosure control: Theory and implementation. *Journal of Official Statistics*, 14(4):463–478, 1998.

[55] Jerrold R. Griggs. Concentrating subset sums at k points. *Bulletin Institute Combinatorics and Applications*, 20:65–74, 1997.

[56] Jerrold R. Griggs. Database security and the distribution of subset sums in $\mathbf{R}^m$. *János Bolyai Math. Soc. 7, Graph Theory and Combinatorial Biology*, pages 223–252, 1999.

[57] Wallis Consulting Group. Community attitudes towards privacy. Internet, August 2007. Prepared for the Office of the Federal Privacy Commissioner. Last Accessed 24th August 2008.

[58] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. *Information Systems*, 25(5):345–366, 2000.

[59] Maria Halkidi. *Data Mining and Knowledge Discovery Handbook*, chapter Chapter 30 - Quality Assessment Approaches in Data Mining. Springer, 2005.

[60] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufman Publishers, 2001.

[61] Health and Medical Research Strategic Review Summary. *The Virtuous Cycle, Working together for health and medical research*. Commonwealth of Australia, 1998.

[62] P. Horak, L. Brankovic, and M. Miller. A combinatorial problem in database security. *Discrete Applied Mathematics*, 91(1-3):119–126, 1999.

[63] U.S. Department of Energy Human Genome Program. Genomics and its impact on science and society: A 2003 primer. Internet, March 2003.

[64] W. H. Inmon. *Building the Data Warehouse*. John Wiley and Sons, fourth edition, 2005.

[65] Ponemon Institute. 2003 privacy trust survey. Web, October 2003. Last visited 23 May 2006.

[66] Md. Zahidul Islam and Ljiljana Brankovic. Detective: A decision tree based categorical value clustering and perturbation technique in privacy preserving data mining. In *Proceedings of the 3rd International IEEE Conference on Industrial Informatics (INDIN 2005)*, Perth, Australia, 2005.

[67] Robert F. Heckard Jessica M. Utts. *Mind on statistics*. Thomson-Brooks/Cole, Belmont, Calif., 2nd edition, 2004.

[68] Audun Jøsang, Claudia Keser, and Theo Dimitrakos. Can we manage trust? *Lecture Notes in Computer Science*, 3477:93107, 2005.

[69] Krishnaram Kenthapadi, Nina Mishra, and Kobbi Nissim. Simulatable auditing. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS'05)*, pages 118–127, New York, NY, USA, 2005. ACM Press.

[70] Jay J. Kim. A method for limiting disclosure in microdata based on random noise and transformation. In *American Statistical Association, Proceedings of the Section on Survey Research Methods*, pages 303–308, 1986.

[71] Jay J. Kim and William E. Winkler. Masking microdata files. In *American Statistical Association, Proceedings of the Section on Survey Research Methods*, pages 114–119, 1995.

[72] Jon Kleinberg, Christos Papadimitriou, and Prabhakar Raghavan. Auditing boolean attributes. *J. Comput. Syst. Sci.*, 66(1):244–253, 2003.

[73] Bruce R. Korf. *Human genetics: A problem-based approach*. Blackwell Science, 1996.

[74] Sverker Jansson Lars Rasmusson. Simulated social control for secure internet commerce. In Catherine Meadows, editor, *Proceedings of the 1996 New Security Paradigms Workshop*. ACM, 1996.

[75] David A. Leiman, Claire Twose, Teresa Y. H. Lee, Alex Fletcher, and Johns Hopkins Terry S. Yoo. Rendering an archive in three dimensions. In *Proceedings of Medical Imaging 2003, Visualization. Image Guided Processing and Display*, San Diego, CA., USA, February 16-21 2003.

[76] Roy J. Lewicki, Daniel J. McAllister, and Robert J. Bies. Trust and distrust: New relationships and realities. *The Academy of Management Review*, 23(3):438–458, 1998.

[77] Yingjiu Li, Lingyu Wang, Xiaoyang Sean Wang, and Sushil Jajodia. Auditing interval-based inference. In *Proceedings of 14th International Conference*

on Advanced Information Systems Engineering, CAiSE 2002, pages 553–567, Toronto, Canada, 2002.

[78] Chong K. Liew, Uinam J. Choi, and Chung J. Liew. A data distortion by probability distribution. *ACM Transactions on Database Systems*, 10(3):395–411, 1985.

[79] Tjen-Sien Lim, Wei-Yin Loh, and Yu-Shan Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Mach. Learn.*, 40(3):203–228, 2000.

[80] Niklas Luhmann. *Trust and Power*. Wiley, Chichester, 1979.

[81] Niklas Luhmann. *Trust: Making and Breaking Cooperative Relations*, chapter Familiarity, Confidence, Trust: Problems and Alternatives, pages 213–237. Basil Blackwell, New York, USA, 1988.

[82] Francesco M. Malvestuto and Mauro Mezzini. Auditing sum queries. In *Proceedings of 9th International Conference on Database Theory, ICDT 2003*, pages 126–142, Siena, Italy, 2003.

[83] Francesco M. Malvestuto and Mauro Mezzini. Privacy preserving and data mining in an on-line statistical database of additive type. In *Proceedings of Privacy in Statistical Databases: CASC Project International Workshop, PSD 2004*, pages 353–365, Barcelona, Spain, 2004.

[84] O. L. Mangasarian and W. H. Wolberg. Cancer diagnosis via linear programming. *SIAM News*, 23(5):1 & 18, 1990.

[85] Stephen Marsh and Mark R. Dibben. Trust, untrust, distrust and mistrust - an exploration of the dark(er) side. In Peter Herrmann, Valérie Issarny, and Simon Shiu, editors, *Proceedings of Third International Conference on Trust Management (iTrust)*, volume 3477 of *Lecture Notes in Computer Science*, pages 17–33. Springer, May 23-26 2005.

[86] Stephen P. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, Department of Computing Science and Mathematics, University of Stirling, April 1994.

[87] Josep Maria Mateo-Sanz, Francesc Sebé, and Josep Domingo-Ferrer. Outlier protection in continuous microdata masking. In *Proceedings of Privacy in Statistical Databases: CASC Project International Workshop, PSD 2004*, pages 201–215, Barcelona, Spain, 2004.

[88] Damien McAullay, Graham Williams, Jie Chen, Huidong Jin, Hongxing He, Ross Sparks, and Chris Kelman. A delivery framework for health data mining

and analytics. In Vladimir Estivill-Castro, editor, *Proceedings of Twenty-Eighth Australasian Computer Science Conference (ACSC2005)*, volume 27, pages 381–387, 2005.

[89] D. Harrison McKnight and Norman L. Chervany. Trust and distrust definitions: One bite at a time. *Lecture Notes in Computer Science*, 2246:27–54, 2001.

[90] Bernadette McSherry. Ethical issues in health*Connect*'s shared electronic health record system. *Journal of Law and Medicine*, 12(1):60–68, 2004.

[91] M. Miller. A model of statistical database compromise incorporating supplementary knowledge. In *Databases in the 1990's*, pages 258–267, 1991.

[92] M. Miller, I. Roberts, and J. Simpson. Application of symmetric chains to an optimization problem in the security of statistical databases. *Bulletin of the ICA*, 2:47–58, 1991.

[93] M. Miller, I. Roberts, and J. Simpson. Prevention of relative compromise in statistical databases using audit expert. *Bulletin of the ICA*, 10:51–62, 1994.

[94] M. Miller and J. Seberry. Relative compromise of statistical databases. *The Australian Computer Journal*, 21(2):56–61, 1989.

[95] James H. Moor. Just consequentialism and computing. *Ethics and information technology*, 1:65–69, 1999.

[96] Lik Mui, Mojdeh Mohtashemi, and Ari Halberstadt. A computational model of trust and reputation. In *Proceedings of the 35th Annual Hawaii Conference on System Sciences*, pages 2431–2439, 2002.

[97] Krishnamurty Muralidhar, Rahul Parsa, and Rathindra Sarathy. A general additive data perturbation method for database security. *Management Science*, 45(10):1399–1415, 1999.

[98] Krishnamurty Muralidhar and Rathindra Sarathy. An enhanced data perturbation approach for small data sets. *Decision Sciences*, 36(3):513–529, 2005.

[99] Krishnamurty Muralidhar, Rathindra Sarathy, and Rahul Parsa. An improved security requirement for data perturbation with implications for e-commerce. *Decision Science*, 32(4):683–698, 2001.

[100] Kieron O'Hara. *Trust: From Socrates to Spin*. Icon Books, Cambridge, UK, 2004.

[101] Michael A. Palladino. *Understanding the Human Genome Project*. Benjamin Cummings, San Francisco, CA, USA, 2002.

[102] Judy Pearsall, editor. *The Concise Oxford Dictionary*. Oxford University Press, New York, USA, tenth, revised edition, 2001.

[103] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, 1993.

[104] James H. Davis Roger C. Mayer and F. David Schoorman. An integrative model of organizational trust. *The Academy of Management Review*, 20(3):709–734, 1995.

[105] Lior Rokach. *Data Mining and Knowledge Discovery Handbook*, chapter Decision Trees. Springer Science & Business Media, Inc., 2005.

[106] Pak Sham. *Statistics in Human Genetics*. Arnold, London, UK, 1998.

[107] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech J.*, 27:379–423, 1948.

[108] C. J. Skinner and M. J. Elliot. A measure of disclosure risk for microdata. *Journal of the Royal Statistical Society*, 64:855–867, 2002.

[109] Ann Sommerville and Veronica English. Genetic privacy: orthodoxy or oxymoron? *Journal of Medical Ethics*, 25(2):144–150, 1999.

[110] Latanya Sweeney. Guaranteeing anonymity when sharing medical data, the datafly system. In *AMIA, Proceedings of Fall Symposium*, pages 51–55, Washington, DC, 1997.

[111] Herman T. Tavani. *Ethics, Computing, and Genomics*, chapter Ethics at the Intersection of Computing and Genomics. Jones and Bartlett Publishers, 2006.

[112] Patrick Tendick. Optimal noise addition for preserving confidentiality in multivariate data. *Journal of Statisical Planning and Inference*, 27:341–353, 1991.

[113] Inc. The TJX Companies. Web site: The tjx companies, inc. The Internet, 2007. http://www.tjx.com/index.html (Last Accessed: 27th August 2007).

[114] Daniel Ting, Stephen Fienberg, and Mario Trottini. Romm methodology for microdata release. In *UNECE/Eurostat Work Session on Statistical Data Confidentiality*, Geneva, Switzerland, 2005.

[115] Dennis Trewin. Managing statistical confidentiality and microdata access - draft principles and guidelines of good practice. In *UNECE/Eurostat Work Session on Statistical Data Confidentiality*, Geneva, Switzerland, 2005.

[116] Mario Trottini. Assessing disclosure risk and data utility: A multiple objectives decision problem. In *Joint ECE/Eurostat Work Session on Statistical Confidentiality*, Luxembourg, 2003.

[117] The Wellcome Trust. Uk biobank: A study of genes, environment and health. Internet. Accessed 29th November 2007.

[118] Peter van Onselen and Wayne Errington. Development and operation: Major party voter databases. In *Proceedings of Australasian Political Studies Association Conference*, Hobart, Australia, 2003.

[119] Poorvi L. Vora. Information theory and the security of binary data perturbation. In *Proceedings of 5th International Conference on Cryptology in India, INDOCRYPT 2004*, pages 136–147, Chennai, India, 2004.

[120] Lingyu Wang, Sushil Jajodia, and Duminda Wijesekera. Securing olap data cubes against privacy breaches. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 161–, 2004.

[121] Lingyu Wang, Yingjiu Li, Duminda Wijesekera, and Sushil Jajodia. Precisely answering multi-dimensional range queries without privacy breaches. In *Proceedings of 8th European Symposium on Research in Computer Security, ESORICS 2003*, pages 100–115, Gjøvik, Norway, 2003.

[122] Lingyu Wang, Duminda Wijesekera, and Sushil Jajodia. Cardinality-based inference control in data cubes. *Journal of Computer Security*, 12(5):655–692, 2004.

[123] Stanley S. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of American Statistical Association*, 60(309):63–69, March 1965.

[124] Leon Willenborg and Ton de Waal. *Statistical Disclosure Control in Practice*. Lecture Notes in Statistics. Springer, New York, USA, 1996.

[125] Leon Willenborg and Ton de Waal. *Elements of Statistical Disclosure Control*. Lecture Notes in Statistics. Springer, New York, USA, 2001.

[126] Adepele Williams and Ken Barker. Controlling inference: Avoiding p-level reduction during analysis. In Ljiljana Brankovic and Chris Steketee, editors, *Fifth Australasian Information Security Workshop (Privacy Enhancing Technologies) (AISW 2007)*, volume 68 of *CRPIT*, pages 193–200, Ballarat, Australia, 2007. ACS.

[127] William E. Winkler. Masking and re-identification methods for public-use microdata: Overview and research problems. In *Proceedings of Privacy in Statistical Databases: CASC Project International Workshop, PSD 2004*, pages 231–246, Barcelona, Spain, 2004.

[128] I. H. Witten, E. Franck, L. Trigg, M. Hall, G. Holmes, and S. J. Cunningham. Weka: Practical machine learning tools and techniques with java implementations. In *Proceedings of ANNES99 International Workshop on emerging*

Engineering and Connectionnist-based Information Systems, pages 192–196, 1999.

[129] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[130] William E. Yancey, William E. Winkler, and Robert H. Creecy, editors. *Disclosure Risk Assessment in Perurbative Microdata Protection*. Lecture Notes in Computer Science: Inference Control in Statistical Databases. Springer, 2002.

[131] Eric Yu and Lin Liu. Modelling trust for system design using the i* strategic actors framework. *Lecture Notes in Computer Science*, 2246:175–194, 2001.

[132] Eric Siu-Kwong Yu. *Modelling strategic relationships for process reengineering*. PhD thesis, University of Toronto, Canada, 1996.