Investigating attribute risk associated with noise-multiplied microdata and noise-infused tabular data, and constructing linkage error model for probabilistically-linked data

Yue Ma

Follow this and additional works at: https://ro.uow.edu.au/theses1

Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following: This work is copyright. Apart from any use permitted under the Copyright Act 1968, no part of this work may be reproduced by any process, nor may any other exclusive right be exercised, without the permission of the author. Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material.

Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

Unless otherwise indicated, the views expressed in this thesis are those of the author and do not necessarily represent the views of the University of Wollongong.

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
Investigating attribute risk associated with noise-multiplied microdata and noise-infused tabular data, and constructing linkage error model for probabilistically-linked data

Yue Ma

This thesis is presented as part of the requirements for the conferral of the degree:

Ph.D of Applied Statistics

Supervisors:
A.Prof Yan-Xia Lin & A. Prof James Chipperfield

Co-supervisor:
Dr. Pavel Krivitsky

The University of Wollongong
School of Mathematics and Applied Statistics

September 2, 2020
Declaration

I, Yue Ma, declare that this thesis is submitted in partial fulfilment of the requirements for the conferral of the degree Ph.D of Applied Statistics, from the University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

Yue Ma

September 2, 2020
The thesis presents our research developments throughout the course of my Ph.D study. My research focuses on two aspects. The first aspect is using data perturbation to achieve data confidentiality. We identified and discussed value disclosure risk issues associated with noise-multiplication masking scheme. The main achievement is that we developed measures which could help a data provider with the process of noise generating variable selection in practice, so that the data provider could produce noise-multiplied data with a desired utility-risk tradeoff. We also studied how output perturbation can be used to protect data privacy in a query system, especially the effect of a differencing attack. We developed a perturbation algorithm which effectively protect against the differencing attack. The second research aspect is analysing probabilistically-linked data. Because data linked by a computerised linkage algorithm contains linkage errors, various approaches have been proposed in the literature for analysing linked data so that unbiased estimates could be obtained. Our research is based on using linkage error model to correct the estimation bias. Our research achievement is that we developed a new linkage error model which is more efficient than other models, and unbiased estimates could be obtained while other models fail to do so.
Acknowledgements

I would like to thank my supervisor, Yan-Xia Lin, for her tremendous help to me towards all my research throughout the course of my Ph.D. study, and insightful comments on the earlier versions of the thesis. I would like to thank my associate supervisor, James Chipperfield, for his tremendous help towards my Ph.D. study and all the opportunities to work with the Australian Bureau of Statistics. I would also like to thank Professor Rathindra Sarathy, my co-supervisor Pavel Krivitsky, and fellow student Bradley Wakefield for all the discussions and feedbacks on the papers I have written.

I would like to thank all anonymous referees for providing valuable feedbacks on all the papers that I have submitted. Their feedbacks greatly helped to improve the quality of my papers. I would like to say that all their feedbacks have been carefully considered when I revised my papers.

I would like to thank the University of Wollongong for providing me with the Ph.D. opportunity. I am very grateful for the support of the Australian Government Research Training Program Scholarship.
# Contents

Abstract iv

1 Introduction 1
   1.1 Data confidentiality .................................................... 1
      1.1.1 Noise multiplication masking method and disclosure risk ...... 5
      1.1.2 Protecting contributor values of business tabular data ...... 7
   1.2 Analysing probabilistically-linked data ............................... 10
   1.3 Research achievement overview and thesis structure ............... 11

2 Noise multiplication masking scheme, data utility loss and disclosure risk measures overview 13
   2.1 Noise-multiplication masking scheme overview .................... 14
      2.1.1 Case 1: $\sigma_C^2$ is public ....................................... 14
      2.1.2 Case 2: $f_C$ is public ........................................... 15
   2.2 Utility loss and disclosure risk measures overview ................ 17

3 Correlation-Attacks on Continuous Microdata 20
   3.1 The correlation-attack strategy ....................................... 22
   3.2 Discussion on $\hat{Y}_i$ and $Y_i^*$ .................................... 23
   3.3 Disclosure risk and data utility loss measures ..................... 27
   3.4 Simulations .................................................................... 31
      3.4.1 Simulation 1 ............................................................. 31
      3.4.2 Simulation 2 ............................................................. 35
   3.5 Conclusion and future work .............................................. 40

4 Simulations studies on noise candidates selection 43
   4.1 Simulation 1 ................................................................. 44
      4.1.1 Simulation 2 ............................................................. 47
      4.1.2 Simulation 3 ............................................................. 51
      4.1.3 Simulation 4 ............................................................. 52
   4.2 Application .................................................................. 54
4.3 Conclusion and Future work ...................................... 56

5 Quantifying protection level of a noise candidate 57
5.1 Definition of overall data utility loss ......................... 58
5.2 The optimal estimator ............................................ 59
5.3 Derivation of $Z_{opt}^i$ ............................................. 61
5.4 Finding $R_{opt}(y, C, \delta)$ ....................................... 62
5.5 Discussion .......................................................... 64
5.6 Simulation study .................................................... 66
5.7 Conclusion .......................................................... 68

6 Attacking Noise-multiplied Data given extrema information 72
6.1 Definition of overall data utility loss ......................... 73
6.2 Estimating the extrema of the original data from the noise-multiplied data 73
6.3 An attacking strategy .............................................. 74
6.4 Data provider’s approaches against the proposed attacking strategy .... 78
6.5 Simulation .......................................................... 80
   6.5.1 Simulation 1 ..................................................... 80
   6.5.2 Simulation 2 ..................................................... 82
   6.5.3 Simulation 3 ..................................................... 85
6.6 Conclusion and future work ..................................... 86

7 An Algorithm for Protecting Tabulated Business Data 88
7.1 Attacking scenarios ............................................... 90
7.2 Measuring Disclosure Risk and Utility Loss ............... 91
   7.2.1 Differencing Attack ........................................ 91
   7.2.2 Defining Disclosure ....................................... 92
   7.2.3 Defining Utility Loss ..................................... 92
7.3 The algorithm in Thompson et al. (2013) .................... 93
7.4 A New Algorithm to Generate Perturbed Cell Estimates ... 97
7.5 The new algorithm v.s. Thompson et al. algorithm ........ 101
   7.5.1 Controlled Utility Loss of Perturbing a Cell .......... 101
   7.5.2 A Conjectured Wider Applicability ................... 101
   7.5.3 Better Utility-Disclosure Trade-offs .................. 103
7.6 Conclusion .......................................................... 104

8 Conditional linkage error model 108
8.1 Latent model and probabilistic-linkage algorithm.......... 112
   8.1.1 Linking fields and agreement pattern ................ 112
   8.1.2 Latent model and match weight ....................... 113
8.1.3 Probabilistic Linkage algorithm .................................. 114
8.1.4 Blocking variable ..................................................... 114

8.2 Using linkage error model to correct estimation bias ................. 115
  8.2.1 LEM overview ....................................................... 115
  8.2.2 Linear Regression ................................................ 116
  8.2.3 Logistic Regression .............................................. 116

8.3 Conditional Linkage Error Model ..................................... 116
  8.3.1 ELE model .......................................................... 117
  8.3.2 Conditional Linkage Error model .............................. 118

8.4 Comparison between CLE, ELE and MEM ............................ 120
  8.4.1 Numerical example ................................................. 120
  8.4.2 Matching Error Model ............................................ 121
  8.4.3 Lahiri and Larsen (2005)’s MEM ............................... 121
  8.4.4 Unbiased estimators using MEM ............................... 122
  8.4.5 Discussion on using MEM to correct estimation bias ........ 122

8.5 Simulation ................................................................ 124

8.6 Conclusion ................................................................. 127

9 Conclusion ................................................................ 128
Chapter 1

Introduction

Statistical agencies such as the Australian Bureau of Statistics collect a large number of data from the public through different means such as sample surveys. To fulfill the role of public services, a statistical agency may either disseminate the collected data to the public for data analysis, or use the data itself to gain knowledge of the population for public issues. When carrying out these two jobs, there are two practical issues faced by the statistical agency. The first issue is data confidentiality. As is required by law in many countries, the statistical agency needs to ensure that when releasing data to the public, private information of data respondents such as personal incomes, are not revealed to the public. The second issue is longitudinal data analysis. That is, the information of the same data respondent may appear in different sample surveys. Consequently, the response variable and the explanatory variables of a regression model may appear in different datasets. It makes data analysis difficult as the statistical agency cannot always match records in different datasets perfectly. The thesis presents my developments on these two issues. The following two sections introduce each issue separately, and outline my research area on each topic.

1.1 Data confidentiality

Data can be released to the public either as microdata or as tabular summaries. Microdata contains information of data respondents across several attributes, such as personal income. Identifiers associated with each record, such as the social security numbers, are removed from the microdata in order to prevent direct identification of a record. Tabular data contains aggregate information of several data respondents, such as the annual income of all businesses in an area. Releasing microdata provides data users with a wider range of statistical analysis than tabular data but has higher disclosure risks than releasing tabulated data because it contains detailed information of a data respondent. In practice, the statistical agency may decide the type of data to be released to the public according to
CHAPTER 1. INTRODUCTION

Table 1.1: Hypothetical microdata released by a statistical agency.

<table>
<thead>
<tr>
<th>Record</th>
<th>Age</th>
<th>Gender</th>
<th>Postcode</th>
<th>Marital Status</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record1</td>
<td>58</td>
<td>M</td>
<td>2224</td>
<td>Divorced</td>
<td>40000</td>
</tr>
<tr>
<td>Record2</td>
<td>55</td>
<td>M</td>
<td>2210</td>
<td>Divorced</td>
<td>45000</td>
</tr>
<tr>
<td>Record3</td>
<td>31</td>
<td>F</td>
<td>2001</td>
<td>Married</td>
<td>48000</td>
</tr>
<tr>
<td>Record4</td>
<td>32</td>
<td>F</td>
<td>2552</td>
<td>Married</td>
<td>59000</td>
</tr>
<tr>
<td>Record5</td>
<td>29</td>
<td>M</td>
<td>2313</td>
<td>Single</td>
<td>35000</td>
</tr>
<tr>
<td>Record6</td>
<td>25</td>
<td>M</td>
<td>2220</td>
<td>Single</td>
<td>36000</td>
</tr>
</tbody>
</table>

the sensitivity of the data. For instance, the Australian Bureau of Statistics only release tabulated business data to the public as business data are considered highly sensitive. On the contrary, the U.S. Census Bureau releases household microdata containing household incomes, property tax, etc to the public by means of microdata because household information is less sensitive.

Ideally statistical agency only releases data to legitimate data users for data analysis. A legitimate data user is interested in knowing population information from released data. However, a malicious data intruder is interested in knowing private information of data respondents (such as a person or a business). Some measures have been adopted by statistical agencies, including the Office for the National Statistics in the United Kingdom to only allow approved researchers to have access to sensitive data (Abrahams and Mahony 2008). However, it is often the case that both legitimate data users and malicious data intruders have access to data released to the public by a statistical agency.

Broadly speaking, there are two types of risks: identity risk and attribute risk. Identity risk is the risk that a record in microdata is correctly associated with the sampling unit in the population by a data intruder. In terms of microdata, identity disclosure occurs if a malicious data intruder correctly re-identifies a sampling unit in a microdata. For instance, A quasi-identifier is an attribute which is not a unique identifier, but could be used in combination with other quasi-identifiers to uniquely identify a record. A data intruder might be able to learn the identity of a particular record using quasi-identifiers. Examples of quasi-identifiers are age, sex and education status. To illustrate identity disclosure, suppose a statistical agency releases a set of hypothetical microdata in Table 1.1. A data intruder knows that his target sampling unit is in the microdata, and he knows that the sampling unit has values (29, M, Single) across three attributes (Age, Gender, Marital Status). Then the data intruder could link the sampling unit to Record 5 in Table 1.1. In that case, Record 5 is re-identified. Sweeney (2000) showed that, 18% of the population in the United States could be re-identified using a combination of three quasi-identifiers: county, gender and date of birth.

Attribute risk is the risk that an attribute value of a sampling unit is disclosed by a data
intruder. Attribute disclosure occurs if something new about about an attribute of a data respondent is learnt by a data intruder. For the above example, the data intruder learns the income value of his target sampling unit from the released microdata. As a result, the attribute value of income for the sampling unit is disclosed. Attribute disclosure was discussed in the U.S. Department of Commerce (1978). Identity disclosure and attribute disclosure were discussed and compared in Skinner (1992).

Tables might be created for count data or magnitude data. For count data, each table cell shows the number of observations which satisfy certain criteria. For instance, a table cell might be the total number of sampling units which are male and less than 18 years old. For magnitude data, each table cell provides the aggregate value of an attribute over all sampling units contributing to the cell. For instance, a magnitude table cell might be the total annual profit across several businesses in a particular area. For tabular data, identity disclosure might occur if the number of contributor values to a table cell is small. For instance, a table cell might be the total annual profit of all mining businesses in a remote area. However there is only one mining business in that area. In that case the identity of the contributor value to the cell is disclosed. Another example is that, suppose a cell of business totals contains three contributors, and the cell value is 100,000. The data intruder knows that the business value of the largest business in the sampling frame is over 80,000 and the business values of all the other businesses are below 20,000. In that case the data intruder knows that the largest business is included in the cell.

Attribute disclosure might occur if a data intruder learns the value of a target unit using an attacking strategy. For instance, suppose a data intruder wants to learn a confidential value of a target business. Suppose the data intruder knew that there are two table cells of business totals: Cell 1 and Cell 2. The contributor units for these two cells are the same except that the target business is in Cell 1 but not in Cell 2. In this case, the data intruder might infer the confidential value of the target business by using the value of Cell 1 minus the value of Cell 2. For a table cell, a dominance rule might be used to evaluate attribute risk for the table cell. For instance, in the Australian Bureau of Statistics, a dominance rule finds cells where the cell value is dominated by a small number of data respondents. If a cell fails this rule, then further investigation is needed to ensure that attributes of predominant data respondents are not disclosed.

Data quality describes whether a set of data contains rich and accurate information. Good data quality means that the data is representative of the real-world and it fits the intended purpose of use. Data users could extract accurate population information from high quality data and use statistical outputs for various purposes. Data utility describes the amount of population information carried in a data release. A higher data utility means a higher analytical validity, i.e. data users could extract more accurate population information from the data. Data sensitivity concerns with information of data respondents that need to be protected against unauthorised disclosure. Sensitive data, such as an attribute
Table 1.2: Hypothetical microdata released by a statistical agency.

<table>
<thead>
<tr>
<th>Record</th>
<th>Age</th>
<th>Gender</th>
<th>Postcode</th>
<th>Marital Status</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record1</td>
<td>≥ 45</td>
<td>M *</td>
<td>Divorced</td>
<td>≥ 40000</td>
<td></td>
</tr>
<tr>
<td>Record2</td>
<td>≥ 45</td>
<td>M *</td>
<td>Divorced</td>
<td>≥ 40000</td>
<td></td>
</tr>
<tr>
<td>Record3</td>
<td>30 ≤ Age &lt; 45</td>
<td>F *</td>
<td>Married</td>
<td>≥ 40000</td>
<td></td>
</tr>
<tr>
<td>Record4</td>
<td>30 ≤ Age &lt; 45</td>
<td>F *</td>
<td>Married</td>
<td>≥ 40000</td>
<td></td>
</tr>
<tr>
<td>Record5</td>
<td>≤ 30</td>
<td>M *</td>
<td>Single</td>
<td>30000 &lt; Income &lt; 40000</td>
<td></td>
</tr>
<tr>
<td>Record6</td>
<td>≤ 30</td>
<td>M *</td>
<td>Single</td>
<td>30000 &lt; Income &lt; 40000</td>
<td></td>
</tr>
</tbody>
</table>

of a person, must be protected in a data release. When releasing data to the public, the statistical agency needs to ensure that legitimate data users could obtain population information from released data, while sensitive information of survey respondents has low disclosure risk. To achieve these two objectives, a common practice used by the statistical agency is to mask the data first, and then release masked data to the public. The masked data normally has a lower level of disclosure risk at the expense of a higher data utility loss. In other words, there is a disclosure-utility tradeoff in the masked data. Duncan (2002) discussed the tension between data utility and data sensitivity and technical procedures for addressing the tension.

To mask a set of original data, the statistical agency might apply one or a combination of the Statistical Disclosure Control (SDC) methods. Different SDC methods achieve the disclosure-utility tradeoff differently. Commonly used SDC methods for both microdata and tabular data include micro-aggregation, suppression, data perturbation, etc. For microdata, additional SDC methods include synthetic data, data shuffling (Muralidhar and Sarathy 2006), top and bottom coding, k-anonymity (Sweeney 2002), noise-addition (Kim 1986) and noise-multiplication. Top and bottom coding has been studied in many papers, e.g. Klein et al. (2014). More information on could be found in Willenborg and de Waal (2001) and Hundepool et al. (2012). Each SDC method attempts to protect sensitive data by either reducing identification risk (such as k-anonymity), or by reducing attribute risk (such as top and bottom coding), or both (such as noise-addition and noise-multiplication). For instance, k-anonymity masks microdata in a way that, for each record, there are at least $k - 1$ other records with the same combination of attributes value. To achieve 2-anonymity for Table 1.1, the statistical agency could anonymise Table 1.1 to produce Table 1.2. In this case a date intruder with knowledge of a sampling unit (29, M, Single) for attributes (Age, Gender, Marital Status) will only have 50% chance of correctly link the sampling unit to the corresponding record.

For each set of original data, the statistical agency needs to determine an appropriate SDC method to be used so that the masked data achieves desired disclosure-utility tradeoff. In order for statistical agencies to make correct decisions, the capability of each SDC
method in terms of its disclosure risk control and utility preservation needs to be fully understood.

Utility loss measures could help statistical agencies to evaluate the amount of information carried in the masked data. Utility loss measure is not unique, and there is no universal measure for it. Different utility loss measures are proposed under specific contexts, see Shlomo (2010), Domingo-Ferrer and Torra (2001) and Agrawal and Aggarwal (2001) for instance. In this thesis, as we consider different masking scenarios in each chapter, we will introduce different utility loss measures for illustration purpose in each chapter. We note that the main focus of the thesis is on the development of disclosure risk control.

My research area is based on two aspects: the first aspect is on quantifying disclosure risk associated with the noise-multiplication masking scheme for microdata. It involves identification of potential disclosure risks associated with releasing noise-multiplied data to the public, and development of disclosure risk measures which could be used by statistical agencies for decision-making when producing noise-multiplied data. Another aspect is on developing output perturbation methods to protect contributor values of tabulated business data. The following two subsections introduces the background of the two aspects in detail.

1.1.1 Noise multiplication masking method and disclosure risk

Multiplicative noise is a data perturbation method for numeric data. Using multiplicative noise for data protection has been advocated by many researchers because of its appealing features. Multiplicative noise provide uniform protection, in terms of the coefficient of variation of the noises, to all sensitive observations (Nayak et al. 2011). Multiplicative noise are also more suitable for economic modeling of income data in some situations (Kim and Winkler 2003). The masking mechanism is easy to implement in practice and a balanced utility-risk tradeoff is achieved by selecting an appropriate noise generating variable, which is referred to as “tuning mechanism” in Klein et al. (2014). The masking method has been used in practice by the U.S. Energy Information Administration and the U.S. Bureau of Census (Kim and Jeong 2008).

The masking scheme protects the original data by perturbing each original observation by multiplying it with a random noise term generated from a noise generating variable $C$. The data provider releases the noise-multiplied data together with the density function, $f_c$, of the noise generating variable $C$ to the public. Methodologies for analysing noise-multiplied data have been developed by taking into account the information of $f_c$ such that population parameter estimates could be recovered from noise-multiplied data (Nayak et al. 2011; Sinha et al. 2011; Lin and Wise 2012; Klein et al. 2014). We will review some of the analysing methods in Chapter 2.
The noise multiplication masking method could significantly reduce identification rate caused by record-linkage techniques if numeric attributes are involved in the linkage process (Kim and Winkler 2003; Oganian and Karr 2011; Muralidhar and Domingo-Ferrer 2016). More on record linkage will be introduced later. However, the masking method may not reduce identity disclosure risk caused by quasi-identifiers if they are non-numeric. For instance, a data intruder might just use non-numeric quasi-identifiers, such as (Gender, Marital Status, Education level, Postcode) to re-identify a record. It is unlike k-anonymity, which could be applied to both numeric and categorical data to reduce re-identification risk caused by quasi-identifiers. To reduce re-identification risk, a statistical agency might use a combination of multiplicative noise and k-anonymity. That is, the statistical agency might choose to apply multiplicative noise on numeric attributes and k-anonymity on categorical attributes.

In this thesis we focus on attribute disclosure. We assume that a numeric variable in a microdata is considered sensitive. The numeric variable is positive, and protected by multiplicative noise. The density function of noise generating variable is released along with the microdata to the public. To learn the sensitive attribute value of a target sampling unit, a data intruder might attempt to unmask a target value from its noise-multiplied counterpart. Attribute disclosure might occur if the intruder re-identified a target sampling unit using non-numeric quasi-identifiers, and infers the target attribute value from its noise-multiplied counterpart with high accuracy. To reduce attribute risk in this case, we aim to perturb the attribute in a way that it is difficult for a data intruder to infer a target value with high accuracy. For the data intruder, he might have no prior knowledge about the unperturbed microdata at all, or he might have some knowledge about the unperturbed microdata. We will consider different cases in each chapter.

To reduce attribute risk, two things need to be determined. The first is intrusion behaviours. That is, to infer the sensitive value of a target record which is protected by multiplicative noise, a data intruder might attempt to develop reasonable attacking strategies to unmask the target value. The statistical agency might need to be aware of these attacking strategies. The second is disclosure risk measure. That is, the statistical agency needs a way to quantify attribute risk associated with each original value. However, as noted in Nayak et al. (2011), evaluating attribute risk is difficult because different data intruders may have different target values as well as different prior knowledge about the original data. Based on the level of knowledge, the data intruder might use different attacking strategies to unmask a target value. To understand attribute risk of a data masking mechanism, a common approach in the literature is to model intrusion behaviour. For instance, certain intrusion behaviours have been developed for unmasking noise-added data (Agrawal and Srikant 2000; Domingo-Ferrer et al. 2004; Liu et al. 2008). Correspondingly, appropriate actions could be made during data masking stages such that the released masked data is protected against these behaviours.
CHAPTER 1. INTRODUCTION

Under the context of multiplicative noise perturbation, some intrusion behaviours have been modelled under specific contexts. For instance, Nayak et al. (2011) considered the scenario that multiplicative noises are used to perturb contributor values of tabular data, and modelled an intrusion behaviour for disclosing a target contributor value of a table cell. In terms of using multiplicative noises to perturb microdata, Klein et al. (2014) considered the scenario where a response variable is sensitive and masked by multiplicative noises while explanatory variables are unmasked and public. The data intruder targets the value of the response variable using the knowledge of explanatory variables. The authors showed that the data intruder may use the predicted value based on the generalised regression model to estimate a target value. Another intrusion behaviour is recognised in Nayak et al. (2011) and Lin and Wise (2012), that the data provider may simply use the noise-multiplied value as an estimate of an original value, provided that the mean of the noise generating variable is 1. This is because in this case, the noise-multiplied value is in fact an unbiased estimate of the original value. This fact is also recognised in Kim (2007) and Kim and Jeong (2008) and these authors proposed several noise distributions in order to reduce the disclosure risk. Corresponding disclosure risk measures against these intrusion behaviours were proposed in Klein et al. (2014) and Lin and Wise (2012) so that data providers could control the disclosure risks by choosing a suitable noise generating variable during data masking stage.

In this thesis, we study several attacking strategies that might be attempted by a data intruder. Consequently, we propose solutions to reduce attribute risk against these attacking strategies. We identify three intrusion behaviours, under specific contexts, and we propose disclosure risk measures or other disclosure avoidance methods which help statistical agencies to make appropriate decisions during the data masking stage, especially the noise generating variable selection process. Our developments might help statistical agencies to produce more secured noise-multiplied data in practice. We will present these developments in Chapter 3, 4, 5 and 6.

1.1.2 Protecting contributor values of business tabular data

Business data is difficult to mask because of its nature. Typically, some industries will be dominated by large businesses whose information is difficult to conceal by existing data masking methods. Non-perturbative data masking methods, such as top coding (Klein et al. 2014), suppression (Salazar-González 2005) and micro-aggregation (Defays and Nanopoulos 1993), significantly reduce information of continuous data items such as turnover or profit, which are of key interest to data users. Perturbation methods, such as data swapping (Moore 1996), synthetic data (Rubin 1993) and noise addition (Kim and Winkler 1995), cannot efficiently protect businesses with distinct continuous-valued characteristics. As a result, most statistical agencies have taken a cautious approach to
releasing business data, and the majority of business data is still released in the form of broad-level tables.

The emergence of remote access facilitates but imposes a challenge of releasing business microdata. Remote access (Blakemore 2001; Reiter 2004) is a virtual system that provides a data analyst with access to a remote system built by a statistical agency. The statistical agency stores microdata in the remote system, and the data analyst communicates with the remote system through a query system. The analyst is restricted from viewing the underlying microdata. Instead, the analyst could only obtain statistical outputs of underlying microdata through the following model (see O’Keefe and Chipperfield 2013; Chipperfield and O’Keefe 2014): (1) an analyst submits a query (i.e. request for a table) to the remote system; (2) the remote system modifies or restricts estimates using an automatic algorithm; (3) the system sends the modified output to the analyst. An example of remote access system is American FactFinder (Hawala et al. 2004), which releases perturbed tabulations of census data to data users.

In this thesis we consider algorithms for producing perturbed business totals via a remote system. We assume a remote system could allow users to specify and request tables created from business microdata. This is considered in TableBuilder, a remote system built by the Australian Bureau of Statistics (Chipperfield et al. 2018). Each cell of a table contains a perturbed survey estimate of total computed from a set of surveyed business values (such as annual revenue) specified by a data user. The set of surveyed business values are called contributor values to a table cell.

The reason for a remote system to release perturbed statistics is to prevent disclosure of confidential values via various methods of attack, the most significant of which is a differencing attack (Lucero et al 2009, Sect. 4.1). A differencing attack reveals a confidential value by taking the difference of two cell totals whose contributing values differ by one. A differencing attack could be very effective on a remote system as the attacker is able to obtain statistical outputs of different underlying microdata with a high degree of freedom. Therefore, a safe and effective automatic perturbation algorithm is needed to produce perturbed cell totals against differencing attacks.

Perturbation of tabular data could perform either through input perturbation (Evens. 1996; Nayak et al. 2011), or through output perturbation (Dwork et al. 2006; Chipperfield et al. 2019). For input perturbation, a random noise with expectation 0 is added to each original business value to produce a perturbed business value. As a result, a set of perturbed business data is produced from the underlying business data. The perturbed business data is stored in the remote system and all responses of queries are calculated based on the perturbed data. For output perturbation, the response to a query is calculated based on the underlying original data first. Then, a random noise with expectation 0 is produced and added to the true response to produce a perturbed response. The advantages of using output perturbation for business totals in a remote system are outlined in
Chipperfield et al. (2019): 1. because the estimates, not the underlying micro-data, are modified, relationships in the micro-data are essentially retained; 2. the degree to which an estimate is modified depends upon the output itself. For example, modification of an estimate may be relatively high if a cell is dominated by a single business and relatively low if a table cell has many small businesses of roughly equal size; 3. because an analyst is restricted from viewing the micro-data, less modification is needed than would otherwise be the case; 4. it allows users to gain rapid access to estimates they request; 5. the modification algorithm assures a specified level of protection is guaranteed if certain restrictions on the remote system are imposed. We note that for Point 4, as soon as the user submits a query, the remote system will generate a perturbation amount to be used for output perturbation. This process will cause delay for users to gain outputs. We also note that Point 5 requires certain restrictions to be imposed on a query system. There are concerns with a remote system when it allows an infinite number of queries to be sent. One example is the tracker problem. It is recognised by many researchers (e.g. Sarathy and Muralidhar 2011). Roughly speaking, the tracker problem states that as the number of queries goes large, the protection level offered by a query system to an original value could go down dramatically. Another example is that a data intruder might refine his estimate of a target value by sending multiple identical queries. For instance, to learn the value of \( x_1 \), the data intruder could send multiple queries for the total of \((x_1, x_2, x_3)\), and then send multiple queries for the total of \((x_2, x_3)\). The intruder’s estimate of \( x_1 \) is the average of the responses for \( x_1 + x_2 + x_3 \) minus the average of the responses for \( x_2 + x_3 \). As the expectation of random perturbation to each table cell is 0, by increasing the number of repeated queries, the data intruder is able to refine his estimate of \( x_1 \). For the repeated query problem, the remote system used by the Australian Bureau of Statistics works in a way that the same perturbation amount is always used to perturb identical queries. To protect against the tracker problem, a statistical agency might need to add more and more noise as the number of queries increases, or impose restrictions on the total number of queries a data user could send. In the thesis we assume certain restrictions are imposed on a remote system to address these problems.

Thompson et al. (2013) proposed an algorithm for perturbing cells of business totals. The algorithm adds noise to a table cell according to the top few contributor values to the cell. The algorithm ensures that, as the contribution of the top contributors gets larger, a larger amount of perturbation is generated. Chipperfield et al. (2019) discussed this algorithm and showed that it effectively protects the top contributor values against various attacking strategies. Another potential output perturbation algorithm, which adds Laplace noises to table cells (Dwork et al. 2006), does not work well for business totals as doing so might cause a large amount of information loss (see Sarathy and Muralidhar 2011 for numerical examples). In Chapter 7 of the thesis, we review Thompson et al.’s perturbation algorithm, and we propose a new algorithm which could be more effective against
differencing attacks for a large amount of table cells.

1.2 Analysing probabilistically-linked data

Statistical agencies often have access to administrative datasets about a set of individuals. The aim of record linkage is to identify and consequently link records, across two (or more) data sets, that contain the same set of individuals. Linking datasets allows more statistical analysis to be performed because the linked dataset contains more analysis variables than in each individual dataset. Record-linkage is the procedure of finding records belonging to the same entity (such as individual or business) across different data files. Perfect linkage, i.e. records of the same entity will be linked, might be difficult to achieve if unique identifiers such as social security number are not available.

A record pair consists of one record from each file. A record pair is said to be a match if the records in the pair belong to the same entity. To find matches, Fellegi and Sunter (1969) introduced a record linkage model for linking two data files when unique identifiers are not available. The decision rule of the model is based on agreement/disagreement across certain attributes which are common to both data files. These attributes are called linking fields. Typical linking fields include date of birth, age, gender, etc. A record pair does not need to agree exactly on a linking field in order for the record pair to be an agreement on the linking field. A comparison function defines what constitutes an agreement. A comparison function could require linking field values of two records to be exactly the same in order for them to be an agreement on the linking field. Alternatively, a comparison function might require the distance between two linking field values to be less than a pre-specified amount in order to classify them as an agreement on the linking field. Christen and Churches (2005) discussed different comparison functions.

The agreement pattern of a record pair is an indicator vector of \( \{x_l\}_{l=1}^k \), where \( x_l = 1 \) if the pair agrees on the \( l \)-th linking field and 0 otherwise, \( k \) is the total number of linking fields. Under the Fellegi-Sunter record linkage model, an agreement pattern is a realisation from either one of two latent classes: Match Class or Unmatch Class. Each record pair has a match weight. It is the \( \log_2 \) of the ratio of the probability of the agreement pattern given that the record pair is a match and the probability of the agreement pattern given that the record pair is not a match. More on the latent model will be described in Chapter 8. A record pair is classified as a match if the match weight is over a threshold value, and is classified as a non-match if the match weight is below a threshold value.

Chipperfield et al. (2018) described a probabilistic linkage algorithm. It is assumed that each record on one file has at most one matching record on the other file. The probabilistic linkage algorithm makes 1-1 assignment. That is, each record on one file could only be linked to at most one record on the other file. A record pair is said to be a link if the pair is linked by the probabilistic linkage algorithm. However, a link is not necessarily a match.
There is a chance that a non-match record pair is linked. In other words, linkage errors might be present in a linked dataset.

Analysing linked data with linkage errors will lead to biased inferences if linkage errors are ignored. It is studies in many contexts, e.g. Scheuren and Winkler (1993). Adjustments need to be made to standard analysing procedure in order for unbiased inferences to be made. Scheuren and Winkler (1993) showed that unbiased regression coefficients estimates could be obtained if the probability for a given record pair to be a match is known. Many models for estimating this probability have been proposed in the literature. Following Chipperfield (2019), a model for estimating this probability is called a Match Error Model (MEM). Lahiri and Larsen (hereafter referred to as LL) (2005) proposed an MEM which estimates this probability by using the posterior probability for a record pair to be a match given the observed agreement pattern. LL’s MEM is studied in Hof and Zwinderman (2012).

Chambers (2009) considered the case where two data files are linked by the probabilistic linkage algorithm. The author considered the scenario that a “secondary file” is linked to a “benchmark file”. The sampling units in the secondary file is a subset of the sampling units in the benchmark file. The author showed that unbiased regression estimate could be obtained if the probability for a sampling unit in the benchmark file to be linked to a sampling unit in the secondary file could be calculated. Following Chipperfield (2019), a model for estimating this probability is called a link Error Model (LEM). Chambers (2009) proposed an Exchangeable Linkage Error (ELE) model for estimating this probability. Under the ELE model, all matching record pairs have the same probability of being linked, and all non-matching record pairs have the same probability of being linked.

There are limitations with both LL’s MEM model and the ELE model. Chipperfield and Chambers (2015) showed that estimates obtained using the MEM model can be biased. For the ELE model, our simulation showed that it will lead to biased regression coefficients estimates in some cases.

In Chapter 8 we introduce a new LEM. The new LEM is constructed by conditioning on the observed linking fields on the benchmark files. We will compare the new LEM with the ELE model and LL’s MEM via simulation. We will show that the new LEM outperforms both models.

1.3 Research achievement overview and thesis structure

There have been five papers came out of my research. The first paper is about a new algorithm for perturbing business data in a query system. The paper was presented in Privacy in Statistical Databases (PSD) 2016 conference and was published in PSD2016 conference proceedings (Ma et al. 2016). The paper is an extension of a joint work between the Australian Bureau of Statistics and us (Chipperfield et al. 2019). The content of our work
is presented in Chapter 7. The second paper is the identification of “correlation-attack” which could be used to attack noise-multiplied data. The paper has been accepted by Sankhya B (Ma et al. 2019). The content of the paper is presented in Chapter 3. The third paper regards value disclosure risk of noise-multiplied data should the knowledge of the maximum or minimum of the original data be possessed by data intruders. The paper had been submitted to Transactions on Data Privacy (TDP), and is now being revised and will be send back to TDP for reconsideration of publication. The content of the paper is presented in Chapter 6. The fourth paper is about quantification of the protection level associated with a noise generating variable for noise-multiplication masking method. The paper was accepted by PSD 2018 and was presented in the conference and published in the conference proceedings (Ma et al. 2018). The content of the paper is presented in Chapter 5. The fifth paper is about a linkage error model which corrects estimation bias of probabilistically-linked data. This is a working paper. The main content of the paper was presented in a workshop of the Australian Bureau of Statistics, and is presented in Chapter 8.

Here is the outline of the thesis: Chapter 2 overviews the noise multiplication masking method, utility loss and disclosure risk measures; Chapter 3 introduces the correlation-attack which could be used to attack noise-multiplied data; Chapter 4 investigates the performance of different noise variables against the correlation-attack through simulations; Chapter 5 proposes a way for quantifying the protection level a noise generating variable offers to the original data; Chapter 6 introduces another value disclosure risk associated with releasing noise-multiplied data to the public by recognising that the knowledge of the maximum or the minimum of the original data could be learnt by some data intruders; Chapter 7 introduces a new output perturbation algorithm for tabulated business totals; Chapter 8 introduces a new linkage error model for obtaining unbiased estimates from probabilistically-linked data; Chapter 9 concludes the thesis.
Chapter 2

Noise multiplication masking scheme, data utility loss and disclosure risk measures overview

There are several ways for using multiplication noises to create a perturbed version of the original data. For instance, Oganian and Karr (2011) considered using multiplicative noises to perturb the original data first, and then apply a linear transformation on the noise-multiplied data to create a final set of perturbed data. The final perturbed data could be regarded as a synthetic version of the original data. In the final perturbed data, the first two moments properties of the original data are preserved, which means that data users could obtain this information by applying standard data analysis on the perturbed data. Another method is called Matrix Multiplicative Data Perturbation (MMDP) (Liu et al. 2008). MMDP uses an orthogonal matrix to be multiplied with the original data matrix to create a perturbed data matrix. The perturbed data matrix preserves Euclidean distances for any two data columns. As a result, data users could apply data mining algorithms to the perturbed data matrix to obtain results similar to the ones obtained from the original data matrix. The multiplication masking method we consider in the thesis follows Kim and Winkler (2003), Nayak et al. (2011), Lin and Wise (2012) and references therein. The features of our noise-multiplication masking method are: 1. Independent noises are generated from an underlying noise generating variable, and are multiplied by each original observation to create noise-multiplied data; 2. The data provider releases the noise-multiplied data together with the partial or full information about the masking procedure to the public, such as the density function of the noise generating variable; 3. Data users use analysing methods which are developed for analysing noise-multiplied data exclusively, so that statistical estimates of the population could be recovered from the noise-multiplied data.
CHAPTER 2. NOISE MULTIPLICATION MASKING SCHEME, DATA UTILITY LOSS AND DISCLOSURE RISK MEASURES OVERVIEW

2.1 Noise-multiplication masking scheme overview

In the following we describe the mathematical setup of the noise multiplication masking method and review methodologies for obtaining statistical inferences from noise-multiplied data. These inferential methods are important for defining utility loss measures we adopt in this thesis. We base our discussion on one-dimensional data. For a set of univariate original data, the noise multiplication masking method works as follows:

Suppose a set of original data \( y = \{y_i\}_{i=1}^n \) are independent realizations from \( Y \). To mask \( y \), the data provider chooses a noise generating variable \( C \) with \( E(C) = 1 \). \( Y \) and \( C \) are independent. A set of noise terms \( c = \{c_i\}_{i=1}^n \) are independently drawn from \( C \) and multiplied with \( y \) to produce the noise-multiplied data \( y^* = \{y_i^*\}_{i=1}^n = \{y_i c_i\}_{i=1}^n \). \( \{y_i^*\}_{i=1}^n \) could be treated as independent realizations from \( Y^* \), where \( Y^* = YC \). The data provider releases \( y^* \) together with other information of \( C \) (such as variance \( \sigma_C^2 \) or density function \( f_C \)) to the public. We assume both \( Y \) and \( C \) are positive and continuous throughout the thesis.

We note that it is possible for multiplicative noises to be only applied to sensitive values of an attribute, leaving the non-sensitive values unchanged. For instance, Klein et al. (2014) considered the case where only extreme values are masked using multiplicative noises. The results we derived in this thesis could also be applied to these cases.

In the following, we review methods for analysing noise-multiplied data assuming that all original values are perturbed by multiplicative noises. We consider two scenarios: in the first scenario, the data provider releases \( y^* \) and \( \sigma_C^2 \) to the public. This setting is considered in Kim and Winkler (2003) and Lin and Wise (2012); in the second scenario, the data provider releases \( y^* \) and \( f_C \) to the public. This scenario is considered more generally because releasing \( f_C \) allows data users to recover more statistical information from the noise-multiplied data.

2.1.1 Case 1: \( \sigma_C^2 \) is public

If \( \sigma_C^2 \) is public, Nayak et al. (2011) showed that the first two moments of \( Y \) could be unbiasedly recovered from \( Y^* \). That is

\[
E(Y) = E(Y^*)
\]

and

\[
E(Y^2) = \frac{E[(Y^*)^2]}{E(C^2)} = \frac{E[(Y^*)^2]}{(\sigma_C^2 + 1)}.
\]
Denote $E(Y) = \mu_Y$ and $\text{Var}(Y) = \sigma_Y^2$. The authors also showed that, $\mu_Y$ and $\sigma_Y^2$ could be unbiasedly estimated by $\hat{\mu}_Y$ and $\hat{\sigma}_Y^2$ using $y^*$, where

$$\hat{\mu}_Y = \frac{\sum_{i=1}^{n} y_i^*}{n}$$  \hspace{1cm} (2.1)$$

and

$$\hat{\sigma}_Y^2 = \frac{1}{n(n-1)}[(n+\sigma_C^2)\sum_{i=1}^{n} (y_i^*)^2 - (\sum_{i=1}^{n} y_i^*)^2].$$  \hspace{1cm} (2.2)$$

In summary, when $\sigma_C^2$ is public, data users could unbiasedly recover estimates of the first two moments and variance of $Y$. However, it is not true for other information of $Y$, such as quantiles and higher order moments as estimating this information requires $f_C$ to be public (Nayak et al. 2011; Sinha et al. 2011; Lin 2014).

### 2.1.2 Case 2: $f_C$ is public

When $f_C$ is public, data users could obtain unbiased higher order moments estimates of $Y$, and quantile estimates of $Y$ in addition to those introduced before. Moreover, releasing $f_C$ allows estimates of the density function of $Y$ using reconstruction algorithms.

**Moments estimation:** Nayak et al. (2011) showed that all higher order moments of $Y$ could be unbiasedly obtained from the noise-multiplied data. Specifically, since the original moments $E[Y^j] = E[Y^*|j]/E[C^j]$, where $j$ is a positive integer. Therefore, for large-sized noise-multiplied data, data users could estimate any finite $j$-th moment of $Y$ using $\frac{\sum_{i=1}^{n} y_i^*}{nE[C]}$, where $E[C^j]$ is obtained directly from the density function of $C$ and $\frac{\sum_{i=1}^{n} y_i^*}{n}$ is the sample estimate of $E[Y^*|j]$.

**Quantile estimation:** Sinha et al. (2011) proposed a Bayesian approach to estimate quantiles of $Y$ from the noise multiplied data $y^*$. Specifically, the conditional posterior distribution of $Y_i|y_i^*$ has the following form:

$$Y_i|y_i^* \sim \frac{1}{y_i} f_C(\frac{y_i^*}{y_i})\pi(y_i)$$  \hspace{1cm} (2.3)$$

where $\pi(y_i)$ is the prior of $Y_i$. Denote the posterior random vector as $\{Y_i|y_i^*\}_{i=1}^{n}$, the data intruder could then draw $N$ replicates of samples from $\{Y_i|y_i^*\}_{i=1}^{n}$. Denote the $j$-th replicate as $(\hat{y}_{i,j}, \hat{y}_{2,j}, \cdots, \hat{y}_{nj})$, where $\hat{y}_{ij}$ is a realization from $Y_i|y_i^*$ for $i = 1, 2, \cdots, n$. Denote further that the ordered $j$-th replicate as $(\hat{y}_{[1],j}, \hat{y}_{[2],j}, \cdots, \hat{y}_{[nj],j})$, where $\hat{y}_{[ij]}$ is the $i$-th order statistic in the $j$-th replicate. The authors suggest using the $N$ replicates to estimate order statistics for the original sample. For instance, to estimate $y_{[1]}^*$, the smallest order statistic in $y$, the data intruder could use $N$ the replicates of $\{\hat{y}_{[1],j}\}_{j=1}^{N}$, and use their mean and standard deviation to construct a point estimate as well as a confidence interval.

**Density reconstruction algorithm:** Using reconstruction algorithms allows data users to obtain density approximations of $Y$ from perturbed data (noise-added data, noise-
multiplied data). Under our context of multiplicative noise perturbation, Lin (2014) proposed a sample-moment-density reconstruction algorithm. The author showed that, the original density $f_Y$ could be expressed as $f_Y(y) = \sum_{k=1}^{\infty} \lambda_k P_k(y)$, where $\lambda_k$ is a function of all moments up to the $k$-th moment of $Y$, and $P_k(y)$ is a Legendre polynomial of degree $k$ in $y$. Consequently, the density function $f_Y$ could be estimated by $f_{Y,K}(y) = \sum_{k=0}^{K} \lambda_k P_k(y)$, where $K$ is the order of the moments of $Y$ that are used to approximate $f_Y$.

As data users only observe $y^*$ and $f_C$, the information of the moments of $Y$ have to be approximated by $y^*$ and $f_C$. This could be done using the moments estimation method we introduced before. As a result, the original density $f_Y(y)$ could be estimated by $f_{Y,K|y^*,C}(y) = \sum_{k=0}^{K} a_k(y) \frac{(y^*)^k}{nE[C^k]}$, where $a_k(y)$ is a continuous function of $y$ and order $k$. Lin (2014) showed that the density approximation $f_{Y,K|y^*,C}(y)$ uniformly converges almost surely to the true underlying density function of the original variable $f_Y(y)$. Lin and Fielding (2015) developed an R-package MaskDensity14 for data users to implement the density reconstruction algorithm on noise-multiplied data in R. We will use MaskDensity14 to estimate the density of $Y$ for all our simulations in the thesis. Quantile estimates of $Y$ could also be obtained if $f_Y$ could be estimated by $f_{Y,K}$. In that case, data users could estimate quantiles of $Y$ based on $f_{Y,K}$. The advantage of estimating quantiles in this way over Sinha et al.’s method is that it requires no prior knowledge about the original data. Numerical examples in Lin (2014) and Lin and Fielding (2015) showed that accurate quantile estimates could be obtained in this way. Also see Lin (2018) for more information.

**MaskDensity14:** We use Lin’s density reconstruction R-program for two purposes in the thesis: 1. From the data provider’s perspective, we use the density reconstruction algorithm to evaluate the overall data utility loss for noise-multiplication masking scheme. The overall data utility loss will be introduced in Chapter 5; 2. From data intruder’s perspective, we investigate how density reconstruction could be used to estimate the bounds of $y$.

The inputs for MaskDensity14 in R are: 1. the noise multiplied data $y^*$. 2. a large noise file which contains random noise drawn from $C$. For large enough noise files it is almost equivalent to releasing $f_C$ to the public. 3. parameters $a$ and $b$, where $a$ is the lower bound of the density function $Y$, $b$ is the upper bound of the density function of $Y$. Lin and Fielding (2015) suggested that the noise.bin file should be consisted of the noise terms which were used to mask the original data. In order for the noise file to be large enough (say 100,000), each noise term might be replicated a random number of times. All the noise terms along with their identical copies are stored in the file The noise terms and parameters $a$ and $b$ are concealed in a ‘noise.bin’ file which is only readable by MaskDensity14. Even though the underlying density function of $Y$ might be unbounded, Lin (2014) showed that if the sample size is large, a very good approximation of $f_Y$ could be achieved if $a$ is chosen to be the minimum of the original data, and $b$ is chosen to be the
maximum of the original data. Lin and Fielding (2015) showed through empirical study that the program performs better if \( a \) is slightly lower than the minimum of the original data, and \( b \) is slightly larger than the maximum of the original data. The noise.bin file is supplied by the data provider. The R-package has been improved recently, see Lin and Krivitsky (2018) and Lin et al. (2018).

To illustrate the program, suppose we have a set of original data \( \{y_i\}_{i=1}^{1000} \) independently drawn from \( Y \sim U[100,500] \). We use a set of noise terms \( \{c_i\}_{i=1}^{1000} \) independently drawn from \( C \sim U[0.5,1.5] \) to mask the original data. The noise-multiplied data is denoted as \( \{y^*_i\}_{i=1}^{1000} \). \( f_Y \) is bounded between 100 and 500, so we let \( a = 100 \) and \( b = 500 \). The noise file contains 100,000 noise terms made from \( \{c_i\}_{i=1}^{1000} \). As a result, the noise.bin file contains 100,000 noise terms as well as \( a \) and \( b \). We obtain three density estimates. The first density estimate \( K_Y \) is the density estimate of \( Y \) using kernel density approximation using the original data; the second density estimate \( \hat{f}_{Y^*} \) is the kernel density estimate of the noise-multiplied variable \( Y^* \) using the noise multiplied data; the third density estimate \( \hat{f}_Y \) is the density estimate of \( Y \) using the Maskdensity14. The three density plots are given in Figure 2.1.

2.2 Utility loss and disclosure risk measures overview

For perturbed microdata, it suffers from both identification disclosure risk and attribute risk. The study of identification disclosure for perturbed microdata mainly focus on how the matching rate would be if record-linkage algorithm is used to link the records in the original microdata with the records in the perturbed microdata. It is normally assumed
that a data intruder has the true knowledge of some records of the original data from other public sources (Kim and Winkler 2003; Duncan et al. 2004). Identification occurs if the data intruder correctly matches an original record with a perturbed record using certain unmasked linking fields. Studies on how data perturbation reduces identification rate could be found in Kim and Winkler (1995, 2003); Oganian and Karr (2011); Muralidhar and Domingo-Ferrer (2016). Should identification of a record occur, the only protection to the original value is the infused noise. Predictive disclosure, or value disclosure risk, is the risk that a masked value of a sampling unit might be reasonably inferred by a data intruder using released data. A low value disclosure risk means that, even if a record is being identified by a data intruder, the private information associated with the record could not be confidently learned by the data intruder. Discussions on value disclosure risk can be found in Nayak et al. (2011), Lin and Wise (2012), Klein et al. (2014) and Ma et al. (2018).

In this thesis we mainly focus on value disclosure risk. For a masking mechanism, we aim to find ways to reduce value disclosure risk for each record. In various discussions in the literature, value disclosure occurs if a data intruder’s estimate of a target value is close to the true value (see Duncan et al. (2004), Sarathy and Muralidhar (2011), Liu et al. (2008) and references therein). For continuous data, which is the focus of our research, it is tough to guess the exact value of the original datum, and it is not necessary to achieve it. Using an Acceptance Rule to decide if an estimate \( \hat{y}_i \) can be accepted as a correct guessing of the original value \( y_i \) is sufficient. A formal definition of value disclosure is introduced in Lin and Wise (2012), which states the following:

*Suppose a data intruder uses \( \hat{y}_i \) as an estimate of the original value \( y_i \). The expression of \( \hat{y}_i \) is determined by his own attacking strategy. To classify the \( \hat{y}_i \) as a valid estimate of \( y_i \), it is sufficient for \( \hat{y}_i \) to be reasonably close to \( y_i \). Disclosure of \( y_i \) occurs if \( \hat{y}_i \) satisfies \( \left| \frac{\hat{y}_i - y_i}{y_i} \right| \leq \delta \), where \( \delta \) is the acceptance rule and is a small positive number. For instance, for a positive observation \( y_i \), if we set \( \delta = 0.05 \), we say that \( \hat{y}_i \) discloses \( y_i \) if \( 0.95y_i \leq \hat{y}_i \leq 1.05y_i \). The value of \( \delta \) is determined by the data provider according to the requirement of disclosure risk control.*

Using acceptance rule to define value disclosure is adopted by the Australian Bureau of Statistics for tabulated business data (Chipperfield et al. 2018). Based on this definition, Lin and Wise (2012), Klein et al. (2014) and Chipperfield et al. (2018) considered a probabilistic value disclosure risk measure for evaluating the disclosure risk of \( y_i \) against an attacking estimator \( \hat{Y}_i \). That is:

\[
R(\hat{Y}_i, \delta|Y_i = y_i) = P\left(\frac{|\hat{Y}_i - Y_i|}{Y_i} < \delta|Y_i = y_i\right),
\]

where \( \hat{Y}_i \) is an attacking estimator which could be used by a data intruder to attack the value of \( y_i \). This disclosure risk measure is to be used by a statistical agency for assessing
the value disclosure risk associated with each original value $y_i$. Other ways for quantifying value disclosure risks include using confidence intervals (Agrawal and Srikant 2000; Nayak et al. 2011) and using the mean squared errors (Duncan et al. 2004) for the same purpose. In the thesis, we adopt the probabilistic value disclosure risk for quality control of a masking mechanism.

In the following we review data utility loss measures. Data users could only obtain perturbed estimates from masked data. The perturbed estimates are normally less accurate than the ones data users would obtain by analysing the original data (or unperturbed estimate). Data utility loss measures such loss of accuracy for a population parameter due to data masking. An **overall data utility loss** is an aggregate measure of the utility losses across several parameters of the population.

There is no unique way to measure overall data utility loss. In the literature, the way of measuring overall data utility loss varies according to different data masking scenarios as well as which parameters estimates could be recovered from masked data. For instance, Yancey et al. (2002) proposed to use the average of relative distances between perturbed estimates and unperturbed estimates across several parameters as an overall data utility loss measure under the context of additive-noise perturbation. Those parameters include population means, covariances and correlations, as those parameter estimates could be accurately recovered from noise-added microdata. Other overall utility loss measures are available, see Shlomo (2010), Domingo-Ferrer and Torra (2001) and Agrawal and Aggarwal (2001). We note that in different masking scenarios, the definitions of overall data utility loss could be different because different masking scenarios preserve different pieces of information of the population. In the thesis, the discussion on data utility loss is for illustration purposes only in order to achieve a balanced discussion. Therefore, we will define different utility loss measures in each chapter in order to facilitate our discussion.
Chapter 3

Correlation-Attacks on Continuous Microdata

For univariate noise-multiplied data, Nayak et al. (2011) and Lin and Wise (2012) showed that the data provider may simply use the noise-multiplied value as an estimate of an original value because the noise-multiplied variable is in fact an unbiased estimator of the original value. This fact is also recognised in Kim (2007) and Kim and Jeong (2008) and the authors proposed several noise distributions so that the corresponding value disclosure risk could be controlled. This chapter introduces an intrusion behaviour for attacking noise-multiplied data. The intrusion behaviour only uses information of the noise-multiplied data and $\sigma_C$. Because of its simplicity, the attacking strategy could always be used as an attempt to unveil a noise-multiplied value. The attacking strategy exploits the correlation between the original data and the noise-multiplied data, therefore we name it as “correlation-attack”. The intuition behind the correlation-attack is that if the correlation between the original data and the noise-multiplied data is high, then a simple linear regression model might be adequate to explain the relationship between the two sets of data. The correlation between the original data and the noise-multiplied data could be reasonably estimated by a data intruder, as we show later. On the basis of the mean squared errors (MSEs), both the unbiased estimator and the correlation-attack estimator tend to be more accurate as the correlation gets larger. Therefore, a high correlation value might motivate a data intruder to use either the unbiased estimate or the correlation-attack estimate to disclose a target value. Correspondingly, we propose a disclosure risk measure to be used by data providers, so that for each original value, the disclosure risks from both estimators could be simultaneously evaluated. The disclosure risk measure could help data providers with noise generating variable selection during data masking stage. We note that other regression models might also be considered to explain the relationship between the original data and the noise-multiplied data, but we do not consider them in this chapter.

The idea of using regression models to attack private values has also been consid-
Li and Sarkar (2011) considered the possibility of using a regression tree to attack a protected value of a target record. By treating the target perturbed attribute as a response variable and other perturbed attributes as explanatory variables, the authors showed that it is possible to reveal the private information of a target record. The authors referred to this attack as “regression-attack”. The regression-attack is similar to the idea of using the predicted value based on a generalised linear regression model to attack a noise-multiplied value proposed in Klein et al. (2014). We feel that the major difference between the regression-attack and our correlation-attack is that the correlation-attack only uses information of a noise-multiplied attribute itself to attack a noise-multiplied value in the attribute. We note that when several attributes in a microdata are protected by multiplicative noises, the idea of the regression-attack might be used to attack a particular noise-multiplied value. In that case, the regressors of an attacking regression model are some or all attributes (either noise-multiplied or not) in the microdata. This idea could be explored in the future.

We assume all original data and noise candidates are positive and continuous, and all noise candidates have expectation 1. The sample size of the original data is large enough so that estimates of population parameters, such as population variance, could be accurately recovered from the noise-multiplied data. We assume the following simple scenario: the data provider releases the noise-multiplied data together with the variance of the noise generating variable used to perturb a set of original data to the public (same as the assumption in Section 2.1.1), so that data users could unbiasedly recover the first two moments estimates of the population (i.e. population mean and variance) from the noise-multiplied data. We note that assuming preservation of the first two moments estimates only is consistent with many other masking schemes in the literature (Brand 2002; Yancey et al. 2002; Kim and Winkler 2003; Oganian and Karr 2011). For instance, the data perturbation method proposed in Oganian and Karr (2011) only allows data users to obtain unbiased estimates of population means and covariance matrix from perturbed microdata. The assumption simplifies discussion of this chapter especially when we define overall data utility loss. The correlation-attack proposed in this chapter could also be used under Case 2 scenario specified in Section 2.1.2.

This chapter is organized as follows: Section 3.1 introduces the correlation-attack strategy. Section 3.2 discusses the correlation-attack estimator and the unbiased estimator. Section 3.3 proposes a disclosure risk measure which could be used by data providers for noise generating variable selection and introduces the definition of overall data utility loss we adopt in this paper. Section 3.4 presents two simulation studies. Section 3.5 concludes the chapter.
3.1 The correlation-attack strategy

In this section we introduce the correlation-attack strategy which could be used by a data intruder to obtain an estimate of an unobservable original value $y_i$. The correlation-attack follows from the idea that, if the sample correlation between $y^*$ and $y$ is high, then a simple linear regression model may adequately explain the relationship between $y$ and $y^*$. Consequently, the data intruder could use the predicted value based on the simple linear regression model to estimate a target original value $y_i$ from $y_i^*$. 

From the data intruder’s perspective, the original data $y$ is unobservable. Therefore, the sample correlation between $y$ and $y^*$, denoted as $r_{yy^*}$, cannot be known. However, when the sample size $n$ is large, $r_{yy^*}$ is close to $\rho_{YY^*}$, where $\rho_{YY^*}$ is the population correlation coefficient between $Y$ and $Y^*$. It can be shown that $\rho_{YY^*}$ takes the following form:

$$\rho_{YY^*} = \frac{\text{Cov}(Y^*, Y)}{\sqrt{\text{Var}(Y^*)\text{Var}(Y)}} = \sqrt{\frac{E(CY^2) - E(C)E(Y)}{\text{Var}(CY)}} = \sqrt{\frac{E(Y^2) - E(Y)^2}{E(C^2Y^2) - E(C)^2E(Y)^2}} = \sqrt{\frac{\sigma_Y^2}{E(C^2)E(Y^2) - E(Y)^2}} = \sqrt{\frac{(\sigma_C^2 + 1)(\sigma_Y^2 + \mu_Y^2) - \mu_Y^2}{\sigma_C^2(\sigma_C^2 + 1) + \mu_Y^2\sigma_C^2}}$$

(3.1)

In the above expression, $\mu_Y$ and $\sigma_Y^2$ are unknown. However, the data intruder could unbiasedly estimate these two terms from $y^*$ by $\hat{\mu}_Y$ and $\hat{\sigma}_Y^2$ using Equation (2.1) and (2.2). Therefore, $\rho_{YY^*}$ could be approximated by $\tilde{r}_{yy^*}$, where

$$\tilde{r}_{yy^*} = \sqrt{\frac{\hat{\sigma}_Y^2}{\hat{\sigma}_Y^2(\hat{\sigma}_C^2 + 1) + \hat{\mu}_Y^2\hat{\sigma}_C^2}}$$

(3.2)

As a result, if the sample size $n$ is large, then $\tilde{r}_{yy^*}$ could be used to approximate the sample correlation $r_{yy^*}$. If $\tilde{r}_{yy^*}$ is large, then it might motivate the data intruder to fit a simple linear regression model between $y$ and $y^*$, and to use the predicted value based on the linear model to attack the unobservable $y_i$. The predicted value of $y_i$ based on $y_i^*$ has the following expression:

$$\hat{y}_i = \hat{\alpha} + \hat{\beta}y_i^*,$$

where $\hat{\alpha}$ and $\hat{\beta}$ are least squares estimates of the intercept and slope terms. The estimates
can be evaluated as follows:
\[ \hat{\beta} = r_{yy^*} \frac{s_y}{s_{y^*}} \]
and
\[ \hat{\alpha} = \bar{y} - \hat{\beta} \bar{y^*}, \]
where \( s_y^2 \) and \( s_{y^*}^2 \) are sample variances of \( y \) and \( y^* \), respectively; \( \bar{y} \) and \( \bar{y^*} \) are sample means of \( y \) and \( y^* \), respectively.

For the data intruder, \( \bar{y}, s_y \) and \( r_{yy^*} \) cannot be known directly as \( y \) is not available. Therefore \( \hat{\alpha} \) and \( \hat{\beta} \) cannot be obtained directly. However, when the sample size \( n \) is large, \( \bar{y} \) is close to \( \mu_y \), \( s_y \) is close to \( \sigma_y \), and \( r_{yy^*} \) is close to \( \rho_{yy^*} \). The values \((\mu_y, \sigma_y, \rho_{yy^*})\) could be approximated from \( y^* \) by \((\hat{\mu}_y, \hat{\sigma}_y, \hat{r}_{yy^*})\) using Equation (2.1), (2.2) and (3.2). Therefore, the unknown quantities \( \bar{y}, s_y \) and \( r_{yy^*} \) could be approximated by \( \hat{\mu}_y, \hat{\sigma}_y \) and \( \hat{r}_{yy^*} \) respectively. As a result, the least squares estimates \( \hat{\alpha} \) and \( \hat{\beta} \) could be approximated by \( \tilde{\alpha} \) and \( \tilde{\beta} \), where
\[ \tilde{\beta} = \hat{r}_{yy^*} \frac{\hat{\sigma}_y}{s_{y^*}}, \]
and
\[ \tilde{\alpha} = (1 - \hat{r}_{yy^*} \frac{\hat{\sigma}_y}{s_{y^*}}) \bar{y^*}. \]

Therefore, the correlation-attack estimate \( \tilde{y}_i \) which could be obtained by the data intruder to attack \( y_i \) takes the following form:
\[ \tilde{y}_i = \tilde{\alpha} + \tilde{\beta} y_i^* = (1 - \hat{r}_{yy^*} \frac{\hat{\sigma}_y}{s_{y^*}}) \bar{y^*} + \hat{r}_{yy^*} \frac{\hat{\sigma}_y}{s_{y^*}} y_i^*. \]

### 3.2 Discussion on \( \tilde{Y}_i \) and \( Y_i^* \)

In this section we discuss the correlation-attack estimator \( \tilde{Y}_i \) and the unbiased estimator \( Y_i^* \) for estimating a target value \( y_i \). We discuss the two estimators together because both \( \tilde{y}_i \) and \( y_i^* \) could easily be obtained by a data intruder to attack \( y_i \) given basic knowledge of \( y^* \) and \( \sigma_{C_i}^2 \). Because of the generality, we propose that the disclosure risks from both estimators should always be evaluated and controlled for most noise-multiplied data. We base our discussion on the mean squared errors (MSEs) of these estimators.

From the last section, we introduced the correlation-attack estimate \( \tilde{y}_i \). We denote the correlation-attack estimator as \( \tilde{Y}_i \). The exact expression for \( \tilde{Y}_i \) cannot be easily found without making a large sample assumption. By noting that when sample size is large, \( \hat{r}_{yy^*} \frac{\hat{\sigma}_y}{s_{y^*}} \) is close to \( \rho_{yy^*} \frac{\sigma_y}{\sigma_{y^*}} = \rho_{yy^*}^2 \), and \( \bar{y^*} \) is close to \( \mu_y \). Then, \( \tilde{Y}_i \) is approximated to be the following
\[ \tilde{Y}_i \approx (1 - \rho_{yy^*}^2) \mu_y + \rho_{yy^*}^2 y_i^*. \] (3.3)
We use this expression of \( \tilde{Y}_i \) for discussion throughout this section. On the other hand, the unbiased estimator \( Y_i^* \) is simply the noise-multiplied variable and \( Y_i^* = Y_iC_i \). It is an unbiased estimator of \( y_i \) given \( Y_i = y_i \) because \( E(Y_i^*|Y_i = y_i) = y_i \). It can be seen directly that \( \tilde{Y}_i \) is partially determined by \( Y_i^* \). The MSE of the correlation-attack estimator \( \tilde{Y}_i \) conditional on \( Y_i = y_i \) is given as:

\[
MSE(\tilde{Y}_i|Y_i = y_i) = E \left[ (1 - \rho_{YY}^2)\mu_y + \rho_{YY}^2CY_i - Y_i \right]^2|Y_i = y_i = (1 - \rho_{YY}^2)^2(\mu_y - y_i)^2 + \rho_{YY}^4y_i^2\sigma_C^2.
\]

The MSE of the unbiased estimator \( Y_i^* \) conditional on \( Y_i = y_i \) is given as:

\[
MSE(Y_i^*|Y_i = y_i) = E[(CY_i - Y_i)^2|Y_i = y_i] = y_i^2E(C - 1)^2 = y_i^2\sigma_C^2.
\]

**Accuracies of \( \tilde{Y}_i \) and \( Y_i^* \):** We first consider a special case where \( \rho_{YY}^* \) is approximately equal to 1. We see that based on Equation (3.3), if \( \rho_{YY}^* \) is approximately equal to 1, then the two estimators \( \tilde{Y}_i \) and \( Y_i^* \) are approximately equal. We note that from Equation (3.1), \( \sigma_C^2 = \frac{1 - \rho_{YY}^2}{\rho_{YY}^2(1 + \mu_y^2/\sigma_y^2)} \). Therefore, a \( \rho_{YY}^* \) close to 1 indicates that \( \sigma_C^2 \) is close to 0, which means that all noise terms stay very close around 1. Therefore, in this case the multiplicative noises do not provide enough protection to the original data, as both the correlation-attack estimate and the unbiased estimate are approximately equal to the unmasked \( y_i \).

We now consider a general case where \( \rho_{YY}^* \in [0, 1] \). As \( \sigma_C^2 = \frac{1 - \rho_{YY}^2}{\rho_{YY}^2(1 + \mu_y^2/\sigma_y^2)} \), therefore \( \sigma_C^2 \) and \( \rho_{YY}^* \) have an inverse relationship. The MSEs of both estimators tend to decrease as \( \sigma_C^2 \) decreases, meaning that they tend to be more accurate for estimating \( y_i \) as \( \rho_{YY}^* \) gets larger. It is straightforward to see it for \( Y_i^* \) based on its MSE. However, it is not straightforward to see it for \( \tilde{Y}_i \). To show this, we substitute \( \sigma_C^2 \) by \( \frac{1 - \rho_{YY}^2}{\rho_{YY}^2(1 + \mu_y^2/\sigma_y^2)} \) in the expression of \( MSE(\tilde{Y}_i|Y_i = y_i) \), so we have

\[
MSE(\tilde{Y}_i|Y_i = y_i) = (k_1 - k_2)\rho_{YY}^4 + (k_2 - 2k_1)\rho_{YY}^2 + k_1,
\]

where \( k_1 = (\mu_y - y_i)^2 \), \( k_2 = \frac{\mu_y^2}{1 + h} \), \( h = \mu_y^2/\sigma_y^2 \). So the MSE is a parabola in terms of \( \rho_{YY}^2 \). The symmetric axis is \( S = 1 + \frac{k_2}{2(k_1 - k_2)} \). The parabola is monotonically decreasing in \( \rho_{YY}^* \in [0, 1] \) in almost all cases, meaning that the correlation-attack estimator \( \tilde{Y}_i \) is more accurate as \( \rho_{YY}^* \) increases. The only exception is when \( k_2 > 2k_1 \). Under this condition, the symmetric axis \( S \) is within \([0,1]\) so the function is not monotone in \([0,1]\). That means when \( y_i \in \left( \frac{2\mu_y(1 + h) - \mu_y \sqrt{2(1 + h)}}{1 + 2h}, \frac{2\mu_y(1 + h) + \mu_y \sqrt{2(1 + h)}}{1 + 2h} \right) \), \( MSE(\tilde{Y}_i|Y_i = y_i) \) will increase first as \( \rho_{YY}^* \) goes from 0 to \( S \), but it eventually decreases to 0 as \( \rho_{YY}^* \) increases from \( S \) to 1. Therefore, both estimators tend to be more accurate as \( \rho_{YY}^* \) gets larger. A large \( \rho_{YY}^* \) value might motivate a data intruder to use either the unbiased estimator or the
correlation-attack estimator to attack $y_i$.

**Comparison between $\tilde{Y}_i$ and $Y^*_i$:** We first note that, the absolute relative distance between the two estimators are given by the following:

$$ARD = \left| \frac{\tilde{Y}_i - Y^*_i}{Y^*_i} \right| = \left| \frac{(1 - \rho_{YY^*}^2)\mu_Y}{Y^*_i} + \rho_{YY^*}^2 - 1 \right|$$

Based on $ARD$, if we wish the relative distance between the correlation-attack estimate and the unbiased estimate to be less than $\gamma$ (a small number such as 0.05), the following condition needs to be satisfied:

**Condition 1:**

$$-\gamma + 1 - \frac{\mu_Y}{Y^*_i} < (1 - \frac{\mu_Y}{Y^*_i})\rho_{YY^*}^2 < \gamma + 1 - \frac{\mu_Y}{Y^*_i}$$

Given a small $\gamma$, if the above condition is satisfied, then it does not matter much which attacking estimate is to be used to attack $y_i$, as the two estimates are very close to each other.

An estimator predicts the unknown value better if it yields a smaller MSE. To compare which estimator predicts $y_i$ better, we set $MSE(\tilde{Y}_i|Y_i = y_i) - MSE(Y^*_i|Y_i = y_i) < 0$. Those conditions which satisfy this inequality mean that under the conditions, the correlation-attack estimator $\tilde{Y}_i$ predicts $y_i$ with better accuracy. After solving the inequality, we let

$$a = \frac{\sigma_Y^2 - \mu_Y \sqrt{\sigma_Y^2 + \mu_Y^2}}{\mu_Y^2},$$
$$b = \frac{\sigma_Y^2 + \mu_Y \sqrt{\sigma_Y^2 + \mu_Y^2}}{\mu_Y^2},$$
$$c = \frac{\mu_Y \rho_{YY^*}^2 (\sigma_Y^2 + \mu_Y^2) - \mu_Y \sigma_Y \rho_{YY^*} \sqrt{(1 + \rho_{YY^*}^2) (\sigma_Y^2 + \mu_Y^2)}}{\rho_{YY^*}^2 \mu_Y^2 - \sigma_Y^2},$$
$$d = \frac{\mu_Y \rho_{YY^*}^2 (\sigma_Y^2 + \mu_Y^2) + \mu_Y \sigma_Y \rho_{YY^*} \sqrt{(1 + \rho_{YY^*}^2) (\sigma_Y^2 + \mu_Y^2)}}{\rho_{YY^*}^2 \mu_Y^2 - \sigma_Y^2}.$$

$e = \min(c,d)$ and $f = \max(c,d)$. Then we have the following result:

**Result 1:** Based on MSEs, for large-sized sample, when $\rho_{YY^*} < a$ or $\rho_{YY^*} > b$, an observation $y_i$ is more vulnerable to $\tilde{Y}_i$ if $y_i \in (e,f)$; when $\rho_{YY^*} \in (a,b)$, $y_i$ is more vulnerable to $\tilde{Y}_i$ if $y_i < e$ or $y_i > f$; when $\rho_{YY^*}$ equals $a$ or $b$, $y_i$ is more vulnerable to $\tilde{Y}_i$ if $y_i > \mu_Y/2$. 

Depending on the distribution of the original data and the noise variance, the difference of the MSEs of the two estimators varies and could be substantial. For instance, if the original variable \( Y \sim U(100,600) \), and noise variance \( \sigma_c^2 \) is 0.433 such that \( \rho_{YY^*} = 0.5 \), the ratio of \( MSE(Y^*_i | Y_i = y_i) \) over \( MSE(\tilde{Y}_i | Y_i = y_i) \) could go over 15 if an original observation \( y_i \) takes value around 350. It means that the correlation-attack estimate is far more accurate than the unbiased estimate according to the magnitudes of the MSEs. On the contrary, when \( y_i \) takes value around 100, the ratio is only around 0.125, meaning that the unbiased estimate is far more accurate according to the MSEs. The ratio of the two MSEs is around 1 if \( y_i \) takes value around 200, meaning that the two estimates perform similarly according to the magnitudes of the MSEs.

Based on the above results, we have the following discussions from both the data intruder and the data provider’s perspectives:

Discussion 1: If \( \rho_{YY^*} \) is high, then the data intruder may use either the correlation-attack estimator or the unbiased estimator to attack a target value \( y_i \). The values of \((a,b,e,f)\) depend on three parameters \((\mu_Y, \sigma_Y, \rho_{YY^*})\). From the data intruder’s perspective, these parameters could easily be estimated from \( y^* \) using Equations (2.1), (2.2) and (3.2). Therefore, the data intruder could obtain estimates \((\hat{a}, \hat{b}, \hat{e}, \hat{f})\), and use Result 1 to make a decision on which particular estimator should be used for attacking \( y_i \). In order to do this, the data intruder needs to make an initial guess about the location of \( y_i \) in terms of \((\hat{e}, \hat{f})\). For instance, if the data intruder’s estimate \( \hat{e}_{yy^*} \) is within \((\hat{a}, \hat{b})\), and the data intruder has a strong belief that \( y_i \) is greater than \( \hat{f} \), then logically speaking the data intruder would use the correlation-attack estimate to attack the value of \( y_i \). In the simulation study in Section 3.4.2, it can be shown that the original values over 26317.6 are more vulnerable to the correlation-attack estimator according to Result 1. If a noise-multiplied value is significantly greater than 26317.6, say 200000, it is unlikely that the corresponding original value is less than 26317.6. Therefore, it is very likely that the data intruder uses the correlation-attack estimator to attack this value.

As the data intruder could obtain estimates of \( \mu_Y \) and \( \rho_{YY^*}^2 \) from the noise-multiplied data, the intruder could easily check whether Condition 1 is satisfied for a target \( y_i \). If it is satisfied, then it does not matter much which estimator is to be used to attack this particular \( y_i \).

The ratio of \( MSE(Y^*_i | Y_i = y_i) \) over \( MSE(\tilde{Y}_i | Y_i = y_i) \) cannot be obtained by the data intruder as doing so requires the knowledge of \( y_i \). However, given \((\mu_Y, \sigma_Y, \rho_{YY^*})\) which could all be estimated by the data intruder, the ratio of \( MSE(Y^*_i | Y_i = y_i) \) over \( MSE(\tilde{Y}_i | Y_i = y_i) \) is a function of \( y_i \). By treating \( y_i \) as a variable, the ratio of the MSEs could be known for each possible value of \( y_i \). Taking the previous example where the original variable \( Y \) follows \( U(100,600) \) for instance, the ratio of \( MSE(Y^*_i | Y_i = y_i) \) over \( MSE(\tilde{Y}_i | Y_i = y_i) \) is above 5 if \( y_i \) is between 280 and 500, meaning that the correlation-attack estimator
should be far more accurate than the unbiased estimator according to the magnitudes of the MSEs. If the data intruder believes that \( y_i \) is within (280, 500), then logically speaking he would use the correlation-attack estimate to attack the unobservable \( y_i \).

Alternatively, if noise density \( f_C \) is public and a data intruder has no prior knowledge about the location of the unobservable \( y_i \), one possible choice is that the data intruder may attempt to compute the expected value of \( y_i \) by using \( E(Y_i|Y_i^* = y_i^*) \). Assuming \( Y \) is positive, the conditional expectation takes the following form:

\[
E(Y_i|Y_i^* = y_i^*) = \int_0^\infty y_i f_{Y_i|Y_i^*}(y_i)dy_i = \int_0^\infty f_C(y_i^*/y) f_Y(y)dy
\]

In the above expression, the density function \( f_Y \) is unknown. To estimate \( f_Y \), the data intruder might either come up with a prior based on past experience, or the data intruder might attempt to estimate \( f_Y \) using the sample-moment-based density reconstruction algorithm as mentioned in Section 2.1.2. The data intruder might then substitute \( f_Y(y) \) by its estimate to estimate \( E(Y_i|Y_i^* = y_i^*) \). As a result, the data intruder could have a rough idea about the expected value of \( y_i \) based on \( y_i^* \), and make a decision about which estimator to use accordingly.

**Discussion 2:** From the data provider’s perspective, the above result means that the correlation-attack estimator \( \tilde{Y}_i \) could yield a higher disclosure risk than the unbiased estimator \( Y_i^* \) for some \( y_i \) and vice versa. To protect all observations, the safest way is to make sure that for each original value \( y_i \), the disclosure risks from both estimators are simultaneously below an acceptable level. In that way, all original observations could be protected regardless of which estimator a data intruder uses.

In the next section, we propose a disclosure risk measure which could be used by data providers to measure disclosure risks against these two estimators simultaneously. The proposed disclosure risk measure could help data providers with noise generating variable selection during the data masking stage.

### 3.3 Disclosure risk and data utility loss measures

For noise multiplication masking method, the noise generating variable \( C \) plays the role of balancing data utility loss and disclosure risk, or “tuning mechanism” in Klein et al. (2014). That is, for a set of original data, the data provider needs to decide on an appropriate noise generating variable which achieves the required utility-risk tradeoff. In this section we propose a value disclosure risk measure to be used by data providers for noise generating variable selection in practice. We also introduce the definition of overall data utility loss we use in this chapter.
CHAPTER 3. CORRELATION-ATTACKS ON CONTINUOUS MICRODATA

We first note that, Lin and Wise (2012) showed that the value disclosure risk of $y_i$ being disclosed by $Y_i^*$ is given by

$$R_{LW}(Y_i^*, \delta | Y_i = y_i) = P(|C - 1| < \delta).$$

We propose to use the probabilistic disclosure risk measure because it is bounded between 0 to 1 and could easily be modified to measure disclosure risks from other estimators. For instance, Klein et al. (2014) proposed a similar probabilistic disclosure risk measure by replacing $Y_i^*$ by a linear predictor based on an attacking regression model. Following this idea, we propose a disclosure risk measure $R_{\rho}(\tilde{Y}_i, \delta | y)$ to evaluate the disclosure risk of $y_i$ being disclosed by the correlation-attack estimator $\tilde{Y}_i$. That is:

$$R_{\rho}(\tilde{Y}_i, \delta | y) = P\left(\left|\frac{\tilde{Y}_i - y_i}{y_i}\right| < \delta\right).$$

Hereafter, we use $R_{LW}(y_i, \delta)$ to denote $R_{LW}(Y_i^*, \delta | Y_i = y_i)$. We use $R_{\rho}(y_i, \delta)$ to denote $R_{\rho}(\tilde{Y}_i, \delta | y)$. As the exact form of $\tilde{Y}_i$ is not straigt-forward to show, $R_{\rho}(y_i, \delta)$ cannot be computed exactly. For large-sized sample, we propose two approaches for a data provider to estimate $R_{\rho}(y_i, \delta)$ in practice.

**Approach 1:** The data provider may just assume that

$$\tilde{Y}_i \approx (1 - \rho_{YY^*}^2)\mu_Y + \rho_{YY^*}^2 Y_i^*$$

as in Section 3.2. Then, we have

$$R_{\rho}(y_i, \delta) \approx P\left(\frac{(-\delta + 1)y_i - (1 - \rho_{YY^*}^2)\mu_Y}{\rho_{YY^*}^2 y_i} < C < \frac{(\delta + 1)y_i - (1 - \rho_{YY^*}^2)\mu_Y}{\rho_{YY^*}^2 y_i}\right).$$

In this expression, the parameters $\mu_Y$ and $\rho_{YY^*}$ are not known to the data provider but they could be estimated using sample estimates obtained from the original data. That is, $\mu_Y$ could be estimated by the sample mean of the original data, and $\rho_{YY^*}$ could be estimated by plugging in sample estimates of $\mu_Y$ and $\sigma_Y^2$ calculated from the original data into Equation (3.1). The data provider could then use the approximated disclosure risk measure to estimate $R_{\rho}(y_i, \delta)$.

**Approach 2:** The data provider could use Monte-Carlo simulations to approximate $R_{\rho}(y_i, \delta)$. That is, the data provider could firstly produce $N$ samples of noise-multiplied data using the noise candidate $C$, and then perform the correlation-attack by assuming the role of a data intruder. The correlation-attack is applied on each sample of the noise-multiplied data following the steps described in Section 3.1. To estimate $R_{\rho}(y_i, \delta)$, suppose among the $N$ samples, in $q$ of them $y_i$ is disclosed by the corresponding
correlation-attack estimate. Then $R_\rho(y_i, \delta)$ is estimated to be $q/N$.

Hereafter, we use $R_{cor}(y_i, \delta)$ to denote an approximate of $R_\rho(y_i, \delta)$ regardless of which approach the data provider uses to estimate $R_\rho(y_i, \delta)$. From Section 3.2, we have shown that some original observations are more vulnerable to the correlation-attack estimator while the others are more vulnerable to the unbiased estimator. We note that even though Result 1 provides a guidance about which estimator is more effective for predicting each $y_i$, it is calculated based on MSEs. Under the probabilistic disclosure risk measure we proposed, our simulation results with different sets of synthetic data show that an estimator with a lower MSE does not necessarily result in a larger disclosure risk. Therefore, to protect $y_i$, the safest way is to make sure that the disclosure risks from both estimators are simultaneously below an acceptable level. In that way, we say that $y_i$ is protected against the two estimators. We propose the following disclosure risk measure to evaluate the disclosure risk against the two estimators:

$$R(y_i, \delta) = \max\{R_{LW}(y_i, \delta), R_{cor}(y_i, \delta)\}$$ (3.5)

A sufficiently low $R(y_i, \delta)$ value means that $y_i$ is protected against both the correlation-attack estimator and the unbiased estimator. For a set of original data $y$ and a noise candidate $C$, the data provider collects a set of disclosure risks $\{R(y_i, \delta)\}_{i=1}^n$. The data provider could use the set of disclosure risks as a reference for noise generating variable selection. In some cases, it might be sufficient to say that a noise candidate offers an acceptable level of protection to the original data if the average value $\{R(y_i, \delta)\}_{i=1}^n$ is below a threshold value. In some other cases where all observations are highly sensitive, the data provider may require $\max(\{R(y_i, \delta)\}_{i=1}^n)$ to be sufficiently low in order for a noise candidate to be considered. The data provider might need to determine a criteria according to the nature of the original data. We will provide an example in Simulation 2.

We note that the disclosure risk measure is designed for the data provider to understand the disclosure risk of using a noise candidate $C$ to mask the original data. For the data intruder, he/she may rely on Discussion 1 to determine which attacking estimator to use, but the intruder cannot use the disclosure risk measure to evaluate the probability of successfully disclosing $y_i$ using either $Y_i^*$ or $\tilde{Y}_i$. This is because the disclosure risk measure requires the knowledge of the original data. The knowledge is not possessed by the data intruder. We also note that a low $R(y_i, \delta)$ value guarantees that $y_i$ is protected against both the unbiased estimator $Y_i^*$ and the correlation-attack estimator $\tilde{Y}_i$. However, we may not say that $y_i$ has a low value disclosure risk because of other possible attacking strategies. When $y_i$ is subject to other value disclosure risks, the data provider might need to consider using $R(y_i, \delta)$ together with other disclosure risk measures to jointly determine a suitable noise generating variable. Because the unbiased estimator and the correlation-
attack estimator are two basic attacking estimators, we feel that a noise candidate should at least guarantee that the noise-multiplied data is protected against these two attacking estimators first in order for the noise candidate to be considered further for masking a set of original data.

In the remainder of this section, we introduce the overall data utility loss measure we use in this chapter. We note that, when noise-multiplied data $y^*$ and noise variance $\sigma_C^2$ are released to the public, data users might only be able to estimate the first two moments and variance estimates of the population from $y^*$ using estimators developed in Nayak et al. (2011). A data user might be able to estimate other population parameters through other means, but we do not pursue the discussion. In this chapter, we assume a simple case that the overall utility loss measure is computed according to the utility losses of $E(Y)$ and $E(Y^2)$ only. Similar ways for measuring overall utility loss have also been considered in the literature (Kim and Winkler 1995, 2003; Brand 2002). For instance, Kim and Winkler (1995) only considered utility losses for population mean and variance as indications of overall utility loss for noise-added microdata. Based on this assumption, we adopt the utility loss measure proposed in Duncan et al. (2001; 2004), which is a data user’s mean squared error in estimating a population parameter from perturbed data. From Section 2.1.1, $E(Y) = E(Y^*)$ and $E(Y^2) = \frac{E[(Y^*)^2]}{(\sigma^2+1)}$. Therefore, $E(Y)$ could be unbiasedly estimated by data users using $U_Y = \frac{\sum_{i=1}^{n} Y_i^*}{n}$, and $E(Y^2)$ could be unbiasedly estimated using $U_{Y^2} = \frac{\sum_{i=1}^{n} (Y_i^*)^2}{n(\sigma^2+1)}$. Based on Duncan et al (2001; 2004)’s utility loss measure, we have the following:

$$UL_1 = Var(U_Y|y) = \frac{\sigma_C^2 \sum_{i=1}^{n} y_i^2}{n^2}$$

and

$$UL_2 = Var(U_{Y^2}|y) = \frac{[E(C^4) - (\sigma_C^2 + 1)^2] \sum_{i=1}^{n} y_i^4}{n^2(\sigma_C^2 + 1)^2}$$

The expression of $UL_1$ means that, a set of noise candidates with equal variance will result in the same level of utility loss for $E(Y)$. As a result, in this chapter we use the following overall utility loss measure for noise generating variable selection: When selecting among a set of noise candidates, we let the variances of these noise candidates to be equal. Then, we say a noise candidate will result in the lowest level of overall utility loss if it has the lowest $UL_2$ value. We note that the overall utility loss measure we use in this chapter is simple and is for illustration purpose only. The primary purpose of this chapter is the introduction of the proposed disclosure risk measure in Equation (3.5). In practice a data provider may use its own overall utility loss measure in conjunction with the disclosure risk measure we proposed for noise generating variable selection.
3.4 Simulations

In this section we present two simulation studies. In the first simulation, we show that the correlation-attack strategy could be used to disclose perturbed values generated by Oganian-Karr (2011)’s noise multiplication masking method if parameter of noise $k$ is released. In the second simulation, we show that our proposed disclosure risk measure could help a data provider to select an appropriate noise generating variable during the data masking stage. We will use an R-U map (Duncan et al. 2001; 2004) to aid us with decision-making in the second simulation. We assume that the unbiased estimator and the correlation-attack estimator are the only sources of value disclosure risk. We also comment on the protection levels offered by a few noise candidates which guarantee that $R_{LW}$ is very small or is 0 for all original observations.

3.4.1 Simulation 1

In this section we present a study on Oganian and Karr (2011)’s multiplicative noise masking scheme. We would like to show two things: 1. Oganian and Karr’s multiplicative masking scheme is vulnerable to the correlation-attack if parameter of noise $k$ is released. 2. $R_\rho$ could help data providers with the decision on the parameter $k$ to be used during the masking process.

Oganian and Karr (2011) presented a masking scheme which preserves the mean and covariance of an original dataset. Specifically, denote the $n \times d$ dimensional original data as $x_o$. $x_o \sim X_o$. $E(X_o) = \mu_o$ and $Var(X_o) = \Sigma_o$. There is no restriction on the distribution of $X_o$.

The masking scheme works as following:

1. Data providers decide the noise variable $R \sim N(\mu_R, \Sigma_R)$ to be used, where

$$\Sigma_R(i, j) = log(1 + \frac{k \Sigma_o(i, j)}{\Sigma_o(i, j) + \mu_o(i)\mu_o(j)}), \quad i, j = 1, \cdots, d,$$  \hspace{1cm} (3.6)

$$\mu_R(i) = -\Sigma_R(i, i)/2, \quad i = 1, \cdots, d.$$  \hspace{1cm} (3.7)

where $k$ is a parameter set by data providers, which is usually a small number such as 0.15 in order to maintain data utility. The noise variable $R$ depends on the mean and covariance matrix of $X_o$. The value of $k$ need not to be released to data users.

2. The masked data $x_m$ are obtained through the following transformation:

$$x_m = \frac{(\sqrt{1+k} - 1)\mu_o + [x_o \circ exp(R)]}{\sqrt{1+k}}$$  \hspace{1cm} (3.8)
where “◦” denotes elementwise matrix multiplication (Schur or Hadamard product). Suppose \( x_m \sim X_m \), then \( E(X_m) = \mu_0 \) and \( \text{cov}(X_m) = \Sigma_o \).

Now suppose \( x_o \) is masked by Oganian and Karr’s masking scheme. After masking, the masked data \( x_m \) is released to data users. As data providers could decide whether the parameter \( k \) is to be released to data users as well, we consider the following three scenarios:

**Case 1**: The masked data \( x_m \) and parameter \( k \) are released to data users.

**Case 2**: The masked data \( x_m \) and the mean of noise variable \( E(R) \) are released to data users. The parameter \( k \) is not released.

**Case 3**: Only masked data \( x_m \) is released.

Oganian and Karr (2011) discussed whether the parameter \( k \) should be public. The authors showed that data users could only obtain biased third and fourth moments of an attribute by analysing the masked data. Releasing \( k \) could allow data users to adjust for these biases. However, the consequence of releasing this information on data disclosure was not studied. In the following we discuss the consequence on value disclosure risk.

Recall that the correlation-attack is for attacking noise-multiplied data. Therefore it cannot be applied to \( x_m \) directly. However, through transformation of (3.8), data intruders could obtain the noise-multiplied data from \( x_m \). Denote \( r \) as the noise matrix drawn from \( \mathcal{R} \), \( y^* = x_o \circ \exp(r) \).

By rearranging (3.8), we have

\[
y^*_j = (\sqrt{1 + k})x_m - (\sqrt{1 + k} - 1)u_o
\]

The dimension of \( y_* \) is \( n \times d \). For convenience, we denote the \( j \)th characteristics of \( x_o \) as \( x^j_o \), the \( j \)th characteristics of \( y_* \) as \( y^*_j \) and the \( j \)th characteristics of \( x_m \) as \( x^j_m \), \( j \)th column of noise matrix \( r \) as \( r^j \) with \( r^j \sim \mathcal{R}^j \). Then it can be seen that \( y^*_j \) can be regarded as the noise-multiplied version of \( x^j_o \), with \( \{y^*_j = x^j_o r^j_i\}_{i=1}^n \). However, the expression of \( y_* \) requires knowledge of \( k \). In **case 3** scenario, \( k \) is not released. Therefore releasing \( x_m \) under **Case 3** is safe from the correlation-attack. We discuss the feasibility of the correlation-attack under **Case 1** and **Case 2** respectively.

**Under Case 1**: Since \( x_m \) and \( k \) are released, the data intruder could obtain \( y_* \). Now suppose the data intruder targets on observations of \( j \)th characteristics of \( x_o \). Denote the original data as \( \{x^j_{oi}\}_{i=1}^n \), the noise multiplied data as \( \{y^j_{oi} = x^j_o r^j_i\}_{i=1}^n \), \( r^j_i \sim \mathcal{R}^j \), where \( E(R^j) = \Sigma_R(j,j)/2, \text{Var}(R^j) = \Sigma_R(j,j) \), the masked data which is passed to data users is \( \{x^j_{mi}\}_{i=1}^n \).

The correlation-attack needs three pieces of information: noise multiplied masked data
\{y_{s,i}^j\}_{i=1}^n$, the mean of the noise generating variable $E(R^j)$ and the variance of the noise generating variable $\text{Var}(R^j)$. Information of $E(R^j)$ and $\text{Var}(R^j)$ are not given, but these values can be directly estimated by substituting $k$ into (3.6) and (3.7), which yield $\hat{E}(R^j)$ and $\hat{\text{Var}}(R^j)$.

That is

$$\hat{\text{Var}}(R^j) = \log(1 + \frac{ks_i^2}{s_{x_i}^2 + \hat{\mu}_{x_i}^2}),$$

$$\hat{E}(R^j) = \frac{\hat{\text{Var}}(R^j)}{2},$$

where $\hat{\mu}_{x_i} = \frac{\sum_{i=1}^n x_{mi}^j}{n}, s_{x_i}^2 = \frac{1}{n-1} \sum_{i=1}^n (x_{mi}^j - \hat{\mu}_{x_i})^2$.

The noise generating variable used to generate noise terms is $\exp(R^j)$, where

$$E\{\exp(R^j)\} = \exp\{E(R^j) + \text{Var}(R^j)/2\} = 1$$

and

$$\text{Var}\{\exp(R^j)\} = \{\exp[\text{Var}(R^j)] - 1\} \exp[2E(R^j) + \text{Var}(R^j)].$$

(3.9)

As a result, the data intruder could carry out the correlation-attack by using the two pieces of information $\{y_{s,i}^j\}_{i=1}^n$ and $\text{Var}\{\exp(R^j)\}$, where $\text{Var}\{\exp(R^j)\}$ is obtained by substituting $\hat{\text{Var}}(R^j)$ and $\hat{E}(R^j)$ into Equation (3.9).

**Under Case 2:** Even though $k$ is not released, the data intruder has information of $E(R^j)$. From (3.7), if $E(R^j)$ is available, $k$ can be estimated through:

$$\hat{k} = \frac{\{\exp[2E(R^j)] - 1\} \{\text{Var}(X_{d_i}^j) + E(X_{d_i}^j)^2\}}{\text{Var}(X_{d_i}^j)}.$$

The value of $E(X_{d_i}^j)$ and $\text{Var}(X_{d_i}^j)$ can be estimated through $\hat{\mu}_{x_i} = \frac{\sum_{i=1}^n x_{mi}^j}{n}, s_{x_i}^2 = \frac{1}{n-1} \sum_{i=1}^n (x_{mi}^j - \hat{\mu}_{x_i})^2$, where $\hat{\mu}_{x_i}$ and $s_{x_i}^2$ are sample mean and sample variance of $x_{mi}^j$.

Using $\hat{k}$, the data intruder could estimate $\{y_{s,i}^j\}_{i=1}^n$ through

$$\tilde{y}_{s,i}^j = \left(\sqrt{1 + \hat{k}} x_{mi}^j - (\sqrt{1 + \hat{k}} - 1) \hat{\mu}_{x_i}, i = 1 \cdots n \right)$$

and the variance of $R_j$ can be easily estimated through

$$\hat{\text{Var}}(R_j) = \log\left(1 + \frac{\hat{k}s_{x_i}^2}{s_{x_i}^2 + \hat{\mu}_{x_i}^2}\right).$$

Then, the data intruder could follow the steps we described under Case 1 to conduct the correlation-attack.
We use the following simulation to show the relationship between value disclosure risk and the choice of \( k \): Suppose the two-dimensional original variable \( X_o = (X_1^o, X_2^o) \). Let \( X_o \sim N(\mu_o, \Sigma_o) \), \( \mu_o = (10, 20) \) and \( \Sigma_o = \begin{bmatrix} 5 & 4.5 \\ 4.5 & 7 \end{bmatrix} \).

We drew a sample of size 1000 from \( X_o \) as our original data which requires masking, and denote the sample as \( \{x_{oi}^1, x_{oi}^2\}_{i=1}^{1000} \).

Now suppose a data provider would like to mask original data \( \{x_{oi}^1, x_{oi}^2\}_{i=1}^{1000} \) using Oganian and Karr’s masking scheme to produce masked data \( \{x_{mi}^1, x_{mi}^2\}_{i=1}^{1000} \). Denote the masked variable as \( X_m = (X_1^m, X_2^m) \). The noise generating variable to be considered is \( R \sim N(\mu_R, \Sigma_R) \), where

\[
\mu_R = \{0.5\log(1+k/21), 0.5\log(1+7k/407)\}
\]

and

\[
\Sigma_R = \begin{bmatrix} \log(1+k/21) & \log(1+9k/409) \\ \log(1+9k/409) & \log(1+7k/407) \end{bmatrix}.
\]

Suppose the data provider releases masked data and the value of \( k \) to data users, but the data provider needs to decide which value of \( k \) to be used during the data masking process. As releasing \( k \) leads to the correlation-attack, the data provider needs to assess the disclosure risks of different \( k \) using the disclosure risk measure \( R_\rho \) for \( \{x_{oi}^1\}_{i=1}^{1000} \) and \( \{x_{oi}^2\}_{i=1}^{1000} \) respectively. Suppose the data provider uses \( R_\rho(\hat{\rho}_{oi}^1, 0.1|\{x_{oi}^1\}_{i=1}^{1000}) \) and \( R_\rho(\hat{\rho}_{oi}^2, 0.1|\{x_{oi}^2\}_{i=1}^{1000}) \) to measure disclosure risks against the correlation-attack.

We carried out a simulation with 5000 iterations, recording the summary statistics of \( R_\rho(\hat{\rho}_{oi}^1, 0.1|\{x_{oi}^1\}_{i=1}^{1000}) \) and \( R_\rho(\hat{\rho}_{oi}^2, 0.1|\{x_{oi}^2\}_{i=1}^{1000}) \) for each iteration, where \( \hat{\rho}_{oi}^1 \) is the estimated population correlation between the original variable \( X_1^o \) and the masked variable \( X_1^m \). We reported the average of \( \hat{\rho}_{oi}^1 \) as \( \hat{\rho}_{oi}^1 \). Similarly we recorded \( \hat{\rho}_{oi}^2 \) on each iteration and reported the average as \( \hat{\rho}_{oi}^2 \).

Tables 3.1 and 3.2 show the summary statistics of \( \{R_\rho(\hat{\rho}_{oi}^1, 0.1|\{x_{oi}^1\}_{i=1}^{1000}) \} \) and \( \{R_\rho(\hat{\rho}_{oi}^2, 0.1|\{x_{oi}^2\}_{i=1}^{1000}) \} \), respectively. Taking Table 3.1 with \( k = 0.15 \) for example, the mean of \( \{R_\rho(\hat{\rho}_{oi}^1, 0.1|\{x_{oi}^1\}_{i=1}^{1000}) \} \) is 0.772, which means roughly speaking, 77% observations could be disclosed by the correlation-attack if the acceptance rule \( \delta \) is 0.1. The value of \( \hat{\rho}_{oi}^1 \) when \( k = 0.15 \) is 0.933, which indicates a strong linear relationship between \( X_1^o \) and \( X_1^m \).

As the correlation could be inferred by the data intruder, such a high correlation may motivate the data intruder to use the correlation-attack to unveil the original observations. When \( k \) gets lower, the corresponding value disclosure risk against the correlation-attack gets lower as well. It can be shown that the value of \( \hat{\rho}_{oi}^1 \) and \( \hat{\rho}_{oi}^2 \) are the same for the
Table 3.1: Summary statistics of \( \{ R_\rho (\tilde{x}_{oi}^1, 0.1) \}_{i=1}^{1000} \) for various \( k \) values.

<table>
<thead>
<tr>
<th>( k )</th>
<th>( Q_1 )</th>
<th>( Q_2 )</th>
<th>mean</th>
<th>( \tilde{R}_{X_1, X_1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>0.958</td>
<td>0.962</td>
<td>0.954</td>
<td>0.965</td>
</tr>
<tr>
<td>0.10</td>
<td>0.873</td>
<td>0.852</td>
<td>0.881</td>
<td>0.910</td>
</tr>
<tr>
<td>0.15</td>
<td>0.777</td>
<td>0.803</td>
<td>0.772</td>
<td>0.815</td>
</tr>
<tr>
<td>0.20</td>
<td>0.713</td>
<td>0.749</td>
<td>0.714</td>
<td>0.768</td>
</tr>
<tr>
<td>0.25</td>
<td>0.664</td>
<td>0.709</td>
<td>0.668</td>
<td>0.734</td>
</tr>
<tr>
<td>0.30</td>
<td>0.624</td>
<td>0.680</td>
<td>0.635</td>
<td>0.709</td>
</tr>
<tr>
<td>0.35</td>
<td>0.589</td>
<td>0.653</td>
<td>0.606</td>
<td>0.689</td>
</tr>
<tr>
<td>0.40</td>
<td>0.563</td>
<td>0.633</td>
<td>0.583</td>
<td>0.674</td>
</tr>
</tbody>
</table>

Table 3.2: Summary statistics of \( \{ R_\rho (\tilde{x}_{oi}^2, 0.1) \}_{i=1}^{1000} \) for various \( k \) values.

<table>
<thead>
<tr>
<th>( k )</th>
<th>( Q_1 )</th>
<th>( Q_2 )</th>
<th>mean</th>
<th>( \tilde{R}_{X_2, X_2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
</tr>
<tr>
<td>0.10</td>
<td>0.986</td>
<td>0.989</td>
<td>0.985</td>
<td>0.990</td>
</tr>
<tr>
<td>0.15</td>
<td>0.958</td>
<td>0.968</td>
<td>0.959</td>
<td>0.972</td>
</tr>
<tr>
<td>0.20</td>
<td>0.927</td>
<td>0.945</td>
<td>0.929</td>
<td>0.953</td>
</tr>
<tr>
<td>0.25</td>
<td>0.896</td>
<td>0.924</td>
<td>0.900</td>
<td>0.935</td>
</tr>
<tr>
<td>0.30</td>
<td>0.866</td>
<td>0.905</td>
<td>0.874</td>
<td>0.920</td>
</tr>
<tr>
<td>0.35</td>
<td>0.840</td>
<td>0.887</td>
<td>0.851</td>
<td>0.907</td>
</tr>
<tr>
<td>0.40</td>
<td>0.814</td>
<td>0.871</td>
<td>0.831</td>
<td>0.896</td>
</tr>
</tbody>
</table>

same \( k \) value and are equal to \( 1/\sqrt{(1+k)} \). Therefore we see that the last columns of both tables are the same.

If we ignore the impact of \( k \) on data utility, then the two tables suggest that the data provider should use a very large \( k \) during the data masking process so that the level of value disclosure risk against the correlation-attack is acceptable. This simulation showed that \( R_\rho \) could be used by the Oganian and Karr’s masking scheme to decide the value of \( k \) to be used during the masking process.

### 3.4.2 Simulation 2

In this section we show how our proposed disclosure risk measure in (3.5) could help data providers with the process of noise generating variable selection. Following Klein et al. (2014), we use the public use data from the 2000 Current Population Survey (CPS) March supplement (available from http://www.census.gov/cps/). The entire data set contains household, family, and individual records. In this section, we consider all positive household income values under household income attribute as the original data. The original data contains 50661 positive observations ranging from 1 to 768742, with mean 53007 and variance 2411407246. The data is skewed to the right. In the following we denote the original data as \( \{ y_i \}_{i=1}^{50661} \).
We consider five noise candidates \( \{C_i\}_{i=1}^5 \) with equal variance. We aim to use the proposed disclosure risk and overall utility loss measure to determine an appropriate noise generating variable for masking \( \{y_i\}_{i=1}^{50661} \). We set the acceptance rule \( \delta = 0.1 \) throughout this section. The distributions of the noises are:

\[
C_1 \sim 0.5U(0.5, 0.9) + 0.5U(1.1, 1.5), \quad U(a, b) \text{ is a uniform random variable with two parameters } a \text{ and } b; \\
C_2 \sim U(1 - 0.5\sqrt{93/75}, 1 + 0.5\sqrt{93/75}); \\
C_3 \text{ follows a truncated normal distribution. That is, it follows } N(1, 0.146135) \text{ which is truncated at } 0.3 \text{ and } 1.7, \text{ meaning that } C_3 \text{ has support } (0.3, 1.7); \\
C_4 \sim 0.5N(0.7, 4/300) + 0.5N(1.3, 4/300); \\
C_5 \sim 0.5T(1.1 - \sqrt{0.56}/4, 0.9, 0.9) + 0.5T(1.1, 0.9 + \sqrt{0.56}/4, 1.1), \text{ where } T(a, b, c) \text{ is a triangular random variable, } a \text{ is the lower limit of the distribution, } b \text{ is the upper limit of the distribution, } c \text{ is the mode of the distribution.}
\]

The distributions of these noise candidates have been proposed or used in the literature for producing noise-multiplied data. Specifically, \( C_1 \) follows a mixture of uniforms distribution and was considered in the simulation study in Klein et al. (2014); \( C_2 \) follows a uniform distribution, which was proposed and discussed in Sinha et al. (2011); \( C_3 \) follows a truncated normal distribution, which was considered in Kim and Winkler (2003); \( C_4 \) follows a mixture of normals distribution, which was proposed in Lin and Wise (2012); \( C_5 \) follows a truncated triangular distribution, which was proposed in Kim (2007) and Kim and Jeong (2008). Note that \( C_1 \) and \( C_5 \) lead to \( R_{LW} = 0 \) and \( C_4 \) leads to a small \( R_{LW} \) (0.0414) for all original observations. Therefore they seem to be good noise candidates if the unbiased estimator is the only source of value disclosure risk for \( y_i \). For \( C_4 \), there is a very small probability that it produces a negative noise. We note that negative noise terms were generated in this simulation. However, it has very little impact on our simulation result. We also tried to only use positive noise terms for this simulation. For both cases, the results were very similar.

Now we assume the role of the data provider and we aim to find an appropriate noise generating variable to mask \( \{y_i\}_{i=1}^{50661} \). We assume the following criteria for disclosure risk control: a noise candidate needs to guarantee that \( \max(R(y_i, 0.1))_{i=1}^{50661} < 0.3 \), i.e. the disclosure risk of any original value is less than 0.3. When multiple noise candidates satisfy this criteria, we say a noise candidate is better than the others in terms of disclosure risk control if it offers the lowest average disclosure risk \( \{R(y_i, 0.1)\}_{i=1}^{50661} \). We use Approach 1 described in Section 3.3 to estimate \( R_p \) using \( R_{cor} \). We also use the overall utility loss measure introduced in Section 3.3 to compare the overall utility loss levels of the noise candidates. That is, since all the noise candidates have the same variance, a noise candidate with a lower \( UL_2 \) has a lower level of overall utility loss under our overall
utility loss measure.

The disclosure risk against income value plots for all five noise candidates are given in Figure 3.1, Figure 3.2 and Figure 3.3. To comment on the plots, we see that different noise candidates protect the original values differently. For instance, we see that $C_2$ (Figure 3.1(b)) offers uniform protections to most observations while the others do not. The noise candidates $C_1, C_4$ and $C_5$, which were previously thought to be good noise candidates because they guarantee that $R_{LW} = 0$, are actually not that ideal for this data set when our proposed disclosure risk measure $R(y_i, \delta)$ is used. For these noise candidates, $R(y_i, \delta) = R_{cor}(y_i, \delta)$, i.e. the value disclosure risk comes entirely from the correlation-attack. We see that in this simulation, observations around the sample mean or that are extremely large are not protected well by these noise candidates. It may suggest that the merit of those noise candidates with $R_{LW} = 0$ may need to be reconsidered due to the correlation-attack. We note that in practice, statistical agencies are more concerned about extreme values in a set of data compared with values around the sample mean. Values around the sample mean might be difficult to re-identify because there are many such observations.

We note that, for this simulation, noise candidates $C_1, C_4$ and $C_5$ do not provide a good level of protection to observations around the sample mean or that are extremely large. It might not be true if the original data follows a different distribution. For instance, if the original data follows $U(10,2000)$, then observations around the population mean are well protected by these noise candidates while observations around 600 are not. As we see in Equation 3.4, the level of protection a noise candidate $C$ offers to an observation $y_i$ depends on $\delta, \sigma_Y, \mu_Y$, the distribution of $C$ and the value of $y_i$ itself.

For each noise candidate, we also considered the average disclosure risk $\{R(y_i, \delta)\}_{i=1}^{50661}$ and the overall level of data utility loss it produces. We use an R-U map to visualize the utility-risk tradeoffs, which is presented in Figure 3.4. We see that, $C_3$ results in the lowest overall utility loss but the highest average disclosure risk. $C_2$ and $C_5$ result in lower average disclosure risks but higher overall utility losses. $C_1$ and $C_4$ yield very similar and the lowest average disclosure risks but $C_1$ yields a lower overall utility loss than $C_4$. However, the overall utility losses produced by these noise candidates are very similar as the highest utility loss (produced by $C_2$) is only 1.2% larger than the lowest utility loss (produced by $C_3$).

Based on all these results, we could make a decision about which noise candidate is appropriate to use in this context. Based on Figure 3.1, Figure 3.2 and Figure 3.3, we see that $C_1, C_4$ and $C_5$ do not satisfy our criteria for disclosure risk control because they cannot guarantee that every original observation has a disclosure risk less than 0.3. Therefore, we could only choose between $C_2$ and $C_3$. From Figure 3.4, we see that $C_2$ results in a much lower average disclosure risk level than $C_3$ at the expense of only a slightly higher level of overall data utility loss. As a result, $C_2$ seems to be the best choice in this context.
(a) disclosure risk plot for $C_1$.

(b) disclosure risk plot for $C_2$.

Figure 3.1: Disclosure risks $\{R(y_i, 0.1)\}_{i=1}^{50661}$ against income values $\{y_i\}_{i=1}^n$ plots for noise candidates $C_1$ and $C_2$. 
Figure 3.2: Disclosure risks \( \{R(y_i,0.1)\}_{i=1}^{50661} \) against income values \( \{y_i\}_{i=1}^n \) plots for noise candidates \( C_3 \) and \( C_4 \).
CHAPTER 3. CORRELATION-ATTACKS ON CONTINUOUS MICRODATA

40

(a) disclosure risk plot for \( C_5 \).

Figure 3.3: Disclosure risks \( \{ R(y_i, 0.1) \}_{i=1}^{50661} \) against income values \( \{ y_i \}_{i=1}^{n} \) plots for each noise candidate \( C_5 \).

3.5 Conclusion and future work

In this chapter we introduced the correlation-attack which could be used by data intruders to attack noise-multiplied data. The correlation-attack only uses information contained in the noise-multiplied data and the variance of noise terms, hence it could be applied to attack noise-multiplied data in most situations. The correlation-attack estimator is discussed along with the unbiased estimator and we propose that disclosure risks from both estimators need to be controlled. Correspondingly, we proposed a disclosure risk measure for data providers to evaluate the disclosure risks against both attacking estimators simultaneously for each noise candidate. The proposed disclosure risk measure could help data providers with decision-making on noise generating variable selection.

A noise candidate which produces a low level of average value disclosure risk against the unbiased estimator may produce a high level of average value disclosure risk against the correlation-attack estimator and vice versa. For instance, if the original data follows \( LN(5, 0.12^2) \), we can show that \( C_4 \) results in a low level of average disclosure risk against the unbiased estimator (0.0414) but a high level of average disclosure risk against the correlation-attack estimator (roughly 0.619). Similarly, we can show that if the original data follows \( LN(3, 2^2) \), the noise candidate \( C_3 \) will result in a low level of average disclosure risk against the correlation-attack estimator (0.112) but a high level of average disclosure risk against the unbiased estimator (0.221). An ideal noise candidate should
Figure 3.4: Utility-Risk tradeoffs of the noise candidates.
protect against both estimators effectively. It might be interesting to find out these ideal noise candidates given different distributions of original data in the future.

Identification of other attacking strategies are necessary for disclosure risk control of noise-multiplied microdata. For instance, it may be the case that in microdata, multiple attributes are protected by multiplicative noises. In that case, the idea of the regression-attack proposed in Li and Sarkar (2011) might be used to attack a noise-multiplied value. That is, to attack $y_{ij}$, which is the original value of the $i$-th record on the $j$-th attribute, a data intruder may regress $y_{ij}$ on the noise-multiplied values of other attributes of the record, i.e. regress $y_{ij}$ on $(y^*_{i1}, y^*_{i2}, \ldots, y^*_{ip})$, where $p$ is the number of regressors used to attack $y_{ij}$. This idea could be explored in the future.
Chapter 4

Simulations studies on noise candidates selection

When using multiplicative noise to protect sensitive original observations, the data provider faces the problem of choosing an appropriate noise generating variable from a set of noise candidates. As a top priority, the noise generating variable should produce a low disclosure risk to the original data. In the literature, the effect of different noise distributions on the disclosure risk against the unbiased estimator has been studied. As a result, several noise distributions have been proposed such that the disclosure risk against the unbiased estimator is low. Those distributions include mixtures of normal distributions (Lin and Wise 2012), truncated triangular distributions (Kim 2007), and mixtures of uniform distributions (Klein et al. 2014). In this chapter, we investigate the effect of noise distributions on the disclosure risk against the correlation-attack estimator on log-normal data through simulations. We note that log-normal distribution can be used for modelling many real-life data, such as income data (Klein et al. 2014) and business data in an organisation (Muralidhar et al. 1995) and references therein. Log-normal distribution has two parameters $\mu$ and $\sigma$. It is non-negative and its shape could go from roughly normal to heavily skewed. We aim to propose a guide about the noise distributions to use for protecting log-normal data against the correlation-attack estimator. Unlike the protection against the unbiased estimator where a noise variable offers uniform protection to all original observations, not all original observations receive the same magnitude of protection. Because in practice, some records are more sensitive than others, sometimes the data provider needs to choose a noise variable such that highly sensitive observations are well protected. For business data or personal income data, extremely large observations are more vulnerable to identity disclosure risk. Therefore, those observations require better protection against value disclosure risk. We use simulation to investigate this issue. We apply our findings using the same set of original data as we used in Section 3.4.2. We note that the distribution of the original data is approximately log-normal. Therefore our findings in this chapter should be applicable in this case.
To start with, for a noise candidate \( C \) and an acceptance rule \( \delta \), the disclosure risk of an original observation \( y_i \) against the unbiased estimator is given as:

\[
R_{ LW}(Y^*_i, \delta | Y_i = y_i) = P\left(\left| Y^*_i - Y_i \right| < \delta | Y_i = y_i \right) = P\left(|C - 1| < \delta\right).
\]

The disclosure risk of an original observation \( y_i \) against the correlation-attack estimator is:

\[
R_{ \rho}(\tilde{Y}_i, \delta | \{y_j\}_{j=1}^n) = P\left(\left| \tilde{Y}_i - y_i \right| < \delta \right).
\]

We use \( R_{ LW}(y_i, \delta) \) to denote \( R_{ LW}(Y^*_i, \delta | Y_i = y_i) \). We use \( R_{ \rho}(y_i, \delta) \) to denote \( R_{ \rho}(\tilde{Y}_i, \delta | \{y_j\}_{j=1}^n) \). We use Approach 2 mentioned in Section 3.3 to estimate \( R_{ \rho}(y_i, \delta) \).

That is, for a set of original data, we firstly produce \( N \) multiple samples of noise-multiplied data using the noise candidate \( C \). Then we perform the correlation-attack by assuming the role of the data intruder. The correlation-attack is applied on each sample of the noise-multiplied data. To estimate \( R_{ \rho}(y_i, \delta) \), suppose among the \( N \) samples, in \( q \) of them \( y_i \) is disclosed by the corresponding correlation-attack estimate. Then \( R_{ \rho}(y_i, \delta) \) is estimated to be \( q/N \).

### 4.1 Simulation 1

In this simulation, we aim to show the following: for a noise candidate, even though it produces a low \( R_{ LW}(y_i, \delta) \), the value of \( R_{ \rho}(y_i, \delta) \) could still be very high, and vice versa. This observation is important because it means that the data provider needs to consider both attacking estimators and their disclosure risks when selecting noise generating variables. We also computed \( R_{ \rho}(y_i, \delta) \) using the Approach 1 we discussed in Section 3.3. Theoretically speaking, for large-sized data, both approaches lead to similar results. We aim to show this point in this simulation.

Denote the original data as \( y = (y_1, y_2, \ldots, y_{1000}) \). The observations are independently drawn from \( Y \sim U(100, 200) \). We consider the following four mixtures of uniforms noise generating variables to mask the data:

- \( C_{1,1} \): \( 0.5U(0.8, 0.9) + 0.5U(1.1, 1.2) \),
- \( C_{1,2} \): \( 0.5U(0.7, 0.9) + 0.5U(1.1, 1.3) \),
- \( C_{1,3} \): \( 0.5U(0.6, 0.9) + 0.5U(1.1, 1.4) \),
- \( C_{1,4} \): \( 0.5U(0.5, 0.9) + 0.5U(1.1, 1.5) \).

The noise candidates \( C_{1,1} \) and \( C_{1,4} \) were used in Klein et al. (2014) for generating noise-multiplied data.

We assume the acceptance rule \( \delta = 0.1 \). It can be easily shown that, all four noise gener-
Table 4.1: Summary statistics of \( \{R_\rho(y_i, 0.1)\}_{i=1}^{1000} \) if \( y \) is masked by \( C_{1,1}, C_{1,2}, C_{1,3} \) and \( C_{1,4} \), respectively.

<table>
<thead>
<tr>
<th>noise</th>
<th>( \tilde{r}_{yy^*} )</th>
<th>Min</th>
<th>( Q_1 )</th>
<th>( Q_2 )</th>
<th>mean</th>
<th>( Q_3 )</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{1,1} )</td>
<td>0.768</td>
<td>0.0864</td>
<td>0.496</td>
<td>0.504</td>
<td>0.531</td>
<td>0.583</td>
<td>0.709</td>
</tr>
<tr>
<td>( C_{1,2} )</td>
<td>0.661</td>
<td>0</td>
<td>0.492</td>
<td>0.503</td>
<td>0.477</td>
<td>0.574</td>
<td>0.663</td>
</tr>
<tr>
<td>( C_{1,3} )</td>
<td>0.570</td>
<td>0</td>
<td>0.383</td>
<td>0.498</td>
<td>0.447</td>
<td>0.584</td>
<td>0.713</td>
</tr>
<tr>
<td>( C_{1,4} )</td>
<td>0.496</td>
<td>0</td>
<td>0.252</td>
<td>0.494</td>
<td>0.422</td>
<td>0.591</td>
<td>0.785</td>
</tr>
</tbody>
</table>

...ing variables yield \( R_{LW}(y_i, \delta) = 0 \) for all \( \{y_i\}_{i=1}^{1000} \). To estimate \( R_\rho(y_i, \delta) \), we simulated the correlation-attack process 5000 times for each noise candidate. On each iteration, we recorded \( \tilde{r}_{yy^*} \), the estimated population correlation coefficient between \( Y \) and \( Y^* \). We recorded the correlation-attack estimates of the original values, which are used to estimate \( R_\rho \) for each \( y_i \). We computed the average of the 5000 samples of \( \tilde{r}_{yy^*} \), which is denoted as \( \bar{\tilde{r}}_{yy^*} \). We report the summary statistics of \( \{R_\rho(y_i, 0.1)\}_{i=1}^{1000} \) for each noise candidate as well as \( \bar{\tilde{r}}_{yy^*} \) in Table 4.1.

Table 4.1 gives the summary statistics of \( \{R_\rho(y_i, 0.1)\}_{i=1}^{1000} \). To interpret the table, for instance, the first quartile \( Q_1 \) of \( \{R_\rho(y_i, 0.1)\}_{i=1}^{1000} \) for \( C_{1,1} \) is 0.496, meaning that 75% of the original values have probabilities of more than 49.6% of being disclosed by the corresponding correlation-attack estimates. Similarly, \( Q_2 \) of \( \{R_\rho(y_i, 0.1)\}_{i=1}^{1000} \) is 0.503 for \( C_{1,2} \), indicating that 50% observations have more than 50.3% probabilities of being disclosed.

We also computed \( R_\rho(y_i, \delta) \) using the **Approach 1** mentioned in Section 3.3. That is, computed \( R_\rho(y_i, \delta) \) using

\[
R_\rho(y_i, \delta) \approx P\left( \frac{(-\delta + 1)y_i - (1 - \rho_{YY^*}^2)\mu_Y}{\rho_{YY^*}^2 y_i} < C < \frac{(\delta + 1)y_i - (1 - \rho_{YY^*}^2)\mu_Y}{\rho_{YY^*}^2 y_i} \right).
\]

The summary statistics of the disclosure risks are given in Table 4.2. We see that the results in Table 4.1 and Table 4.2 are very close. It is expected because in this simulation, the sample size is quite large. The result might suggest that data providers could use either **Approach 1** or **Approach 2** to calculate the disclosure risk.

To sum up, Table 4.1 shows that, a smaller value of \( \tilde{r}_{yy^*} \) leads to a lower average disclosure risk from the correlation-attack estimator, as given by the mean of \( \{R_\rho(y_i, 0.1)\}_{i=1}^{1000} \). This observation agrees with our discussion in Section 3.2 regarding the accuracy of the correlation-attack estimator. It might suggest that, the data provider could use a noise generating variable with a large variance to reduce the risk from the correlation-attack estimator. The disclosure risks from the correlation-attack estimator are quite large for all noise candidates, even though \( R_{LW}(y_i, 0.1) \) are uniformly 0 for all \( \{y_i\}_{i=1}^{1000} \).

Now we consider original data sets which follow log-normal distributions. Suppose...
CHAPTER 4. SIMULATIONS STUDIES ON NOISE CANDIDATES SELECTION

Table 4.2: Summary statistics of \( \{ R_{cor}(y_i, 0.1) \}_{i=1}^{1000} \) if \( y \) is masked by \( C_{1,1}, C_{1,2}, C_{1,3} \) and \( C_{1,4} \), respectively.

<table>
<thead>
<tr>
<th>noise</th>
<th>( \rho_{YY^*} )</th>
<th>Min</th>
<th>( Q_1 )</th>
<th>( Q_2 )</th>
<th>mean</th>
<th>( Q_3 )</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{1,1} )</td>
<td>0.778</td>
<td>0.241</td>
<td>0.500</td>
<td>0.500</td>
<td>0.529</td>
<td>0.571</td>
<td>0.652</td>
</tr>
<tr>
<td>( C_{1,2} )</td>
<td>0.672</td>
<td>0</td>
<td>0.500</td>
<td>0.500</td>
<td>0.476</td>
<td>0.567</td>
<td>0.608</td>
</tr>
<tr>
<td>( C_{1,3} )</td>
<td>0.581</td>
<td>0</td>
<td>0.381</td>
<td>0.500</td>
<td>0.443</td>
<td>0.572</td>
<td>0.655</td>
</tr>
<tr>
<td>( C_{1,4} )</td>
<td>0.507</td>
<td>0</td>
<td>0.250</td>
<td>0.500</td>
<td>0.415</td>
<td>0.579</td>
<td>0.723</td>
</tr>
</tbody>
</table>

Table 4.3: Summary statistics of \( \{ R_{cor}(y_i, 0.1) \}_{i=1}^{1000} \) if \( y \) is masked by \( C_{1,5}, C_{1,6}, C_{1,3} \), respectively.

<table>
<thead>
<tr>
<th>noise</th>
<th>( \rho_{YY^*} )</th>
<th>Min</th>
<th>( Q_1 )</th>
<th>( Q_2 )</th>
<th>mean</th>
<th>( Q_3 )</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{1,5} )</td>
<td>0.979</td>
<td>0.096</td>
<td>0.109</td>
<td>0.112</td>
<td>0.112</td>
<td>0.115</td>
<td>0.125</td>
</tr>
<tr>
<td>( C_{1,6} )</td>
<td>0.865</td>
<td>0.080</td>
<td>0.089</td>
<td>0.091</td>
<td>0.091</td>
<td>0.094</td>
<td>0.108</td>
</tr>
</tbody>
</table>

we have a set of original data \( \{ y_i \}_{i=1}^{1000} \) independently drawn from \( LN(3, 2) \). We consider the following noise candidates:

\( C_{1,5} \): \( N(1, 0.146) \) truncated at 0.3 and 1.7.

\( C_{1,6} \): \( N(1, 0.2) \) truncated at 0.1 and 1.9.

Assuming \( \delta = 0.1 \). We can show that, for \( C_{1,5}, R_{LW} = 0.221 \). For \( C_{1,6}, R_{LW} = 0.185 \). However, the corresponding disclosure risks against the correlation-attack are lower. The corresponding disclosure risks against the correlation-attack are given in Table 4.3.

Table 4.3 shows that, the disclosure risks against the correlation-attack are very low if \( C_{1,5} \) and \( C_{1,6} \) are used to mask the original data. The average disclosure risks against the correlation-attack are 0.112 and 0.091, respectively. However, the disclosure risks against the unbiased estimator are higher (0.221 and 0.185, respectively).

Overall, Simulation 1 shows that, the disclosure risk from the correlation-attack and the unbiased estimator need to be examined at the same time. A noise candidate which yields a low disclosure risk against the unbiased estimator might yield a high disclosure risk against the correlation-attack estimator, and vice versa. Therefore, the data provider will generally need to evaluate the disclosure risk against the correlation-attack. Because of this, the data provider needs guidance about the noise candidates to be used for different distributions of the original data. In the next two sections, we present a simulation study for original data sets which follow log-normal distributions. We aim to provide a rough guide about the types of noise candidates to be used for log-normally distributed original
CHAPTER 4. SIMULATIONS STUDIES ON NOISE CANDIDATES SELECTION

4.1.1 Simulation 2

Recall that

\[ R_{\rho} (y_i, \delta) \approx P \left( \frac{(-\delta + 1)y_i - (1 - \rho_{YY}^2)\mu_Y}{\rho_{YY}^2 y_i} < C < \frac{(\delta + 1)y_i - (1 - \rho_{YY}^2)\mu_Y}{\rho_{YY}^2 y_i} \right). \]

Based on this expression, we see that the disclosure risk of \( y_i \) against the correlation-attack estimator is determined by \( \rho_{YY} \), \( \mu_Y \), and the value of \( y_i \). Therefore, the distributions of the original data and the noise generating variable jointly determine the disclosure risk of \( y_i \). We note that the log-normal distribution could go from roughly symmetric to heavily skewed. In this simulation we want to investigate the following questions: 1. Which kinds of log-normal distribution (e.g. symmetric or heavily skewed) are vulnerable to the correlation-attack. 2. Which noise generating variables provide more protection against the correlation-attack for original data sets which follow different log-normal distributions. 3. Which subset of the observations are more vulnerable to the correlation-attack. For example, do a few large observations have a higher risk of being attacked by the correlation-attack?

We conducted the following simulation: We chose four different original data sets \( y_1, y_2 \) and \( y_3 \), each with sample size 1000. Each data set is created by randomly drawing 1000 observations from a log-normal distribution \( LN(0, \sigma^2) \), and then multiply each observation by 10. The parameters of \( \sigma \) used for \( y_1, y_2 \) and \( y_3 \) are 0.5, \( \sqrt{2}/2 \) and \( \sqrt{3}/2 \), respectively. The distribution of \( y_1 \) is roughly symmetric, the distribution of \( y_2 \) is mildly skewed, the distribution of \( y_3 \) is heavily skewed. We chose five different noise generating variables to mask each set of data. The five noise generating variables we use are:

1. \( C_{2,1} \sim 0.5U(0.6, 0.9) + 0.5U(1.1, 1.4) \).

2. \( C_{2,2} \sim U(1 - \sqrt{0.84}/2, 1 + \sqrt{0.84}/2) \).

3. \( C_{2,3} \sim N(1, 0.07) \).

4. \( C_{2,4} \sim T(0.4755, 1.7245, 0.8) \).

5. \( C_{2,5} \sim T(0.2755, 1.5245, 1.2) \).

Note that \( C_{2,4} \) follows a left skewed triangular distribution and \( C_{2,5} \) follows a right skewed triangular distribution. For all 5 noise candidates, the means are 1 and the vari-
Table 4.4: Average disclosure risk of $y_1$ for each noise candidate.

<table>
<thead>
<tr>
<th>noise setting</th>
<th>average disclosure risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{2,1}$</td>
<td>0.385</td>
</tr>
<tr>
<td>$C_{2,2}$</td>
<td>0.394</td>
</tr>
<tr>
<td>$C_{2,3}$</td>
<td>0.410</td>
</tr>
<tr>
<td>$C_{2,4}$</td>
<td>0.399</td>
</tr>
<tr>
<td>$C_{2,5}$</td>
<td>0.406</td>
</tr>
</tbody>
</table>

Table 4.5: Average disclosure risk of $y_2$ for each noise candidate.

<table>
<thead>
<tr>
<th>noise setting</th>
<th>average disclosure risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{2,1}$</td>
<td>0.216</td>
</tr>
<tr>
<td>$C_{2,2}$</td>
<td>0.267</td>
</tr>
<tr>
<td>$C_{2,3}$</td>
<td>0.311</td>
</tr>
<tr>
<td>$C_{2,4}$</td>
<td>0.298</td>
</tr>
<tr>
<td>$C_{2,5}$</td>
<td>0.286</td>
</tr>
</tbody>
</table>

Table 4.6: Average disclosure risk of $y_3$ for each noise candidate.

<table>
<thead>
<tr>
<th>noise setting</th>
<th>average disclosure risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{2,1}$</td>
<td>0.164</td>
</tr>
<tr>
<td>$C_{2,2}$</td>
<td>0.232</td>
</tr>
<tr>
<td>$C_{2,3}$</td>
<td>0.278</td>
</tr>
<tr>
<td>$C_{2,4}$</td>
<td>0.265</td>
</tr>
<tr>
<td>$C_{2,5}$</td>
<td>0.246</td>
</tr>
</tbody>
</table>

ences are 0.07, and therefore the correlations between each set of original data and the noise candidates are the same. The five noise candidates have the following features:

1. We use a mixture of uniforms random variable $C_{2,1}$ to represent the group of noise generating variables with $R_{L,W} = 0;$ 2. $C_{2,2}$ and $C_{2,3}$ represent balanced noise generating variables. 3. $C_{2,4}$ and $C_{2,5}$ represent skewed noise generating random variables.

For each set of original data and noise candidate combination, we calculated the disclosure risk of each original value against the correlation-attack through 5000 iterations. For the three sets of original data, the correlations between the original data and the five noise candidates are 0.678, 0.876, 0.926, respectively. We calculated the average of the disclosure risks for all observations. The results are given in Table 4.4, Table 4.5, and Table 4.6.

In the following we use the average disclosure risk to quantify the protection level of each noise candidate. To interpret the tables, we see that for each choice of noise can-
didate, \( y_1 \) has the highest average disclosure risk. It might suggest that log-normally distributed data sets which are roughly symmetric are more vulnerable to the correlation-attack than log-normally distributed data sets which are skewed. Moreover, even though the correlation between \( y_4 \) and its noise-multiplied counterpart is the highest, the disclosure risk for \( y_4 \) is the lowest among the three sets of data. It might suggest that a higher correlation between the original data and the noise-multiplied data does not necessarily mean a higher disclosure risk. Results from Simulation 1 and other unreported simulations show that for two sets of original data, a higher correlation between the original data and its noise-multiplied version is associated with a higher disclosure risk if the two sets of data are distributed similarly.

In terms of which noise candidates protect the original data better, we see that in all cases, \( C_{2,1} \) gives the best protection to the original data. It might suggest that those noise candidates with \( R_{LW} = 0 \) also offer good protections to log-normally distributed original data in many cases. This observation is also showed in our simulation study in Section 3.4.2. In Section 3.4.2, Figure 3.4, we see that \( C_1 \) and \( C_4 \) offer the best protection to the original data. In that simulation, both \( C_1 \) and \( C_4 \) has \( R_{LW} \) very close to 0. In summary, it seems that the for a log-normally distributed data set, it is more vulnerable to the correlation-attack if its distribution is roughly symmetric, and it seems that those noise candidates with a low \( R_{LW} \) protect log-normally distributed data sets better in most cases.

Another question we intend to answer is, whether a particular subset of original observations are vulnerable to the correlation-attack. To investigate this question, for \( y_1, y_2 \) and \( y_3 \), we divided the original observations into 10 groups according to their magnitude. That is, the smallest first 100 observations are in the first group, the 101st smallest observations to the 200th smallest observations are in the second group, etc. We recorded the average disclosure risk of observations in each group. The results are given in Figure 4.1, Figure 4.2 and Figure 4.3.

Based on Figure 4.1, we see that for \( y_1 \) and noise candidates \( C_{2,3}, C_{2,4} \) and \( C_{2,5} \), the highest risk regions are around the median. However, the disclosure risk for \( C_{2,4} \) peaks to the left of the median, and the disclosure risk for \( C_{2,5} \) peaks to the right of the median. In other words, \( C_{2,4} \) provides more protection to large valued observations and \( C_{2,5} \) provides more protection to lower valued observations. Notice that \( C_{2,4} \) is right skewed and \( C_{2,5} \) is left skewed. Interestingly the peak of the disclosure risk shifts according to the skewness of the noise candidate.

Based on Figure 4.2 and Figure 4.3, we see that small original observations are more vulnerable to the correlation-attack than large observations if \( C_{2,4} \) is used. The reverse is true if \( C_{2,5} \) is used. This observation agrees with our results given in Figure 4.1. Therefore, these results may suggest that, for log-normally distributed original data sets, in order to provide more protections to small original observations, the data agency should choose a right-skewed noise generating variable; similarly, the data provider should choose a left-
Figure 4.1: average disclosure risk of $y_1$ in each region.

Figure 4.2: average disclosure risk of $y_2$ in each region.
skewed noise generating variable if the data agency wants to provide more protections to large-valued original observation. We will use this idea in the application study.

To summarize, simulations in this section might suggest the following: 1. The correlation-attack is more efficient for attacking data sets which are log-normally distributed and roughly symmetric, and less efficient for attacking data sets which are log-normally distributed and heavily skewed. 2. It looks like noise candidates with the value of $R_{LW}$ very close to 0 provide better protection against the correlation-attack than other noise candidates. 3. Given that the original data is log-normally distributed, it looks like a right-skewed noise generating variable is more efficient for protecting large-valued original observations, and a left-skewed noise generating variable is more efficient for protecting small-valued original observations.

4.1.2 Simulation 3

In Simulation 2 we have seen that the mixtures of uniforms noise generating variable ($C_{2,1}$) provides a better protection than all the other noise candidates. We speculated in Simulation 2 that noise candidates with $R_{LW} = 0$ provide better protections against the correlation-attack as well. In this simulation we nominate a few noise candidates with $R_{LW}$ being 0. We aim to compare the levels of protection offered by these noise candidates, and interpret our simulated result. We consider the following noise generating variables:

$C_{3,1}: 0.5T(0.6815127, 0.6815127, 0.9) + 0.5T(1.1, 1.3184873, 1.3184873)$.

$C_{3,2}: 0.5T(0.501669, 0.9, 0.9) + 0.5T(1, 1.1, 1.1, 1.498331)$. 
\[ C_{3,3} : 0.5T(1, 0.61197, 0.9, 0.755985) + 0.5T(1, 1.1, 1.38803, 1.244015). \]

\[ C_{3,4} : 0.5U(0.624, 0.9) + 0.5U(1.1, 1.376). \]

\( C_{3,1}, \) \( C_{3,2} \) and \( C_{3,3} \) follow mixtures of triangular distributions. This type of noise distribution was proposed in Kim (2007) and Kim and Jeong (2008) for generating noise terms. The means of these noise candidates are 1 and and the variances are 0.063. The original data we aim to protect is \( y_4 \). \( y_4 \) is generated by drawing 1000 observations from \( LN(0, 0.4) \), and then multiply each observation by 10.

Following the same steps as we did before, we recorded the average disclosure risk of \( y_4 \) for each noise candidate. The results are given in Table 4.7. The table shows that, even though all these noise candidates have \( R_{LW} = 0 \), their protection against the correlation-attack can still differ dramatically. In this simulation, \( C_{3,1} \) provides the best protection to \( y_4 \). We suggest that in practice, the data provider could come up with many differently distributed noise candidates with \( R_{LW} = 0 \), and assess their protection to the original data individually. Kim and Jeong (2008) proposed many candidate distributions with \( R_{LW} = 0 \). These noise distributions could all perform differently in different practical situations. The more choices the data provider has, the better the chance that the data provider will find an ideal noise candidate such that the desired utility-risk tradeoff is achieved.

**Table 4.7:** Average disclosure risk of \( y_4 \) for each noise candidate

<table>
<thead>
<tr>
<th>noise setting</th>
<th>average disclosure risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{3,1} )</td>
<td>0.238</td>
</tr>
<tr>
<td>( C_{3,2} )</td>
<td>0.280</td>
</tr>
<tr>
<td>( C_{3,3} )</td>
<td>0.244</td>
</tr>
<tr>
<td>( C_{3,4} )</td>
<td>0.262</td>
</tr>
</tbody>
</table>

4.1.3 Simulation 4

In this simulation we compare the performance of different “mixture of uniforms” noise generating variables. In the above simulations, we only tried mixtures with equal weights noise generating variables. However, it may be possible to use mixtures with unequal weights noise generating variables in practice. As we showed in Simulation 1, a skewed noise generating variable has the advantage of providing greater protection to certain portions of the original observations for log-normally distributed original data. We speculate that using mixtures with unequal weights noise generating variables could achieve the same outcome in many situations. For instance, for log-normally distributed original
Table 4.8: average disclosure risk of $y_4$ for each noise candidate.

<table>
<thead>
<tr>
<th>noise setting</th>
<th>average proportions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_{4,1}$</td>
<td>0.254</td>
</tr>
<tr>
<td>$c_{4,2}$</td>
<td>0.302</td>
</tr>
<tr>
<td>$c_{4,3}$</td>
<td>0.301</td>
</tr>
</tbody>
</table>

data, if the top 10% of the original observations require strong protection, then we may consider using a right-skewed mixtures with unequal weights noise generating variables to mask the whole original data, giving the top 10% original observations more protection and still certain levels of protection to the rest of the observations. In practice the data provider only needs to release the variance of the noise variable to the public, therefore the data intruder may not know that skewed noise variable has been used and more extreme observations have been given stronger protection. Therefore, in the following we assume that a correlation-attack is performed on the entire noise-multiplied data as an attempt to reveal confidential values.

We use the following three mixture of uniforms noise candidates with means equal to 1 and variances equal to 0.103 in this simulation:

$C_{4,1} : 0.5U(0.5, 0.9) + 0.5(1.1, 1.5)$.

$C_{4,2} : 0.7U(0.1904168, 0.9) + 0.3U(1.1, 1.289821)$.

$C_{4,3} : 0.7U(0.7101786, 0.9) + 0.3U(1.1, 1.809583)$.

The original data we use is $y_4$, the same as we used in Simulation 3. Note that $C_{4,1}$ is symmetric; $C_{4,2}$ is skewed to the left; $C_{4,3}$ is skewed to the right. Following the steps described in Simulation 1, we first computed and recorded the average disclosure risk for each noise candidate in Table 4.8. Then, we divided $y_4$ into 10 regions, with region 1 corresponding to the first 100 smallest observations. We calculated the average disclosure risk of observations in each region. The results are given in Table 4.9.

The results agree with our speculation. Even though $C_{4,1}$ provide the best protection, $C_{4,3}$ provide the best protection the better protections for large observations. Therefore we suggest that if, in a set of log-normally distributed original data, only extremely large (or small) observations require strong protection, the data provider could consider using skewed noise variables with $RLW$ being 0 to protect the original data.
Table 4.9: Average proportion of $y_4$ being disclosed by the correlational attack according to regions.

<table>
<thead>
<tr>
<th>noise setting</th>
<th>region1</th>
<th>region2</th>
<th>region3</th>
<th>region4</th>
<th>region5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_{4,1}$</td>
<td>0.022</td>
<td>0.294</td>
<td>0.405</td>
<td>0.310</td>
<td>0.211</td>
</tr>
<tr>
<td>$c_{4,2}$</td>
<td>0.084</td>
<td>0.144</td>
<td>0.138</td>
<td>0.105</td>
<td>0.075</td>
</tr>
<tr>
<td>$c_{4,3}$</td>
<td>0.000</td>
<td>0.181</td>
<td>0.643</td>
<td>0.703</td>
<td>0.611</td>
</tr>
<tr>
<td>noise setting</td>
<td>region6</td>
<td>region7</td>
<td>region8</td>
<td>region9</td>
<td>region10</td>
</tr>
<tr>
<td>$c_{4,1}$</td>
<td>0.174</td>
<td>0.177</td>
<td>0.232</td>
<td>0.330</td>
<td>0.390</td>
</tr>
<tr>
<td>$c_{4,2}$</td>
<td>0.190</td>
<td>0.428</td>
<td>0.658</td>
<td>0.699</td>
<td>0.503</td>
</tr>
<tr>
<td>$c_{4,3}$</td>
<td>0.388</td>
<td>0.150</td>
<td>0.079</td>
<td>0.109</td>
<td>0.140</td>
</tr>
</tbody>
</table>

4.2 Application

In this section we present an application study. The setting we use is the same as the setting considered in the real-life application study in Klein et al. (2014). We aim to show two things: 1. The noise candidates considered in their application study are not good for protecting against the correlation-attack. 2. In order to protect extreme income values, which is the requirement considered in their application, we apply some of the conclusions we got through our above simulations, so that extreme income values are protected against the correlation-attack.

In the following we describe the application setting. The data set is the public use data from the 2000 Current Population Survey (CPS) March supplement (available from http://www.census.gov/cps/). The entire data set contains household, family, and individual records. Klein et al. (2014) only considered household records whose income are positive as the original data set. We note that the original data roughly follows a log-normal distribution. Only the largest 10% of income values are sensitive and require protection. Denote household income data as $y = (y_1, y_2, \ldots, y_{50661})$ with $y_1 > y_2 > \cdots > y_{50661} > 0$. Extreme income observations $(y_1, y_2, \cdots, y_{5066})$ are sensitive. Klein et al. (2014) only applied multiplicative noise to these observations. Denote the noise-multiplied counterparts as $(z_1, z_2, \cdots, z_{5066})$. In Klein et al. (2014), they proposed four mixtures of uniforms noise generating variables $h_1, h_2, h_3$ and $h_4$. The distributions of these variable take the following form:

$$h_i \sim (1 - p) \cdot U(\xi_1, \xi_2) + p \cdot U(\xi_3, \xi_4).$$

The parameters used by each noise generating variables are:

$h_1$: $\xi_1 = 0.8, \xi_2 = 0.9, \xi_3 = 1.1, \xi_4 = 1.2, p = 0.5$
Table 4.10: Summary statistics of $R_\rho(y, 0.1)$ and $R_\rho(y, 0.2)$.

<table>
<thead>
<tr>
<th>noise setting</th>
<th>correlation</th>
<th>$\delta = 0.1$</th>
<th>$\delta = 0.2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$Q_1$</td>
<td>$Q_2$</td>
</tr>
<tr>
<td>$h_1$</td>
<td>0.94</td>
<td>0.21</td>
<td>0.33</td>
</tr>
<tr>
<td>$h_2$</td>
<td>0.74</td>
<td>0.26</td>
<td>0.41</td>
</tr>
<tr>
<td>$h_3$</td>
<td>0.72</td>
<td>0.23</td>
<td>0.35</td>
</tr>
<tr>
<td>$h_4$</td>
<td>0.50</td>
<td>0.05</td>
<td>0.19</td>
</tr>
</tbody>
</table>

$h_2$: $\xi_1 = 0.5, \xi_2 = 0.9, \xi_3 = 1.1, \xi_4 = 1.5, p = 0.8$

$h_3$: $\xi_1 = 0.5, \xi_2 = 0.9, \xi_3 = 1.1, \xi_4 = 1.5, p = 0.5$

$h_4$: $\xi_1 = 0.1, \xi_2 = 0.8, \xi_3 = 1.2, \xi_4 = 1.5, p = 0.8$

It is assumed that the data provider releases the perturbed data to the public. That is, $(z_1, z_2, \cdots, z_{5066}, y_{5067}, \cdots, y_{50661})$ are released to the public. In the following we consider the Case I in Klein et al. (2014). For Case I, the data provider releases the perturbed data, together with indications of which observations are noise-multiplied observations. When proposing these noise candidates, the authors did not consider disclosure risk from the correlation-attack. In this simulation we measure the disclosure risks against the correlation-attack for these noise candidates.

The correlation-attack is performed on $(z_1, z_2, \cdots, z_{5066})$ to disclose their original values. We set $\delta = 0.1$ and 0.2 and measure the disclosure risk respectively. We used Approach 2 mentioned in Section 3.3 to measure disclosure risks against the correlation-attack. We report the average correlation between $y$ and $z$ and the summary statistics of disclosure risks for $\delta = 0.1$ and $\delta = 0.2$ in Table 4.10.

To interpret Table 4.10, take noise setting $h_2$ for example. The $Q_2$ value when $\delta = 0.2$ is 0.71. It means that if the data provider use $h_2$ to mask $(y_1, y_2, \cdots, y_{5066})$, then 50% of observations have a disclosure risk of more than 0.71. Based on the table, we see that when $\delta = 0.1$, it looks like $h_4$ is the only reasonable choice as the average disclosure risk is 0.24, whereas other noise candidates result in relatively high disclosure risks. When $\delta$ is set to be 0.2, we see that even in the best case the average disclosure risk is 0.46, which means that none of the noise candidates provide good protection.

To reduce the disclosure risks of $(y_1, y_2, \cdots, y_{5066})$, we propose the following masking scheme: Instead of only perturbing $(y_1, y_2, \cdots, y_{5066})$, we perturb the entire data $(y_1, y_2, \cdots, y_{50661})$ using multiplicative noise. The noise generating variable we propose is $C_{4,3}$, the right-skewed mixture of uniforms noise generating variable in Simulation 4. The reason is that based on our observation in Simulation 1 and Simulation 4, it is
likely that a right-skewed mixture of uniforms noise variable provides better protection to largest 10% of the original observations. When $C_{4,3}$ is used, the correlation between the noise-multiplied data and original data is roughly 0.9. The correlation is very high, which might motivate a data intruder to use the correlation-attack. Suppose the correlation-attack is performed by using the entire data. We calculated the disclosure risk for $(y_1, y_2, \cdots, y_{5066})$, and we report the summaries statistics of the disclosure risks in Table 4.11.

We see from Table 4.11 that, $(y_1, y_2, \cdots, y_{5066})$ are now well protected against the correlation-attack. The average disclosure risks are reduced to 0.07 and 0.15 for $\delta = 0.1$ and 0.2, respectively. In summary, this application showed again the advantage of using skewed noise generating variables to achieve the desired level of protection for certain portions of original observations.

### 4.3 Conclusion and Future work

In this chapter we proposed several questions with regard to protecting against the correlation-attack for log-normally distributed original data sets. We presented several simulation results of using different types of noise generating variables to protect different log-normally distributed data sets. We have the following observations: First of all, extensive simulations show that noise candidates with $R_{LW} = 0$ offer better protection to the original data against the correlation-attack; Secondly, the simulations show that using right-skewed noise candidates could help the data provider to better protect larger observations. Similarly, using left-skewed noise candidates could help the data provider to better protect smaller observations. We presented an application of using skewed noise candidates to achieve a desired level of protection in a hypothetical scenario. The observations in this chapter could guide the process of noise generating variable selection for data providers in practice.
Chapter 5

Quantifying protection level of a noise candidate

From this chapter onward we assume the case where the density function of the noise generating variable $f_C$ is public knowledge. In Chapter 3, we proposed a disclosure risk measure which could be used by data providers for noise generating variable selection. However, besides the unbiased estimator and the correlation-attack estimator, there exist other estimators for attacking noise-multiplied data. Therefore, the disclosure risk measure we introduced in Chapter 3 might be too limited in its applicability. In this chapter we propose a measure for evaluating the protection level of a noise candidate. The proposed measure could be of practical use for noise generating variable selection.

Because the data provider has no control over which attacking strategy a data intruder might use, an ideal noise generating variable should provide enough protection to each original observation against all attacking strategies. The data provider might say that a noise candidate provides enough protection to the original data if for any original observation, the maximum value disclosure risk against any attacking strategy is below a threshold value. However, it is difficult to find such a noise candidate. Alternatively, the data provider might look at the average value disclosure risk for quantifying the protection level that noise candidate offers to the original data against an attacking strategy, which is the mean of the value disclosure risks of all original observations against the attacking strategy. For large-sized data, the average value disclosure risk provides a rough idea about the proportion of original observations which could be disclosed by the attacking strategy in a set of noise-multiplied data. The data provider might want the proportion to be below an acceptable level, especially if all observations might be identified or targeted. Under this setting, the data provider might say that a noise candidate offers enough protection to the original data if the average value disclosure risks against several attacking strategies are all below a given threshold level.

We consider the following setting in this chapter: to attack an original observation $y_i$, the data intruder uses an attacking estimator $g(Y_i^*)$, where $Y_i^* = y_iC$, $C$ is the noise
generating variable, and $g(Y_i^*)$ is a function of $Y_i^*$. We will show in Section 5.5, that there are many attacking estimators which satisfy this setting. We also assume the data protector uses the average value disclosure risk to quantify the protection level that a noise candidate offers to the original data against an attacking strategy. In practice, to evaluate the protection level of a noise candidate, the data provider might need to evaluate value disclosure risks against several attacking estimators. This is a complicated job for the data provider. In this chapter, we derive an attacking estimator, namely the optimal estimator, which maximizes the average value disclosure risk. Therefore, for a noise candidate, instead of measuring its value disclosure risks against several attacking estimators which are functions of $Y_i^*$, the data provider could simply measure the average value disclosure risk against the optimal estimator to understand the protection level the noise candidate provides to the original data. The data provider could use this single measure for noise generating variable selection in practice. We note that other attacking estimators which are not functions of $Y_i^*$ exist, such as the attacking estimator introduced in Klein et al. (2014). These cases are not considered in this chapter.

As noted in Chapter 3, some distributions have been proposed as good noise distributions in the literature because they offer strong protections against the unbiased estimator for any set of original data. We will show in the simulation study that these distributions might not provide good protections to the original data according to our protection level quantification measure.

This chapter is organised as follows: Section 5.1 introduces the utility loss measure we use in this chapter. Section 5.2 introduces the optimal estimator. Section 5.3 derives the optimal estimator. Section 5.4 introduces methods for evaluating value disclosure risks against the optimal estimator. Section 5.5 discusses the optimal estimator from the data intruder’s point of view. Section 5.6 presents a simulation study. Section 5.7 concludes the chapter.

### 5.1 Definition of overall data utility loss

As we noted in Chapter 2, moments and quantile estimates of $Y$ could be accurately estimated from the noise-multiplied data. For quantile estimates, we note that Sinha et al. (2011)’s approach requires a data user to have prior knowledge about the distribution of $Y$. On the contrary, using the reconstruction algorithm to estimate this information does not require prior knowledge and is very accurate. We note the following: 1. for large-sized sample, moments estimates of $Y$ are normally estimated accurately regardless of the distribution of $C$; 2. under the reconstruction algorithm approach, the accuracy of the quantile estimates of $Y$ depends on the accuracy of the reconstructed density estimate $\hat{f}_Y$. Unlike moments estimates, the accuracy of $\hat{f}_Y$ is largely affected by the distribution of $C$. (See results in Agrawal and Aggarwal (2001)). Therefore, in this chapter we adopt the
overall data utility loss measure proposed in Agrawal and Aggarwal (2001). The authors defined the overall data utility loss measure according to how much the reconstructed density estimates and the original density function overlap. Mathematically, the overall data utility loss is defined as:

\[ UL_{Y,C} = \frac{1}{2} E[ \int_{-\infty}^{\infty} |f_Y(x) - \hat{f}_{Y,C}(x)| dx ], \]

where \( \hat{f}_{Y,C}(x) \) is the reconstructed density function of \( f_Y \) given the data masked by \( C \). \( UL_{Y,C} \) is bounded between 0 and 1. This utility loss measure is similar to the pMSE which could be seen as a loss measure for synthetic data (Snoke et al. 2018). A low \( UL_{Y,C} \) value means that many parameter estimates of \( Y \) could be accurately estimated from the noise-multiplied data if \( C \) is the noise generating variable. We note that functionals of the distribution might be estimated directly better than by going through a density estimator (Bickel and Ritov 2003). We do not pursue this discussion in this chapter.

In practice, \( UL_{Y,C} \) could be computed in the following way: 1. the data provider estimates \( f_Y \) from \( y \) using either moment based density approximation or kernel density estimation. Denote the estimate as \( \hat{f}_Y \); 2. the data provider independently generates \( q \) sets of noise-multiplied data from \( Y^* \); 3. the data provider applies the sample-moment-based density approximation method (Lin 2014) on these sets of noise-multiplied data to obtain \( q \) realizations of \( \hat{f}_{Y,C} \); 4. For the \( k \)-th reconstructed density function \( \hat{f}_{Y,C}^k \) where \( k = 1, 2, \cdots, q \), the data provider computes \( \int_{-\infty}^{\infty} |\hat{f}_Y(y) - \hat{f}_{Y,C}^k(y)| dy \). The data provider estimates \( UL_{Y,C} \) using \( \frac{1}{2q} \sum_{k=1}^{q} \int_{-\infty}^{\infty} |\hat{f}_Y(y) - \hat{f}_{Y,C}^k(y)| dy \). We will use these steps to compute the overall utility loss in the simulation.

Once the disclosure risks and expected losses of several noise candidates have been estimated, the data provider can choose a suitable noise generating variable to mask the original data. The data provider could then apply the chosen \( C \) once and release the noise-multiplied data to the public.

5.2 The optimal estimator

In this section we introduce the optimal estimator which maximizes the average value disclosure risk. We assume the original data is only subject to attacking estimators which are functions of \( Y_i^* \). According to Chapter 2, the probabilistic value disclosure risk measure for evaluating the disclosure risk of \( y_i \) against an attacking estimator \( \hat{Y}_i \) takes the following form:

\[ R(\hat{Y}_i, \delta|Y_i = y_i) = P\left( \frac{|\hat{Y}_i - Y_i|}{Y_i} < \delta \big| Y_i = y_i \right), \]

where \( \hat{Y}_i = g(Y_i^*), Y_i^* = CY_i \).

We assume the data provider uses the average value disclosure risk to quantify the pro-
tection level a noise candidate $C$ offers to the original data against the attacking estimator $\tilde{Y}_i$. The average disclosure risk is defined as:

$$\{R(\tilde{Y}_i, \delta | Y_i = y_i)\}_{i=1}^{n} = \frac{1}{n} \sum_{i=1}^{n} R(\tilde{Y}_i, \delta | Y_i = y_i).$$

The data provider has no control over which attacking strategy a data intruder might use to attack the original data, therefore, to assess the protection level of a noise candidate, the data provider might need to evaluate the average value disclosure risk against many potential attacking estimators. A noise candidate is qualified for masking the original data if the average value disclosure risk is below a certain level. To facilitate the process, we propose an optimal estimator which maximizes the average probabilistic value disclosure risk. The optimal estimator takes the following form:

$$Z_i^{opt} = \arg\max_{g(Y_i^*)} \int_{g(Y_i^*)(1-\delta)}^{g(Y_i^*)(1+\delta)} f_{Y_i^*}(x)dx,$$

where $f_Y$ is the density function of $Y$, $Y_i^* = Y_iC$ and $f_{Y_i^*|Y_i^*}(x) = \frac{1}{x} f_C(\frac{Y_i^*}{x}) f_Y(x)$. The derivation of $Z_i^{opt}$ is given in the next section. Correspondingly, the value disclosure risk of $y_i$ against the optimal estimator $Z_i^{opt}$ is defined as

$$R_{opt}(y_i, C, \delta) = P(\left| \frac{Z_i^{opt} - Y_i}{Y_i} \right| < \delta | Y_i = y_i)$$

The average value disclosure risk of using $C$ to mask the original data against the optimal estimator is $R_{overall}(y, \delta, C)$. That is:

$$R_{overall}(y, \delta, C) = \frac{1}{n} \sum_{i=1}^{n} R_{opt}(y_i, C, \delta)$$

We propose that the data provider measures $R_{overall}(y, \delta, C)$ to quantify the level of protection a noise candidate $C$ offers to the original data. If $R_{overall}(y, \delta, C)$ is below an acceptable level, then $C$ offers enough protection to the original data against any other estimators which are functions of $Y_i^*$. We note that when $y_i$ is subject to other forms of attacks which are not solely functions of $Y_i^*$, such as the one proposed in Klein et al. (2014), the data provider might need to use $R_{overall}(y, \delta, C)$ in conjunction with other corresponding average value disclosure risk measures to quantify the protection level of $C$. In this paper we assume the original data is only subject to attacking estimators which are functions of $Y_i^*$. 
5.3 Derivation of $Z_i^{opt}$

Suppose for a set of original data $\{y_i\}_{i=1}^n$, the following probabilistic disclosure risk measure is used by the data provider:

$$P(\frac{\bar{y}_i - y_i}{y_i} < \delta) = P(\frac{\bar{y} - y}{y} < \delta), i = 1, \ldots, n$$

We assume $Y > 0$, $C > 0$, $\bar{Y} = g(Y^*)$, where $Y^* = CY$. We observe that the disclosure risk of an observation $t$ is given as:

$$P\left(\frac{|g(Y^*) - Y|}{Y} < \delta | Y = t\right) = \int_0^\infty I\left(\frac{g(t^*)}{1+\delta} < t < \frac{g(t^*)}{1-\delta}\right) f_{Y^*|Y}(t^*|Y = t) dt^*,$$

where $t^*$ is the noise-multiplied version of $t$. The conditional probability $P\left(\frac{|g(Y^*) - Y|}{Y} < \delta | Y\right)$ is a function of random variable $Y$.

The average disclosure risk for the original data is

$$R_{overall} = \frac{\sum_{i=1}^n P\left(\frac{|g(Y^*) - Y|}{Y} < \delta | Y = y_i\right)}{n}$$

Suppose $E_Y\left(P\left(\frac{|g(Y^*) - Y|}{Y} < \delta | Y\right)\right)$ exists, therefore we have

$$R_{overall} \xrightarrow{n \to \infty} E_Y\left(P\left(\frac{|g(Y^*) - Y|}{Y} < \delta | Y\right)\right)$$

as $n \to \infty$.

The objective is to find an expression of $g(Y^*)$ which maximizes $R_{overall}$ as $n \to \infty$.

Because $\{Y^*|Y = t\} = tC$, therefore $f_{Y^*|Y}(t^*|Y = t) = \frac{1}{t} fc\left(\frac{t^*}{t}\right)$. We observe the following:

$$E_Y\left(P\left(\frac{|g(Y^*) - Y|}{Y} < \delta | Y\right)\right) = \int_0^\infty \int_0^\infty I\left(\frac{g(x^*)}{1+\delta} < x < \frac{g(x^*)}{1-\delta}\right) f_{Y^*|Y=x^*}(x^*) dx^* f_Y(x) dx$$

$$= \int_0^\infty \int_0^\infty I\left(\frac{g(x^*)}{1+\delta} < x < \frac{g(x^*)}{1-\delta}\right) \frac{1}{x} fc\left(\frac{x^*}{x}\right) f_Y(x) dx dx^*$$

$$= \int_0^\infty f_{Y^*}(x^*) \int_0^\infty I\left(\frac{g(x^*)}{1+\delta} < x < \frac{g(x^*)}{1-\delta}\right) f_{Y|Y^*=x^*}(x) dx dx^*$$

$$= \int_0^\infty f_{Y^*}(x^*) \int_0^{\frac{g(x^*)}{1+\delta}} f_{Y|Y^*=x^*}(x) dx dx^*$$

Therefore, $E_Y\left(P\left(\frac{|g(Y^*) - Y|}{Y} < \delta | Y\right)\right)$ is maximized if $\int_0^{\frac{g(x^*)}{1+\delta}} f_{Y|Y^*=x^*}(x) dx$ is maximized.

The form of $g(x^*)$ which maximizes $\int_0^{\frac{g(x^*)}{1+\delta}} f_{Y|Y^*=x^*}(x) dx$ is
CHAPTER 5. QUANTIFYING PROTECTION LEVEL OF A NOISE CANDIDATE

In this section we discuss how the data provider could find \( R_{opt}(y_i, C, \delta) \) in practice.

**Case 1: \( f_y \) is known.** If \( f_y \) is known to the data provider then the mathematical expression of \( Z_i^{opt} \) might be derived analytically. Suppose \( Q(y) \) is the antiderivative of \( \frac{1}{y} f_c(Y/y) f_y(y) \), \( H(x) = \int_{\frac{1}{x}}^{\frac{1}{x+1}} \frac{1}{y} f_c(Y/y) f_y(y) dy \), then

\[
H(Z^{opt}_i) = \int_{\frac{1}{Z^{opt}_i}}^{\frac{1}{Z^{opt}_i+1}} \frac{1}{Z^{opt}_i} f_c(Y_i) f_y(y) dy = Q(\frac{Z^{opt}_i}{1-\delta}) - Q(\frac{Z^{opt}_i}{\delta+1})
\]

To find the argument \( Z_i^{opt} \) where \( H(Z_i^{opt}) \) is maximized, we take derivative of \( H(Z_i^{opt}) \) to yield:

\[
\frac{dH(Z^{opt}_i)}{dZ^{opt}_i} = \frac{1}{Z^{opt}_i f_c(Y_i Z_i^{opt}) f_y(Z_i^{opt})} - \frac{1}{Z^{opt}_i f_c(Y_i (\delta+1) Z_i^{opt}) f_y(Z_i^{opt})}
\]

We set \( \frac{dH(Z^{opt}_i)}{dZ^{opt}_i} = 0 \) to find \( Z_i^{opt} \). As an example, if \( Y \sim LN(\mu_1, \sigma_1^2) \), \( C \sim LN(\mu_2, \sigma_2^2) \), we can show that

\[
Z^{opt}_i = \exp(\frac{(\sigma_1^2 + \sigma_2^2)\ln[(1+\delta)(1-\delta)] + 2\sigma_1^2\ln(y_i C) - 2\sigma_1^2 \mu_2 + 2\sigma_2^2 \mu_1}{2(\sigma_1^2 + \sigma_2^2)})
\]

Correspondingly, we could obtain \( R_{opt}(y_i, C, \delta) \) for each \( y_i \) if the expression of \( Z_i^{opt} \) is available.

**Case 2: \( f_Y \) is unknown.** In practice, the data provider may only have a set of original data \( \{y_i\}_{i=1}^n \) available without knowing the true underlying distribution of \( Y \). To find \( R_{opt}(y_i, C, \delta) \) in this case, the data provider might need to estimate the density function \( f_y \). For instance, the data provider might obtain a kernel density estimate \( K_Y \) based on \( \{y_i\}_{i=1}^n \), and replace \( f_y(Y/Y_i) \) by \( \frac{1}{\gamma} f_c(Y/Y_i) K_Y(y) \) in Equation (5.1) to approximate \( Z_i^{opt} \), and then approximate \( R_{opt}(y_i, C, \delta) \) for each \( y_i \).
CHAPTER 5. QUANTIFYING PROTECTION LEVEL OF A NOISE CANDIDATE

The data provider might have different strategies to approximate $R_{opt}(y_i, C, \delta)$. In the simulation study, we assume $Y$ is bounded between $[a, b]$ and $C$ is bounded between $[c, d]$, where $a$ and $b$ are the minimum and maximum of the original data, and $(a, b, c, d) > 0$. Therefore, we note that $Y^*$ is bounded between $[ac, bd]$. We adopt the following strategy to approximate $R_{opt}(y_i, C, \delta)$:

- Step 1: We generate $n_1$ realizations of $y_i^*$, where $y_i^* \sim y_i C$. Denote them as $\{y_i^*\}^{n_1}_{i=1}$;

- Step 2: We note that the support of $f_{Y|Y^*}$ is $[max(a, \frac{y_i^*}{Y}), min(b, \frac{y_i^*}{Y})]$. Therefore, for each $y_i^*$, we take $n_2$ points between $[max(a, \frac{y_i^*}{d}), min(b, \frac{y_i^*}{c})]$ with equal increment $Q$. Denote the points as $\{h_v\}^{n_2}_{v=1}$;

- Step 3: For each $h_v$, we attempt to estimate $\int_{\frac{h_v}{1+\delta}}^{\frac{h_v}{1-\delta}} f_{Y|Y^*}(y)dy$. $Y$ is bounded between $[a, b]$, therefore $\frac{h_v}{1+\delta}$ cannot be lower than $a$ and $\frac{h_v}{1-\delta}$ cannot be greater than $b$. So

$$\int_{\frac{h_v}{1+\delta}}^{\frac{h_v}{1-\delta}} f_{Y|Y^*}(y)dy = \int_{\max(a, \frac{h_v}{1+\delta})}^{\min(b, \frac{h_v}{1-\delta})} f_{Y|Y^*}(y)dy.$$

To estimate this integral, we take $n_3$ points between $max(a, \frac{h_v}{1+\delta})$ and $min(b, \frac{h_v}{1-\delta})$ with equal increment $\Delta$. Let $m_j = max(a, \frac{h_v}{1+\delta}) + (j-1)\Delta$. We approximate the integral by $I_{h_v} = \sum_{j=1}^{n_3} \frac{1}{m_j} f(c(y_i^*/m_j))K_y(m_j)\Delta$. As a result, for $\{h_v\}^{n_2}_{v=1}$, we obtain a corresponding set $\{I_{h_v}\}^{n_2}_{v=1}$;

- Step 4: We let $z_{i,y_i^*}^{opt} = \arg\max_{h_v} \{I_{h_v}\}^{n_2}_{v=1}$. Then $z_{i,y_i^*}^{opt}$ is the optimal estimate of $y_i$ given $y_i^*$;

- Step 5: We repeat the above process for each $y_i^*$, $l = 1, 2, \cdots, n_1$. As a result, we obtain $\{z_{i,y_i^*}^{opt}\}^{n_1}_{i=1}$;

- Step 6: We count the number of $z_{i,y_i^*}^{opt}$ such that $z_{i,y_i^*}^{opt} \in [y_i(1-\delta), y_i(1+\delta)]$. Denote the number as $n_4$;

- Step 7: We estimate $R_{opt}(y_i, C, \delta)$ by $n_4/n_1$. 
We use the above steps for estimating $R_{opt}(y_i, C, \delta)$ in the simulation study. We note that the above approach is naive and might not be the most efficient way for estimating $R_{opt}(y_i, C, \delta)$. However, it is sufficient for illustration purposes.

5.5 Discussion

In this section we discuss the optimal estimator and other possible attacking estimators mainly from data intruders’ point of view. For illustration, we use another three attacking estimators for estimating an original value $y_i$. The three attacking estimators will also be used in the simulation detailed in Section 5.6:

**Unbiased estimator:** Nayak et al. (2011) showed that $Y^*_i = CY_i$ is an unbiased estimator of $y_i$ as $E(Y^*_i|Y_i = y_i) = y_i$. Therefore, $Y^*_i$ is the unbiased estimator for $y_i$.

**Correlation-attack estimator:** In Chapter 3 we showed that when the population correlation between $Y$ and $Y^*$ is high, then a simple linear model might be adequate to explain the relationship between the two variables. The correlation-attack estimator $\hat{Y}^*_i$ takes the following form: $\hat{Y}^*_i = (1 - \rho^2_{Y^*Y})\mu_Y + \rho^2_{Y^*Y} Y^*_i$, where $\rho^2_{Y^*Y}$ is the population correlation between $Y$ and $Y^*$, and $\mu_Y$ is the population mean of $Y$.

**Maximum a posteriori (MAP) estimator:** The MAP estimator is simply the mode of $f_{Y_i|Y^*_i}(y)$, where $f_{Y_i|Y^*_i}(y) = \frac{1}{y} f_C(\frac{Y_i}{y}) f_Y(y)$. Denoting the estimator as $Z_i$, then $Z_i = \text{argmax}_y f_{Y_i|Y^*_i}(y)$.

**Discussion 1:** Interpretation of $z^opt_i$ from the data intruder’s point of view

Given $y^*_i$, a realization of $Z^opt_i$ takes the following form:

$$z^opt_i = \text{argmax}_{\hat{Y}^*_i} \int \frac{g(\hat{Y}^*_i)}{1 + \delta} f_Y(\hat{Y}^*_i = y^*_i) dy = \text{argmax}_{\hat{Y}^*_i} \int \frac{g(\hat{Y}^*_i)}{1 + \delta} \frac{1}{y} f_C(\frac{\hat{Y}^*_i}{y}) f_Y(y) dy.$$

From the data intruder’s point of view, suppose the data intruder has no prior knowledge of $y_i$ and assumes that $y_i \sim Y_i$. Given $y^*_i$, suppose the data intruder uses an estimate $g(y^*_i)$ to attack $y_i$, then the probability that $g(y^*_i)$ discloses $y_i$ could be expressed as

$$P(|\frac{g(y^*_i) - Y_i}{Y_i} - \delta| Y^*_i = y^*_i) = P(\frac{g(Y^*_i)}{1 + \delta} < Y_i < \frac{g(Y^*_i)}{1 - \delta} | Y^*_i = y^*_i).$$

The expression of $z^opt_i$ means that $z^opt_i$ maximizes the above posterior probability of
disclosing \( y_i \). Therefore, if the data intruder wants to maximize the posterior probability of disclosing \( y_i \), the data intruder may use \( z_i^{opt} \) to attack \( y_i \).

To find \( z_i^{opt} \), the data intruder needs information of \( f_C(\frac{Y_i}{\gamma}) \) and \( f_Y(y) \). Because \( f_C \) and \( y_i^* \) are public knowledge, the data intruder knows \( f_C(\frac{Y_i}{\gamma}) \). However, \( f_Y(y) \) is unknown and needs to be estimated. The data intruder could estimate \( f_Y \) by using Lin’s reconstruction algorithm which we reviewed in Chapter 2. As a result, the data intruder could obtain an approximated value of \( z_i^{opt} \) by replacing \( f_Y(y) \) by \( \hat{f}_Y(y) \) in the expression of \( z_i^{opt} \).

**Discussion 2: Estimator selection from the data intruder’s point of view**

To attack \( y_i \), the data intruder might use any of the four estimators \( Y_i^*, \hat{Y}_i, Z_i^{opt} \) and \( Z_i \). For the data provider, it could use the corresponding value disclosure risk measures to tell which attacking estimator is more effective for attacking \( y_i \). For instance, in the simulation study, we can show that if \( y_i \) is around 185000, then it has a value disclosure risk of 1 against the optimal estimator if \( C_1 \) is used to mask the original data. Therefore, the data provider knows that if the optimal estimator is used to attack the corresponding noise-multiplied value \( y_i^* \), it will lead to value disclosure. However, the data intruder might not know that the optimal estimator will surely lead to value disclosure of \( y_i^* \).

To attack \( y_i^* \), the data intruder might come up with his own rule for determining which attacking estimator to use to disclose \( y_i \). For instance, the data intruder might use the correlation-attack estimator if the estimated sample correlation between the noise-multiplied data and the original data is very high. Alternatively, as we argued in Discussion 1, \( Z_i^{opt} \) maximizes the data intruder’s posterior probability of disclosing \( y_i \). If the probability is used as a decision rule, then the data intruder will use the optimal estimator to attack \( y_i \). Regardless, the decision rule used by a data intruder does not necessarily lead to the best choice of estimator for attacking \( y_i \). Therefore, the data provider might be less worried about those original values which suffer very high disclosure risks against an attacking estimator when evaluating the protection level of a noise candidate \( C \), especially if the corresponding records have low identity disclosure risks.

**Discussion 3: Value disclosure risk measure and the worst-case DP**

When releasing noise-multiplied data to the public, the worst-case differential privacy (DP) could be achieved. In that way, no private information could be revealed even if an adversary knows all information about the data except the private information. This requirement could be achieved in some statistical limitation methods, such as \( \epsilon \)-differentially private synthetic data (Snoke et al. 2018). Our value disclosure risk measure aims to ensure a sufficient level of uncertainty when a data intruder with no prior knowledge attacks the noise-multiplied data, but it does not guarantee the worst-case DP. Releasing data which satisfy the worst-case DP might render a high loss of data utility.
and therefore might not be considered by a statistical agency. How to generate noise-multiplied data which achieves the worst-case DP while maintaining enough data utility requires future study.

5.6 Simulation study

In this section we present a simulation study using the data we used in Simulation 2 of Chapter 3. However in this case we only consider the top 5% household income values under household income attribute as the original data. We consider several noise candidates. The original data contains 2533 positive observations ranging from 140000 to 768742. We denote the original data as \(\{y_i\}_{i=1}^{2533}\). We set the acceptance rule \(\delta = 0.1\) throughout this section. We consider the following four noise candidates:

\[
C_1 \sim 0.5U(0.5,0.9) + 0.5U(1.1,1.5);
\]

\[
C_2 \sim 0.5U(0.3,0.9) + 0.5U(1.1,1.7);
\]

\[
C_3 \sim U(1 - 0.5\sqrt{93/75},1 + 0.5\sqrt{93/75});
\]

\[
C_4 \sim U(0.245,1.755).
\]

Among the four noise candidates, \(C_1\) and \(C_3\) have the same variance 0.103. \(C_2\) and \(C_4\) have the same variance 0.190. \(C_1\) and \(C_2\) represent those types of noise candidate which have been advocated by researchers as they offer strong protections against the unbiased estimator.

We first measured the value disclosure risks of each \(y_i\) against each of the four attacking estimators. To find \(R_{opt}(y_i,C,0.1)\), we obtained a kernel density estimate \(K_Y\) based on the original data by using the function ‘density()’ with default parameters in R. We computed the average value disclosure risks against each attacking estimator. The results are given in Table 5.1. To comment on the table, we see that \(C_1\) and \(C_2\) provide very good protection to the original data against the unbiased estimator because the corresponding average value disclosure risks are 0. However, comparing \(C_1\) with \(C_3\) (with the same variance), we see that \(C_3\) protects the original data better against the other three estimators than \(C_1\). A similar story can be seen when we compare \(C_2\) and \(C_4\). The result shows that a noise candidate which protects the original data well against one attacking estimator may not provides sufficient protection against other estimators. The result may also suggest that, those noise candidates represented by \(C_1\) and \(C_2\) which have been advocated by some researchers might not provide better protections than other noise candidates.

Denote \(R_{cor}(y_i,C,0.1)\) as the value disclosure risk of \(y_i\) against the correlation-attack estimator if \(C\) is used to mask \(y_i\). We provide value disclosure risks \(\{R_{cor}(y_i,C_1,0.1)\}_{i=1}^{2533}\)
against the original data $\{y_i\}_{i=1}^{2533}$ plot in Figure 5.1, and we provide $\{R_{opt}(y_i, C_1, 0.1)\}_{i=1}^{2533}$ against $\{y_i\}_{i=1}^{2533}$ plot in Figure 5.2. We also provide similar plots under the case of $C_3$ in Figure 5.3 and 5.4. We see that, for an attacking strategy, a noise candidate might not be able to protect all observations. For instance, we see in Figure 5.2 that some observations have value disclosure risks of 1 against the optimal estimator. We also see that, a noise candidate which protects an original observation well against one attacking estimator might not protect it well against another estimator. For instance, we see in Figure 5.4 that observations around 200000 only have value disclosure risks of around 0.2 against the optimal estimator. However, the corresponding value disclosure risks against the correlation-attack estimator are above 0.3. The story is reversed for observations greater than 500000. While it might be desirable for a noise candidate to be able to protect all observations against all attacking strategies, these two figures showed that it is difficult to find such a noise candidate. Therefore, using average value disclosure risk to quantify the level of protection of a noise candidate offers to the original data against an attacking estimator, and ensuring that the mean disclosure risk is below a certain level seems to be one reasonable criteria for noise generating variable selection. There are other alternatives. For instance, the data provider might select a noise candidate which ensures that the maximum value disclosure risk against several attacking estimators for each observation is below a certain level.

For each of the noise candidates, we computed the level of overall utility loss. To compute this value, take $C_1$ for instance, we firstly produced $q$ samples of noise-multiplied data using $C_1$. For the $i$-th sample of noise-multiplied data, we used the R-package MaskDensity14 (Lin and Fielding 2015) to generate the reconstructed density function $\hat{f}_{Y,C_1}^i$. Thus, we obtained $q$ reconstructed density functions. We estimated $UL(f_Y, C_1)$ by

$$\frac{1}{2q} \sum_{i=1}^{q} \int_{-\infty}^{\infty} |K_Y(y) - \hat{f}_{Y,C_1}^i(y)| dy,$$

where $K_Y$ is the kernel density estimate of $Y$. Denote $\hat{f}_{Y,C}(y) = \sum_{i=1}^{q} \hat{f}_{Y,C_1}^i / q$. Figure 5.5 and 5.6 provide $\hat{f}_{Y,C}(y)$ and $K_Y(y)$ plots for illustrating the accuracy of the reconstructed density functions. The overall utility losses for the four noise candidates are 0.190, 0.209, 0.230 and 0.204 respectively. Except for $C_4$, we see that there is a trade-off between $R_{overall}(C)$ (last column of Table 5.1) and the overall utility loss. We see that $C_1$ offers the lowest level of overall utility loss but the worst protection to the original data, while $C_3$ offers a better protection at the expense of a higher amount of utility loss. The data provider could choose a noise candidate which achieves the desired utility-risk tradeoff for masking the original data.

In this simulation, we see from Table 5.1 that $R_{overall}(C_4) = 0.519$, meaning that it offers the highest level of protection to the original data among the four noise candidates. To find a noise candidate with better protection, we tried various noise distributions with
variances larger than the variances of the above four noise candidates and we found several noise candidates. For instance, we found that if $C \sim LN(-0.144, 0.536^2)$ truncated between 0 and 5 is considered, then $R_{overall}(C)$ is only 0.424, which provides a better protection than $C_4$. In practice the data provider could decide its own threshold level such that $R_{overall}(C)$ needs to be below the threshold level in order for $C$ to be considered.

5.7 Conclusion

In this chapter we proposed a measure for quantifying the protection level of a noise candidate $C$ for noise multiplication masking scheme. To attack the original data from its noise-multiplied version, the data intruder might adopt different attacking strategies. From the data provider’s perspective, we argued that to quantify the protection level of a noise candidate against a particular attacking strategy, the data provider might look at the average value disclosure risk. We proposed an optimal attacking estimator which maximizes the average value disclosure risk, and the corresponding maximized average value disclosure risk is the measure for quantifying the protection level a noise candidate $C$ offers to the original data. As a result, the data provider could use this single measure instead of using multiple value disclosure risk measures for noise generating variable selection in practice. We note that in this chapter we assumed an attacking estimator is a function of the noise-multiplied variable. Relaxing this assumption for more generalised results requires future study.

Table 5.1: The average value disclosure risks against each attacking estimator

<table>
<thead>
<tr>
<th>noise</th>
<th>unbiased estimator</th>
<th>correlation-attack estimator</th>
<th>MAP estimator</th>
<th>optimal estimator</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>0</td>
<td>0.357</td>
<td>0.543</td>
<td>0.628</td>
</tr>
<tr>
<td>$C_2$</td>
<td>0</td>
<td>0.307</td>
<td>0.508</td>
<td>0.567</td>
</tr>
<tr>
<td>$C_3$</td>
<td>0.179</td>
<td>0.327</td>
<td>0.501</td>
<td>0.547</td>
</tr>
<tr>
<td>$C_4$</td>
<td>0.124</td>
<td>0.301</td>
<td>0.484</td>
<td>0.519</td>
</tr>
</tbody>
</table>
CHAPTER 5. QUANTIFYING PROTECTION LEVEL OF A NOISE CANDIDATE

**Figure 5.1:** \( \{R_{cor}(y_i, C_1, 0.1)\}_{i=1}^{2533} \) against income values plot.

**Figure 5.2:** \( \{R_{opt}(y_i, C_1, 0.1)\}_{i=1}^{2533} \) against income values plot.
CHAPTER 5. QUANTIFYING PROTECTION LEVEL OF A NOISE CANDIDATE

**Figure 5.3:** $\{R_{cor}(Y_i, C_3, 0.1)\}_{i=1}^{2533}$ against income values plot.

**Figure 5.4:** $\{R_{opt}(Y_i, C_3, 0.1)\}_{i=1}^{2533}$ against income values plot.
CHAPTER 5. QUANTIFYING PROTECTION LEVEL OF A NOISE CANDIDATE

Figure 5.5: $K_Y$ and $\hat{f}_{Y,C}$ plots for $C_1$. The solid lines are density plots of the original data. The dashed lines are reconstructed density plots.

Figure 5.6: $K_Y$ and $\hat{f}_{Y,C}$ plots for $C_3$. The solid lines are density plots of the original data. The dashed lines are reconstructed density plots.
Chapter 6

Attacking Noise-multiplied Data given extrema information

In this chapter we recognise that even though only noise-multiplied data is released to the public, the values of the extrema of the unpublished original data, i.e. the maximum and minimum of the original data, could be learned by some data intruders in some contexts. The first context is protecting business microdata. The noise multiplication masking method has the potential to be applied to business microdata. A typical character of business data is that it contains a few very large observations relative to the rest observations. Multiplicative noise might be considered as a good way to perturb each individual observation in the business data as it offers uniform protections to all observations (Nayak et al. 2011) and multiplicative noise is more suitable for economic modeling of income data (Kim and Winkler 2003). When masking business microdata, a typical problem is that the top few respondents of an attribute, such as the top two respondents, know the ranks of their responses in the original data. Another context is that, the extrema could be reasonably estimated from noise-multiplied data using the quantile estimation methods we reviewed in Chapter 2.

This chapter has two purposes. The first purpose is to identify an attacking strategy by assuming that the extrema of the original data are available. The second purpose is to introduce and investigate a modification of the traditional noise multiplication masking method, namely controlled noise multiplication masking which guarantees that no noise-multiplied observation could be attacked by the above strategy. The chapter is organised as follows: Section 6.1 introduces the utility loss measure we adopt in this chapter. Section 6.2 reviews methods for estimating the extrema of the original data from the noise-multiplied data. Section 6.3 introduces the attacking strategy. Section 6.4 introduces methods for protecting against the attacking strategy. Section 6.5 presents three simulation studies. Section 6.6 concludes the chapter.
6.1 Definition of overall data utility loss

We adopt the same utility loss measure as we did in Chapter 4. That is, suppose the original observations are independently drawn from $Y$. The data user could get an estimate of $f_Y$ based on the perturbed data. The accuracy of the estimate of $f_Y$ depends on the data provider’s choice of noise variable $C$. Denote the data user’s estimate of $f_Y$ by $\hat{f}_{Y,C}$. The overall data utility loss is defined as:

$$UL(f_Y, C) = \frac{1}{2} E\left[\int_{-\infty}^{\infty} |f_Y(y) - \hat{f}_{Y,C}(y)| dy\right]$$

$UL(f_Y, C)$ is bounded between 0 and 1, with 0 indicating no utility loss.

6.2 Estimating the extrema of the original data from the noise-multiplied data

We consider univariate data. For most data intruders, the extrema of the original data are not known. However, due to the two quantile estimation methods we reviewed in Chapter 2, it is possible for data intruders to estimate the extrema of the original data from the noise-multiplied data. In this section, we discuss how to estimate this piece of information using the quantile estimation methods.

Estimating the extrema using R-package MaskDensity14

As we have described in Section 2.1.2, for MaskDensity14 to work, the data provider releases a noise.bin file. The noise.bin file contains two parameters $a$ and $b$. The values of $a$ and $b$ might be very close to the minimum and maximum of the original data. As a result, the data intruder might simply use the values of $a$ and $b$ to estimate the minimum and maximum of the original data.

Estimating the extrema through Sinha et al’s quantile estimation

The quantile estimation method has been introduced in Section 2.1.2. To data intruder might use the method to estimate the minimum and maximum of the original data.

The authors showed in their simulation study how different combinations of prior/noise affect estimations of order statistics of the original data which is roughly symmetrically distributed. For instance, the authors noted that in their example, when a normal prior is used, the quantile estimation method produces very accurate estimates over the minimum, maximum and median of the original data if the original data was perturbed by normal noise or log-normal noise. We will present a simulation study in Section 6.5.2.
6.3 An attacking strategy

Without loss of generality, we assume that the original data and noise terms are positive. Suppose the data intruder knows the extrema of \( y \), which are denoted as \( y_{\min} \) and \( y_{\max} \). At the moment we assume all noise terms are bounded between \((c_l, c_u)\). That is, suppose \( \{c_i\}_{i=1}^n \) are noise terms used to mask \( y \), with \( c_{\min} = \min(\{c_i\}_{i=1}^n) \) and \( c_{\max} = \max(\{c_i\}_{i=1}^n) \). Then \( c_l \leq c_{\min} \) and \( c_u \geq c_{\max} \). In the following we assume the data intruder knows \( c_l \) and \( c_u \). In practice, the data intruder might use the lower bound and upper bound of \( C \) as estimates for \( c_l \) and \( c_u \).

The values in \( y \) are bounded by \( y_{\min} \) and \( y_{\max} \) and the noise terms are bounded by \( c_l \) and \( c_u \). Thus for sufficiently low and sufficiently high published values there are restricted bounds for the noise terms.

Let \( \varepsilon_l \) be a constant value. If a published noise-multiplied value \( y_i^* \) is below \( y_{\min}(c_l + \varepsilon_l) \), i.e. \( y_i c_l \leq y_{\min}(c_l + \varepsilon_l) \). Then we have \( c_l \leq \frac{y_{\min}}{y_i}(c_l + \varepsilon_l) \).

As \( y_{\min} \leq y_i \), therefore \( c_l \leq c_l + \varepsilon_l \). In other words, \( y_i \) is bounded between \( c_l \) and \( c_l + \varepsilon_l \). The data intruder might use the mid-point between \( c_l \) and \( c_l + \varepsilon_l \) as an estimate of \( c_i \). Denoting the estimate as \( \tilde{c}_i \), \( \tilde{c}_i = c_l + 0.5\varepsilon_l \). Consequently, the data intruder’s estimate of \( y_i \) is \( \tilde{y}_i = \frac{y_i^*}{c_l + 0.5\varepsilon_l} \).

The relative error between \( y_i \) and \( \tilde{y}_i \) is \( |\frac{\tilde{y}_i}{y_i} - 1| \). It is maximized if \( c_l \) is equal to either \( c_l \) or \( c_l + \varepsilon_l \). To limit the relative error to a given value \( \alpha \), i.e. \( \frac{c_l + \varepsilon_l}{c_l + 0.5\varepsilon_l} - 1 = \alpha \), the value of \( \varepsilon_l \) is equal to \( \frac{2\varepsilon_l\alpha}{1 - \alpha} \).

Similarly, let \( \varepsilon_u \) be a constant value. If a published noise-multiplied value \( y_j^* \) is above \( y_{\max}(c_u + \varepsilon_u) \), that means \( c_u - \varepsilon_u \leq c_j \leq c_u \). The data intruder could then estimate \( c_j \) using \( \tilde{c}_j \), where \( \tilde{c}_j = c_u - 0.5\varepsilon_u \). The data intruder’s estimate of \( y_j \) is \( \tilde{y}_j = \frac{y_j^*}{c_u - 0.5\varepsilon_u} \). To limit the relative error of the estimate to a given value \( \alpha \), \( \varepsilon_u \) is equal to \( \frac{2\varepsilon_u\alpha}{1 - \alpha} \).

Therefore, suppose the data intruder would like to unmask certain noise-multiplied values with relative errors less than \( \alpha \). The data intruder could adopt the following attacking strategy:

**If a noise multiplied value \( y_i^* \) is below \( y_{\min}(c_l + \varepsilon_l) \), then the data intruder could estimate \( y_i \) using \( \tilde{y}_i \), where \( \tilde{y}_i = \frac{y_i^*}{c_l + 0.5\varepsilon_l} \). If a noise multiplied value \( y_j^* \) is above \( y_{\max}(c_u + \varepsilon_u) \), then the data intruder could estimate \( y_j \) using \( \tilde{y}_j \), where \( \tilde{y}_j = \frac{y_j^*}{c_u - 0.5\varepsilon_u} \).**

Since the density function of \( C \) is available, we could work out the expected relative error of the estimates. That is,

\[
E(\tilde{D}_j) = \int_{c_u - \varepsilon_u}^{c_u} \left| \frac{c}{c_u - 0.5\varepsilon_u} - 1 \right| \left[ \frac{f_c(c)}{\int_{c_u - \varepsilon_u}^{c_u} f_c dc} \right] dc
\]

When \( C \) is uniformly distributed over \((c_u - \varepsilon_u, c_u)\), which is the case for Simulation 1
in Section 6.5, we have that

\[
E(\hat{D}_j) = \int_{c_u-\epsilon_u}^{c_u} \left| \frac{c}{c_u-0.5\epsilon_u} - 1 \right| \frac{1}{\epsilon_u} dc
\]

\[
= \int_{c_u-0.5\epsilon_u}^{c_u} \left( \frac{c}{c_u-0.5\epsilon_u} - 1 \right) \frac{1}{\epsilon_u} dc + \int_{c_u-\epsilon_u}^{c_u-0.5\epsilon_u} \frac{1}{\epsilon_u} dc
\]

\[
= \frac{0.25\epsilon_u}{2(c_u-0.5\epsilon_u)} + \frac{0.25\epsilon_u}{2(c_u-0.5\epsilon_u)}
\]

\[
= \epsilon_u/(4c_u - 2\epsilon_u) = \alpha/2
\]

In situations where the interval \((c_u - \epsilon_u, c_u)\) is very small, we may roughly assume multiplicative noises are uniformly distributed over the interval and expect the relative error to be roughly \(\alpha/2\). Similarly, we define the relative error of \(\tilde{y}_i\) as

\[
\tilde{d}_i = \frac{\tilde{y}_i - y_i}{y_i},
\]

we could easily show that

\[
E(\tilde{D}_i) = \epsilon_l/(4c_l + 2\epsilon_l) = \alpha/2.
\]

As a simple example, suppose a uniform noise variable is used, and a data intruder sets \(\alpha\) to be 0.05. Then the intruder could get \(\epsilon_l = \frac{10}{92} c_l\) and \(\epsilon_u = \frac{10}{105} c_u\). Those \(y_i^*\) such that \(y_i^* \leq y_{\text{min}}(c_l + \epsilon_l)\) or \(y_i^* \geq y_{\text{max}}(c_u - \epsilon_u)\) could be disclosed by the intruder using the attacking strategy with relative errors no greater than 0.05 with an expected relative error of 0.025. We have the following discussions about implementing the attacking strategy in practice:

**Discussion 1**: When the noise generating variable \(C\) is unbounded, the data intruder cannot know \(c_l\) and \(c_u\). However, we argue that the data intruder could still use the proposed attacking strategy to attack a considerable amount of noise-multiplied observations with a high success rate. In the following we assume that the data intruder only wishes to attack extremely small noise-multiplied observations. The data intruder could apply the attacking strategy to the noise-multiplied data \(y_i^*\) through the following steps:

**Step 1.** The data intruder estimates a value \(q\) such that \(P(C < q) \leq k\). \(k\) should be a small number, for instance 0.5%.

**Step 2.** Replace \(c_l\) by \(q\) in Step 2 of the attacking strategy, and use the attacking strategy to unmask observations.

Following the above steps, we claim that the expected proportion of successful unmaskings will be worse than \((1 - k)\%\) but it should be greater than \((1 - n_k/m)\%\), where \(n_k = nk\), \(m\) is the number of observations which are attacking (those noise-multiplied observations falling below \(y_{\text{min}}(q + \epsilon_q)\), where \(\epsilon_q = \frac{2a\alpha}{1-a}\)). To justify it, note that in order
for our attacking strategy to be successful for unmasking $y_i^*$, two conditions need to be hold:

**Condition 1:** $y_i^* \leq y_{\min}(q + \varepsilon_q)$;

**Condition 2:** $q$ is the true lower bound for $c_i$ such that $c_i$ is bounded by $[q, q + \varepsilon_q]$.

Since there are $m$ noise-multiplied observations falling below $y_{\min}(q + \varepsilon_q)$, therefore we could attack those observations using the attacking strategy. Due to regression to the mean, the expected number of attacked observations with noise terms that are less than $q$ will be greater than $mk$. As a result, the expected proportion of successful unmaskings will be worse than $(1 - k)\%$. However note that the expected number of noise terms that are less than $q$ is $n_k$. In an extreme case where all these $n_k$ noise terms were used to mask those $m$ observations which were being attacked, the success rate of unmasking will be $(1 - n_k/m)\%$. Therefore in general we should expect to see the success rate of unmaskings to be between $(1 - n_k/m)\%$ and $(1 - k)\%$. An example will be given in Simulation 1 in Section 6.5.1.

**Discussion 2:** If multiplicative noises were applied to business data, the attacking strategy could also be effectively used by the largest and second largest data respondents. It is because for business data, it is likely that the top few largest data respondents know the ranks of their responses in the data (Hundepool et al. 2012).

For example, for the annual profit data in a typical business microdata, suppose the second largest data respondent knows that its annual profit is the second largest in the unobservable original data. Denote the unobservable original data as $(y_{[max]}, y_{[2]}, y_{[3]}, \cdots, y_{[n]})$, where $y_{[k]}$ is the $k$-th largest observation. For the second largest respondent, it knows that its contribution is $y_{[2]}$, and it knows that $y_{[2]}$ is the true upper bound for $(y_{[3]}, y_{[4]}, \cdots, y_{[n]})$. In this case, it could substitute $y_{max}$ by $y_{[2]}$ in the attacking strategy as an attempt to disclose some observations of $(y_{[3]}, y_{[4]}, \cdots, y_{[n]})$.

Specifically, if a noise-multiplied business value $y_i^* = y_i c_i > y_{[2]}(c_u - \varepsilon_u)$, therefore we have that $\frac{y_{[2]}}{y_i}(c_u - \varepsilon_u) \leq c_i \leq c_u$. Except for $y_i = y_{max}$, in all other scenarios we have $\frac{y_{[2]}}{y_i} > 1$. Therefore for these scenarios we have $(c_u - \varepsilon_u) \leq c_i \leq c_u$ for sure. As a result, for these scenarios, we could use $\hat{y}_i = \frac{y_i^*}{c_u - 0.5\varepsilon_u}$ to estimate $y_i$ with a desired error bound.

Therefore, suppose the second largest data respondent could find $h$ noise-multiplied observations falling above $y_{2}(c_u - \varepsilon_u)$. Among the $h$ noise-multiplied observations, in the worst scenario the noise-multiplied counterpart of $y_{max}$ is included. In this case the attacking strategy is still valid for unmasking the remaining $h - 1$ noise-multiplied observations. It is also possible that the noise-multiplied counterpart of $y_{max}$ is not included in the $h$ noise-multiplied observations. In this case the attacking strategy is
valid for unmasking all $h$ observations. Therefore, the success rate of unmasking these $h$ noise-multiplied observation is either $100(1 - 1/h)\%$ or $100\%$.

**Discussion 3:** When the data intruder does not have the knowledge of the extrema values $y_{\min}$ and $y_{\max}$, the data intruder needs to estimate them mathematically, for instance using the methods in Section 6.2. The data intruder may obtain estimates of $\hat{y}_{\min}$ and $\hat{y}_{\max}$, and use the estimates to replace the true values when implementing the attacking strategy. We note that in order for the attacking strategy to return estimates with desired error bounds, we need:

(a) $\hat{y}_{\min} \leq y_{\min}$ and (b) $\hat{y}_{\max} \geq y_{\max}$.

For instance, if we use Step 2 to unmask a noise-multiplied value $y_{i}^* = y_{i}c_{i}$, in order for our attacking strategy to work we need to have that $\frac{y_{\min}}{y_{i}} < 1$, so that $c_{i}$ is bounded between $[c_{l}, c_{l} + \epsilon_{l}]$. However, if we replace $y_{\min}$ by $\hat{y}_{\min}$ in Step 2 of the attacking strategy, our attacking strategy only works if $\frac{\hat{y}_{\min}}{y_{i}} < 1$ as well, so that we could unmask $y_{i}^*$ with the desired error bound.

If we have that $\hat{y}_{\min} < y_{\min}$, our attacking strategy works well. However, since the number of noise-multiplied observations falling below $\hat{y}_{\min}(c_{l} + \epsilon_{l})$ is less than or equal to those falling below $y_{\min}(c_{l} + \epsilon_{l})$, the data intruder could only attack fewer noise-multiplied observations and vice versa for the other side.

If (a) and (b) are not satisfied, i.e. $\hat{y}_{\min} > y_{\min}$ or $\hat{y}_{\max} < y_{\max}$, the relative errors of the estimates could be larger than $\alpha$, but the estimates are still good if $\hat{y}_{\min}$ is close to $y_{\min}$ and similarly for $\hat{y}_{\max}$. We show this point by calculating the upper bound and the expected relative errors of the estimates. Suppose $\hat{y}_{\max} \in [(1 - \eta)y_{\max}, y_{\max}]$. The data intruder replaces $y_{\max}$ by $\hat{y}_{\max}$ in the attacking strategy and unmasked several noise-multiplied observations whose values are greater than $\hat{y}_{\max}(c_{u} - \epsilon_{u})$. Denote a particular attacked noise-multiplied observation as $y_{j}^* = y_{j}c_{j}$, and denote the estimate of its original value $y_{j}$ as $\hat{z}_{j}$. The relative error of $\hat{z}_{j}$ is $|\frac{c_{j}}{c_{u} - 0.5\epsilon_{u}} - 1|$. Since $\hat{y}_{\max} \geq (1 - \eta)$, we have that $c_{j} \in [(1 - \eta)(c_{u} - \epsilon_{u}), c_{u}]$. The upper bound of the relative error of $\hat{z}_{j}$ occurs when $c_{j} = (1 - \eta)(c_{u} - \epsilon_{u})$, and it can be shown that the upper bound is $\frac{\eta(c_{u} - \epsilon_{u}) + 0.5\epsilon_{u}}{c_{u} - 0.5\epsilon_{u}}$.

If multiplicative noises are assumed to be uniformly distributed in $[(1 - \eta)(c_{u} - \epsilon_{u}), c_{u}]$, then the expected relative error of $\hat{z}_{j}$ is

$$ERE = \frac{(K\eta + 0.5\epsilon_{u})^2 - K^2 - 0.75\epsilon_{u}^2 - 2K\epsilon_{u} + c_{u}^2}{2(K + 0.5\epsilon_{u})(\eta K + \epsilon_{u})},$$

where $K = c_{u} - \epsilon_{u}$. Both results get smaller as $\eta$ goes smaller, meaning that the intruder’s estimate $\hat{z}_{j}$ is more accurate if the intruder’s estimate $\hat{y}_{\max}$ is closer to $y_{\max}$. The relationship between the ERE and $\eta$ for $c_{u} = 2$ and $\alpha = 0.05$ is given in Figure 6.1.
CHAPTER 6. ATTACKING NOISE-MULTIPLIED DATA GIVEN EXTREMA INFORMATION

Figure 6.1: This Figure shows the relationship between ERE and $\eta$, given that $c_u = 2$ and $\alpha = 0.05$. It shows that as $\eta$ goes smaller, the expected relative error ERE goes smaller as well. Because a smaller $\eta$ means that $\hat{y}_{\text{max}}$ is closer to $y_{\text{max}}$, therefore the attacking strategy is more effective if $\hat{y}_{\text{max}}$ is closer to $y_{\text{max}}$.

Therefore, even if the true values of $y_{\text{max}}$ and $y_{\text{min}}$ are not available, the attacking strategy still imposes disclosure risk if the extrema can be accurately estimated.

6.4 Data provider’s approaches against the proposed attacking strategy

As different data intruders possess different knowledge about the original data, it is reasonable to assume that some data intruders possess the knowledge of the extrema of the original data. Moreover, as we will show in the simulation study, the extrema could be reasonably estimated by data intruders from the noise-multiplied data using the methods we introduced in Section 6.2. In both cases the data provider needs to avoid disclosure risk from the attacking strategy. In the following we discuss how to generate noise-multiplied data which are free from the attacking strategy.

Approach 1: The data provider might only supply moments of $C$ instead of $f_C$ to the public. In this way, the lower and upper bounds of $C$ are not available to the data intruder. For instance, the data provider could release the noise-multiplied data $y^*$ together with $\{E(C), E(C^2), E(C^3), E(C^4)\}$ to the public. In this way, data users could still obtain many useful statistics, such as the first four moments estimates by using the moment estimation method proposed in Nayak et al. (2011). However, some statistical inferences which requires $f_C$ cannot be drawn. For instance, Sinha et al. (2011)’s quantile estimation approach calculates the conditional posterior distribution of $Y_i | y_i^*$ (see Equation 2.3). If
$f_C$ is not available, then this estimation approach cannot be used.

**Approach 2:** For a set of noise-multiplied data, the data provider might apply the attacking strategy to the noise-multiplied data first by setting $\alpha = \delta$. Note that for the data provider, $y_{\text{min}}$ and $y_{\text{max}}$ are known. Suppose only a few noise-multiplied observations could be attacked by the attacking strategy. In that case, for each of those original values, the data provider might generate a new noise term to be multiplied with the original value so that the noise-multiplied value cannot be attacked by the strategy. Formally speaking, the data provider might generate the noise-multiplied data in the following way:

1. Generate a set of noise-multiplied data $y^*$ and apply the attacking strategy to $y^*$.

2. If there is only a small proportion of original observations whose noise-multiplied versions either fall below $y_{\text{min}}(c_l + \varepsilon_l)$ or above $y_{\text{max}}(c_u - \varepsilon_u)$, then reproduce the noise terms from $C$ for those original observations to generate new noise-multiplied observations. Leave the remaining noise-multiplied observations (those falling between $y_{\text{min}}(c_l + \varepsilon_l)$ and $y_{\text{max}}(c_u - \varepsilon_u)$) unchanged.

3. Repeat process 2 until there are no observations falling below $y_{\text{min}}(c_l + \varepsilon_l)$ or above $y_{\text{max}}(c_u - \varepsilon_u)$.

We name the above process **controlled noise multiplication masking method**. Denote the noise-multiplied data generated in this way as $z^*$. The data provider releases $z^*$ together with $f_C$ to the public. To differentiate, we name the standard noise multiplication masking method as **ordinary noise multiplication masking method**. We note that a similar perturbation method which combines noise addition with minimum data swapping method was investigated in Kim and Winkler (1995). The authors recognised that applying noise addition masking alone does not provide enough protection to the CPS-IRS linked file, and an additional data swapping step is needed to further reduce the re-identification rate. The authors showed that the utility loss due to the data swapping step is acceptable. The advantage of the Approach 2 over the Approach 1 is that it might allow data users to obtain more statistical information from the noise-multiplied data.

When consider using the controlled noise multiplication masking method, the data provider might say that the noise-multiplied data produced by the Approach 2 carries enough information if the utility losses between the controlled noise-multiplied variable $Z^*$ and the ordinary noise-multiplied variable $Y^*$ are similar. We will show an application in Simulation 3 and we adopt the measure introduced in Section 4.1 for measuring utility loss.
6.5 Simulation

In this section we present three simulation studies: In Simulation 1 we demonstrate the attacking strategy by applying it to the simulation study in Klein et al. (2014). In Simulation 2 we estimate the extrema of the original data by using Sinha et al. (2011)'s quantile estimation approach. In Simulation 3 we illustrate the controlled noise multiplication method.

6.5.1 Simulation 1

We firstly illustrate an application of the attacking strategy under a case study in Klein et al. (2014). to attack a few extremely small noise-multiplied observations. Klein et al. (2014) considered the following masking scenario (Corresponding to Case I noise-multiplied data in their paper): consider an income attribute which contains $n$ observations, and denote the $i$-th person’s income value as $y_i$. An income observation is considered sensitive if it is extremely large (exceeding a threshold value $t$). All sensitive income values are masked by multiplicative noise. Therefore, denote the income value for the $i$-th record in the noise-multiplied attribute as $z_i$, then

$$z_i = \begin{cases} 
\frac{y_i c_i}{y_i} & y_i > t, \\
y_i & y_i \leq t.
\end{cases} \quad (6.1)$$

The data provider releases the following information to the public: 1. the noise-multiplied income data $\{z_i\}_{i=1}^n$; 2. the threshold value $t$; 3. indication of whether an observation is noise-multiplied. With this information, Klein et al. (2014) developed an Expectation-Maximization method to obtain log-normal regression coefficients by treating the income attribute as the response variable.

If the data provider uses a noise generating variable with a lower bound $C_l$ to mask the original data, which is public information, then the data intruder could use $C_l$ as the lower bound of the noise terms. If this is the case, then with the knowledge of the threshold value $t$, the data intruder could carry out the attacking strategy to unmask a small amount of noise-multiplied income observations which fall below a threshold level, because $t$ could be treated as the minimum of the original data which are masked by multiplicative noise. To illustrate this, we consider the simulate study in Section 5 of Klein et al. (2014). The simulation study uses public use data from the 2000 Current Population Survey (CPS) March Supplement. The original data contains positive incomes of 50661 households. Denote the original data as $(y_1, y_2, \cdots, y_{5066}, \cdots, y_{5066})$, where $y_1 \geq y_2 \geq \cdots \geq y_{5066}$. In this example, threshold is defined as the 90th percentile of the income data, therefore $t=106305$. Therefore, observations $y = (y_1, y_2, \cdots, y_{5066})$ are masked by multiplicative noise, where $y_1 \geq y_2 \geq \cdots \geq y_{5066} \geq t$. The data provider releases noise-multiplied data
Table 6.1: Results of applying the attacking strategy to Klein et al.’s simulation.

<table>
<thead>
<tr>
<th>noise candidate</th>
<th>mean no.attacked</th>
<th>sd of mean no.attacked</th>
<th>mean rel.errs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>749.63</td>
<td>20.31</td>
<td>0.051</td>
</tr>
<tr>
<td>$h_2$</td>
<td>58.79</td>
<td>7.46</td>
<td>0.049</td>
</tr>
<tr>
<td>$h_3$</td>
<td>147.10</td>
<td>11.40</td>
<td>0.049</td>
</tr>
<tr>
<td>$h_4$</td>
<td>6.77</td>
<td>2.59</td>
<td>0.049</td>
</tr>
</tbody>
</table>

$\{z_i\}_{i=1}^{50661}$ to the public, where $z_i$ is as defined in Equation (6.1). The authors proposed four mixture of uniforms noise generating variables $h_1, h_2, h_3$ and $h_4$ to perturb $y$. The distributions of these variables are:

$h_1$: $0.5U(0.8,0.9) + 0.5U(1.1,1.2)$;

$h_2$: $0.8U(0.5,0.9) + 0.2U(1.1,1.5)$;

$h_3$: $0.5U(0.5,0.9) + 0.5U(1.1,1.5)$;

$h_4$: $0.8U(0.1,0.8) + 0.2U(1.2,1.5)$;

To illustrate that the setting is vulnerable to the attacking strategy, for each noise candidate, we generated 5000 samples of noise-multiplied data and applied the attacking strategy on each sample by assuming the role of the data intruder. We assume $\alpha = 0.1$, i.e. the data intruder would like to use the attacking strategy to disclose some noise-multiplied values with relative error bounds less than 0.1. For noise candidate $h_1$, since the lower bound of the noise terms is $c_l = 0.8$, $\varepsilon_l = 8/45$, where $\varepsilon_l$ is defined in Section 6.3. According to the attacking strategy, noise-multiplied values which are below $t(c_l + \varepsilon_l) = 103,942.67$ could be unmasked with error bound less than 0.1. For each sample of the noise-multiplied data, we recorded the total number of observations being attacked, and the relative errors of the estimates. We computed the average of the relative errors. Based on the 5000 results, we calculated the mean and standard deviation of the total number of observations being attacked in each sample, and the mean of the average relative errors of all the samples. We did the same process for all the noise generating variables. The results are given in Table 6.1.

To interpret Table 6.1, taking noise candidate $h_2$ for instance, the average number of noise-multiplied observations being attacked in each sample (mean no.attacked) is 58.79 with a standard deviation (sd of mean no.attacked) 7.46, meaning that roughly speaking, for each sample of the noise-multiplied data, around 59 noise-multiplied observations could be unmasked by the attacking strategy with relative errors less than 0.1. The
standard deviation is relatively small, meaning that the number of observations being attacked in each sample is consistent. The mean of the averages of relative errors of estimates (mean rel.errs.) is around 0.05, which agrees with the expected relative errors we showed in Section 6.3 as in this case the noises are uniformly distributed over the interval \((c_i, c_i + \varepsilon_i)\).

In practice, the data provider could use an unbounded noise generating variable to perturb the original data. We argued in Discussion 1 of Section 6.3 that the attacking strategy could still be used under this case. We illustrate it through the following example: We consider an unbounded mixture of normals noise generating variable \(h_5\), where \(h_5 \sim 0.5N(0.7, 0.01) + 0.5N(1.3, 0.01)\). Suppose \(h_5\) was used to mask \(y = (y_1, y_2, \cdots, y_{5066})\). Following the steps in Discussion 1, we set \(k = 0.31\%\). We drew \(n\) sets of noise samples from \(h_5\), and we found that \(q = 0.45\). In other words, on average only around 15.79 noise terms were below 0.45 in each set of noise sample. Therefore, \(q = 0.45\) is the lower bound for approximately 5050 noise terms which were used to perturb \(y\). We repeated the masking-attacking process 1000 times. During the attacking process, we replace \(c_l\) by \(q = 0.45\) when calculating \(\varepsilon_l\) and the threshold value. The result shows that when \(\alpha = 0.15\), on each iteration there were on average 92.27 observations being attacked (noise-multiplied value below the threshold value \(t(q + \varepsilon_l)\)). Following Discussion 1, it is expected that the proportion of successful unmaskings will be worse than 99.69% but greater than 82.6%. Our simulation result showed that the average rate of successful unmaskings across the 1000 iterations was around 89.4%, which agrees with our claim.

### 6.5.2 Simulation 2

In this section we illustrate Sinha et al.’s quantile estimation approach for estimating extrema of the original data. We do not illustrate using MaskDensity14 to estimate extrema of the original data because many numerical examples are presented in Lin (2014) and Lin and Fielding (2015). We consider the quantile estimation method because we want to show that under certain contexts, it is possible for some intruders to come up with a reasonable prior which could lead to accurate estimation of the extrema of the original data. We wish to alert the data provider that the potential disclosure risk due to this fact needs to be evaluated during the data masking stage, and the choice of noise generating variable plays a role in determining the accuracy of the estimates.

The distribution of income data could be modelled as a log-normal distribution (Klein et al. 2014 and references therein). An intruder could use this form of prior in conjunction with Equation (2.3) for quantile estimation. The log-normal distribution is characterised by two parameters \(\mu\) and \(\sigma\), which could easily be estimated by the data intruder from the released noise-multiplied data. We still consider the data we used in Simulation 1 as the original data.
Recall that the original data consists of 50661 observations. Denote the original income data as \((y_1, y_2, \cdots, y_{50661})\), where \(y_1 \geq y_2 \geq \cdots \geq y_{50661}\). Now we suppose that the data provider uses \(h_3\) to produce noise-multiplied data, and releases the noise-multiplied data \((z_1, z_2, \cdots, z_{50661})\) to the public, where \(z_i\) is as defined in Equation (6.1) with \(t = 106305\). We assume the role of the data intruder and aim to estimate the largest original value \(y_1\) using the quantile estimation method with a log-normal prior. To decide the two parameters of the prior, we model the distribution of \((y_1, y_2, \cdots, y_{50661})\) as \(LN(\mu, \sigma^2)\), where \(\mu\) and \(\sigma^2\) are estimated using \(\hat{\mu}\) and \(\hat{\sigma}^2\). To find \(\hat{\mu}\) and \(\hat{\sigma}^2\), since the population mean and variance of the log-normal random variable are \(exp(\mu + \sigma^2/2)\) and \([exp(\sigma^2) - 1]exp(2\mu + \sigma^2)\), we set \(exp(\hat{\mu} + \hat{\sigma}^2/2)\) equal to the mean of the original data, \([exp(\hat{\sigma}^2) - 1]exp(2\hat{\mu} + \hat{\sigma}^2)\) equal to the variance of the original data. We have \(\hat{\mu} = 10.568\) and \(\hat{\sigma}^2 = 0.619\). The mean and variance of the original data could be estimated from the noise-multiplied data \((z_1, z_2, \cdots, z_{50661})\) by using \(\hat{\mu} = \sum_{i=1}^{50661} z_i / \sum_{i=1}^{50661} z_i^2 + \sum_{j=50661}^{50661} z_j^2 - \hat{\mu}^2\). Consequently, the prior distribution of \(y = (y_1, y_2, \cdots, y_{50661})\) is set to be \(LN(10.568, 0.619)\) truncated at \([106304, \infty)\).

We now apply the masking-estimating process 50 times. On each iteration the original data \(y = (y_1, y_2, \cdots, y_{50661})\) was masked by \(h_3\) to generate noise-multiplied data \(y^* = (y_1^*, y_2^*, \cdots, y_{50661}^*)\). Then, we estimated the value of \(y_1\) by using the quantile estimation approach with the truncated log-normal prior. We drew 50 replicates of sample from \(\{Y_i | y_i^*\}_{i=1}^{50661}\) and estimated the value of \(y_1\) using the sample mean of the last order statistics of the 50 replicates. We repeated the masking-estimating process 50 times and obtained 50 estimates of \(y_1\). The real value of \(y_1\) is 768742, and the average of the 50 estimates we obtained was 696467.5 with standard deviation 71353.32. Among the 50 estimates, 18 of them were within 10% of the true value of \(y_1\). The distribution of the 50 estimates is given in Figure 6.2(b).

In this simulation, the prior distribution is reasonable, but there is still a difference between the prior distribution (dotted black lines) and the actual density of \(y\) (black line) as we see in Figure 6.2(a). However, the posterior distribution of \(y\) (black points) is closer to the real distribution of \(y\). The average of the estimates of \(y_1\) is not too far from its real value. The results indicate that the estimates of the extrema could be very accurate if a good prior distribution is available.

We can show that the choice of noise generating variable plays a role in terms of the accuracy of the estimates. If \(h_1\) was used as the noise generating variable, we can show that the average of the estimates of \(y_1\) is around 744597, while if \(h_2\) was used as the noise generating variable, then the average of the estimates is around 716754. Therefore \(h_1\) leads to a better accuracy of the estimates of \(y_1\), and hence a higher risk against the attacking strategy. The purpose of this simulation is to illustrate that the extrema of the original data could be estimated, and therefore the data provider needs to consider the disclosure risk that might occur due to this fact.
Figure 6.2: Figure 6.2(a) shows the distribution of the original data (solid black line), distribution of the prior (dotted black line), posterior distribution of the original data based on a particular set of noise multiplied data (black points); Figure 6.2(b) shows the distribution of the 50 estimates of $\hat{y}_1$. The true value of $y_1$ is 768742.
6.5.3 Simulation 3

In this section we show an application of using the controlled noise multiplication masking method to generate noise-multiplied data.

The current performance of MaskDensity14 works the best when the original data is symmetric. For this simulation, the original data we use is still the CPS household income data (all 50661 observations), but after a logarithm transformation. Moreover, we cut off those transformed values which are less than 7.5 in order to further reduce the skewness in the original data. After these steps, the original data is distributed as shown in Figure 6.3 (black triangles). Denote the new original data as \( y \). We have \( y_{\text{min}} = 7.501 \) and \( y_{\text{max}} = 13.552 \).

Suppose the data provider feels that the knowledge of \( y_{\text{min}} \) and \( y_{\text{max}} \) is possessed by some data intruders, the data provider might consider using the controlled noise multiplication masking method to produce noise-multiplied data. In the following we assume the role of the data provider and we set acceptance rule \( \delta = 0.075 \). It means that we do not want data intruders to be able to attack any noise-multiplied observation with a relative error less than 7.5% of the true value using the proposed attacking strategy. We use a uniform variable \( U(0.7, 1.3) \) to generate noise terms. Initial assessment shows that if the ordinary noise multiplication masking method was applied to the original data, then only around 0.66% noise-multiplied observations could be disclosed by the attacking strategy.

We generated 30 sets of controlled noise-multiplied data, and measure the overall utility loss to determine if the amount of information carried by the controlled noise-multiplied variable \( Z^* \) is sufficient.

To measure the overall data utility loss, we applied MaskDensity14 to reconstruct the distribution of the original data from each set of the controlled noise-multiplied data. We recorded the difference of the areas enveloped by the two density curves. Based on all 30 sets of noise-multiplied data, we computed the overall data utility loss \( UL(f_y, Z^*) \), where

\[
UL(f_y, Z^*) = \frac{1}{60} \sum_{k=1}^{30} \int_{-\infty}^{\infty} |f_y(y) - f_{Y,C}^k(y)| \, dy,
\]

where \( f_{Y,C}^k(y) \) is the density approximant of \( f_y \) obtained from the \( k \)-th sample of the controlled noise-multiplied data. When we applied MaskDensity14, we set the two parameters \( a = 0.99y_{\text{min}} \) and \( b = 1.01y_{\text{max}} \) to achieve a better density reconstruction, where \( a \) and \( b \) are the lower and upper bounds of the reconstructed density functions. The result showed that the overall data utility loss due to applying the controlled noise multiplication is 23.4%.

For parallel comparison, we also generated 30 sets of ordinary noise-multiplied data using the ordinary noise multiplication masking method. The result showed that the overall data utility loss of applying ordinary noise multiplication is 20.4%. That means that...
the controlled noise multiplication masking method only results in a slightly higher data utility loss in this case. To visualise the results, we computed the average of the reconstructed density functions \( \hat{f}_Y(v) \), where \( \hat{f}_Y(v) = \sum_{k=1}^{30} \hat{f}_{Y,C}^k(v) / 30 \), \( \hat{f}_{Y,C}^k(v) \) is the density at point \( v \) of the reconstructed density function obtained from the \( k \)-th set of controlled noise-multiplied data. Similarly we computed the average of the reconstructed density functions based on the 30 sets of ordinary noise multiplied data. The results of density reconstructions are presented in Figure 6.3. As we can see, the average reconstructed density function based on the controlled noise-multiplied data is similar to the average reconstructed density function based on the ordinary noise-multiplied data.

To numerically compare the statistical analysis results (moments estimates and quantile estimates) that could be obtained from the two types of noise-multiplied data, we analysed the 30 sets of the controlled noise-multiplied data and also the 30 sets of the ordinary noise-multiplied data. For the original data, the first three moments were (10.53, 111.64, 1191.66). The first three quartiles were (9.96, 10.6, 11.15). By analysing the controlled noise-multiplied data, the averages of the first three moments estimates obtained from the 30 sets of controlled noise-multiplied data were (10.51, 111.22, 1183.19) with standard errors (0.006, 0.139, 2.24) respectively. The averages of the first three quartiles were (9.93, 10.63, 11.23) with standard errors (0.104, 0.085, 0.083). By analysing the ordinary noise-multiplied data, the averages of the first three moments estimates were (10.53, 111.66, 1192.04) with standard errors (0.0004, 0.095, 1.56). The averages of the first three quartiles estimates were (9.90, 10.58, 11.24) with standard errors (0.054, 0.051, 0.062). In summary, the averages of these statistics obtained from the two types of the noise-multiplied data were quite similar. The results suggested that the controlled noise multiplication masking method works well in this simulation.

### 6.6 Conclusion and future work

In this chapter we introduced an attacking strategy against multiplicative noise protection by recognising that the extrema of the original data could be learned by some data intruders. We showed through simulation that the proposed attacking strategy could effectively unmask certain noise-multiplied observations. We discussed methods to avoid the disclosure risk from the data provider’s point of view, especially the controlled noise multiplication masking method. We showed through simulation that it is able to preserve high data utility in certain contexts, and therefore could be used as an alternative masking approach when the ordinary noise multiplication masking method cannot deliver satisfactory protection to the original data.
Figure 6.3: The Figure shows the density plot of the original data (black triangles); the plot of the average of the reconstructed densities based on 30 sets of controlled noise-multiplied data (solid black line); the plot of the average of the reconstructed densities based on 30 sets of ordinary noise-multiplied data (dotted black line). The plots show that the level of utility loss due to the controlled noise multiplication masking method is similar to that of the ordinary noise multiplication masking method.
Chapter 7

An Algorithm for Protecting Tabulated Business Data

Releasing business microdata to the public is dangerous because the records could easily be identified. The reason is that business values (such as annual profit) possessed by dominant enterprises could be very large compared with other business values, and might easily lead to identity disclosure (Chipperfield et al. 2019). Methods such as top and bottom coding can be effective at reducing risk, but often represent a significant loss of data utility. Alternatively, statistical agencies could rely on a remote server (query system) for releasing information of business data while protecting sensitive information.

The feature of a remote server is that, data users cannot view the underlying data. The data users could only communicate with the remote server by sending queries. A simple model for a remote server (O’Keefe and Chipperfield 2013; Chipperfield and O’Keefe 2014) is: (1) an analyst submits a query to the remote server; (2) the remote server processes the query to generate a secure value; (3) the server returns the secure value to the user. For the first step, a query could be for some quantity such as a sample mean. For the second step, even though the user is prevented from seeing the raw data, it is still usual to perturb statistical outputs to protect against e.g. differencing attacks (see Section 7.2.1).

When a remote server is used, the statistical agency has the option of whether to store perturbed data or unperturbed data in the remote server. Storing perturbed data is known as input perturbation (Evans et al. 1998). That is, each underlying data value is perturbed by a random noise (either through noise addition or through noise multiplication). Each operation is performed on the perturbed data when responding to a query. For instance, suppose the original data is \((y_1, y_2, \ldots, y_n)\). Instead of storing the original data, a set of perturbed data \((y_1 + \varepsilon_1, y_2 + \varepsilon_2, \ldots, y_n + \varepsilon_n)\) is stored, where \(\{\varepsilon_i\}_{i=1}^{n}\) are random noise terms. Suppose a query of aggregate total is received, the query system will return the perturbed total \(\sum_{i=1}^{n} (y_i + \varepsilon_i)\) to the data user.

Alternatively, the statistical agency could store unperturbed data in the remote system. When responding to a query, a random noise is added to the unperturbed output to pro-
duce the perturbed output. The perturbed output is then returned to the data user. This mechanism is called output perturbation. Output perturbation is used in some statistical agencies, such as the Australian Bureau of Statistics.

The advantages of using a query system with output perturbation is described in Chipperfield et al. (2019). The following is a direct quote from the paper: "First, because the estimates, not the underlying micro-data, are modified, relationships in the micro-data are essentially retained. Second, the degree to which an estimate is modified depends upon the output itself. For example, modification of an estimate may be relatively high if a cell is dominated by a single business and relatively low if a table cell has many small businesses of roughly equal size. Third, because an analyst is restricted from viewing the micro-data, less modification is needed than would otherwise be the case. Fourth, it allows users to gain rapid access to estimates they request. Fifth, the modification algorithm assures a specified level of protection is guaranteed".

In this chapter, we consider the case where survey estimate of population totals of business data are released through a remote system using output perturbation. A random noise is generated through an automatic algorithm, and is added to the true aggregate total to produce a perturbed total. The perturbed output is released in a table cell. For instance, the Australian Bureau of Statistics has developed a remote system called “TableBuilder”. Data users could construct their own tables by sending multiple queries to TableBuilder. Each table cell contains a perturbed output as specified by a data user. For business data, using output perturbation is not easy to implement effectively because business data is normally heavily skewed and dominated by some extremely large observations.

Using output perturbation to protect each individual contributor is studied under the context of $\epsilon$-differential privacy (Dwork et al. 2006). $\epsilon$-differential privacy is a strong requirement that all contributor values are protected regardless of an adversary’s attacking strategy. This can be achieved by adding Laplace noise to the statistical outputs, but it could lead to huge data utility loss (Sarathy and Muralidhar 2011). In this chapter, we note that in a remote system, the most significant attacking strategy is a differencing attack. We consider the perturbation algorithm proposed in Thompson et al. (2013) as our benchmark algorithm. The algorithm is specially designed for protecting business-typed data. We introduce the algorithm, point out the drawbacks associated with the algorithm. Then we introduce a new algorithm for perturbing business totals. The new algorithm allows a better control over cell utility losses, achieves better utility-disclosure tradeoffs for many cells, and is conjectured to be able to produce a wider range of perturbed cells with acceptable levels of utility loss and disclosure risk.

This chapter is organized as follows: Section 7.1 describes three attacking scenarios. Section 7.2 introduces measures of disclosure risk and utility loss. Section 7.3 describes Thompson et al.’s algorithm. Section 7.4 describes the new algorithm. Section 7.5 discusses advantages of the new algorithm compared with the algorithm in Thompson et al.
through simulations. Section 7.6 concludes the chapter.

### 7.1 Attacking scenarios

Chipperfield et al. (2019) described three attacking scenarios that could happen to business data. Suppose in each scenario, The attacker targets a business value. The attacker knows that the sampling weight of the target is equal to one, i.e. the target is in the sample. Moreover, the target is the largest contributor to the cell. In Scenario 1 and 2, the attacker does not have prior knowledge of any contributor values. In Scenario 3, the attacker is the second largest contributor to the cell. The three scenarios are described below:

Attack Scenario 1: The table cell value is used as an estimate of the largest contributor value.

Attack Scenario 2: The attacker uses the difference between two cell estimates as an estimate of the largest contributor value. The first table cell consists of all business values and the second table cell contains the same set of contributor values as the first table cell except that the largest contributor value is excluded.

Attack Scenario 3: This is the same as Attack Scenario 1 except that the attacker is also the second largest contributor to the cell. The attacker can use the cell total minus its own contribution to estimate the largest contributor value.

Scenario 2 is an example of a differencing attack. Differencing attack can be particularly effective when used via a remote server since, at least in the case of TableBuilder, the attacker is relatively free to request tables of their choice. More detailed discussions on disclosure by differencing can be obtained in Brown (2004) and a list of defences against differencing attacks via remote servers can be found in O’Keefe and Chipperfield (2013).

In order for an attack to succeed the attacker needs the following:

1. To know that the target is in the sample. A large business might have a higher probability to be included in a sample than smaller businesses when stratified sampling is used. For smaller businesses, sampling may provide some protection since an attacker will not know if a particular business is selected in the sample. Since the underlying micro-data are not observed, it would be necessary to submit a series of queries in order to confirm whether or not a small business is actually in the sample (see Chipperfield and
2. In the case of Attack Scenario 2, to know how to uniquely identify the target in terms of a set of quasi-identifiers. This allows the attacker to “drop” the target business from the cell in a table. To conduct Attack Scenario 1 and 3, the business does not have to be uniquely identified, often referred to as identification, only that the target business dominates the cell.

3. To circumvent TableBuilder’s confidentiality protections and disclose an attribute of the business.

TableBuilder gives users a high degree of flexibility in choosing a table’s dimensions and scope. There is often considerable information about large businesses in the public domain which may in turn make identification likely (e.g. there may only be one private hospital in a small area). Accordingly, we conservatively assume that 1. and 2. occur with certainty. So for large businesses at least, the only protection available in TableBuilder is perturbation. Consequently, perturbation is the focus of how disclosure risk is measured. We also measure disclosure risk in terms of a differencing attack only because it is easy to perform and is very effective.

### 7.2 Measuring Disclosure Risk and Utility Loss

#### 7.2.1 Differencing Attack

Consider the following scenario: suppose a continuous valued characteristic of the $i$th sample unit is $y_i$ and there are $n$ sample units. An estimation weight is the reciprocal of the probability that the unit is included in the sample when probability sampling is used. The estimation weight of $y_i$ is $w_i$. Define $y = (y_1, y_2, \cdots, y_n)$ and $y_1w_1 \geq y_2w_2 \geq \cdots \geq y_nw_n > 0$ with $w_1 = 1$. The attacker estimates $y_1$ by taking the difference of two perturbed estimates. That is, suppose we have:

Query 1: $\hat{S}^* = \sum_{i=1}^{n} y_iw_i + R^*$, where $R^*$ is a random perturbation variable for perturbing Query 1.

Query 2: $\hat{S}^*_{-1} = \sum_{i=2}^{n} y_iw_i + R^*_{-1}$, where $R^*_{-1}$ is a random perturbation variable for perturbing Query 2.

Under a differencing attack, the attacker’s estimate of $y_1$ is
\[ \hat{Y}_1 = \hat{S}^* - \hat{S}_{-1}^* = y_1 + R^* - R_{-1}^*. \]

In this chapter, we assume a differencing attack is the only source of disclosure risk. Throughout the paper, without loss of generality, we assume the attacker’s target is the largest contributor value \( y_1 w_1 \), and the attacker also knows the weight of \( y_1 \) is equal to one \( (w_1 = 1) \). The reasons for these assumptions are that, normally speaking, the largest contributor value \( y_1 w_1 \) has the highest disclosure risk against a differencing attack than any other contributor value, and for the largest business, its estimation weight is 1 because it might be unique in the population. Therefore, if we could find ways to effectively protect the largest contributor, all other contributors would be protected as well.

To define disclosure risk, we conservatively assume that: 1. the target is in the sample; and 2. the attacker could uniquely identify the target through a set of quasi-identifiers. So the only protection available in a remote system is perturbation. Consequently, perturbation is the focus of how disclosure risk is measured.

### 7.2.2 Defining Disclosure

**Disclosure Risk:** Following the definition of value disclosure risk in Chapter 2, we say that the disclosure risk against a differencing attack is the probability that a realization of \( \hat{Y}_1 \) is within 100\( \alpha \)% of the true value \( y_1 \). If we define disclosure risk of attacking target value \( y_1 \) as \( D(y_1, \alpha) \), then

\[ D(y_1, \alpha) = P(|R^* - R_{-1}^*| < \alpha y_1) \]

We say that \( \alpha \) is the the acceptance rule and \( DL \) is the acceptable level of disclosure risk. Different values of \((DL, \alpha)\) could be adopted on the basis of whether the attack is likely to occur. We say that perturbed cell estimates have an acceptable disclosure risk if \( D(y_1, \alpha) \) is less than \( DL \).

### 7.2.3 Defining Utility Loss

We define the utility loss of perturbing a cell as the relative distance between the perturbed cell value and unperturbed cell value. Measuring utility loss by percentage difference between the perturbed estimate and the unperturbed estimate has been widely used in many applications. It is formally introduced by Domingo-Ferrer and Torra (2001) and widely used by other authors in their studies (see Kim and Winkler 2003; Yancey et al. 2002).

For each table cell, it is perturbed by a random noise variable. The distribution of the noise variable is determined by a perturbation algorithm. As an algorithm produces a perturbed cell value randomly, in order to assess and compare the performances of
CHAPTER 7. AN ALGORITHM FOR PROTECTING TABULATED BUSINESS DATA

different perturbation algorithms, we look at the **average utility loss**. The average utility loss is the expected utility loss of perturbing the cell using the algorithm. In other words, for the Cell 1 described in Section 7.2.1, the average utility loss is \( E(|R|)/\sum_{i=1}^{n} y_iw_i \). It is preferable that the average utility loss to be as low as possible given that \( D(y_1) < DL \).

### 7.3 The algorithm in Thompson et al. (2013)

In this section we introduce an output perturbation algorithm for protecting aggregate totals of business values. The application of this perturbation algorithm is investigated extensively in Chipperfield et al. (2019).

As before, consider any particular cell in a table and let there be \( n \) sample units \((y_1w_1, \cdots, y_nw_n)\) contributing to the cell. The survey estimate of the total is \( \hat{s} = \sum_{i=1}^{n} y_iw_i \). We assume \( y_1w_1 \geq y_2w_2 \geq \cdots \geq y_nw_n \).

The algorithm basically uses a noise random variable \( P^* \) to perturb each cell. The expression of \( P^* \) is \( P^* = \sum_{i=1}^{K} (m_iD_i^*H_i^*)y_iw_i \), where \( K \) is the number of top contributors in the cell that are used in calculation of \( P^* \); \( m = (m_1, \cdots, m_K) \) is vector of non-negative real numbers; \( D_i^* \) is a random variable taking the value -1 and 1 with equal probability; \( H_i^* \) is a random variable with mean 1 and for the purpose of this chapter we set \( H_i \) to have a symmetric triangular probability density function centered at 1 with width 0.6. The \( m \) vector is used to balance utility loss and disclosure risk for each table cell.

The algorithm in Thompson et al. (2013) generates a perturbed cell value in the following way:

1. The algorithm identifies the parameters \((K, m)\) to be used.

2. The algorithm generates a perturbation amount \( p^* \) from a random variable \( P^* \), and adds \( p^* \) to the total \( \hat{s} \) to generate a perturbed cell value \( \hat{s}^* \).

The distinct features of using this perturbation random variable is that:

1. The perturbation amount is relatively high if a cell is dominated by a few large businesses.

2. It does not add bias to the final table. e.g. \( E(P^*) = 0 \).

For the first point, we note that the perturbation amount is proportional to the magnitude of the top few contributor values. It could guarantee that the top few contributor values are protected against the differencing attack. Mathematically, recall that when \( w_1 = 1 \),
CHAPTER 7. AN ALGORITHM FOR PROTECTING TABULATED BUSINESS DATA

attacker’s estimate of $y_1$ is

$$\hat{y}_1 = y_1 + P^* - P^*_{-1}$$

where $P^*_{-1} = \sum_{i=2}^{K+1} (m_i D_i^* H_i^*) y_i w_i$. Therefore, $y_1$ is protected by

$$P^* - P^*_{-1} = (m_1 D_1^* H_1^*) y_1 + \sum_{i=2}^{K} (m_i D_i^* H_i^*) y_i w_i - \sum_{i=2}^{K+1} (m_i D_i^* H_i^*) y_i w_i.$$

The magnitude of $P^* - P^*_{-1}$ is directly determined by the magnitude of $y_1$, which means that the perturbation amount goes higher for larger $y_1$. The optimal set of parameters to be used for each cell depends on the distribution of contributing values to the cell estimate. The optimal set of parameters guarantees that the perturbed estimates have the lowest average utility loss subject to having an acceptable level of disclosure risk $DL$. Examples of the optimal choices of magnitude vector when $K = 3$ for different contributor values are given in Table 7.1. Chipperfield et al. (2019) showed that these parameters result in the lowest utility loss subject to having at most 15% disclosure risk, assuming $\alpha = 0.11$.

For a given cell of a table, the algorithm achieves its best performance if the optimal set of parameters for perturbing the cell is used. As developing a program of searching the optimal set of parameters to be used to perturb each cell value is non-trivial, Chipperfield et al.(2019) investigated the outcomes of using one set of parameters to perturb all cells. The set of parameters was selected upon satisfying the requirement of disclosure risks for a few benchmark cells, and the study examined its impact on utility losses of different cells through empirical studies. However, fixing the set of parameters certainly limits the efficacy of the algorithm as the choice of parameters is not always the optimal one for perturbing many cells (see Chipperfield et al. (2019)). In the following, we assume that the same $m$ vector is used for all cells.

There are issues with this configuration. The issues are:

1. The algorithm could produce a very perturbed table cell because the upper bound of the support of $P^*$ could be very large. It might happen that the magnitude of perturbation produced by the algorithm is extremely large.

2. The protection to $y_1$, i.e. $P^* - P^*_{-1}$ could cancel each other out (i.e. taking values close to 0) with high probability if the top $K + 1$ contributor values are uniformly distributed.

To illustrate the first issue, suppose the contributor values $(y_1 w_1, y_2 w_2, \cdots, y_8 w_8) = (25, 25, \cdots, 25)$. If the parameter $m = (0.5, 0.4, 0.3)$, then the expected utility loss of perturbing this cell using the algorithm is $E(P^*)/200 = 7.54\%$. This value is generally acceptable for perturbing a table cell. However, the actual utility loss of perturbing this
cell using the algorithm is bounded between \((-24\%, 24\%\)). That means, it is possible for the perturbation algorithm to produce perturbed response with utility loss 24%. Should it happen, information reflected by the perturbed value could be misleading to some data users. The distribution of a random perturbed value is given in Figure 7.1.

For the second issue, we note that the disclosure risk of \(y_1\) is given below:

\[
D(y_1, \alpha) = P(|P^* - P^*_{-1}| < \alpha y_1)
\]

Without loss of generality we consider a special case where the contributor values are \((y_1 w_1, y_2 w_2, \cdots, y_8 w_8) = (25, 25, \cdots, 25)\). We set \(m = (0.5, 0, 0)\). Assuming \(w_1 = 1\). Using a differencing attack to attack \(y_1\), denote the estimator as \(\hat{Y}_1\). Then we have

\[
\hat{Y}_1 = y_1 + P^* - P^*_{-1}
\]

For this example, \(P^*\) is distributed as \(12.5D_1 H_1\) and \(P^*_{-1}\) is distributed as \(12.5D_2 H_2\). The modes for the two distributions are both at \(\pm 12.5\). Problem arises if \(p^*\) is close to \(p^*_{-1}\) because \(y_1\) is protected by \(P^* - P^*_{-1}\). As \(P^*\) and \(P^*_{-1}\) have the same modes, it is likely to happen that the values for both \(p^*\) and \(p^*_{-1}\) are around one of the modes, providing little protection to \(y_1\).

Generally speaking, when all contributors are identical, we have that \(P^* \overset{d}{=} P^*_{-1}\). The distribution of \(P^* - P^*_{-1}\) is likely to peak at 0, meaning that there is a high chance that \(y_1\) is not getting sufficient protection. As an example, disclosure risk plot for \(y_1\) when contributor values are \(\{25\}_{i=1}^{8}\) and \(m = (0.5, 0.4, 0.3)\) is given in Figure 7.2. The corresponding disclosure risk of \(y_1\) is given in the area shaded red.

We note that in general, cells with identical contributor values are not considered as sensitive as those cells which contain dominant contributors. However, for a remote system, it might occur that output perturbation is used for all table cells. Therefore, those cells with identical contributor values need to be discussed in this chapter.

From the data intruder’s point of view, if the data intruder knows the perturbation algorithm and knows that the top \(K\) contributor values were equal, then the data intruder might be aware that using a differencing attack for \(y_1\) might be effective based on the above argument. Knowing that the top \(K\) contributor values are equal might motivate the data intruder to use a differencing attack.
Table 7.1: Magnitude values that guarantee at most 15% disclosure risk given $\alpha = 0.11$ and minimise the average utility loss for different distributions of top contributor values.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Relative Size of Top Contributors</th>
<th>Optimal Magnitude Values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st</td>
<td>2nd</td>
</tr>
<tr>
<td>1</td>
<td>90</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>70</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>60</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>40</td>
</tr>
<tr>
<td>6</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td>8</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>25</td>
<td>25</td>
</tr>
</tbody>
</table>

Figure 7.1: Distribution of $\sum_{i=1}^{8} 25 + P^*$ given $m = (0.5, 0.4, 0.3)$ for the table cell with contributor values $(25, 25, \cdots, 25)$. 
7.4 A New Algorithm to Generate Perturbed Cell Estimates

We introduce a new algorithm to generate perturbed cell estimates. Compared with Thompson et al’s algorithm, the advantage of the new algorithm is that:

1. The utility loss for perturbing each table cell is bounded according to the requirement of the data provider.

2. If $y_1$ is not strongly dominating a table cell, then the disclosure risk of $y_1$ is lower.

Suppose an ordinary cell has contributor values $(y_1w_1, y_2w_2, \cdots, y_nw_n)$, $w_1 = 1$. The statistical agency specifies a threshold, $\beta$, on the average utility loss of a table cell. $\hat{s}$ is the unperturbed cell value, i.e. $\hat{s} = \sum_{i=1}^n y_iw_i$. The new algorithm perturbs the cell value as follows:

1. The statistical agency sets the value of $\beta$, e.g. 0.05.

2. Define $\lambda = \hat{s}\beta$, where $\hat{s}$ is the cell value. If $n$ is an even number, a noise variable $R_1^*$ is used. $R_1^* \sim 0.5U(-0.5\lambda, 0.5\lambda) + 0.25U(1.5\lambda, 2\lambda) + 0.25U(-2\lambda, -1.5\lambda)$. If $n$ is an odd number, a noise variable $R_2^*$ is used. $R_2^* \sim 0.5U(0.5\lambda, 1.5\lambda) + 0.5U(-1.5\lambda, -0.5\lambda)$.
The value of $\beta$ is set according to the requirements of the statistical agency. The maximum data utility loss of the table cell is $2\beta$. This is because the maximum amount of perturbation (in magnitude) produced by the algorithm is $2\lambda$. The corresponding maximum utility loss is $2\lambda/\hat{s} = 2\beta$.

The disclosure risk of $y_1$ could also be mathematically determined. That is, the disclosure risk of $y_1$ is given by:

$$D(y_1, \alpha) = P(|R_1^* - R_2^*| < \alpha y_1)$$

$$= \int_{-\alpha y_1}^{\alpha y_1} \int_{-1.5\lambda}^{1.5\lambda} f_{R_1^*}(x+t)f_{R_2^*}(t)dt\,dx$$

The density functions of $R_1^*$ and $R_2^*$ are given in the following:

$$f_{R_1^*}(r) = \begin{cases} \frac{1}{2\lambda} & \text{if } |r| \in (0, 0.5\lambda) \text{ or } |r| \in (1.5\lambda, 2\lambda) \\ 0 & \text{otherwise} \end{cases}$$

$$f_{R_2^*}(r) = \begin{cases} \frac{1}{2\lambda} & \text{if } |r| \in (0.5\lambda, 1.5\lambda) \\ 0 & \text{otherwise} \end{cases}$$

The disclosure risk of $y_1$ is a function of $(y_1, \alpha, \beta, \lambda, \lambda_1)$, where $\lambda = \beta \sum_{i=1}^{n} y_iw_i$, $\lambda_1 = \beta \sum_{i=2}^{n} y_iw_i$. The mathematical expressions of the disclosure risk of $y_1$ is conditional on the value of $y_1$, and the relationship between $\alpha$ and $\beta$. The mathematical expressions are given in Tables 7.2 and 7.3.

To interpret the tables, when $\beta \geq 2\alpha$, and $n$ is an odd number, and if the value of $y_1$ satisfy that

$$\frac{\lambda}{4\beta - 2\alpha} < y_1 < \min\left(\frac{\lambda}{1.5\beta + \alpha}, \frac{\lambda}{2\alpha}\right)$$

Then based on Table 7.3, the disclosure risk of $y_1$ is $P_{C21}$, where the expression of $P_{C21}$ is given in Table 7.2. That is, $P_{C21} = \frac{\alpha y_1}{2\lambda}$.

The idea of splitting up odd and even cases is addressing the differencing attack by guaranteeing that the counts of contributors for the two queries (for the total and total minus the target value) will go from odd to even or even to odd. It makes it much harder for the perturbation under the set of $n$ contributors and the set of $n-1$ contributors to cancel out if the largest contributor is not strongly dominating the cell. In this case the $K-1$ noise components that could be equal under Thompson et al.’s approach will no longer be equal. To see this, suppose we have a set of 8 contributors $(y_1, y_2w_2, \cdots, y_8w_8) = (25, 25, \cdots, 25)$. Assuming $\beta = 0.05$, and suppose the data attacker use a differencing attack to attack $y_1$ by sending two queries. The response the
CHAPTER 7. AN ALGORITHM FOR PROTECTING TABULATED BUSINESS DATA

The algorithm in Thompson et al. (2013).

By differencing the two outputs, the attacker’s estimate of \( y_1 \) is

\[
\hat{y}_1 = y_1 + R_1^* - R_2^*
\]

The distributions of \( R_1^* \) and \( R_2^* \) are given in Figure 7.3 and 7.4. The distribution of \( R_1^* - R_2^* \) and the disclosure risk of \( y_1 \) are given in Figure 7.5. Compared with Figure 7.2, we see that the disclosure risk of \( y_1 \) is smaller because the distribution of \( R_1^* - R_2^* \) does not peak at 0.

In the next section, we discuss the advantages of using the new algorithm compared to the algorithm in Thompson et al. (2013).

---

**Table 7.2:** The following terms are to be used for Table 7.3. \( \lambda_1 = \delta_1 \beta \).

<table>
<thead>
<tr>
<th>( P_{c_{11}} )</th>
<th>( \frac{1}{22\lambda_1} \alpha y_1^2 \beta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_{c_{12}} )</td>
<td>( \frac{1}{22\lambda_1} (0.125 \lambda^2 + 0.5 \lambda \alpha y_1 + 0.5 \alpha y_1^2 - 0.25 \lambda \lambda_1 + 0.125 \lambda^2 - 0.5 \alpha y_1 \lambda_1) )</td>
</tr>
<tr>
<td>( P_{c_{13}} )</td>
<td>( \frac{1}{22\lambda_1} (1.125 \lambda^2 - 1.5 \lambda y_1 + 0.5 \alpha y_1^2 - 2.25 \lambda \lambda_1 + 1.125 \lambda^2 - 1.5 \lambda y_1 \lambda) )</td>
</tr>
<tr>
<td>( P_{c_{21}} )</td>
<td>( \frac{1}{22\lambda_1} (0.75 \lambda \lambda_1 + 0.5 \alpha y_1 \lambda_1 - 0.875 \lambda^2) )</td>
</tr>
<tr>
<td>( P_{c_{22}} )</td>
<td>( \frac{1}{22\lambda_1} (-0.5 \alpha y_1^2 + \lambda_1 y_1 \alpha + 3 \lambda \lambda_1 - 2 \lambda_1^2 - 1.125 \lambda^2 + 1.5 \lambda \alpha y_1) )</td>
</tr>
<tr>
<td>( P_{c_{23}} )</td>
<td>( \frac{3 \alpha y_1 \beta}{22\lambda_1} )</td>
</tr>
<tr>
<td>( P_{c_{24}} )</td>
<td>( \frac{1}{22\lambda_1} (0.125 \lambda^2 \alpha y_1^2 + 0.5 \alpha y_1 \lambda_1 - 0.25 \lambda \lambda_1 + 0.125 \lambda^2 - 0.5 \alpha y_1 \lambda) )</td>
</tr>
<tr>
<td>( P_{c_{25}} )</td>
<td>( \frac{1}{22\lambda_1} (1.125 \lambda^2 - 0.5 \alpha y_1 \lambda_1 + 1.5 \alpha y_1 \lambda - 2.25 \lambda \lambda_1 + 1.125 \lambda^2 - 1.5 \lambda \alpha y_1 \lambda) )</td>
</tr>
</tbody>
</table>

**Table 7.3:** Disclosure risks of the largest contributor value \( y_1 \) if table cells are perturbed by the new algorithm, given different conditions of the largest contributor value \( y_1 \) and different relationships between \( \alpha \) and \( \beta \).

<table>
<thead>
<tr>
<th>( n ) is odd, ( \frac{\lambda}{4^3 + 2^\alpha} &lt; y_1 &lt; \min(\frac{\lambda}{1.5^\beta + \alpha}, \frac{\lambda}{2^\alpha}) )</th>
<th>( \beta \geq 2\alpha )</th>
<th>( \frac{\alpha}{1.5^\beta} &lt; \beta &lt; 2\alpha )</th>
<th>( \beta \leq \frac{\alpha}{1.5^\beta} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_{c_{21}} )</td>
<td>( P_{c_{24}} + P_{c_{21}} )</td>
<td>not possible</td>
<td></td>
</tr>
<tr>
<td>( n ) is odd, ( \frac{\lambda}{4^3 + 2^\alpha} &lt; y_1 &lt; \frac{\lambda}{4^3 - 2^\alpha} )</td>
<td>( P_{c_{22}} )</td>
<td>( P_{c_{24}} + P_{c_{22}} )</td>
<td>not possible</td>
</tr>
<tr>
<td>( n ) is odd, ( \frac{\lambda}{4^3 + 2^\alpha} &lt; y_1 &lt; \frac{\lambda}{4^3 - 2^\alpha} )</td>
<td>( P_{c_{22}} )</td>
<td>( P_{c_{24}} + P_{c_{22}} )</td>
<td>( P_{c_{24}} + P_{c_{25}} )</td>
</tr>
<tr>
<td>( n ) is odd, ( y_1 &lt; \frac{\lambda}{4^3 + 2^\alpha} )</td>
<td>( P_{c_{23}} )</td>
<td>( P_{c_{24}} + P_{c_{23}} )</td>
<td>( P_{c_{24}} + P_{c_{26}} )</td>
</tr>
<tr>
<td>( n ) is even, ( y_1 &lt; \frac{\lambda}{4^3 + 2^\alpha} )</td>
<td>( P_{c_{11}} )</td>
<td>( P_{c_{12}} )</td>
<td>( P_{c_{12}} + P_{c_{13}} )</td>
</tr>
</tbody>
</table>

the two queries are:

First query: \( \sum_{i=1}^{8} 25 + R^*_1 \); Second query: \( \sum_{i=2}^{8} 25 + R^*_2 \).

By differencing the two outputs, the attacker’s estimate of \( y_1 \) is

\[
\hat{y}_1 = y_1 + R_1^* - R_2^*
\]
Figure 7.3: Distribution of $R_1^*$. 

Figure 7.4: Distribution of $R_2^*$. 
Figure 7.5: Distribution of $R^*_1 - R^*_2$ and the corresponding disclosure risk of $y_1$ (red area), assuming $\alpha = 0.11$. Compared with Figure 7.2, we see that the disclosure risk of $y_1$ is smaller because the distribution of $R^*_1 - R^*_2$ does not peak at 0.

7.5 The new algorithm v.s. Thompson et al. algorithm

7.5.1 Controlled Utility Loss of Perturbing a Cell

For a given cell with cell value $\hat{s}$, it can be easily seen that the perturbation amount $p^*$ generated by the new algorithm is bounded in $(-2\beta\hat{s}, 2\beta\hat{s})$. As a result, the utility loss of perturbing the cell through the new algorithm is bounded in $(0, 2\beta)$. As a result, both the average utility loss and the maximum utility loss of perturbing the cell could be controlled.

To illustrate this advantage, suppose the contributor values to a cell are $(y_1w_1, \cdots y_8w_8) = (25, 25, 25, 25, 25, 25, 25, 25)$. From Table 7.1, the optimal magnitude values are $(m_1, m_2, m_3) = (0.5, 0.4, 0.3)$. If the cell is perturbed by the algorithm in Thompson et al. with the optimal magnitude values, the average utility loss is 7.54%. However, as we have showed above, the maximum possible utility loss is 24%. A perturbed result with such a high utility loss may mislead some data users. In contrast, perturbing the cell by the new algorithm with $\beta = 0.0754$ also gives an average utility loss of 7.54%. The maximum utility loss is $2\beta = 15.08\%$, which is lower than the counterpart produced by Thompson et al (2013)’s algorithm.

7.5.2 A Conjectured Wider Applicability

It is conjectured that the new algorithm could legitimately perturb a wider range of cells. A legitimately perturbed cell value should satisfy the requirements of both disclosure risk and utility loss. Recall that we assume the attacker’s target is the largest contributor value.
y_1$, and $w_1 = 1$. In the following, we say that a cell could be legitimately perturbed by an algorithm if the average utility loss is less than $T$, and the disclosure risk of $y_1$ against differencing attack is less than $DL$ given a specified acceptance rule $\alpha$. We illustrate this conjectured wider applicability by comparing the performances of the two algorithms on different cells.

Suppose the statistical agency set $(T, DL, \alpha)$ to be $(10\%, 15\%, 11\%)$ for a perturbed cell estimate to be legitimate. Recall that in a recent study in Chipperfield et al. (2019), we used one set of parameters for the algorithm in Thompson et al. (2013) for perturbing all cells as it is more practical. We repeat this here and set the parameters to be $K = 3$, $m = (0.4, 0.3, 0.2)$. We set $\beta$ equal to 0.1. Recall that $\beta = 0.1$ means that the expected utility loss to a cell is $10\%$.

**Cell 1:** The contributor values of cell 1 are $(y_1w_1, y_2w_2, \cdots, y_6w_6) = (30, 30, 30, 10, 5, 5)$. We measure the average utility losses and disclosure risks resulting from the two perturbation algorithms, respectively. We applied Thompson et al.’s algorithm and the new algorithm to the data 5000 times. On each iteration, and for each approach, we generated estimates of $y_1$ as the difference in perturbed totals. To estimate disclosure risk for each algorithm, we record the number of times that an estimate of $y_1$ discloses $y_1$, i.e. the number of times that the estimate is within $((1 - \alpha)y_1, (1 + \alpha)y_1)$. Let the number be $k$. Then the estimated disclosure risk is $k/5000$. For Thompson et al.’s algorithm, we recorded the utility loss to the cell on each iteration, and we computed the average utility loss across all 5000 samples of utility losses. The average utility loss is our estimate of the expected utility loss.

Using the algorithm in Thompson et al., the average utility loss and disclosure risk of releasing perturbed cell estimates are 12.4% and 9.4%, respectively. It means that, even though the perturbed cell estimates satisfy a required level of disclosure risk, they do not carry enough data utility as required by the statistical agency. Using the new algorithm, the average utility loss and disclosure risk of releasing perturbed cell estimates are 10% and 6.5%. This means that, both the requirements of utility loss and disclosure risk are satisfied, and it is legitimate to release a perturbed cell value generated by the new algorithm.

**Cell 2:** The contributor values of cell 2 consist of $(y_1w_1, y_2w_2, \cdots, y_7w_7) = (25, 25, 25, 25, 1, 1, 1)$. Using the algorithm in Thompson et al., the average utility loss and disclosure risk of releasing perturbed cell estimates are 10.9% and 12.0%, respectively. That means perturbed cell estimates do not carry enough data utility as required by the statistical agency. Using the new algorithm, the average utility loss and disclosure risk of releasing perturbed cell estimates are 10% and 11.4%. Both the requirements of utility loss and disclosure risk are satisfied, and it is legitimate to release a perturbed cell
value generated by the new algorithm.

The above two cells are used to show that the new algorithm could help to generate legitimate cell estimates that are not achievable by the algorithm in Thompson et al. (2013). However, we next show that it is possible that, when a cell contains a dominant contributor value, the algorithm in Thompson et al. (2013) is better. For illustration, we set \((T, DL, \alpha)\) to be \((15\%, 12\%, 11\%)\), and \(\beta = 0.15\) for the next cell.

**Cell 3:** The contributor values of cell 3 consist of \((y_1w_1, y_2w_2, \ldots, y_9w_9) = (60, 20, 15, 15, 10, 10, 10, 10)\). The average utility loss and disclosure risk of releasing perturbed cell estimates generated by the algorithm in Thompson et al. are 14.1\% and 9.5\%, while the counterparts generated by the new algorithm are 15\% and 13.1\%. In this case the algorithm in Thompson et al. (2013) is the better algorithm for perturbing the cell.

The reason the new algorithm is not favourable for **Cell 3** is that, when the ratio \(y_1w_1/\hat{s}\) gets large, the disclosure risk of using the new algorithm goes up dramatically. To see this, without loss of generality, we assume \(n\) is even, \(\beta \geq 2\alpha\), \(y_1w_1 < \min(\lambda\beta + \alpha, \lambda\alpha)\). From Table 7.3, the disclosure risk is \(P_{C11} = \frac{1}{2\lambda\alpha}\alpha y_1w_1^2\beta\). It is evident that the ratio \(y_1w_1/\hat{s}\) has a large impact on the value of disclosure risk. Possible future research would be to use either the algorithm in Thompson et al. or the new algorithm to perturb a cell estimate subject to a condition involving the value of \(y_1w_1/\hat{s}\).

### 7.5.3 Better Utility-Disclosure Trade-offs

We compare the utility-disclosure tradeoffs of the two algorithms on cells with different contributor values through simulations. In order to obtain utility-disclosure plots, we gradually changed the values in the magnitude vector \(m\) used by the algorithm in Thompson et al. (2013) and the parameter \(\beta\) used by the new algorithm. We recorded the average utility losses and disclosure risks given different parameter values. Moreover, we provide utility-disclosure plots for \(\alpha = 0.11\) and \(\alpha = 0.18\), respectively.

**Simulation 1:** The contributor values of a cell are \((y_1w_1, y_2w_2, \ldots, y_8w_8) = (25, 25, 25, 25, 25, 25, 25)\). We set the magnitude vector to be \(m = (0.3 + 0.01i, 0.2 + 0.01i, 0.1 + 0.01i), \) where \(i = 1, 2, \ldots, 40\). We recorded the utility loss and disclosure risk of releasing perturbed cell estimates generated by the algorithm in Thompson et al. (2013) for each value of \(i\) for generating the utility-disclosure plot. When \(i = 20\), \(m = (0.5, 0.4, 0.3), \) which is the optimal magnitude vector as shown in Table 7.1. Similarly we set \(\beta = 0.04 + i/400, i = 1, 2, \ldots, 40; \) and we obtained the utility-disclosure plot for the new algorithm. We use boxes to represent the utility-disclosure plot of the algorithm in Thompson et al. (2013) and dots to represent the utility-disclosure
plot of the new algorithm and these symbols also apply to Figures 7.7 and 7.8 discussed in Simulations 2 and 3. The utility-disclosure plots for $\alpha = 0.11$ and 0.18 are provided in Fig. 7.6.

Simulation 2: The contributor values of a cell are $(y_1w_1, y_2w_2, \ldots, y_9w_9) = (40, 20, 15, 15, 10, 10, 10)$. We set the magnitude vector to be $m = (0.15 + 0.01i, 0.1 + 0.01i, 0.05 + 0.01i)$, where $i = 1, 2, \ldots, 40$. The parameter of new algorithm is set to be $\beta = 0.03 + i/300, i = 1, 2, \ldots, 40$. We follow the same procedure as in Simulation 1 to obtain the utility-disclosure plots of the two algorithm for $\alpha = 0.11$ and 0.18. The plots are given in Fig. 7.7.

Simulation 3: The contributor values are $(y_1w_1, y_2w_2, \ldots, y_9w_9) = (60, 20, 10, 10, \ldots, 10)$. We set the magnitude vector to be $m = (0.05 + 0.05i, 0.05i, 0.05i)$, where $i = 1, 2, \ldots, 40$. The parameter of the new algorithm is set to be $\beta = 0.01 + i/150, i = 1, 2, \ldots, 40$. The utility-disclosure plots for $\alpha = 0.11$ and 0.18 are given in Fig. 7.8.

From Fig. 7.6, we see that the new algorithm leads to a better utility-disclosure trade-off when the contributor values to a cell estimate are uniformly distributed. From Fig. 7.7, we see that this advantage is reduced when the largest contributor value dominates the cell estimate. From Fig. 7.8, we see that the new algorithm again offers a better utility-disclosure trade-off even though the largest contributor value is significantly larger than all other contributor values, as in this case where the largest contributor value does not dominate the cell total.

7.6 Conclusion

In this chapter we introduced a new algorithm to generate perturbed survey estimates of population totals via a remote system. The perturbed totals are released in table cells. The advantages of the new algorithm are discussed compared with the algorithm in Thompson et al. (2013). It is conjectured that the new algorithm could be widely used in many remote systems for creating aggregate totals from business microdata. Possible future research would be on combining the new algorithm with the algorithm in Thompson et al. to perturb business totals such that the more efficient utility-risk tradeoff could be achieved.
Figure 7.6: Utility-disclosure plots for Simulation 1 with different $\alpha$ values. The boxed plot represents results generated by the Thompson et al. algorithm and the dotted plot represents results generated by the new algorithm.
(a) Utility-Disclosure plots for Simulation 2 with $\alpha = 0.11$

(b) Utility-Disclosure plots for Simulation 2 with $\alpha = 0.18$

**Figure 7.7:** Utility-disclosure plots for Simulation 2 with different $\alpha$ values.
 CHAPTER 7. AN ALGORITHM FOR PROTECTING TABULATED BUSINESS DATA

(a) Utility-Disclosure plots for Simulation 3 with $\alpha = 0.11$

(b) Utility-Disclosure plots for Simulation 3 with $\alpha = 0.18$

Figure 7.8: Utility-disclosure plots for Simulation 3 with different $\alpha$ values.
Chapter 8

Conditional linkage error model

For statistical agencies, they might have collected multiple datasets which contain information on the same individuals. For instance, for the same individuals, three datasets might have been created based on survey responses. The first dataset might contain information of age, gender and personal income of the individuals. The second dataset might contain information of gender, education status, and daily expenditures of the individuals. The third dataset might contain information of age, gender, marital status and ethnic group of the individuals. When recording these information, the identity of the data respondents are generally removed from the dataset for data privacy concerns. However, it might be interesting to know the relationship between two attributes which are not in the same dataset, for instance personal income and ethnic group.

When the personal identities of the data respondents are known, the statistical agency could easily match the records from different datasets, and create a larger dataset containing all the attributes contained in the smaller datasets. The statistical agency could then investigate the relationships between attributes originally located in different datasets. However, because uniquely identifying information is generally removed from these datasets, the statistical agency cannot match records from distinct datasets with certainty. Investigating the relationships by collecting new data could be very expensive. Alternatively, the statistical agency could link the records, and then investigate the relationship between those attributes. Record linkage is relatively cheap and easy to conduct. Therefore, it is a daily routine for many statistical agencies to match records from different datasets via the process of record linkage. In this chapter, we consider the case where only two datasets are linked together through the process.

Record linkage is the act of identifying records from two files that belong to the same unit (individual or business). Ideally, record pairs that belonged to the same unit could be identified with certainty. In this way, data analyst could extract accurate statistical information between analysing variables in the linked-dataset. This is only possible when unique identifiers, such as social security number, are available and error-free. However, such information is normally not available in each dataset to be linked in order to protect
respondents confidentiality. Therefore, for a record pair, i.e. one record from each dataset, it is not known that whether it is a match (the two records in the record pair come from the same unit), or a non-match (the two records come from different units).

Fellegi and Sunter (1969) developed a record linkage model for linking records of two datasets where unique identifiers are not available. The record linkage model utilises a set of attributes called link fields. Link fields are a subset of attributes which are present in both datasets. They contain quasi-identifying information for a unit, such as age, gender, education status etc. For a record pair, Fellegi and Sunter’s (1969) record linkage model compares the values of a record pair across their link fields. To compare the similarity, an agreement pattern is created. An agreement pattern is a vector contains agreement outcomes. If the record pair agrees on the first link field, then the agreement outcome is 1. Otherwise it is 0. For instance, suppose there are three link fields. The record pair agrees on the first two link fields, but disagree on the third link field. Then the agreement pattern is \((1,1,0)\). The match weight for a record pair is the \(\log_2\) of the probability of observing the agreement pattern if the record pair is a match minus the \(\log_2\) of the probability of observing the agreement pattern if the record pair is a non-match. Suppose there are \(n\) records in each dataset. Then there are \(n^2\) record pairs to consider. For each record pair, a match weight is computed.

In practice, the statistical agency needs to decide on upper and lower thresholds on the match weight, \(U\) and \(L\). The decision rule is: If for a record pair, its match weight is above \(U\), then we say the record pair is a match. If its match weight is below \(L\), we say the record pair is a non-match. If the match weight is in between, then we say the record pair is a potential match. Generally the vast majority of record pairs are classified as non-matches.

Fellegi and Sunter (1969) showed that the optimal \(U\) and \(L\) could be determined in the sense that the set of potential matches is minimized, subject to pre-specified error levels for false links (linked non-matches) and false non-links (non-linked matches). Other methods for determining these two thresholds can be found in Tuoto (2016). All potential matches are passed to clerical review, which means statistical agents need to manually determine whether a potential match is a match by comparing more information of the record pair based on some extra information. For instance, suppose a record pair disagrees on address. A statistical agent could manually check whether the disagreement on address is due to spelling errors, and then make a decision on whether the record pair is a match. However, the clerical review process is costly, time-consuming, and is also subject to judgement error (Scheuren and Winkler 1993).

For the purposes of this chapter, we only consider one-to-one correspondence, i.e. for each record in one file there is exactly one match in another file. We assume the following process for record linkage, which could be seen as an application of matching constraint. That is, firstly, the files are linked by a probabilistic linkage algorithm, which is described
in Chipperfield and Chambers (2015). Next, the match weights of linked record pairs are compared with the threshold values $U$ and $L$ to determine matches or non-matches, same as in Fellegi-Sunter’s approach. The probabilistic linkage algorithm is a one-to-one assignment algorithm. One-to-one assignment means that, for each record in one file, it is assigned a link with a record in the other file. Records on each file could only be linked once. The probabilistic linkage algorithm links the record pairs iteratively. On each iteration the record pair with the highest match weight is linked, and the record pairs which contain one of the linked records will not be linked on subsequent iterations. A linked record pair with match weight greater than $U$ is considered as a match. A linked record pair with match weight below $L$ is considered as a non-match. A linked pair with match weight in between is considered as a potential match and is to go to clerical review.

Once the clerical review process is done, those record pairs which are considered to be matches are linked together. However, linked data almost inevitably contains errors. Analysing linked data with linkage errors can result in large levels of estimation bias. For instance, Scheuren and Winkler (1993) showed that the ordinary least squares estimator would lead to biased estimation of regression coefficients when linked data contains errors. Consequently, methods for correcting estimation bias have been studied extensively in the past (Scheuren and Winkler 1993; Lahiri and Larsen 2005; Chambers 2009; Hof and Zwinderman 2012; Chipperfield and Chambers 2015). Generally speaking, these methods allow data users to perform data analysis on probabilistically-linked data (e.g. data linked by the probabilistic linkage algorithm). Consequently, the need for clerical review is reduced. It is important because clerical review is costly and is subject to linkage errors.

Broadly speaking, there are two main methods for correcting estimation bias. The first method uses a Matching Error Model (MEM). The second method uses a Linkage Error Model (LEM). An alternative approach is developed in Goldstein et al. (2012), where the authors suggest replacing all potential matches with synthetic records generated by multiple imputation. For a MEM, the probability for a record pair to be a match is computed for each record pair, and the probabilities are summarized in a matrix. The probability could be computed based on the observed agreement pattern or the observed match weight of the record pair. The use of MEM for correcting estimation bias is thoroughly discussed in Scheuren and Winkler (1993). The authors assumed that an accurate MEM could be constructed, and showed that using the MEM and a bias-corrected estimator would lead to unbiased estimates of linear regression coefficients. Lahiri and Larsen (2005) developed an unbiased linear regression coefficient estimator which incorporates a MEM. The authors also proposed a new MEM which computes the posterior probability for a record pair to be a match. Hof and Zwinderman (2012) developed several unbiased regression coefficient estimators for different regression models.

Different MEMs have been proposed in the literature. In addition to the MEM proposed in Lahiri and Larsen (2005), Scheuren and Winkler (1993) proposed a MEM by using the
match weight of a record pair. A similar MEM was proposed in Goldstein et al. (2012). Lahiri-Larsen (LL)’s MEM is studied in Hof and Zwinderman (2012) and Chipperfield and Chambers (2015). In this chapter we consider LL’s MEM for comparison purposes.

On the other hand, LEM calculates the probability that two sampling units, when properly indexed, are linked by the probabilistic linkage algorithm. The probabilities for each pair of sampling units are collected in a matrix. Calculating these probabilities might require simulating the process of probabilistic linkage iteratively, such as the bootstrap method in Chipperfield and Chambers (2015). Chambers (2009) proposed several bias-adjusted estimators by incorporating LEM, under one-to-one correspondence. The author showed that, when an LEM is constructed accurately, data users could use the bias-adjusted estimators to obtain unbiased linear regression coefficient estimates from probabilistically-linked data. Subsequent works by Chambers and Kim (2010), Chipperfield and Chambers (2015) extended the result for other regression models and other linkage scenarios than one-to-one correspondence. All these works made it possible for a data analyst to obtain unbiased results directly from probabilistically-linked data. We note that for LL’s MEM, it does not matter whether the record pairs are probabilistically-linked, as the calculation of the MEM only depends on the observed agreement patterns.

Chipperfield and Chambers (2015) showed that LL’s MEM perform poorly under one-to-one correspondence. Our simulation showed that LL’s MEM tends to assign a higher probability of being a match to non-match pairs and lower probability of being a match to match pairs compared with a LEM. We note that a record pair being linked by the probabilistic linkage algorithm provides strong evidence that the record pair is a non-match. As a result, the record pair should have a low probability of being a match. However, LL’s MEM does not consider this fact. Consequently, LL’s MEM might not be accurate in some situations. Our simulation showed that using LL’s MEM will result in larger relative standard errors for linear regression coefficient estimates compared with using a LEM.

Constructing an accurate LEM might be difficult. Chambers (2009) constructed an LEM by assuming that all matches have the same probability $\lambda$ of being linked, while all non-matches have the same probability of $(1 - \lambda)/(n - 1)$ being linked, where $n$ is the total number of records in each file. For high quality probabilistically-linked data, $\lambda$ should be close to 1. This LEM is called “Exchangeable Linkage Error (ELE) Model”. The ELE model has only one parameter $\lambda$. $\lambda$ is also the proportion of links which are matches, i.e. the proportion of correct links. To estimate this parameter, Chambers (2009) proposed to use a good audit sample drawn from the probabilistically-linked data. A clerical review process is needed for estimating the proportion of correct links in the sample. Chipperfield and Chambers (2015) proposed a bootstrap method for estimating this parameter without using an audit sample. The authors showed that using the bootstrap method leads to very good estimation of this parameter. However, because the ELE made the assumption on the structure of linkage errors, it will cause estimation bias in many
This chapter introduces a new LEM, namely Conditional Linkage Error (CLE) model, under one-to-one correspondence. Following the convention in Chambers (2009) and Chipperfield and Chambers (2015), we assume a secondary file is being linked to a “benchmark file”. The benchmark file contains all explanatory variables of a regression model, the secondary file contains the response variable of the regression model. The CLE model is constructed by conditioning on the link fields in the benchmark file. The advantage of the CLE model over the ELE model is that: (a) it produces unbiased regression coefficient estimates in wider scenarios; (b) by conditioning on more information, the CLE model could lead to estimates with a lower mean squared error than the ELE model for a regression coefficient. In line with Scheuren and Winkler (1993), Lahiri and Larsen (2005), and Chipperfield and Chambers (2015), we assume all link fields are categorical and discrete.

This chapter is organised as follows: Section 8.1 introduces the latent model and probabilistic linkage algorithm. Section 8.2 reviews the work of Chambers (2009). Section 8.3 introduces the ELE model and the CLE model. Section 8.4 discusses the ELE model, the CLE model and LL’ MEM. Section 8.5 presents a simulation study. Section 8.6 concludes the chapter.

8.1 Latent model and probabilistic-linkage algorithm

In this section we introduce how two datasets are linked by the probabilistic linkage algorithm. Suppose we wish to link two files, File \( A \) and File \( B \), and under one-to-one correspondence. Each file contains \( n \) records. Following Lahiri and Larsen (2005); Chambers (2009); Kim and Chambers (2010), we assume that File \( A \) contains all explanatory variables of a regression model, and File \( B \) contains the response variable. File \( A \) is the benchmark file.

8.1.1 Linking fields and agreement pattern

A typical dataset contains details of units across several attributes. Linking fields are fields that are common to both datasets. Examples of linking fields are first name, date of birth, postcode. Records are linked based on their linking fields. Suppose both files contain \( k \) linking fields. Define a record pair to be a record on File \( A \) and a record on File \( B \). As the matching status of a record pair is not known, each record pair is a potential match. For each record pair we observe an agreement pattern on the set of linking fields. The observed agreement pattern for the \( i \)th record on File \( A \) and the \( j \)th record on File \( B \), called here the \((A_i, B_j)\) record pair, is denoted as \( a_{ij}^o = (a_{ij1}^o, a_{ij2}^o, \ldots, a_{ijk}^o) \), where \( a_{ijl}^o = 1 \) if both records share the same value on the \( l \)-th linking field and 0 otherwise. For
instance, suppose there are three linking fields which take the values (2,2,1) for record $A_i$ and take the values (2,3,1) for record $B_j$. In this case, the observed agreement pattern is $a_{ij}^o = (1,0,1)$. The observed agreement pattern $a_{ij}^o$ can be treated as a realization of an underlying random variable $a_{ij}$, where $a_{ij} = (a_{ij1}, a_{ij2}, \ldots, a_{ijk})$. We say that $a_{ijk}$ is the agreement outcome of $(A_i, B_j)$ on the $k$-th link field. The observed agreement patterns of all record pairs are stored in a observed comparison outcome matrix $A^o$ (the superscript $o$ is to emphasise that the matrix is observed). The underlying random variable of $A^o$ is $A$, where $A = (a_{11}, a_{12}, \ldots, a_{1n}, a_{21}, a_{22}, \ldots, a_{2n}, \ldots, a_{n1}, a_{n2}, \ldots, a_{nn})^T$, $T$ denotes vector transpose. That is, we treat $A^o$ as a realisation from the random variable $A$. It is often assumed in the literature that the agreement patterns in $A$ are independent. The distribution of an agreement pattern is described by a latent model which will be shown next.

### 8.1.2 Latent model and match weight

A latent model is used to classify the matching status of a record pair. That is, each record pair belongs to one of two latent (or unobserved) classes: match or non-match. Denote the match class by $M$ and the non-match class as $U$. Denote the conditional probability of observing $a_{ij}^o$ if $(A_i, B_j)$ is a match by $P(a_{ij} = a_{ij}^o | M)$, and denote the conditional probability of observing $a_{ij}^o$ if $(A_i, B_j)$ is a non-match by $P(a_{ij} = a_{ij}^o | U)$.

It is often assumed (Fellegi and Sunter 1969; Lahiri and Larsen 2005), and is also adopted in many real-life applications (such as the Death Registrations to Census Linkage Project in the Australian Bureau of Statistics) that, for a record pair, the agreement outcome on one linking field is independent of the agreement outcome on other linking fields, given its latent class. Following this assumption, the probability of observing an agreement pattern $a_{ij}^o$ for $a_{ij}$ that is a match or non-match, respectively, is given below:

\[
P(a_{ij} = a_{ij}^o | M) = \prod_{l=1}^{k} M_l^{a_{ijl}} (1-M_l)^{(1-a_{ijl})} \tag{8.1}
\]

\[
P(a_{ij} = a_{ij}^o | U) = \prod_{l=1}^{k} U_l^{a_{ijl}} (1-U_l)^{(1-a_{ijl})}
\]

where the parameter $M_l$ is the probability that a match record pair agrees on the $l$-th linking field, and the parameter $U_l$ is the probability that a non-match record pair agrees on the $l$-th linking field. The parameters $M_l$ and $U_l$ could be estimated through either supervised learning or unsupervised learning (Jaro. 1989). Supervised learning requires training data with known matching status of the record pairs. The parameters could then be estimated directly from the training data. The most common unsupervised approach uses maximum likelihood estimation of the parameters. Specifically, the likelihood of observing the comparison outcome matrix $A^o$ is given by:
\[ L(A^o) = \prod_{i=1}^{n} \prod_{j=1}^{m} \{ P(a_{ij} = a_{ij}^o | M) \pi + P(a_{ij} = a_{ij}^o | U) (1 - \pi) \}, \]

where \( n \) is the number of records in File A, \( m \) is the number of records in File B. \( \pi \) is the probability that a random record pair is a match and is equal to \( n_0 / mn \), where \( n_0 \) is the number of matches between the two files. Under one-to-one correspondence, \( n = m = n_0 \) and \( \pi = 1/n \). In general, for any \( n_0, m \) and \( n \), the parameters \( M_l, U_l \) and \( \pi \) are estimated via expectation maximization. To simplify our discussion of this chapter, we assume that \( M_l \) and \( U_l \) are known.

From Equation (8.1), we could calculate the match weight for each record pair. Mathematically, the match weight of \( (A_i, B_j) \), denoted as \( w_{ij} \), has the following form:

\[ w_{ij} = \log_2 \left( \frac{P(a_{ij} = a_{ij}^o | M)}{P(a_{ij} = a_{ij}^o | U)} \right). \]

For a record pair, the higher the match weight, the more likely that the record pair is a match. For all record pairs, the match weights are calculated, and the record pairs are ready to be linked by the probabilistic linkage algorithm. In practice, a statistical agency could use any linkage algorithm which ensures one-to-one assignment, as long as the linkage algorithm is reasonable. That is, the linkage algorithm should only link those record pairs with high match weights. The CLE model we propose in this paper could be used for any linkage algorithm. In the following we introduce the probabilistic linkage algorithm.

### 8.1.3 Probabilistic Linkage algorithm

Following Chipperfield and Chambers (2015, Section 4.2), a simple probabilistic linkage algorithm is used in this chapter. The probabilistic linkage algorithm works as follows:

1. Sort all record pairs by their match weights, \( w_{ij} \) for \( i, j = 1, \ldots, n \), from the highest to the lowest. Note that there are \( n^2 \) record pairs;
2. The first record pair in the ordered list is linked;
3. All record pairs containing either of the records linked in Step 2 are removed from the list of potential links, meaning that those removed record pairs will not be linked in the future;
4. Return to Step 2 until all records are linked.

### 8.1.4 Blocking variable

In this paper, we assume the existence of a blocking variable \( Z \). A blocking variable is measured without error on both files, such that a record pair cannot be linked at all if the
two records share different values of the blocking variable $Z$. Records are partitioned into different blocks according to the value of $Z$. Records in the same block share the same $Z$ value. Therefore, only records within the same block have positive probabilities of being linked, i.e. linkage errors only exist within a block.

8.2 Using linkage error model to correct estimation bias

Once two datasets are linked by the probabilistic linkage algorithm, the linked dataset could be used for data analysis. To do this, a Linkage Error Model (LEM) needs to be constructed first. Suppose File $A$ and File $B$ to be linked are of size $n$. An LEM is a matrix, denoted as $Q$, with the following form:

$$Q = \begin{bmatrix}
q_{11} & q_{12} & \cdots & q_{1n} \\
q_{21} & q_{22} & \cdots & q_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
q_{n1} & q_{n2} & \cdots & q_{nn}
\end{bmatrix}$$

where $q_{ij}$ is the probability that $(A_i, B_j)$ is linked by the linkage algorithm. $B_i$ is a record on File $B$. $A_i$ and $B_j$ is a match. $B_i$ is not observable. In the following we review previous works developed by Chamber (2009) on using LEM to correct estimation bias.

Define $B$ be a re-ordering of the records, such that $(A_1, B_1), \ldots, (A_i, B_i), \ldots, (A_n, B_n)$ are matches. We do not observe $B$ because we do not know the matching statuses of record pairs between File $A$ and File $B$. We define $\tilde{a}_{ij}$ as the agreement pattern for $(A_i, B_j)$.

8.2.1 LEM overview

Denote $X_i = (X_{i1}, X_{i2}, \ldots, X_{in})$ be the set of explanatory variables for record $A_i$, $Y_i$ be the response variable such that $E(Y_i|X_i) = f(\beta, X_i)$. $Y_i$ is possessed by $B_i$. After probabilistically linking the data, $X_i$ is linked to $Y_i^*$, which is observable. Denote $Y = (Y_1, Y_2, \ldots, Y_n)^T$, $Y^* = (Y_1^*, Y_2^*, \ldots, Y_n^*)^T$, $X = (X_1, X_2, \ldots, X_n)^T$, where $T$ denotes matrix transpose. Chambers (2009) modelled the relationship between the two random vectors as $Y^* = PY$, where $P$ is a random permutation matrix which shuffles the order of $Y$ into $Y^*$ by applying the linkage algorithm. $P$ is full rank, each row and column contains a value of one and n-1 values of zero. Assuming $P$ and $Y$ are independent given $X$, that is, $E(Y^*|X) = E(P|X)E(Y|X)$. Then $E(P|X)$, which is denoted as $Q$, is the LEM. $Q$ is an $n$ by $n$ matrix with elements $(q_{ij})_{i,j=1,\ldots,n}$, where $q_{ij}$ is the probability that $A_i$ and $B_j$ are linked given $X$. So $q_{ii}$ is the conditional probability that record $A_i$ is correctly linked. In the following we review unbiased estimators for two regression models.
8.2.2 Linear Regression

Let the true linear model be
\[ Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i, \]
where \( E(\epsilon_i) = 0 \) and \( \text{Var}(\epsilon_i) = \sigma^2 \) and \( \text{cov}(\epsilon_i, \epsilon_j) = 0 \) for \( j \neq i \). We observe \((x_{i1}, x_{i2}, \cdots, x_{ip})\) for each record \( A_i \), and we observe \( y_i^* \), which is linked to \( A_i \).

Let \( \beta = (\beta_0, \beta_1, \beta_2, \cdots, \beta_p) \). Chambers et al. (2009) use an LEM to estimate \( \beta \) by
\[
\hat{\beta} = (X_d^T Q^T Q X_d)^{-1} X_d^T Q^T y^*,
\]
where \( y^* = (y_1^*, y_2^*, \cdots, y_n^*) \), \( y_i^* \) is the observed value of the response variable which is linked to \( x_i \). \( X_d^T \) is the transpose of \( X_d \), \( X_d \) is the design matrix, i.e.
\[
X_d = \begin{bmatrix}
1 & x_{11} & x_{12} & \cdots & x_{1p} \\
1 & x_{21} & x_{22} & \cdots & x_{2p} \\
\vdots & \vdots & \ddots & \vdots \\
1 & x_{n1} & x_{n2} & \cdots & x_{np}
\end{bmatrix}
\]

8.2.3 Logistic Regression

Suppose the true model is
\[
y_i = \begin{cases} 
1, & \text{if } \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i > 0 \\
0, & \text{otherwise}
\end{cases}
\]
where \( \{\epsilon_i\}_{i=1}^n \) are independently distributed logistic errors. Denote \( \beta = (\beta_0, \beta_1, \beta_2, \cdots, \beta_p) \).

Chambers et al. (2009) uses an LEM to estimate \( \beta \) by iteratively calculating:
\[
\hat{\beta}^{k+1} = \hat{\beta}^k - (-X_d^T Q^T Q X_d)^{-1} X_d^T Q^T (y^* - Qf),
\]
until convergence, where \( f = \exp(X_d\hat{\beta}^k)/(1 + \exp(X_d\hat{\beta}^k)) \), \( D = \text{diag}(f(I - f)) \), \( I \) is identity vector with same order as \( f \), \( X_d \) is the design matrix, \( X_d^{-1} \) is the inverse of \( X_d \).

8.3 Conditional Linkage Error Model

In this section we review the Exchangeable Linkage Error (ELE) model, which is the only LEM proposed in the literature so far. Then, we will introduce a new LEM, which we call the Conditional Linkage Error (CLE) model.
8.3.1 ELE model

In practice, the form of $Q$ is difficult to know. Chambers (2009) proposed an Exchangeable Linkage Error (ELE) model by assuming that $q_{ii} = \lambda$ for all $i = 1, 2, \cdots, n$ and $q_{ij} = \frac{1-\lambda}{n-1}$ for all $i \neq j$, where the parameter $\lambda$ is the proportion of links that are matches. Under this specification, the ELE model only requires the knowledge of $\lambda$. To estimate $\lambda$, Chambers (2009) suggested to use a good audit training data with known matching status. Alternatively, a model-based estimate of $\lambda$ is described in Chipperfield and Chambers (2015). The authors noted that, even though the agreement pattern for $(A_i, \tilde{B}_i)$ cannot be observed, the distribution of the agreement pattern for $(A_i, \tilde{B}_i)$ is given in Equation (8.1) for all $i, j = 1, 2, \cdots, n$. Let $\tilde{a}_{ij}$ be the random variable of the agreement pattern for $(A_i, \tilde{B}_j)$. The authors proposed to generate agreement patterns $\tilde{a}_{ij}$ using (8.1):

(a) For matches $(A_i, \tilde{B}_i)$ for $i = 1, \ldots, n$, generate an agreement pattern $\tilde{a}_{ii}$ according to $\pi_{\tilde{a}_{ii}|M}$ in (8.1).

(b) For non-matches $(A_i, \tilde{B}_j)$ for $i, j = 1, \ldots, n$ and $i \neq j$, generate an agreement pattern $\tilde{a}_{ij}$ according to $\pi_{\tilde{a}_{ij}|U}$ in (8.1).

Note that we do not need to simulate the linking fields of $\tilde{B}$. We only need to simulate agreement patterns for matches and non-matches according to the latent model. The probabilistic linkage algorithm only needs the agreement patterns of the two files to link the records. The estimate of $\lambda$ is calculated in the following way:

1. Generate plausible agreement patterns for $\tilde{a}_{ij}$ as described in (a) and (b) above for all $i, j = 1, \ldots, n$. The simulated agreement patterns are stored in a simulated comparison outcome matrix $\tilde{A}^s$. Repeat this $T$ times and denote the $t$th set of simulation-generated agreement patterns by the matrix $(\tilde{A}^s)_t$. Note that in $(\tilde{A}^s)_t$, we know the class from which an agreement pattern is simulated.

2. For the $t$-th replicate, use the probabilistic linkage algorithm to link records in File $A$ and File $\tilde{B}$ according to $(\tilde{A}^s)_t$ (instead of $A^o$). Note that for each $A_i$, we know which simulated agreement pattern in $(\tilde{A}^s)_t$ is the simulated agreement pattern for $(A_i, \tilde{B}_j)$ for all $j = 1, 2, \cdots, n$. For the $t$-th replicate, when we link the data according to $(\tilde{A}^s)_t$, say if we know that $(A_i, \tilde{B}_i)$ is linked by the probabilistic linkage algorithm, then we know that it is a correct link. Therefore, for all linked record pairs, we know how many of them are correct links. We calculate $p_t$, which is the proportion of links that are matches for the $t$th replicate. Repeat for $t = 1, \ldots, T$. 
3. The estimated $\lambda$ is $\hat{\lambda} = T^{-1}\Sigma tp_t$, the average of the proportions of correct links across all replicates.

Recall that in Section 8.2.1, Chambers (2009) showed that $Q = E(P|X)$, where $P$ is a random permutation matrix and $X$ is the set of explanatory variables. The construction of the ELE model certainly does not take $X$ into account. Therefore, the ELE model only works if $P$ is independent of $X$. Recall that $P$ reflects the outcome of a linkage algorithm. Therefore, when $P$ and $X$ are independent, that means that $X$ is independent of the linkage outcome. It is true when link fields and covariates are independent, because in all situations the linkage outcome is determined by link fields only, and the linkage outcome is not affected by covariates if link fields and covariates are independent. However in general covariates and linkage outcome are not independent. For instance, ‘Gender’ could used as a link field and could also be one of the covariates in a regression model. Therefore, the ELE model is unlikely to hold in practice. Therefore we propose a new linkage error model, namely “Conditional Linkage Error (CLE) model”, for correcting estimation bias. The CLE model is constructed by conditioning on all link fields of the benchmark file. By doing so, it delivers two advantages: 1. the CLE model is able to correct estimation bias under wider scenarios, such as when covariates $X$ and link fields are not independent; 2. for a regression coefficient, the CLE model leads to a lower mean squared error than the ELE model.

### 8.3.2 Conditional Linkage Error model

The CLE model is constructed by conditioning on observed information of link fields in the benchmark file (File A). Assuming there are $k$ link fields. Denote the link fields in File A as $L = (L_1, L_2, \cdots, L_k)$. The information of $L$ is summarised through the following steps:

Step 1: We construct the agreement patterns of all record pairs in File A. For instance, let $A_i$ and $A_j$ form a record pair $(A_i, A_j)$. Then we have the agreement pattern $a_{ij}^B = (a_{ij1}, \cdots, a_{ijk})$, where $a_{ijl}^A = 1$ if $(A_i, A_j)$ agrees on the $l$-th linking field, and 0 otherwise. The superscript in $a_{ijl}^A$ is used to indicate the agreement outcome is from $(A_i, A_j)$. Following this idea, we could obtain all agreement patterns for all record pairs from the benchmark file. The agreement patterns are stored in $A^A$.

Step 2: We define a set of parameters $\{p_l\}_{l=1}^k$. $p_l$ is the probability that $(A_i, A_j)$ agrees on the $l$-th linking field, or $P(a_{ijl}^A = 1) = p_l$ for all $i \neq j$ and $l = 1, 2, \cdots, k$. In practice, given linking fields of File A, we observe $(A^A)^o = \{(a_{ijl}^A)^o\}_{i=1, \cdots, n; i \neq j}$. The value of $p_l$ could be estimated by $\frac{\sum_{i=1}^n \sum_{j \neq i} (a_{ijl}^A)^o}{n(n-1)}$. 
Recall that Equation (8.1) shows the probability of observing an agreement outcome given each latent class. Because of $\mathbf{A}^\alpha$, we could construct the following set of conditional probabilities for observing an agreement outcome for the non-match class $U$:

\[
P(\tilde{\mathbf{a}}_{ijl} = 0|\mathbf{a}_{ijl} = 0, \tilde{\mathbf{a}}_{jij} = 0, U) = 1 - \frac{U_l - (1-p)M_l - p(1-M_l)}{(1-p)(1-M_l)}
\]

\[
P(\tilde{\mathbf{a}}_{ijl} = 0|\mathbf{a}_{ijl} = 0, \tilde{\mathbf{a}}_{jij} = 1, U) = 1
\]

\[
P(\tilde{\mathbf{a}}_{ijl} = 0|\mathbf{a}_{ijl} = 1, \tilde{\mathbf{a}}_{jij} = 0, U) = 1
\]

\[
P(\tilde{\mathbf{a}}_{ijl} = 0|\mathbf{a}_{ijl} = 1, \tilde{\mathbf{a}}_{jij} = 1, U) = 0
\]

(8.2)

The last three conditional probabilities above are obtained by logical inference. For instance, if $(\mathbf{A}_i, \mathbf{A}_j)$ disagrees on the $l$-th link field, and $(\mathbf{A}_j, \tilde{\mathbf{B}}_j)$ agrees on the link field. That means $\mathbf{A}_i$ and $\tilde{\mathbf{B}}_j$ for sure disagree on the $l$-th link field.

The above two steps show how to make use the information of link fields on the File $\mathbf{A}$. Because of Equation (8.2), we could now generate a plausible agreement pattern for $(\mathbf{A}_i, \tilde{\mathbf{B}}_j)$ for all $i, j = 1, \ldots, n$ based on $\mathbf{A}^\alpha$ in the following way:

(a) For match record pairs $(\mathbf{A}_i, \tilde{\mathbf{B}}_i)$ for $i = 1, \ldots, n$, generate an agreement pattern $\tilde{\mathbf{a}}_{ii}^\alpha$ according to $\mathbf{\pi}_{\tilde{\mathbf{a}}_{ii}|M}$ in (8.1);

(b) For non-match record pairs $(\mathbf{A}_i, \tilde{\mathbf{B}}_j)$ for $i, j = 1, \ldots, n$ and $i \neq j$, generate an agreement pattern $\tilde{\mathbf{a}}_{ij}^\alpha$ according to (8.2).

We propose a bootstrap method for constructing the CLE model:

1. Generate plausible comparison outcomes for $\tilde{\mathbf{a}}_{ij}$ according to (a) and (b) above for all $i, j = 1, \ldots, n$. Repeat this $T$ times and denote the $t$th set of simulated agreement outcome matrix by $(\tilde{\mathbf{A}}^\alpha)^t$.

2. For the $t$-th replicate, use the probabilistic linkage algorithm to link File $\mathbf{A}$ and File $\tilde{\mathbf{B}}$ according to $(\tilde{\mathbf{A}}^\alpha)^t$. Let the $t$-th linkage outcome be $M^\alpha(t)$ (a permutation matrix), where $M^\alpha(t) = (m_1, m_2, \ldots, m_n)^T$. $m_i = (m_{i1}, m_{i2}, \ldots, m_{in})$, $m_{ij} = 1$ if $\mathbf{A}_i$ is linked to $\tilde{\mathbf{B}}_j$, and 0 otherwise. Repeat for $t = 1, \ldots, T$.

3. Estimate $Q$ by $\frac{1}{T} \sum_{t=1}^{T} M^\alpha(t)$.

We note that both the ELE and the CLE could be constructed using the bootstrap approach, i.e. for each record pair we simulate an agreement pattern based on the latent
model. The difference is that, for the CLE the agreement pattern is simulated by using the conditional probabilities in Equation 8.2, whereas for the ELE the agreement pattern is simulated by using the unconditional probabilities in Equation 8.1. The simulated agreement patterns are more plausible if the information of $A^A$ is taken into account. A numerical example for comparing these two models are given in the next section.

### 8.4 Comparison between CLE, ELE and MEM

Simulations show that the CLE model is able to correct estimation bias for various regression models when covariates are a function of link fields, i.e. \( X = f(L) \). A full mathematical proof will be sketched in future works. In this chapter, we will compare the performance of the CLE model with the ELE model via simulations. We use both models to compute linear regression coefficient estimates, and we compare the relative bias and Relative Mean Squared Errors (RMSEs) of the results. Moreover, we will also add a Matching Error Model (MEM) proposed by Larhiri and Larsen (2005) for comparison. The purpose is to show that using MEM is unable to correct estimation bias whereas using LEM is able to correct estimation bias.

#### 8.4.1 Numerical example

We compare the difference between ELE and CLE using a numerical example. Suppose File $A$ contains three records and File $B$ contains three records. The true matches are $(A_1, B_1)$, $(A_2, B_2)$ and $(A_3, B_3)$. Suppose there are three linking fields $(L_1, L_2, L_3)$ with 2 categories each. The linking fields’ values are shown in Table 8.1.

Suppose true parameters of the latent model are $M_l = (0.9, 0.9, 0.8)$ and $U_l = (0.5, 0.5, 0.5)$, $p_l = U_l$. Notice that the values of $M_l$ are very high, meaning that it is very likely that a true match pair agrees on all these linking fields. Using the probabilistic linkage algorithm to link the data, the linkage outcome is $(A_1, B_2)$, $(A_2, B_1)$ and $(A_3, B_3)$. That means, $A_1$ and $B_2$ are linked, $A_2$ and $B_1$ are linked, $A_3$ and $B_3$ are linked. Because only $(A_3, B_3)$ is a correct link, therefore $1/3$ of the records are correctly linked.

Table 8.1: Values of linking fields on File $A$ and File $B$.

<table>
<thead>
<tr>
<th>Record</th>
<th>$L_1$</th>
<th>$L_2$</th>
<th>$L_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$A_2$</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$A_3$</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$B_1$</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$B_2$</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$B_3$</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Under the ELE model, \( q_{ii} = 1/3 \) and off-diagonal elements are uniformly \( q_{ij} = 1/3 \). This means that \( A_1 \) has the same probability of being linked to \( B_2 \) and \( B_3 \). However, by observing Table 8.1, we see that \( A_1 \) and \( A_2 \) agree on more linking fields than \( A_1 \) and \( A_3 \), therefore it is more likely for \( (A_1, B_2) \) to be linked than \( (A_1, B_3) \). Under CLE, we could show that \( q_{12} = 0.192 \) and \( q_{13} = 0.006 \), which is consistent with our argument. The \( Q_{CLE} \) matrix is the \( Q \) matrix under the CLE model, which is obtained by following the bootstrap method at the end of Section 8.3.2.

The two \( Q \) matrices under the two models are given below:

\[
Q_{ELE} = \begin{bmatrix}
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
\frac{1}{3} & \frac{1}{3} & \frac{1}{3}
\end{bmatrix}
\]

\[
Q_{CLE} = \begin{bmatrix}
0.802 & 0.192 & 0.006 \\
0.192 & 0.780 & 0.028 \\
0.006 & 0.028 & 0.966
\end{bmatrix}
\]

### 8.4.2 Matching Error Model

In this subsection we review the Matching Error Model (MEM). The mechanisms of using MEM and LEM for correcting estimation bias are similar. However, the definitions of the two approaches are slightly different.

Using a MEM, an \( n \times n \) matrix is constructed. We denote it as \( Q_{MEM} \), which has the following structure:

\[
Q_{MEM} = \begin{bmatrix}
q_{11} & q_{12} & \cdots & q_{1n} \\
q_{21} & q_{22} & \cdots & q_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
q_{n1} & q_{n2} & \cdots & q_{nn}
\end{bmatrix}
\]

where \( q_{ij} \) is the probability that \( (A_i, B_j) \) is a match.

### 8.4.3 Lahiri and Larsen (2005)’s MEM

To compute the probability of a record pair to be a true link, Lahiri and Larsen (2005) proposed to use the following posterior probability:

\[
q_{ij} = P(M|a_{ij}^o) = \frac{\pi P(a_{ij}^o|M)}{\pi P(a_{ij}^o|M) + (1-\pi)P(a_{ij}^o|U)},
\]
π is the probability that a random record pair is a match. However, normally for the MEM matrix computed in this way, the sum of each row does not equal to 1. Under one-to-one correspondence, for each record in File A there is a match in File B. Therefore, the sum of each row of the MEM matrix should equal to 1. The authors suggest to standardize each row (i.e. by dividing the entries by the row sum) to construct the final $Q_{MEM}$ matrix.

### 8.4.4 Unbiased estimators using MEM

Lahiri and Larsen (2005) proposed the following unbiased estimator for a linear regression’s coefficients:

$$
\hat{\beta} = (X_d^T Q_{MEM}^T Q_{MEM} X_d)^{-1} X_d^T Q_{MEM}^T z,
$$

where $X_d$ is the design matrix as in Section 8.2.2, $z = (z_1, z_2, \cdots, z_n)$. $z_i$ is the observed value of the response variable from $C_j$.

For logistic regression coefficients, Hof and Zwinderman (2012) proposed the following iterative method for getting unbiased estimates:

$$
\hat{\beta}^{k+1} = \hat{\beta}^k + (X_d^T Q_{MEM}^T Q_{MEM} X_d)^{-1} X_d^T Q_{MEM}^T (z - \hat{p}(z|X_d,Q_{MEM},\hat{\beta}^k))
$$

$$
\hat{p}(z|X_d,Q_{MEM},\hat{\beta}^k) = \frac{\exp(Q_{MEM} X_d \hat{\beta}^k)}{1 + \exp(Q_{MEM} X_d \hat{\beta}^k)}
$$

$$
\Omega = \text{diag}(\hat{p}(z|X_d,Q_{MEM},\hat{\beta}^k)(1 - \hat{p}(z|X_d,Q_{MEM},\hat{\beta}^k))),
$$

where $\hat{\beta}^k$ stands for the estimate of $\hat{\beta}$ obtained by iterating the above equation $k$ times. To find $\hat{\beta}$, let $\hat{\beta}^0$ be an initial estimate of $\beta$ set by the data user. Then the data user could update the estimate of $\hat{\beta}$ by substituting $\hat{\beta}^0$ in the above equation to obtain $\hat{\beta}^1$, and then substitute $\hat{\beta}^1$ in the above equation to obtain $\hat{\beta}^2$, etc. The final estimate of $\hat{\beta}$ is obtained if the difference between two consecutive estimates of $\hat{\beta}$ is below a threshold level, e.g. 0.001.

### 8.4.5 Discussion on using MEM to correct estimation bias

Different MEMs have been proposed in the literature (see Rubin and Belin (1991), Lahiri and Larsen (2005), Goldstein et al. (2012)). However, the problem with MEMs is that these models generally do not correct estimation bias under one-to-one correspondence scenario. It is noted in Goldstein et al.(2012), that it is an on-going research problem to find a MEM which corrects estimation bias. Hof and Zwinderman (2012) proposed several methods to improve the accuracy of the LL’s MEM, however these methods generally do not correct estimation bias either.

On the contrary, Chipperfield and Chambers (2015) showed that using the ELE model
could correct estimation bias when link fields and covariates are independent. We will show in the simulation that, the CLE model could correct estimation bias in a wider context where covariates and link fields are not independent. Moreover, as an LEM incorporates the probabilistic linkage algorithm, it results in a lower mean squared error than LL’s MEM for a regression coefficient in general. To see this, we still consider the the numerical example in 8.4.1. The link fields of the two files to be link are given in Table 8.1. There are 9 record pairs, three of them are matches, therefore the probability of a random record pair being a match is \( \pi = \frac{1}{3} \). Given that \( M_l = (0.9, 0.9, 0.8) \) and \( U_l = (0.5, 0.5, 0.5) \), we could compute the probability of each record pair being a match given the observed agreement pattern of the record pair using Equation (8.3). For instance, the observed agreement pattern of \( (B_1, C_1) \) is \( (1, 0, 0) \). Then we have

\[
q_{11} = \frac{0.3 \cdot 0.9^1 \cdot 0.9^0 \cdot 0.8^0}{0.3 \cdot 0.9^1 \cdot 0.9^0 \cdot 0.8^0 + (1 - 0.3) \cdot 0.5^1 \cdot 0.5^0 \cdot 0.5^0} = 0.435
\]

We repeat the above process and calculated the \( Q_{MEM} \), which has the following form:

\[
Q_{LL} = \begin{bmatrix} 0.338 & 0.429 & 0.233 \\ 0.396 & 0.312 & 0.292 \\ 0.329 & 0.259 & 0.412 \end{bmatrix}
\]

We can compare \( Q_{LL} \) with \( Q_{CLE} \). Attention needs to be paid at the diagonal elements of the two matrices. As the matches are \( (A_i, B_i) \), \( i = 1, 2, 3 \), we should see larger diagonal elements than off-diagonal elements, i.e. a match should have a higher probability of being linked or a higher posterior probability of being a match given the observed agreement pattern. We say that for \( Q_{CLE} \), the diagonal elements are much larger, whereas it is not true for \( Q_{LL} \). The agreement pattern for record pair \( (B_1, C_1) \) is \( (1, 0, 0) \), which means that it is very unlikely for this record pair to be a match. This fact is captured by the \( Q_{CLE} \) model, with \( q_{13} = 0.006 \). However, we see that for \( Q_{LL} \), the probability for this record pair being a match is 0.233, which is still very high.

Chambers (2009) showed that, for the ELE model, there is an inverse relationship between the proportion of correct links and the relative standard errors of regression coefficient estimates. Our simulations showed that it is also true for the CLE model. That is, we prefer the diagonal elements to be large so that the relative standard errors of regression coefficient estimates are reduced. For the MEM model, the diagonal elements are normally not as large, probably because it does not consider the probability linkage algorithm. The probability linkage algorithm could effectively identify record pairs that are non-matches, and assign 0 (or close to 0) probabilities to those record pairs in an LEM. It is not true for LL’MEM. We found that LL’s method tends to assign a match record pair with a lower probability in the MEM compared with the LEMs.
8.5 Simulation

The simulation compares the performance of the CLE, ELE and LL’s MEM in terms of estimating regression coefficient estimates. We compare them by setting up a hypothetical one-to-one linkage scenario, which is introduced below.

Two linkage scenarios were simulated in this section. In both scenarios, block size was 10 records and only records within a block could be linked. Both File A and File B contain 5 linking fields. In File A the values for each linking field were assigned independently with equal probability from the set of possible categories. The linking fields on File B, say the $i$-th linking field of record $B_i$, denoted as $F^B_{il}$, was generated independently for each $i$ and $l$ according to:

$$P(F^B_{il} = f^A_{il}) = r$$

and

$$P(F^B_{il} = \text{any value other than } f^A_{il}) = \frac{1-r}{C_l-1};$$

where $f^A_{il}$ is the observed value of the $l$-th linking field for $A_i$, $C_l$ is the number of categories in the $l$-th linking field. $r$ is a random realisation from $U(2M_l - 1,1)$. Under this setting, we have $U_l = 1/C_l$ and $p_l = 1/C_l$.

In the first linkage scenario, $(C_1,C_2,C_3,C_4,C_5) = (5,5,4,4,4)$. It means that $U_l = (0.2,0.2,0.25,0.25,0.25)$. We set $M_l = (0.8,0.6,0.6,0.6,0.6)$. In the second linkage scenario, $(C_1,C_2,C_3,C_4,C_5) = (7,7,7,6,6)$. It means that $U_l = (1/7,1/7,1/7,1/6,1/6)$. We set $M_l = (0.8,0.7,0.7,0.6,0.5)$. In Scenario 1 and 2, 74% and 91% of the links were matches. We assumed the true values $M_l$, $U_l$ and $p_l$ are known.

The following process was repeated 500 times for each linkage scenario. On each iteration 1000 records (100 blocks with 10 records each) were independently simulated for File A and File B according to above settings. File A contains three explanatory variables $X_1$, $X_2$ and $X_3$. Denote $X_{i1}$, $X_{i2}$ and $X_{i3}$ as the covariates for the $i$-th record in File A. Let $F^A_{il}$ be the underlying random variable for $f^A_{il}$. The distributions of $X_{i1}$, $X_{i2}$ and $X_{i3}$ satisfy the following:

1. $X_{i1} \sim U(80,180)$ if $F^A_{i1} + F^A_{i2} \leq 6$, otherwise $X_{i1} \sim U(170,300)$;
2. $X_{i2} \sim U(10,20)$;
3. $X_{i3} \sim 0.5F^A_{i2}$.

Note that $X_1$ and $X_3$ are dependent of the first two linking fields in File A. File B contains two response variables $Y_1$ and $Y_2$, where $Y_{i1} = 30 + X_{i1} + 2X_{i2} + \xi_i$, $\xi_i \sim N(0,2)$; $Y_2 = 1$ if $-4.5 + 2X_{i2} + \delta_i > 0$ and 0 otherwise, where $\delta_i$ is logistic noise with mean 0 and scale 1. Therefore the true parameters for linear regression coefficients were $(\beta_0, \beta_1, \beta_2) = (30,1,2)$ and for logistic regression coefficients were $(\beta_3, \beta_4) = (-4,2.5)$. 
The performance of LL’s MEM, ELE model and CLE model under the two linkage scenarios are compared. For LL’s MEM, each row of the $P$ matrix is standardised so that the elements in each row sum up to 1. For the ELE model, the parameter $\lambda$ were estimated by the bootstrap method (Section 8.3.1) using 50000 iterations. We also computed the naive results which were obtained by performing standard statistical analysis on the linked data. The naive results were biased because of linkage errors.

After all 500 iterations were simulated, the absolute relative mean squared errors (RMSE) and absolute relative biases were computed. The results of linkage scenario 1 is given in Table 8.2 and the results of linkage scenario 2 is given in Table 8.3.

**Table 8.2: Results of linkage scenario 1**

| parameters $\beta$ | model  | RMSE | $|\text{Relative Bias}|$ |
|---------------------|--------|------|-----------------|
| $\beta_0$          | naive  | 75.86| 156.6%          |
|                    | LL     | 43.50| 116.7%          |
|                    | ELE    | 3.047| 19%             |
|                    | CLE    | 3.281| 3.5%            |
| $\beta_1$          | naive  | 0.052| 22.7%           |
|                    | LL     | 0.032| 17.9%           |
|                    | ELE    | 0.002| 3.4%            |
|                    | CLE    | 0.001| 0.5%            |
| $\beta_2$          | naive  | 0.226| 26.3%           |
|                    | LL     | 0.175| 14.7%           |
|                    | ELE    | 0.080| 0.3%            |
|                    | CLE    | 0.130| 0.7%            |
| $\beta_3$          | naive  | 0.303| 25.4%           |
|                    | LL     | 0.024| 1%              |
|                    | ELE    | 0.066| 6.2%            |
|                    | CLE    | 0.045| 0%              |
| $\beta_4$          | naive  | 0.199| 30.8%           |
|                    | LL     | 0.029| 7.8%            |
|                    | ELE    | 0.039| 7.3%            |
|                    | CLE    | 0.027| 0.2%            |
From the tables, we observe the following: 1. LL’s MEM leads to biased results for all parameters; 2. The ELE model and CLE model lead to better results in terms of RMSE and absolute relative bias than LL’s MEM in general. This shows that LEM models are more efficient because they take the probability linkage algorithm into account; 3. While the ELE model leads to biased results except for $\beta_2$, the CLE model leads to low biases for all parameters in both linkage scenarios. The exception is $\beta_0$ in linkage scenario 1, where the absolute relative bias is around 3.5%. Our unreported simulations show that our CLE model leads to estimates with increased bias when linkage quality is bad (say less than 80% correct links). This issue will be investigated in the future. In conclusion, the results suggest that the CLE model leads to better results than LL’s MEM and the ELE model if explanatory variables and linking fields are not independent, and the CLE model is able to correct the estimation bias in this situation.
8.6 Conclusion

In this chapter we introduced a new linkage error model, namely the conditional linkage error (CLE) model, under one-to-one correspondence. Estimates using the CLE model have very low bias even when the covariates and linking fields are correlated—this is not the case for the ELE. Simulation results showed that the CLE model outperformed the ELE and that both LEM approaches (ELE and CLE) outperformed the MEM approach.
Chapter 9

Conclusion

This thesis presents our research achievements throughout the course of my Ph.D study. We derived a few results on the aspect of data confidentiality. Specifically, we showed several disclosure risks associated with using the noise-multiplication masking scheme to protect data confidentiality. We proposed several measures to aid the process of noise generating variable selection, so that a balanced utility-risk tradeoff could be achieved for noise-multiplied data. We also developed a perturbation algorithm which could be used for output perturbation in a query system. The perturbation algorithm is very effective against the differencing attack. Other than the research developments on data confidentiality, our research also covers the area of analysing probabilistically-linked data. We developed a linkage error model which helps to correct estimation bias of probabilistically-linked data. We showed that the linkage error model outperforms other models, and that the linkage error model can deliver parameter estimates with very low bias while other models fail to do so. The research outcomes outlined in the thesis might help to promote future research on the noise-multiplication masking scheme and the development of more powerful linkage error models.
Bibliography


