Spectral Anonymization of Data

by

Thomas Anton Lasko

M.D., University of California, San Diego (2000)
S.M., Massachusetts Institute of Technology (2004)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2007

Author .....
Department of Electrical Engineering and Computer Science
August 6, 2007

Certified by .....
Peter Szolovits
Professor
Thesis Supervisor

Certified by .....
Staal A. Vinterbo
Assistant Professor
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Spectral Anonymization of Data

by

Thomas Anton Lasko

Submitted to the Department of Electrical Engineering and Computer Science
on August 6, 2007, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science

Abstract

Data anonymization is the process of conditioning a dataset such that no sensitive information can be learned about any specific individual, but valid scientific analysis can nevertheless be performed on it. It is not sufficient to simply remove identifying information because the remaining data may be enough to infer the individual source of the record (a reidentification disclosure) or to otherwise learn sensitive information about a person (a predictive disclosure).

The only known way to prevent these disclosures is to remove additional information from the dataset. Dozens of anonymization methods have been proposed over the past few decades; most work by perturbing or suppressing variable values. None have been successful at simultaneously providing perfect privacy protection and allowing perfectly accurate scientific analysis.

This dissertation makes the new observation that the anonymizing operations do not need to be made in the original basis of the dataset. Operating in a different, judiciously chosen basis can improve privacy protection, analytic utility, and computational efficiency. I use the term 'spectral anonymization' to refer to anonymizing in a spectral basis, such as the basis provided by the data's eigenvectors.

Additionally, I propose new measures of reidentification and prediction risk that are more generally applicable and more informative than existing measures. I also propose a measure of analytic utility that assesses the preservation of the multivariate probability distribution. Finally, I propose the demanding reference standard of nonparticipation in the study to define adequate privacy protection.

I give three examples of spectral anonymization in practice. The first example improves basic cell swapping from a weak algorithm to one competitive with state-of-the-art methods merely by a change of basis. The second example demonstrates avoiding the curse of dimensionality in microaggregation. The third describes a powerful algorithm that reduces computational disclosure risk to the same level as that of nonparticipants and preserves at least 4^{th}order interactions in the multivariate distribution. No previously reported algorithm has achieved this combinaton of results.

Thesis Supervisor: Peter Szolovits
Title: Professor

Thesis Supervisor: Staal A. Vinterbo
Title: Assistant Professor

Acknowledgments

This work was funded in part by the National Library of Medicine (grants T15 LM 007092-14 and 2R01LM007273-04A1).

My greatest thanks belong to my wife Kathy. Although her academic contributions to this work were minimal, she supported it in every other way possible. Without her help, I simply would not have reached this point, and her contributions outweigh those of anyone else listed here.

Nevertheless, the contributions of others were not small. Peter Szolovits of the Medical Decision Making group of MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL) was my research advisor, and allowed me the freedom to pursue pretty much any reasonable research direction I wanted. If I needed help he was there, but if I just needed time and space he left me alone. He has facilitated my work and stimulated my imagination, and these are about the best things one can say of an advisor.

Staal Vinterbo of the Harvard-MIT Division of Health Sciences and Technology (HST), the Decision Systems Group (DSG) of Brigham and Women's Hospital and Harvard Medical School was my primary advisor for the specific domain of data anonymization. Staal has spent many hours with me kicking around ideas, suggesting research strategies, and evaluating my work, and has made valuable contributions to it along the way. The measures of disclosure risk are examples of good things that came out of these long discussions.

My introduction to data anonymization was by attending a paper Staal presented at a medical computing conference. Before this I had not considered the idea that it was either desirable or possible to release a set of data that kept the patient's identity secret but all other information public, and after hearing that paper I was completely jealous that he got to work on such a cool problem.

Hal Abelson was the third member of my committee, and provided valuable feedback on drafts of this dissertation. It was an honor to work with someone as brilliant and insightful as Hal. One can understand the popularity of his previous work by his insistence on understanding all issues at a simple, intuitive level. Working with Hal, you quickly learn that if you can't explain things at the intuitive level, then you don't really understand them yourself.

Dan Masys, formerly of the UC San Diego School of Medicine and now Professor and Chair of the Department of Biomedical Informatics at the Vanderbilt University Medical Center introduced me to computational medicine when I was in medical school. By some great fortune, Dan was assigned as my advisor there. During the phase when everyone was choosing a medical specialty, I went to him and asked for suggestions on which specialty would take best advantage of my quantitative background, and which would allow the greatest opportunity for creative, synthetic work. He said, "Have I got the field for you!" He pointed me to the 4^{th}-year rotation in biomedical informatics at the National Library of Medicine, and I was hooked. That experience led to a postdoctoral fellowship in the field and then to this graduate work. From medical school to this point, Dan has probably done more to facilitate and mentor my career than any other person. I also know of nobody with a finger

more perceptively on the pulse of computational medicine than Dan. If you want an insightful view of the future of the field, he's the one to get it from.

Octo Barnett of the Laboratory of Computer Science at Harvard's Massachusetts General Hospital was the director of my postdoctoral fellowship, and it was a privilege to work in his lab. Octo was the first person to put a mainframe computer in a hospital for clinical use, and has a lot of valuable (and some less valuable but hilarious) things to say about the application of computational techniques to medicine.

Lucila Ohno-Machado of HST is the new director of the DSG. She has been a wonderful advisor and mentor during my all of my work in Boston, both at Harvard Med and at MIT. It was Lucila who convinced me that a PhD would be a worthwhile and reachable goal, and she has given me great advice and support along the way. She introduced me to the field of machine learning, which became the focus of my graduate study and the core of this dissertation. I often wonder if I would have taken this direction if it were not for her excellent course on the topic.

Fern DeOliveira has provided wonderful administrative support during my years in Pete's research group. It's been a pleasure to work with such an interesting and thoughtful person. Marilyn Pierce and Janet Fischer in the department's graduate administrative office have also helped a great deal in navigating the waters and avoiding the hidden hazards of graduate school.

I must also thank the doctors, nurses, and support staff at Gundersen Lutheran Medical Center in La Crosse, WI, where I trained clinically. They had never heard of a physician going into computational medicine as a specialty, but they provided me a great clinical foundation. The bulk of what I know about clinical medicine, and many of the ideas I've had about using computing power to solve clinical problems, I acquired during my time there. They could have been annoyed by someone taking this unusual path, but instead they were excited and enthusiastically supportive.

Finally, it is one of Knuth's *Things a Computer Scientist Rarely Talks About*, but I must make the personal observation that I believe all truth and beauty come from God. Some of the mathematics I have encountered in this work, such as spectral kernel methods, are breathtakingly beautiful and broadly powerful. Others have called them magic — I cannot help but see the hand of God in them.

I also agree with many that inspiration is as operative in science as it is in art. If any amount of beauty or truth has somehow found its way into this dissertation, I acknowledge the Source of its inspiration. Of course, anything that in hindsight turns out to be incorrectly interpreted, un-insightful, or just plain wrong, I take credit for all of that.

I set this dissertation as my Even Ha'Azer, my Stone of Helping, for To this point hath the LORD helped me.

Contents

1 Introduction **13**
- 1.1 Data Anonymization . . . 13
- 1.2 Non-technical Overview . . . 18

2 Background **23**
- 2.1 Assessing Privacy Protection . . . 24
  - 2.1.1 Prior Measures . . . 26
- 2.2 Assessing Analytic Utility . . . 30
- 2.3 Existing Anonymization Methods . . . 30
  - 2.3.1 Additive Noise . . . 31
  - 2.3.2 Local Suppression and Global Recoding . . . 34
  - 2.3.3 Data Swapping . . . 35
  - 2.3.4 Microaggregation . . . 37
  - 2.3.5 Data Synthesis . . . 40

3 Proposed Assessment Measures **47**
- 3.1 Disclosure Risk . . . 47
- 3.2 Analytic Utility . . . 49

4 Basic Spectral Anonymization **55**
- 4.1 Theory and Examples . . . 55
  - 4.1.1 Example - Cell Swapping . . . 56
  - 4.1.2 Example - Microaggregation . . . 58
- 4.2 Experiments . . . 59
  - 4.2.1 Methods . . . 59
  - 4.2.2 Results . . . 60
  - 4.2.3 Discussion . . . 63

5 Nonlinear Spectral Anonymization **65**
- 5.1 Theory and Example . . . 65
- 5.2 Experiments . . . 67
  - 5.2.1 Methods . . . 68
  - 5.2.2 Results . . . 69
  - 5.2.3 Discussion . . . 70

6 Conclusions and Open Problems **79**
6.1 Summary 79
6.2 Limitations 80
6.3 Open Problems 81
6.4 Conclusion 83

A Dataset Details **85**

List of Figures

1-1 Individual identifiers specified in the HIPAA privacy rule. 15
1-2 Deidentification and reidentification 17
1-3 Simplified example of anonymization 20

3-1 Example analysis of analytic utility. 53

4-1 The SVD-Swapping algorithm. 57
4-2 The Spectral-RHS algorithm. 59
4-3 Privacy protection of basic spectral anonymization. 61
4-4 Reidentification analysis using the new privacy measures. 62

5-1 The Partitioned ICA Swapping algorithm. 67
5-2 Privacy protection of nonlinear anonymization methods. 69
5-3 Analytic utility of nonlinear anonymization methods. 72

A-1 Continuous Variables. 86

List of Tables

4.1 Utility of Basic Methods 63

5.1 Utility of Nonlinear Methods 71

A.1 Binary Variables 85

Chapter 1

Introduction

1.1 Data Anonymization

The goal of data anonymization is to allow the release of scientifically useful data about individuals while simultaneously protecting their privacy. It is not enough to remove explicit identifiers such as names or phone numbers. We must also remove enough additional information so that an attacker cannot infer an identity based on what remains (a reidentification disclosure) or otherwise infer sensitive information about a person (a prediction disclosure). These disclosures could be made by examining the remaining data for combinations of variables that might uniquely identify an individual. For example, a record that lists a 50 year old man with a new tumor that usually first appears in childhood could easily be enough to identify a particular person. In general, each variable we add to a record narrows the field of individuals who could have produced it, and given enough variables we can always infer an identity.

Data anonymization has been an area of active research for three decades, yet nearly every aspect of it remains an open question: How do we measure privacy protection, and what amount of protection do we want? What is the optimal method of perturbing the data to achieve this protection? How do we measure the impact of the perturbation on scientific analysis, and what is an acceptable impact?

Good anonymization techniques are becoming essential for medical research, and I will use this domain as my primary motivation. Medical research often depends on large disease registries and clinical trial databases. Physician researchers complain that patients are opting out of these registries and trials for fear that their information might be disclosed to unknown — and unscrupulous — third parties. These growing opt-out rates are significantly reducing the usefulness of registries and causing bias in their datasets [1, 2, 3]. Less than 40% of patients, for example, signed informed consent for inclusion in the Canadian Stroke Network Registry, despite a major effort to enroll them[4]. Although simple pledges of confidentiality have not altered opt-out rates in behavioral surveys [5], the idea that a patient's identity could be provably

and irreversibly removed from their data may help assuage fears of that data being used against them. If anonymization can provide a provable and permanent barrier to associating a particular patient with a particular record, then it may even reduce some of the legal burdens associated with creating and maintaining such registries.

The United States Congress has recognized the need to protect patient privacy in the face of rapidly increasing electronic exchange of medical information. It therefore included in the Health Insurance Portability and Accountability Act of 1996 (HIPAA) a mandate to adopt Federal regulations for the protection of medical information. In response to this, the US Department of Health and Human Services issued the HIPAA Privacy Rule that specifies how this information should be protected. Among other things, the Privacy Rule lists 18 data elements that it considers identifying information, that must be removed from a dataset in order to release the dataset without restriction (Figure 1-1) [6].

Removing this identifying information, or *deidentification*, is a necessary but insufficient step in anonymization. The remaining data don't directly reveal individual identities, but they may be combined to form a unique identifier or *key*. The key could be used to link a deidentified record with a record from a different source that *does* include an individual identifier. Thus, a deidentified database can be linked with outside information to *reidentify* its data and expose the individuals they describe. This is exactly what happened to when a reporter reidentified records in released, deidentified AOL search engine data [7].

A well-known example of key formation is that the simple combination of birth date and 5-digit zip code uniquely identified 69% of registered voters in Cambridge, MA [8]. An otherwise deidentified database with this information intact can thus be reidentified by linking it with voter registration records. The HIPAA Privacy Rule restrictions are intended to prevent this sort of thing, and in fact they do not allow data releases that include full dates of birth or 5-digit zip codes without a contractual agreement that the recipient will not attempt to reidentify the data.

Anonymization therefore requires us to prevent reidentification. The only known way to do this is to remove additional information from the data to prevent a link between an anonymized record and an identified record from an outside source. The most powerful known method of making such a link is by forming a unique key that either exactly or approximately matches the same information in an identified record from the third source. For inexact key matches, the closest such match is taken as the correct one. Since any variable of the dataset can potentially contribute to a key, this means we must perturb all variables. Moreover, the more columns there are to contribute to a key, the greater the perturbation must be to prevent the key's formation, since each additional variable adds discriminating power [9, 10]. The impact on analysis of the larger perturbations grows quickly, often exponentially, leading one researcher to conclude that "when a data set contains a large number of attributes which are open to inference attacks, we are faced with a choice of either completely suppressing most of the data or losing the desired level of anonymity"

1. Names.
2. All geographic subdivisions smaller than a state, including street address, city, county, precinct, ZIP Code, and their equivalent geographical codes, except for the initial three digits of a ZIP Code if, according to the current publicly available data from the Bureau of the Census:
   (a) The geographic unit formed by combining all ZIP Codes with the same three initial digits contains more than 20,000 people.
   (b) The initial three digits of a ZIP Code for all such geographic units containing 20,000 or fewer people are changed to 000.
3. All elements of dates (except year) for dates directly related to an individual, including birth date, admission date, discharge date, date of death; and all ages over 89 and all elements of dates (including year) indicative of such age, except that such ages and elements may be aggregated into a single category of age 90 or older.
4. Telephone numbers.
5. Facsimile numbers.
6. Electronic mail addresses.
7. Social security numbers.
8. Medical record numbers.
9. Health plan beneficiary numbers.
10. Account numbers.
11. Certificate/license numbers.
12. Vehicle identifiers and serial numbers, including license plate numbers.
13. Device identifiers and serial numbers.
14. Web universal resource locators (URLs).
15. Internet protocol (IP) address numbers.
16. Biometric identifiers, including fingerprints and voiceprints.
17. Full-face photographic images and any comparable images.
18. Any other unique identifying number, characteristic, or code, unless otherwise permitted by the Privacy Rule for re-identification.

Figure 1-1: Individual identifiers specified in the HIPAA privacy rule.

[11].

Given the difficulty of preventing it, we might be tempted to downplay the risk of forming a key from clinical data, relying on the notion that public sources of detailed, personal clinical information must be rare, and so the risk of an attacker forming a reidentification key with this information must be small (indeed the HIPAA Privacy Rule seems to rely on this notion). In reality, this data is easier to come by than we would like to think. On one hand, the pool of people with *authorized* access to portions of our medical information is greater than most of us realize or desire [3]. And on the other hand, unauthorized disclosures happen with alarming frequency. Medical records have been stolen on desktop computers during break-ins [12], stolen on laptops during normal business operations [13, 14, 15], maliciously faxed to a newspaper and radio station [16], sold to tabloids [17], mistakenly (but repeatedly over six months) faxed to a bank [18], left in a filing cabinet donated to a thrift store [19] and found blowing in the wind by the dozens in downtown Lewiston, Idaho [20], and by the thousands in Mesa, Arizona [21].

These are direct disclosures, and anonymization methods aren't designed to prevent them. But if one of these patients has separately participated in a disease registry or clinical trial, inadequate anonymization of those publicly-released datasets may allow an attacker to link the directly disclosed record with sensitive information in the public datasets.

A simplified (and fictional) example of reidentification is shown in Figure 1-2. In this example, heart rate (`HR`), mean arterial pressure (`MAP`), liver enzymes (`AST` and `ALT`), CD-4 cell count (`CD4`), and HIV viral load (`Vir Ld`) have been measured for these patients. The data has been deidentified by removing the patient names. The attacker has access to outside information that for a particular patient gives the values for the first four variables, which are commonly measured clinical parameters. But he does not have access to the last two variables, which relate to the HIV status of the patient. The information in these last two variables is what the attacker seeks by attempting a reidentification. By forming a key from `HR`, `MAP`, `AST`, and `ALT`, the attacker is able to match a name to the last record, and learn sensitive information about that patient. (In reality, the matches would probably be inexact, especially if the outside information consists of measurements made on a different occasion, but for the moment we'll keep it simple.)

Since we don't know what information an attacker might have, our conservative assumption is that he has the full original versions of the anonymized public datasets. Obviously, if an attacker actually possesses the original dataset he wouldn't need to attempt reidentification because he already has all of the information. But if we assess reidentification risk under the assumption that he has all of the original data, we are assured of equal or smaller risk under the more realistic conditions of the attacker possessing only partial information.

Unfortunately, reidentification is not the only disclosure risk undertaken by participants in a clinical trial or disease registry. There is also the risk that an attacker

Name	HR	MAP	AST	ALT	CD4	Vir Ld
	91	101	11	3	1010	0
	86	106	100	150	800	0
	88	110	220	187	200	110000
	72	95	20	10	950	0
	101	99	432	395	400	30000

(a) Data that has been deidentified by removing names.

Name	HR	MAP	AST	ALT	CD4	Vir Ld
	91	101	11	3	1010	0
	86	106	100	150	800	0
	88	110	220	187	200	110000
	72	95	20	10	950	0
	101	99	432	395	400	30000
Jones, Howard	101	99	432	395		

(b) One record has been reidentified using the combination of `HR`, `MAP`, `AST`, and `ALT` as the key. The attacker has learned the value for `CD4` and `Vir Ld` for this patient.

Figure 1-2: Deidentification and reidentification.

may be able to learn *something* new about a person, even if he can't attach a name to an anonymized record. Based on outside information and the anonymized data, the attacker may be able to predict the value of some particular sensitive variable. This predictive disclosure risk is slightly more difficult to assess than reidentification risk because it is not uniquely borne by the participants in the study, and because it represents only a prediction, not concrete information about an individual. We would like an anonymization scheme to provide some kind of reassuring bound on this risk, but there is currently no consensus on how to measure prediction risk, or what a proper bound on it might be.

1.2 Non-technical Overview

Anonymization Methods Most approaches to anonymization arise from the insight that the analyst or statistician working with the anonymized data is looking for fundamentally different information than an attacker who wants to discover sensitive information about an individual. The analyst is looking for associations between variables: are the patients who took a certain drug less susceptible to HIV infection than the ones who took a placebo? Does the effect depend on age, gender, race, comorbidities, concurrent medication or some combination of these? The attacker is looking for individual information: does my employee have HIV? The perfect anonymization would preserve all of the information of interest to the analyst and none of the information of use to an attacker. Since these are different types of information, we have some hope of being able to do this.

One way that researchers have approached this problem has been to slightly change each little piece of information, such as by adding small random values to numeric data or blanking out parts of each record. The hope is that the changes are small enough that trends and associations can continue to be discerned, but large enough that it is difficult to figure out which individual is associated with each record. The problem is that "small enough for the analyst" and "large enough to stop the attacker" may not actually overlap. In fact, it is common for existing methods to be able to satisfy one but not the other. As the amount of information in each record grows, it eventually becomes impossible to simultaneously satisfy both demands. Chapter 2 discusses many of these existing approaches in detail, and how effective they are at satisfying these demands.

My approach to anonymization aligns with the idea previously proposed by others that instead of changing each little piece of the information, we can create new records using the information from the original records. No new record would belong to any one person, and therefore no attacker could look up his employee, neighbor, or professor to learn sensitive information. Yet the new records would contain (we hope) all of the information of interest to the analyst.

What makes this hard is ensuring that the new records actually do contain all the

needed information. In particular, whatever assumptions we make in the construction of the new records will be exactly what the analyst learns about the process being studied. If we think to include certain effects of race, gender, and comorbidities in our HIV vaccine model, the analyst can learn about those from our anonymzied data, but if we leave out effects due to concurrent medications, the analyst can never discover them. What we need is a way to create new records without making consequential assumptions about the structure of the data.

My approach attempts this by finding pieces of information in the old records that are only associated by chance, and randomly rearranging them into new records. In doing so, our challenge is to keep together truly related pieces of information. If we do this properly, no new record will represent a real individual, but all of the analytic information will remain intact.

For example, suppose we have a dataset that collects the variables a, b, c, d, and e for a thousand people. We may find that variables a, b, and c are related in some way, as are variables d and e, but that the set $\{a, b, c\}$ is not probabilistically related to the set $\{d, e\}$. (That is, if we know $\{a, b, c\}$ for one record, that doesn't tell us anything at all about $\{d, e\}$ for that record.) We would therefore create a new record by combining the set $\{a, b, c\}$ from a randomly chosen original record with the set $\{d, e\}$ from a different random record.

This can get more complicated if we also have a variable f that is partially related to the set $\{a, b, c\}$ and partially to the set $\{d, e\}$. In this case, we would take part of the new value of f from the original record that provided $\{a, b, c\}$, and part from the record that provided $\{d, e\}$. For example, if the value of f was 9 in the first record and 3 in the second, and the original data shows that f is twice as strongly related to $\{a, b, c\}$ as to $\{d, e\}$, then we would create a new f that was $9(\frac{2}{3}) + 3(\frac{1}{3}) = 7$.

Figure 1-3 illustrates these ideas using the example in Figure 1-2. Here, let's say that our analysis finds that the pairs {`HR`, `MAP`}, {`AST`, `ALT`}, and {`CD4`, `Vir Ld`} are (mostly) mutually independent, but the components within each pair are strongly related. The only dependence between pairs is that {`CD4`, `Vir Ld`} depends in some weak but complex way on the other two pairs. The data set therefore gets anonymized in a way that shuffles the pairs among records, but preserves the identified dependencies. The color coding in the figure shows the main source of the information for each value in the newly created records. The first anonymized record gets `HR` and `MAP` from the last original record, but `AST` and `ALT` from the second record, and so on. The anonymized values for `CD4` and `Vir Ld` come mainly from the record indicated by the color code, but are not exact copies of the originals because of the interactions with the other variables. The detailed mechanism of generating the new values is explained in Chapters 4 and 5.

The bulk of the work for this approach is in finding independent sets of information, which is never as clean and simple as the examples above. In Chapter 4, I use Singular Value Decomposition to find the independent sets. In reality, Singular Value Decomposition solves the easier problem of finding uncorrelated sets rather than truly

Original Data:

Name	HR	MAP	AST	ALT	CD4	Vir Ld
	91	101	11	3	1010	0
	86	106	100	150	800	0
	88	110	220	187	200	110000
	72	95	20	10	950	0
	101	99	432	395	400	30000

Anonymized Data:

Name	HR	MAP	AST	ALT	CD4	Vir Ld
	101	99	100	150	989	0
	91	101	220	187	875	0
	88	110	11	3	281	99000
	72	95	432	395	760	0
	86	106	20	10	455	33000

Figure 1-3: Simplified example of anonymization. Colors of anonymized values indicate their primary source in the original data.

independent sets, but it is useful enough to demonstrate the idea, and it turns out to work at least as well as existing anonymization methods. In Chapter 5, I refine the idea so that we first find groups of similar records, and then we look for independent sets separately within each group. Since the groups are smaller, it is easier to find truly independent sets of information, and the grouping allows us to find small regions of the data that show relationships between variables where other regions may show no relationship. For this refinement I use the technique of Independent Components Analysis, which is a newer and for my purposes a more powerful technique than Singular Value Decomposition. It turns out to provide a surprisingly faithful anonymization with strong privacy protection. Chapter 6 summarizes my results and discusses open research problems.

Assessment Methods Of course, to decide how faithful or strong an anonymization is, we need some way to measure it. As with other aspects of anonymization, there is no consensus on these measures, and existing measures have various problems that makes them inadequate for my purposes.

I propose three new measures that are more informative and more generally applicable than existing measures. The first measure, *prediction distance*, determines how closely an attacker can predict the contents of an original record, given the anonymized data. It basically finds the distance between an original record and its closest anonymized record, where 'distance' can be defined any way we like. The second measure, *prediction ambiguity*, characterizes how easy it is to find the best match or prediction of an original record. If there are several equally close matches in the anonymized data, this is high ambiguity, and if one stands out from the rest, this is low ambiguity. The third measure, *prediction uncertainty*, characterizes how much of a difference it makes to choose between the best matches. If there are a bunch of equally-likely matches, but they're all nearly identical, this is low uncertainty (and likely poor privacy protection).

Which of these measures is most important to an anonymization depends on the algorithm used and perhaps the particular context it is used in. For an algorithm that removes any deterministic correspondence between original and anonymized data, so that no anonymized record is the direct image of any particular original record, reidentification risk is zero (subject to subtle caveats discussed in Section 2.1), and prediction risk is the dominant threat to privacy. Therefore prediction distance is probably the primary interest, since as long as the best prediction is far enough from the truth, it may not matter as much how easy it is for an attacker to find it.

On the other hand, an algorithm that does preserve a unique, deterministic relationship between original and anonymized records may be more accurately assessed using prediction ambiguity, which is the most relevant in a matching attack. If the best match clearly stands out from the crowd, it may not matter as much how far away that crowd is.

I also propose a new method of measuring analytic utility that is also generally

applicable and measures whether complex structure within the dataset is preserved. This method uses kernel principal components analysis, a powerful tool that finds the most important nonlinear associations between several variables at once, and allows us to specify the class of nonlinear associations we wish to look for. Each kernel principal component represents a particular nonlinear association within that class. The first component finds the largest association and separates its effects from the rest of the information. The second component finds the second largest association, and so on. My proposed measurement method finds these associations within the original data and then looks within the anonymized data to see if those associations have the same distribution as the original data. A statistical test measures whether the distributions are the same or different. And we can see even more subtle effects by plotting the components against each other.

Chapter 2 describes all of these measures in detail, and Chapters 4 and 5 demonstrate their use.

Chapter 2

Background

Survey statisticians have been concerned about the problem of *disclosure control* since at least as early as the 1920's, but the research efforts aimed at preventing large-scale disclosures began in earnest in the 1970's with the increasing availability of both survey information and computer systems [22]. Since then a surprisingly large number of different and creative anonymization methods have been proposed. These methods have come from the international statistical and census communities [23, 24], the computer science community [25], the cryptography community [26, 27], and probably many others whose origins are not as clear.

Though there are books on the subject [28, 29], I know of no comprehensive survey of the field, and I suspect this is in part due to its breadth. There are proposed methods to prevent disclosure from tabulated data [30, 28], queried database systems [31, 26], and detailed, individual-level data termed *microdata* [24]. This dissertation focuses on the problem of preventing disclosure of sensitive information about individuals contained in a set of detailed microdata, released as a whole into the public domain.

Disclosure control is a difficult problem, and even in the narrower domain of microdata anonymization, none of the existing methods has demonstrated clear superiority over the others for general use. Part of the difficulty is in defining with mathematical precision what makes a good method. It is clear that an anonymization method must provide both privacy protection and what is called *analytic utility*. That is, the method must simultaneously prevent the disclosure of sensitive individual information and allow accurate analysis, usually of statistical trends or associations. One of the contributions of this dissertation is the proposal of precise but general definitions and measures of both analytic utility and privacy protection.

Conventions and Assumptions For the rest of this dissertation, I will be assuming that the data is in the form of either a matrix or a set, with each row or record having a constant number of columns or variables. The variables can potentially be continuous, ordinal, categorical, or binary. For many methods, including my

examples of spectral anonymization, categorical attributes can be accommodated by converting variables with q categories to either q or $q-1$ binary variables. I have found that converting q categories to q binary variables is the most helpful. This includes originally binary variables that are more usefully thought of as two-category categorical variables. For example, a `Gender` variable with `M` and `F` values becomes two binary variables: `Gender-M`, which can be true or false, and `Gender-F`, which can be true or false.

The anonymization of these binary variables may produce continuous values for them. One way to interpret these is to normalize them and consider them probabilistic. I have found that a $\{-1, +1\}$ encoding works best during anonymization, with either simple thresholding at zero to produce binary values or a normalized inverse logit function to produce values in the $[0, 1]$ range. These can be interpreted as anonymizing a categorical or binary variable in part by giving a probability distribution over possible values.

I will call the original dataset A, and the anonymized dataset $\tilde{A}$. Their columns or variables will be a_j and $\tilde{a}_j$, and their rows or records will be A_j or $\tilde{A}_j$. In general, I will refer to a column of a matrix M as m or m_j, and a row as M_j. To refer to a particular cell of a matrix, I will use the usual notation M_{ij}.

I will sometimes call a record a *point* without warning, because I find it helpful to visualize the records of a dataset as points in an n-dimensional space, with each dimension corresponding to a variable of the dataset. In fact the main contribution of this dissertation depends on visualizing the dataset this way. I will also use the terms *anonymization*, *perturbation*, and *masking* interchangeably; all have wide use in the literature.

2.1 Assessing Privacy Protection

The threat to privacy from released anonymized data is technically that of breach of confidentiality. More specifically, it is the threat of specific types of disclosure of information that the data collectors have pledged to keep confidential. I will group these under the term *computational disclosure*, and they are different from direct disclosure, which is the malicious or accidental release of the raw information. Computational disclosure risk can be divided into two types [32, 33]. The primary risk is of reidentification disclosure, where an attacker manages to match a particular person's identity to a particular record in the released anonymized dataset using information that the attacker has learned independently about the person. Attempting to make this match is called a *matching attack*, and it is usually assessed under the extremely conservative assumption that the attacker knows the entire original dataset [34]. Under this assumption, the attack becomes a problem of matching each anonymized record with its corresponding original record. If we can prevent the matching attack from succeeding under these conditions, we can prevent it when the attacker

knows far less about the individuals involved. The easiest way to prevent this attack is to perturb the data such that there is no one-to-one correspondence between an anonymized record and an original record. I will illustrate one way this can be done.

The secondary risk is of predictive disclosure, where an attacker manages to predict the approximate, and perhaps partial, content of a target record with the help of the released anonymized dataset. This risk is secondary, first because it is only a prediction, not a proof, but also because it is not uniquely undertaken by subjects of the data. Similar risk is borne by those in the same underlying population but who did not participate. Furthermore, it is difficult to imagine an application where predictive disclosure is not a desired outcome of the study. In medical applications, the primary motivation for releasing anonymized data is to allow analysts to draw valid conclusions regarding associations between the data's variables. We want the anonymized data to preserve associations between smoking and heart attack, for example, or between a particular drug and its side effects. We want physicians to be able to predict disease risk from patients' symptoms and behavior. This unavoidably allows an attacker to make these same predictions.

The *degree* of predictive disclosure risk is of interest to the anonymizer, however, because if the prediction can be made sufficiently accurate, privacy protection is broken. In fact, high predictive accuracy is fundamentally what facilitates a matching attack. We would like the release of anonymized data to pose no greater risk of predictive disclosure to the participants of the study than it does to non-participants. If, for example, we added an amount of random noise to each variable that was small compared to the variance of that variable, an attacker could predict the information for participants with much greater accuracy than for non-participants.

Under certain conditions, the distinction between reidentification disclosure and prediction disclosure can become blurred. For example, synthetic data methods (see below) anonymize by sampling new records from a model built on the original records. For these methods, there is no deterministic relationship between a particular anonymized record and any original record, meaning that no anonymized record is the unique image of any original record. Under these conditions, reidentification loses its meaning, so we consider their reidentification risk as zero.

But it can happen that the model generates an anonymized record that randomly reproduces (or nearly reproduces) an original record. This synthetic record is then at risk for reidentification. But it's not quite that simple, because other synthetic records may match an attacker's target record on the fields known to the attacker, but not on the unknown fields that the attacker is trying to learn. The attacker does not know whether the match he finds faithfully reproduces the unknown information, so this situation is still more of a prediction than a reidentification, where the attacker knows the unknown fields are (at least close to) the true data.

Some datasets, especially binary or categorical datasets, may be distributed such that there are many identical records generated, and the probability of reproducing an original is high. In this case, an attacker may calculate a very high probability

that a prediction is correct, and the situation looks like a reidentification. Existing methods of quantifying risk don't assess these distinctions very well, but I propose methods in Chapter 3 that distinguish between them. For simplicity, I will describe methods that provide no unique deterministic relationship between an anonymized record and any original as having zero reidentification risk, but it will be understood that the above subtleties apply.

The following sections describe existing methods to quantify disclosure risks. My proposed new methods and their relationships to existing ones are described in Chapter 3.

2.1.1 Prior Measures

In previous work, the empirical reidentification rate [34] and k-anonymity [35] have been common anonymity measures. These measures are suboptimal for several reasons detailed below, but more importantly, they are not general enough to assess all of the methods I use in my examples, and they don't estimate predictive disclosure risk. Both measures assume a one-to-one match between original and anonymized records that does not exist with some of our examples, and k-anonymity additionally requires the anonymization to produce groups of k records that are functionally identical. It is also difficult to make a principled decision of what value of these measures represents adequate protection.

The empirical reidentification rate One common measure of reidentification risk is the estimated success of a matching attack against the anonymized data. I will call this estimate the *empirical reidentification rate*. The matching attack has its origins in the record linkage algorithm of Fellegi and Sunter [36]. The record linkage algorithm was originally intended to identify records from different sources that referred to the same individual, such as might be done when two hospitals merge. This method takes a pair of candidate records a and b that might constitute a match, and examines a set of variables from each record in the pair. These variables form a pattern γ of match closeness. The algorithm examines the pattern γ and calculates the likelihood ratio L of the pair (a, b) belonging to a set G of matching records, where

$$L = \frac{\Pr(\gamma|(a,b) \in G)}{\Pr(\gamma|(a,b) \notin G)}. \tag{2.1}$$

In other words, L is the ratio of the probability that the pattern γ was produced by a pair that is a true match vs. the probability that it was produced by a pair that was not a match. If the ratio is above a chosen threshold, the pair is declared a match, and if it is below a second threshold, it is declared a non-match. Pairs with likelihood ratios between the two thresholds are declared possible matches and held for clerical review.

For merging real records from different sources, the dependencies between variable

match probabilities in (2.1) can be difficult to calculate, especially for non-matches. Some have used the iterative Expectation Maximization (EM) algorithm [37] to estimate the probabilities, considering membership in G to be the latent variable [38]. Researchers have used software based on this method to test some of the anonymization algorithms described below [34, 39]. EM can work for records with few variables, but for high-dimensional data the computation time becomes prohibitive.

Other researchers simply assume independence between the variables in γ, and compute a value for L by multiplying likelihood ratios of each variable alone [40]. The probabilities in (2.1) could also be replaced by some sort of distance or difference measures, although the difficulties with dependent data remain [41].

In a matching attack, one of the pair (a, b) is the known, identified record, and the other is a candidate match from an anonymized dataset. In this case, the task of estimating the probabilities in (2.1) is actually much easier because the attacker presumably knows the mechanisms by which the anonymized table is created from the original. We will see that some methods use univariate masking mechanisms that treat each variable independently, and this makes calculating the required probabilities quite simple. Others modify the original data according to known joint probability distributions that we can use directly in (2.1). Since the anonymization parameters are often included with the dataset to enable statistical analysis, this makes the matching attack a feasible approach to reidentification. Even if the parameters were not included, relying on this to thwart the attack would be attempting "security through obscurity", which is a well-known bad idea.

The anonymizer can also estimate the success of a matching attack, and therefore the risk of reidentification, based on the anonymization parameters. The anonymizer may, for example, simply carry out a matching attack between the original and the anonymized database and measure the reidentification rate. Since the original dataset is the best possible source to use in a matching attack, this empirical reidentification rate can be seen as a conservative estimate of the reidentification risk.

There are some subtle issues with the empirical reidentification rate, however. We might think that we would like this rate to be zero for an anonymized medical database, but it's not actually that simple. A zero empirical reidentification rate would mean that the true match is *never* the most probable match, and therefore it leaks information.

The biggest hurdle in mounting a matching attack is for the attacker to identify which of his matches are likely to be correct and which are not. In the original application of record linkage, a threshold on L was used, but how can an attacker have confidence in his chosen threshold? The problem is amplified when, in the interest of computational simplicity, a distance measure is used in place of the likelihood ratio L. The empirical matching rate does not take these difficulties into account, but instead represents an upper bound on attack success, assuming that the attacker can identify the correct matches.

Instead of designing an anonymization scheme that reduces the empirical matching

rate to zero, a better approach might be to reduce the reidentification risk to zero by removing one-to-one correspondence between the original and anonymized dataset. For methods that don't do this, it would still be better to thwart the matching attack by producing some number of records that are roughly equally good candidate matches for any target. The attack would then come up with a set of candidates that are all approximately equally likely to be the target's match, and an attacker essentially has to choose one at random. If one of these is trivially more likely than the others, and a naive attacker chooses it, he will be correct with a probability of $1/k$, if there are k equally good candidates. The empirical reidentification rate does not provide the information needed to assess this property of an anonymization.

k-anonymity The notion that we want at least k functionally indistinguishable matches for any given target carries the name k-anonymity[35]. The value of k is a parameter of the anonymization, although it is difficult to make a principled argument for the choice of any particular value of k.

This is an important and useful measure of anonymization, and was touched on several times in the literature before being made explicit. It was first hinted at by observations in early work that if there are k datasets that could produce the same summary table (such as might be obtained by swapping certain entries in the dataset), then releasing the summary table does not allow one to reconstruct the dataset with confidence greater than $1/k$ [22]. It was touched on later in the context of data releases [9] by observing that unique records are the ones at most risk, and masking methods should seek to reduce the number of unique records. The concept of uniqueness as the main problem in anonymization caught on and was frequently targeted by masking methods and as a measure of privacy protection [42]. It was incorporated as a stopping parameter (for $k = 2$) where the masking of a particular record was rejected unless there was at least one other masked record that could reasonably be the image of the same original [10]. Finally it was recognized that there are different degrees of non-uniqueness, and a record that is indistinguishable from 2 others is at greater risk of reidentification than one indistinguishable from 100 others [8].

Ensuring k-anonymity seems on its face to be an effective guarantee of privacy, but there are some pitfalls in its application. If all k indistinguishable records are literally identical, for example, then matching an original to the group of k gives an attacker the same information as matching to a single record. This would produce an exact predictive disclosure that is functionally equivalent to a reidentification. If the records have identical values for a few variables, then we have an exact prediction for those variables, with some uncertainty for the remaining ones. This is most problematic when the exact prediction is for a sensitive variable. One solution to this is to measure k-ambiguity instead [43], where the records are indistinguishable on the variables used for matching, but not on the sensitive variables. This presupposes the identification of sensitive variables.

Originally, the paradigm for achieving k-anonymity was the combination of global recoding and local suppression for categorical variables (see Section 2.3). These methods globally broaden categories and erase individual cell values so that each record is an exact match to $k-1$ other records, counting an erased cell as an exact match to any value. Thus the patterns `AB*` and `A*C` both match the pattern `ABC`, where the "`*`" represents an erased cell. Similarly, the patterns `ABC` and `ABB` could both be transformed to the pattern `ABD` where the new element `D` represents the broadened category of `B` $\cup$ `C`. The approach can be extended to continuous data using the technique of microaggregation (see below), where k similar records are clustered and replaced with k copies of a single record that is representative of the entire cluster [44].

Besides not assessing predictive disclosure risk, k-anonymity also fails to describe the continuum of reidentification risk, because suppressed values have a probability distribution that can be estimated from the dataset as a whole. A suppression may render two records logically equivalent matches, but one may be much more likely than the other probabilistically. It would be helpful if our privacy metric could express this difference and identify probabilistic risks, but k-anonymity doesn't do it.

Ensuring k-anonymity is therefore insufficient to ensure a low risk of reidentification, and it turns out also to be unnecessary. As we have seen above, we can have k distinguishable records, but if they are all roughly equally good matches to the target record, this can provide good anonymity.

(c,t)-isolation My prediction ambiguity measure (see Section 3.1) is related to and inspired by the theoretic and powerful (c,t)-isolation bound [45]. The (c,t)-isolation bound refers to how well an attacker can predict a complete original record given the masked database and any amount of auxiliary information, which may include portions of the original database. If an attacker can use the masked data and his auxiliary information to compute a record that approximates a record in the original database, he has succeeded in *isolating* that original record. If the distance from the computed record to a particular original record is a factor of c closer than it is to the t^{th} closest record, then he has (c,t)-isolated the original record. The anonymizer's goal is to prevent an attacker from (c,t)-isolating any original record with a probability more than trivially greater than he could if he didn't see the masked dataset. Claiming a (c,t)-isolation bound for a particular algorithm is a strong statement, and these claims can be difficult to prove [45].

In contrast to the above metrics, however, the (c,t)-isolation bound is a property of an anonymization method and not of an anonymized dataset. We can use it to prove the privacy protection of a method, but not to estimate the risk in releasing a given dataset. For example, we can show that releasing something as seemingly safe as a uniformly spaced histogram allows (c,t)-isolation of a point in an exponential distribution for certain values of c and t [45], but we couldn't compute values for c or t for a particular dataset that had been anonymized using this method. It

appears difficult to prove (c, t)-isolation for other than the simplest anonymization methods and data distributions. Bounds have been proven for histograms of uniform density hypercubes in the absence of auxiliary information, and work is in progress for histograms of other distributions [45].

2.2 Assessing Analytic Utility

The best assessment of analytic utility would be to compare the specific statistics of analytic interest on the original vs. the anonymized data. With data intended for general release, it is not practical to compare all possible statistics of interest, and so other measures must be substituted. There is a large set of measures that have been proposed over the years.

Historically, these measures have often been simple difference measures, such as mean Euclidean distance, mean squared difference or mean absolute difference between the masked and corresponding original data points. Measures of whether various statistics are conserved in the masking have also been used [46, 47].

The conserved statistics are commonly up to 2^{nd}-order moments (means, variances, covariances, correlations), and they are often conserved only asymptotically, meaning if you anonymize many datasets, on average the change in the statistic will be near zero [41]. Most methods only conserve their target statistics asymptotically, so I will use the term *conserved* to describe asymptotic conservation. If the method conserves the statistic in each instance, no matter how small the sample, I will say that the statistic is *exactly conserved*. I will say a statistic is *recoverable* if the value of the statistic for the original data may be different than for the masked data, but the original value can be calculated from the masked data. An example of a recoverable statistic would be covariances that are increased by a known amount with a particular masking method.

2.3 Existing Anonymization Methods

This section is not an exhaustive survey of microdata anonymization, although it is more comprehensive than any survey I have found. It covers all of the different classes of methods of which I am aware — Additive Noise, Local Suppression and Global Recoding, Data Swapping, Microaggregation, and Data Synthesis. Within each class I give several variations on the theme, enough to give an idea of the scope of strengths and weaknesses available within that class. I've included all of the historically important and common methods of which I'm aware, but I'm sure there are more variations waiting to be discovered in unsearched corners of the literature.

In discussing the merits of each method, I use the measures of privacy protection and analytic utility with which they were assessed by their authors or other published analyses. As described above, these measures vary widely among researchers, and not

all of these measures are directly comparable, but I try to summarize what is known about each method despite these differences.

2.3.1 Additive Noise

Adding random noise to the data is perhaps the most obvious perturbation method, and was first investigated over two decades ago [9, 48]. This method adds a random matrix R to A, so that

$$\tilde{A} = A + R. \tag{2.2}$$

Uncorrelated noise The simplest approach is to construct R with independent columns and variance proportional to that of the original column, so that

$$R_{ij} \sim N(0, b\,\mathrm{var}(a_j)), \tag{2.3}$$

where b is the constant of proportionality that controls the overall level of masking. This preserves variable means, and if b is reported, the variances can be recovered with $\mathrm{var}(a_j) = \mathrm{var}(\tilde{a}_j)/(1+b)$. The covariance matrix is also recoverable, since the off-diagonal elements are preserved, and the diagonal elements are the variances. The univariate distributions are not recoverable, however, since we are adding Gaussian noise to an arbitrary original distribution, but they can be approximated using Expectation Maximization methods [49, 50]. Some statistical models such as decision trees can also be successfully constructed from $\tilde{A}$ [50], but in general, higher order effects are not recoverable.

Empirical reidentification rates can be as high as 100% for datasets with 20-30 variables and 85% for only 4-6 variables [9]. Beyond this, the noise can be largely stripped from $\tilde{A}$ using singular value decomposition (SVD) [51]. Therefore this method is unusable in practice, but it can be a benchmark for others.

Correlated noise The first improvement we might make is to use correlated noise where each row R_j of R is drawn from

$$R_j \sim N(0, b\Sigma), \tag{2.4}$$

where $\Sigma = \mathrm{cov}(A)$ is the original covariance matrix. The covariance matrix is recoverable from $\tilde{A}$ by $\Sigma = \mathrm{cov}(\tilde{A})/(1+b)$ [52]. With extra effort, an analyst can perform regression analysis on $\tilde{A}$ [48]. The univariate distributions and higher order effects are unrecoverable.

Alternatively, we can release a corrected matrix $\tilde{A}^c$ with the same mean and covariance as A by converting each row $\tilde{A}_j$ into the row $\tilde{A}^c_j$ by

$$\tilde{A}^c_j = \frac{1}{d_1}\tilde{A}_j + \frac{d_2}{d_1}\mu$$

where μ is the mean row vector of A, $d_1 = \sqrt{1+b}$, and $d_2 = d_1 - 1$ [52, 53].

The privacy protection is stronger than for uncorrelated noise, with resistance to SVD noise stripping, although the protection remains insufficient for medical data. To protect against a matching attack, the magnitude b of the additive correlated noise must grow exponentially with the number of uncorrelated columns, which becomes prohibitive on large databases. A test on a file of 1080 records with 13 masked variables reidentified 31% after masking with $b = 0.1$ and 79% after masking with $b = 0.05$ [39]. The same test on a file of 59,000 records and 8 variables masked with $b = 0.01$ reidentified 6% of the records. It appears from the report that 4% of the records probably represented unpredictably correct matches,and so were actually protected by being members of sets of $k \geq 5$ equally likely matches. This portion probably accounts for more than 4%, but the report doesn't contain enough detail to discern the fraction of matches that were unpredictably correct. Less than 2% were described as "clearly true matches" and were probably predictably correct and vulnerable to reidentification. But even a fraction of a percent would be more than we would like for medical record anonymization.

Correlated noise after normal transformation We can improve this method further and conserve univariate distributions by taking advantage of the fact that the sum of two normal random variables is itself a normal random variable [54, 55, 10]. We first transform a column a to a normal distribution using the CDF P_a of a and the cumulative standard normal distribution Φ, by

$$a^{\mathrm{N}} = \Phi^{-1}(P_a(a)), \tag{2.5}$$

where a^{N} is the new normally-distributed column, and the collection of these columns form the new matrix A^{N}. Categorical variables are transformed using a similar but slightly more complicated procedure that first transforms them to pseudo-uniform random variables and then to normal random variables.

To the new matrix A^{N} we can add correlated normal random noise where

$$R_{ij} \sim N(0, b\,\mathrm{cov}(A^{\mathrm{N}})) \tag{2.6}$$

as before but, unlike previously, where masking resulted in a matrix of unknown distribution, the result $\tilde{A}^{\mathrm{N}} = A^{\mathrm{N}} + R$ in this case remains a multivariate normal distribution. We can now reverse the transformation to the original scale giving

$$\tilde{a} = P_a^{-1}(\Phi(\tilde{a^{\mathrm{N}}})). \tag{2.7}$$

The univariate distribution of a is therefore conserved, and the original covariance matrix Σ is recoverable with $\Sigma = \mathrm{cov}(\tilde{A})/(b+1)$ [55]. Consistent estimators of some higher-order moments can be calculated, although in general they have high variance [10].

I am unaware of any privacy protection tests for this method, although it has been tested in combination with row resampling (see below).

Row resampling Further protection can be provided by examining each row $\tilde{A}_i$ before including it in the dataset. If $\tilde{A}_i$ is too close to A_i, or if there are no other masked rows that would be an equivalent match to A_i, we can reject it, resample the noise row R_i, and reconstruct the masked row $\tilde{A}_i$. This is similar to rejecting rows R_i that are too close to zero. Row resampling could affect the conservation of both the univariate distributions and the covariance matrix if done too frequently, since the rows of noise are no longer from a strictly normal distribution.

Row resampling improves privacy protection by removing ineffective masking. Compared with simple correlated noise, correlated noise after normal transformation and row resampling reduced empirical reidentification rates by roughly an order of magnitude (0.6% vs. 0.07% for 1000 records with 7 variables and the severe masking of $b = 0.5$) [10, 48]. Although the experiment didn't investigate the effect of the row resampling alone, it is a reasonable conjecture that much of the reduction came from rejecting ineffective masking.

Mixture of normals An approach that is similar in spirit to row resampling but more systematic uses a mixture of normal distributions [56]. Instead of constructing R from a single, broad, multivariate normal distribution and rejecting the noise samples near zero, we construct a mixture of narrow distributions spread out over a broad range with means bounded away from zero, ensuring that most perturbations are not too small. Thus we sample the rows R_j with

$$R_j \sim \sum_i \omega_i N(\mu_i, c_i \operatorname{cov}(A)) \tag{2.8}$$

where μ_i is a vector of noise column means randomly generated but constrained away from zero, and the mixture weights ω_i sum to one. The resulting noise matrix R will have an overall covariance matrix $b \operatorname{cov}(A)$, with b determined nonlinearly by the mixture means μ_i, variances c_i, and weights ω_i. A nonlinear optimizer may be used to find a set of parameters that produces a given value of b and fulfills other necessary constraints.

One experiment found that the mixture method of (2.8) reduced the empirical reidentification rate by roughly a factor of 6 over the simple correlated noise method of (2.4) (6% reidentification vs. 33% for 1023 records with 7 variables and masking constant $b = 0.27$), while conserving many direct statistics of the data. The mean absolute variation was very high, however ($\text{mean}_{i,j}[|A_{ij} - \tilde{A}_{ij}|/|A_{ij}|] = 45.3$), since only larger perturbations were generated.

We can save lots of computation, produce the same effect, and eliminate the

nonlinear solver by producing the directly with colored white noise, with

$$R_j = (b^{\frac{1}{2}}\Sigma^{\frac{1}{2}}s)^{\mathrm{T}}, \tag{2.9}$$

where

$$s \sim \sum_i \omega_i N(\mu_i, \sigma^2), \tag{2.10}$$

$$\Sigma = \mathrm{cov}(A), \tag{2.11}$$

and σ^2 is a common variance for all mixture components [39]. The parameters μ_i and σ^2 are randomly chosen and appropriately scaled so that the elements of s have mean 0 and variance 1. The notation $\Sigma^{\frac{1}{2}}$ denotes the square root of the matrix Σ, which is defined as the matrix X that satisfies the equation $XX^{\mathrm{T}} = \Sigma$.

Data-dependent noise The latest refinement of the additive noise idea is the notion that we can add noise that depends on the local data density at a particular point [45]. Where the points are sparse, we add larger random values to better mask them, but where they are dense, much smaller values will do. The earlier normal-transform method produces a similar effect by effectively spreading out dense clusters and condensing sparse clusters before masking, although points near the median are condensed more than points farther away. One experiment has demonstrated that Gaussian mixture parameters that produced original synthetic data are recoverable under this masking method [27].

Analysis of the privacy protection provided by data-dependent noise has not yet been made, although we would expect it to be an improvement over previous methods.

2.3.2 Local Suppression and Global Recoding

These two methods are historically common, and often used together [5, 1, 57, 8, 58], despite the fact that they offer poor privacy protection at a cost of high information loss.

Global Recoding We can reduce record uniqueness by replacing detailed categories with more general categories [5]. We might, for example, replace a 5-digit zip code with the first four digits, taking advantage of the hierarchical nature of zip codes. Or we might replace date of birth with year of birth. If we generalize to sufficiently broad categories for each variable, we can achieve k-anonymity for a categorical table. We could similarly anonymize continuous data by discretizing and then generalizing [59]. The analytic price we pay for this is high, however, as we lose detail for an entire variable each time we recode. A less drastic measure might be to recode only locally, for particular rows, but this imposes a much greater burden on analysis, since masked

records now come from many different sample spaces, and this is difficult to take into account during analysis [60].

Local Suppression If, after global recoding, there are just a few records at risk of reidentification, we can suppress the particular values that produce that risk. Completely suppressing a cell is a drastic step, however, and it can greatly distort analytic properties [57]. For example, since cells are selected for suppression because they are unique or extreme in some way, deleting them will affect statistics as simple as a univariate mean.

We might try to suppress cells in a pattern that minimizes the distortion [5, 35], but not only is this NP-hard [61], it is insecure, as it leaks information that can allow suppressed cells to be reconstructed [62]. Consider, for example, a table of 100 rows, with a gender category of 90 `Male` values and 10 suppressed values. If we know the pattern is a minimal suppression, then we can infer that all 10 suppressed values are `Female`. To see this, assume for contradiction that the suppression is minimal and the suppressed value on a particular record r is `Male`. Un-suppressing the `Male` value in r would not change the anonymity of the table, because any record that matched r before revealing the `Male` value will still match it after, since all records have the value `Male` in that variable. Therefore the pattern was not a minimal suppression, contradicting the assumption. Hence all suppressed values are `Female`. More complex reconstruction is possible if the attacker has prior knowledge of the data — certainly a matching attack would succeed a substantial fraction of the time. Additionally, suppressed values can be approximately reconstructed from information in non-suppressed variables, if dependent sets of variables are not all suppressed at once.

Randomized Suppression To my knowledge, an algorithm that randomly suppresses cells until k-anonymity is achieved has not been proposed or analyzed in the literature. Such an algorithm could eliminate the risk of reconstruction due to pattern-based information leaks and it would reduce the distortion of the data due to suppression. The downside of the method is that in order to sufficiently anonymize the data, it would have to suppress so many cells that it would greatly decrease the statistical power of any analysis. It would also remain vulnerable to approximately reconstructing the suppressed cells using information in remaining variables. These may be two of the reasons why nobody has seriously proposed it.

2.3.3 Data Swapping

Data swapping is a perturbation method that was not originally intended to mask data, but rather to prove that subjects were protected when summary tables were released [22, 63]. The idea was that if there were multiple equivalent databases that could be produced by swapping database cells while keeping the chosen summary

counts constant, then there were multiple possibilities for which records produced the counts of each variable, and individual data would not be leaked by publishing those counts.

Despite its original abstract intentions, data swapping is a competitive practical method for masking data. Since choosing swaps that exactly conserve our desired statistics is often intractable, we generally look for swaps that only approximately conserve them. One way to do this for categorical data is to use the original summary tables as probability distributions and sample new elements from them to produce a masked database [64].

The privacy protection of the data swapping class of anonymization methods depends in general on the types and volume of swaps that are made. If the swapped-in values look random compared to the swapped out values, then this could be effective in preventing a matching attack, given enough swaps. Certainly most elements of most records would have to be perturbed in order to prevent a matching attack, as leaving even a few variables intact in each record can be sufficient to allow reidentification [65], even if they are not the same variables in each record. Moreover, if the swapped-in values are too close to the swapped-out values, the results would be similar to a noise-addition algorithm with small noise values, and would be easily overcome.

Preserving k core variables We can make the problem easier by loosening the constraints such that we only preserve the value of k core variables for each record [66]. This requires swapping only between records that match on the values of those core variables. This method preserves all interactions (up to order $k + 1$) that use only one non-core variable. It does not preserve statistics that use more than one non-core variable.

This is the method used to protect the 1990 U.S. Census 100 percent detail data [67, 66]. This data was grouped by address block and a small number of household records were matched between blocks on some chosen core variables. All non-core variables were then swapped between these records. Smaller-sized blocks were masked more often than larger blocks due to their increased risk of reidentification.

For the 2000 U.S. Census, the procedure was changed to mask only unique household records in small blocks, and then only with some probability and if the household had not been masked using suppression followed by imputation. [68].

This method cannot and was not intended to protect against a matching attack; the administrators relied on the fact that swapping a small fraction of records "has the nice quality of removing any 100% assurance that a given record belongs to a given household" [68].

Similar-value swapping An approach to conserve all statistics would be to swap only similar values [69]. By swapping values that are within a certain percentile of each other, we can hope to ignore the need to meet any other constraint. If the variables are uniformly distributed, then we can choose the matching percentile to

guarantee that no bivariate correlation is reduced by more than a specified constant, and additionally that the 95% Confidence Interval (CI) for any large subset mean is within specified bounds [69]. One experiment found that upper-bound distortion goals were met in most cases despite the fact that the data was not uniformly distributed [69].

The empirical reidentification rate in that experiment was 13% for a maximum reduction of 0.975 in correlation and 95% CI of $\pm 0.02\%$ on large subset means. This was roughly equivalent protection to additive correlated noise that caused about the same correlation distortion (although opposite in sign — correlated noise tends to increase the original correlation).

Ranked column swapping A similar approach is to randomly divide the data into several subsets, rank-order the variables one at a time, and swap entire ordered columns between subsets [70]. This appears to be roughly equivalent to similar-value swapping under a binomial distribution over the difference in rank. Beyond preserving univariate statistics, its statistical guarantees are unclear. It does not protect well against matching attacks, with one experiment finding 61% reidentification of 1000 records with 2 variables, and 100% reidentification of 1000 records with 6 variables [71].

Swapping within clusters In a variation that blurs the line between swapping and microaggregation (see below) we might try to cluster the data into small groups and then swap within the group [72]. We are not aware of any published analysis of the data distortion or privacy protection produced by this method. We expect that the multivariate distortion would probably be reduced compared to similar-value swapping, since the swapping would be done among individuals that are similar on the whole instead of on one variable alone, and we expect that the privacy protection should be at least as good.

Synthetic-guided swapping In variation that blurs the line between data swapping and synthetic data (see below), we can generate synthetic data and then replace the values in each column with the equally-ranked values from the original data. If we use a multivariate normal synthetic distribution generated with the means and covariances of the original data, it will exactly conserve univariate distributions and asymptotically conserve bivariate rank correlations [71]. As with all synthetic methods, this one reduces practical reidentification risk to zero, since there is no unique deterministic association between original and anonymized records.

2.3.4 Microaggregation

Microaggregation groups k similar data points together and replaces them all with a single point that is somehow representative of the group. Depending on the natural

clustering of the data, allowing groups of k to $2k-1$ data points may preserve information a little better than holding strictly to groups of size k [73]. Replacements such as the mean [74] and the group range [72] have been proposed. Since each point in the group is made identical to every other point in the group, this achieves at least k-anonymity. Microaggregation was originally intended for continuous variables, but it can be extended to work with ordinal or categorical data [44].

Microaggregation can be thought of as reducing the effective size of the dataset while maintaining all of its statistical characteristics, subject only to the reduction in size. This reduction is drastic, however, with a significant effect on the power of most statistical tests performed on the dataset and the variance of the computed statistics. To achieve k-anonymity, one must essentially discard $\frac{k-1}{k}$ of the information in the dataset. Intuitively, microaggregation is a way of attempting to represent the data using roughly $1/k$ of the information, but in a more systematic way than a $1/k$ random sampling.

The challenge with masking by microaggregation is that grouping to maximize some measure of within-group similarity is NP-hard [75]. Therefore all microaggregation proposals in the literature represent attempts to approximate the optimal grouping in a reasonable time. Unfortunately, we will see that some of these attempts significantly degrade the privacy protection.

Single variable microaggregation The simplest approach would be to group the data based on a single variable and replace each record in the group with their mean. Optimal grouping on a single variable can be done in polynomial time by reduction to a shortest-path problem [76]. This is efficient and achieves k-anonymity, but it also induces correlation where there was none (such as between two variables that were both originally uncorrelated with the grouping variable), and it distorts regression analyses when the dependent variable was originally uncorrelated with the grouping variable [9].

Univariate microaggregation Another simple approach is univariate microaggregations, where each column is aggregated and replaced independently of the others [74]. This breaks the k-anonymity that microaggregation is meant to produce, and naive reidentification rates were as high as 97% for 1080 records with 13 variables aggregated with k as high as 10 [77]. We also suspect that this masking may be at least partially reversible using SVD filtering.

Single variable projection A slightly more sophisticated approach would be to project multivariate data onto a single variable, such as the first principal component, and perform single-variable microaggregation with the projected variable [73]. This achieves k-anonymity, and is probably an improvement over plain single-variable microaggregation, but it doesn't preserve even low order statistics [77].

Multivariate microaggregation A much better approach is to define a similarity measure that uses the complete data record, and to aggregate records that are the most similar. In this direction, a normalized distance vector would be an obvious choice for a similarity measure. One algorithm [78] finds the two most distant points in the cloud of data, then groups the $k-1$ nearest points with each of them. Using these two groups as starting clusters, it then runs Ward's clustering algorithm on the data , with the exception that it never joins groups that both have more than k members. When the clustering is finished, it recurses on groups with more than $2k$ members [73]. This is time and space intensive, but there are few competing multivariate microaggregation solutions. One evaluation measured an empirical reidentification rate of 6.9% for a dataset of 77,000 rows with 7 variables and $k = 5$ [79]. Another algorithm that may have slightly better information preserving properties is to find the same two starting clusters, remove them, and recurse on the rest of the data [73, 44].

Fuzzy clustering Instead of restricting membership to a single cluster, we can allow each record to have fuzzy membership in multiple clusters. We then choose the record's replacement with probability equal to the membership function for each cluster [80]. The privacy protection and analytic utility haven't yet been well analyzed for this method, but I conjecture that the distortion may be greater than with crisp clustering. I also suspect that fuzzy clustering may provide better protection, since the attacker must now deal with a probability distribution for the cluster membership of an original record.

Randomized Clustering We could use a randomized optimization algorithm such as Genetic Programming to find approximately the best clusters [73]. One experiment found this to provide comparable analytic utility to deterministic clustering, and reduced running time for high dimensional data [73].

Graph Theoretic Methods We can apply graph theory to come up with some good clusters in a reasonable time, where we consider the dataset as a graph with nodes corresponding to data points and edges weighted with a function of the similarity between points. We might, for example, build a minimum spanning tree over the nodes and then remove distance-weighted edges in decreasing order of length, leaving in place edges whose removal would produce a group smaller than k, until all groups are smaller than $2k-1$ [81]. Or we could start with a triangular mesh over the nodes and do the same thing, which might give us a better solution because it starts with a denser graph [82]. Good analysis on the information preserving properties of these methods and their relative merits has not yet been done.

We could also use a graph-theoretic algorithm to build up clusters by making optimum pairwise matches between nodes [72]. Optimal matching on groups of two

is possible in $O(n^3)$ time [83], but is NP-hard for groups of three (see the Maximum Triangle Packing problem [84]). Using recursion, we could approximate optimal matching for groups of 2^m. This takes a long time even for paired matches, so for large tables we might try tracking only the top l nearest neighbors for each row instead of keeping a complete table of distances. The value of l needs to be high enough to get a complete match-up; a value of $l = 23$ was required in one instance for a complete match-up of 17,000 rows [72].

2.3.5 Data Synthesis

The tradeoff between analytic utility and privacy protection is usually presented as inevitable (e.g. [85]). But it is not inevitable. Consider what would happen if instead of releasing our data, we perturb it to match the data of a *different* sample from the same population. The conclusions we could draw from the perturbed data would be equally valid as those from the original data, but the original participants would have zero reidentification risk and prediction risk as low as non-participants. This is the underlying insight to releasing synthetic data [86]. We only have to tweak this procedure so that the released data belongs to subjects that would exist in an appropriate sample from an infinite population, but may not exist in the real world. The synthetic data method is thus in a prime position to provide theoretically perfect computational disclosure control. The limits are those of model accuracy, not a tradeoff between utility and privacy. If we can come up way to provide an accurate synthetic sample, this can provide perfect analytic utility (in the sense that all statistical analyses are as valid as the original data), and perfect privacy protection (in the sense that all computational disclosure risk is the same as non-participants).

To construct our synthetic data, we begin by viewing the original data as a sample from a continuous underlying distribution and attempt to draw an entirely different sample from that same distribution [86]. The challenge lies in accurately modeling the underlying distribution — indeed, this is the central problem of the field of machine learning. All of the work in this area therefore attempts to efficiently provide a reasonably accurate model. The first effort along these lines used the summary tables themselves as conditional probability models for sampling categorical data [64].

The idea of sampling from a model raises the important question of why we wouldn't release the model instead of the samples. The samples cannot contain any more information than the model, after all, so the most an analyst can hope for is to recreate the model from the data [87]. On the one hand, this is good, because it means the subject's data is as protected as if we had released only the model. But on the other hand, is there any benefit to releasing synthetic samples instead of the model itself? Moreover, if a model is imperfect, its synthetic data is useless to the analyst looking for effects that the model doesn't reproduce [87].

Let's first address the issue of an imperfect model. Preserving as much information as possible is one of the two great challenges of computational disclosure control, and

the challenge applies to all of the masking methods described here. Indeed, we've seen that the drawback to many methods is that they can preserve up to 2^{nd}order statistics but nothing beyond that, making them useless for many applications. Other methods don't even preserve 2^{nd}order statistics. The major advantage of the synthetic data methods is that they uncouple analytic utility from privacy protection. We are therefore free to produce the most accurate model we can with virtually no impact on the computational disclosure risk. The fact that the model may not be perfect is small potatoes compared to the analytic hits we are forced to take at the hands of other methods. So while this is certainly a limitation for synthetic data methods, it is less of a limitation than for other existing methods.

So why wouldn't we simply release the model instead of sampling from it? If our model were simple enough, we should do just that, and for some of the methods reviewed below, this would be appropriate and useful. A complicated model, however, may be of no use whatsoever to an analyst, because the information (and limitations) of interest may not be easy to identify or extract. But if we release a set of samples from the generative model, this easily and efficiently transfers the complex information to the analyst in a way that allows unrestricted analysis. The example in Chapter 5 illustrates this point, creating a type of model that would be difficult to describe analytically or draw conclusions from directly.

An interesting irony is that while the argument that one could simply release the model is usually applied to the synthetic anonymization method, it is actually more applicable to any of the other methods that preserve a small number of analytic results such as univariate distributions and covariances. For those methods, the analyst would lose nothing if the anonymizer provided only those statistics. It is only with methods like synthetic methods, that can preserve more complex statistics, that the analyst gains any real benefit from examining the anonymized data.

Historically, when anonymization meant perturbing a small percentage of the data, the analyst could gain from access to the anonymized data, but in today's environment when we must perturb all of the data to prevent a matching attack, this is no longer true in general.

Privacy protection issues are roughly the same for all synthetic methods. The reidentification risk is zero, but there may be undesirable prediction risk, especially for outliers [42]. Since these issues don't vary between synthetic anonymization methods, my descriptions will focus on accuracy issues.

Summary tables The earliest idea for releasing synthetic data was to use summary tables of original categorical data as a simple probability distribution for generating synthetic data. [64]. The summary tables can be used as conditional probability distributions from which samples can be drawn. This can be efficient for small, 2^{nd} order tables, but becomes intractable very quickly if we try to account for higher-order interactions. And, of course, it doesn't work well for continuous data.

Univariate pdfs For continuous data, we would like to fit a probability density function (pdf) and sample from that. Early work used univariate distributions that were fit with piecewise uniform distributions if an appropriate parametric pdf couldn't be found [86]. This was a stepping stone to current methods, but obviously inadequate for investigating multivariate effects.

Preserving generalized k^{th} moments An early attempt to handle multivariate effects aimed at conserving generalized k^{th} degree moments, which are statistics of the form

$$\sum_j A_{1j}^{s_1} A_{2j}^{s_2} \cdots A_{mj}^{s_m},$$

where s_i are nonnegative integers that sum to at most k [88]. Preserving the generalized 2^{nd}moments will preserve, for example, the means, variances and covariances of the data. If we can preserve k^{th} order statistics for arbitrarily high k, we can theoretically meet any constraint on analytic utility (although such constraints may increase disclosure risk).

We therefore seek synthetic replacements in a table of m variables that meet $\binom{m+k}{k} - 1$ constraints to preserve generalized k^{th} moments. If there are more constraints than elements in the table (which is not hard to do given the exponential growth with k), then the original table is the unique solution to the set of constraints, and no replacements can be made that exactly meet the constraints. The constraints form a system of nonlinear equations of maximum degree k, and when an exact solution exists, finding it is an NP-complete problem [88]. A polynomial-time approximation algorithm has been proposed for $k = 2$ that finds random solutions meeting all but one constraint exactly and then uses quadratic programming to maximize the final constraint [88]. Finding even approximate solutions can be difficult for higher k.

We are not aware of any assessments of the privacy protection of this method, but given the difficulties in extending it to larger values of k, it is not a good candidate for practical use.

Latin Hypercube Sampling To model multivariate interactions, we can preserve the original rank correlations using the technique of Latin Hypercube Sampling (LHS) with rank correlation refinement [89]. We start with a matrix R in which each column is a random permutation of $i/(m+1)$, $(i = 1, 2, \ldots, m)$. We generate a properly correlated random matrix S by

$$S = \Sigma_A^{\frac{1}{2}} \Sigma_R^{-\frac{1}{2}} R. \tag{2.12}$$

where Σ_A is the covariance matrix of A, and $Sigma_R$ similarly.

We then produce R^* by rearranging the values in R for each column so that they appear in the same rank order as the values in S. Note that all of its elements are in

the interval $[0, 1]$. We find the masked matrix $\tilde{A}$ by looking up the preimage of R^* in the appropriate CDFs, or

$$\tilde{a} = P_a^{-1}(r^*) \tag{2.13}$$

This preserves univariate distributions and rank correlations, but no higher order interactions. This looks very similar, if not identical to, synthetic-guided swapping (see above).

The CDFs of the columns of A can be fitted as parametric distributions or non-parametric empirical distributions. The usual caveats apply to the decision to parametrically fit a distribution, which would be more accurate if we correctly identify the distribution family, or to non-parametrically fit an empirical distribution, which would be more accurate if if we can't correctly identify the family. The tails of the empirical distribution can be smoothed to provide better protection against reidentification of extreme values.

Copulas A different way to produce essentially the same result is to use copulas [90], whereby variable values are replaced with equally ranked values drawn from a univariate standard normal distribution. That is, we produce a normally distributed column b in matrix B from column a in matrix A by

$$b = \Phi^{-1}(P_a(a)), \tag{2.14}$$

where Φ is the cumulative normal standard distribution.

We then calculate the Spearman rank order correlation matrix R of the data A, and the product moment correlation matrix Q by

$$Q = 2 \sin\left(\frac{\pi R}{6}\right). \tag{2.15}$$

We can then use Q as the covariance matrix in a multivariate normal distribution, where

$$\tilde{B} \sim N(\mu_B, Q), \tag{2.16}$$

and μ_B is the vector of column means of B. From $\tilde{B}$ we use the reverse transformation to return to the original scale of A by

$$\tilde{a} = P_a(\Phi(\tilde{b})). \tag{2.17}$$

This procedure transforms the arbitrary distribution of A into the multivariate normal distribution of B that can be described parametrically, and from which synthetic samples can be drawn efficiently. The rank order correlation matrix of A, B, $\tilde{B}$, and $\tilde{A}$ are all approximately the same. This method preserves univariate distributions and rank-order correlations, but no higher order interactions.

Maximum entropy simulation We can fit a joint distribution that preserves an arbitrary number of user-defined constraints (up to the number of records) using maximum entropy simulation, and sample from that distribution using Markov Chain Monte Carlo methods [91]. This is an attractive approach, because it allows the simulated data to conserve any properties of interest, but it is both labor and CPU intensive. For n constraints, fitting the distribution requires solving an n-dimensional nonlinear optimization problem, each step of which requires a numeric integration. To maintain only means and variances of a 2-variable table, this requires $n = 6$ dimensions for the optimization problem, and adding further constraints such as interval-specific means or moments further increases the complexity. For high-dimensional data this becomes intractable. One way to reduce the computational burden is to fit a regression or other model to the data, sample only the independent variables using maximum entropy simulation, and then calculate the dependent variables with the model. This certainly works to decrease the computational burden, but again is labor intensive [91].

Multiple imputation Independent of our model construction method, if we are uncertain about our fitted probability distribution, we can release data that reflects this uncertainty. Instead of a single dataset of the same size as the original, we can sample m synthetic sets of this size, in a technique named multiple imputation [92, 93]. Multiple imputation is usually used to fill in occasionally missing data in a dataset, and in that application there is usually no need to go much beyond $m = 5$ [94]. But in multiple imputation of full datasets it turns out that this rule of thumb does not apply, and values for m in the hundreds are often needed to get an accurate estimate of the model's uncertainty [95]. As we would desire and expect, the extra uncertainty in the model is reflected in increased standard error of the various statistics calculated on the multiple samples, taking into account the between-sample variability [94].

If we are uncertain about even the appropriate family for the distribution, we can incorporate that uncertainty by constructing hierarchical mixtures of models, and sampling from that mixture [96].

Partially synthetic data In the interest of efficiency, we might try to generate synthetic values for only those variables usable as keys, and then only for records with fewer than k similar neighbors, building a Bayesian model that generates the key values given the non-key values over some local neighborhood [97]. The idea of multiple local models is promising for generating synthetic data, but limiting the protection to only the highest risk records and then only on the most likely key variables would be ineffective at preventing a matching attack. If we relax these limitations and include all records and all variables, we find we are talking about building a full Bayesian model over all records, which, as we have said, is intractable.

Clustered synthetic data A promising idea is to first cluster the data as we would do for microaggregation, and then instead of replacing all elements of the group with a single representative value, we build a separate model for each group and sample from that. This should give us the benefits of of both microaggregation and synthetic data. Preserving the positions of the clusters should preserve much of the high-level and nonlinear structure of the data, modeling each cluster should preserve much of the substructure. We would hope that modeling the individual clusters would be simpler than modeling the data as a whole.

The only known attempt at this found the clusters in a manner similar to the multivariate microaggregation described above [73], where each cluster was found by taking the $k-1$ nearest neighbors of a randomly chosen point and removing them from the data. The models were uniform distributions fit along the principal components of the clusters.

Experiments on some standard datasets showed that this approximately preserved 2^{nd}order statistics, and a k-nearest neighbor classifier worked roughly as well on the masked data as on the original data (sometimes better, probably due to the anonymization removing noise). It shouldn't surprise us that a k-nearest neighbor classifier worked well for an anonymization method based on preserving the k nearest neighbors of random points, but the overall strategy of synthetic data drawn from models of clusters is a promising one that I will extend in Chapter 5.

Chapter 3

Proposed Assessment Measures

3.1 Disclosure Risk

Prior assessment measures for disclosure risk are unsatisfying, because they are either unnecessary, insufficient, or not applicable in many cases (Section 2.1). I propose the new measures *prediction distance*, *prediction ambiguity*, and *prediction uncertainty* to better quantify disclosure risk. Each of these measures applies to a single original data point given the anonymized dataset. We can calculate the measures for each original point and compare distributions over all points to establish the adequacy of privacy protection.

Prediction distance Prediction distance $d(A_j, \tilde{A})$ is the distance from a particular original point A_j to the closest anonymized point in $\tilde{A}$, using some distance measure s. It represents the closest an attacker can get to predicting the values of an original data point. To allow scale- and dimensionality-invariant measures, s can be calculated on standardized data and normalized by the number of dimensions, such as

$$s(x, y) = \left[\frac{1}{m} \sum_{i=1}^{m} (x_i - y_i)^2 \right]^{\frac{1}{2}}, \tag{3.1}$$

where m is the number of dimensions of x and y, and x_i refers to the i^{th} variable in record x. The prediction distance of an original record A_j would then be

$$d(A_j, \tilde{A}) = \min_{\theta} s(A_j, \tilde{A}_\theta)$$

This and the following measures assume a *rational attacker*, meaning that he will chose an anonymized data point as the best prediction, constructing one that is not part of the anonymized database.

In Section 2.1 I discussed some subtle distinctions between prediction disclosure and reidentification disclosure, and mentioned the possibility that a synthetic method

could generate new records that reproduced originals. This situation would give a prediction distance distribution with some probability density at zero distance, indicating probably unacceptable risk, depending on the reference standard we use to decide what is acceptable.

Prediction ambiguity Prediction ambiguity $c(A_j, \tilde{A}, s, k)$ gives the relative distance from the record A_j to the nearest vs. the k^{th}-nearest record in the set $\tilde{A}$. Formally,

$$c(A_j, \tilde{A}, s, k) = \frac{s(A_j, \tilde{A}_{(1)})}{s(A_j, \tilde{A}_{(k)})}$$

where $\tilde{A}_{(i)}$ is the i^{th}-closest record in $\tilde{A}$ to A_j, under the distance measure s.

An ambiguity of zero means A_j was an exact match to some record in $\tilde{A}$, and an ambiguity of one means that the best match from $\tilde{A}$ was a tie among k records. Note that these k records are not necessarily identical, only equidistant from A_j. Intuitively, ambiguity represents the difficulty in selecting the best match from among the k top candidates. Low ambiguity suggests a prominent best match, high ambiguity suggests a crowd of points all equally likely to be the best match.

Prediction uncertainty Prediction uncertainty $u(A_j, \tilde{A}, s, k)$ gives the variation among the k best matches to A_j. Formally,

$$u(A_j, \tilde{A}, s, k) = v(\tilde{A}_{(1:k)})$$

where $\tilde{A}_{(1:k)}$ is the set of k closest matches to A_j under the distance measure s, and v is a measure of variation, such as the average variance of each column in $\tilde{A}_{(1:k)}$. Intuitively, prediction uncertainty measures the impact on making a poor choice of the best match. If columns have different levels of importance, we could use a weighted average of column variances. The measure v could also return a set of uncertainties, once for each column.

Between these three measures of prediction risk we can calculate 1) how accurately an attacker can predict the values of an original record, 2) how sure he will be that he has made the best prediction, and 3) the predictive consequences of choosing among the best possibilities. These properties are missed by the empirical reidentification rate and k-anonymity. Depending on the application, the three measures may not have equal importance. For anonymizations with no unique deterministic relationship between original and anonymized records, there is no reidentification risk, so prediction distance would be of primary importance. If we have strong limits on how closely an attacker can predict the values of a target record, we may be satisfied with weaker limits on how sure he is of the best match or what the range of uncertainty is. For anonymizations that allow unique relationships to remain, we may want tighter

bounds on ambiguity and uncertainty, as reidentification risk would be the biggest risk. If a particular anonymized record is a clear best choice for a reidentification, it may not matter as much to the attacker just how far away that best choice is. In Chapters 4 and 5 I will demonstrate the use of these three measures in assessing prediction risk more completely than the existing measures allow. In Chapter 4 I will also demonstrate their use in assessing reidentification risk.

These measures happily lend themselves to defining a reference standard for what constitutes sufficient protection against predictive disclosure. Consider a second dataset A^* consisting of a second sample from the same population as A, but including none of the same individuals. Releasing this nonoverlapping sample A^* would clearly pose zero reidentification risk to the subjects of A, unless they contained identical individuals. It would pose nonzero prediction risk, however, because the records in A^* are similar to those in A, and associations learned from one would apply to the other.

I therefore propose using A^* as a reference standard for anonymization. If releasing any anonymized dataset $\tilde{A}$ imposes a computational disclosure risk to the subjects of A that is no greater than if we had released A^* instead, $\tilde{A}$ shall be deemed sufficiently protective of its subjects' privacy. This is a high standard, representing the same protection against computational disclosure as we would get by not participating in the study. (Of course, other privacy risks remain, such as accidental or malicious direct disclosure of the original data, but these are very different types of risk, and require preventive measures outside the scope of this dissertation.) We can attempt to meet this standard by requiring that the distributions of our three privacy measures are no smaller for $\tilde{A}$ than for A^*. If subjective assessment of the distribution is unclear on whether the standard is met, a Kolmogorov-Smirnov test can be used with a one-sided null hypothesis that the obtained distribution is equivalent or higher than the reference standard [98, 99]. This would require a statistical definition of 'equivalent' that is meaningful in practical terms, such as for example, a difference of 0.05 between the cumulative distributions. Note that contrary to the common use of p-values, in this case a high p-value would be desirable and indicates sufficient protection.

3.2 Analytic Utility

Previous utility measures assess the conservation of bivariate statistics, commonly 2^{nd}-order moments. This is a good start, but it does not allow analysis of higher-order interactions, such as with the subgroup analyses that are ubiquitous in medical research. For example, if a study includes both men and women experiencing deep-vein thrombosis, we may want to ask the question of how smoking affects deep-vein thrombosis in women taking birth control pills vs. those not taking birth control pills. This would probably not be accurately reflected in the anonymized data that only conserved 2^{nd}-order moments. Univariate distributions can also be important, and

some methods try to conserve them [10]. Changes in many of these statistics due to masking are sometimes averaged into a single measure [41]. Other suggestions for measuring analytic utility include information-theoretic measures of entropy, mutual information, or information loss [24], or other measures that are specific to the anonymization method [73].

I propose that the ultimate measure of analytic utility is how well the multivariate distribution of the data is preserved. For the statistical analyst looking for trends, dependencies, and associations in the variables, this is all that matters. For the computer scientist looking to build a predictive model of the data, the multivariate distribution is the ultimate source of information for the model.

If we could find the multivariate distribution of both the original and the anonymized data, we could compare them and assess whether the anonymized data faithfully reproduces the distribution. Unfortunately, inferring this distribution from the data is hard — it is the central problem of machine learning, and no general solution is in sight. Fortunately, finding the multivariate distributions of the original and anonymized data is not required for deciding if those distributions are statistically 'the same'.

We can make this decision as follows. Let's take the example of a 2-column dataset with variables x, y as the simplest example that illustrates the idea. If we plot this 2-dimensional dataset on a plane, we'll see some regions with dense clusters of data points and some regions with relatively few or no points. If we plot the bivariate density on the third axis, we'll have hills and valleys and ridges winding around the plotted area in some complicated pattern.

Now on this plot we draw an arbitrary, possibly complicated curve, that can describe some particular nonlinear feature important to us, or it can simply be a randomly chosen curve. We project all of the data onto the closest point of this curve, and we calculate the empirical univariate distribution of the projected data along that curve. We then draw the same curve on the plot of the anonymized data. If the univariate distributions of these two curves align, then we have one small piece of evidence that the two datasets might have the same bivariate distribution. We can again use the two-sided Kolmogorov-Smirnov test [99] to get an objective assessment of the statistical equality of the univariate distributions. As with the privacy measure comparison, a high p-value indicates equivalent distributions.

We can keep doing this for many arbitrary curves, and eventually we may convince ourselves that the two distributions must be statistically equal, or at least close enough for what we need. But there are (at least) two problems with this. First, the number of curves we must draw may be so large as to make the problem intractable, especially when the data have high dimension. Second, there is always the chance that our curves may have missed the one important region of the data where there is a difference between the original and anonymized distributions.

We can use the power of kernel methods [100] to handle these problems. Kernel methods allow us to perform certain operations on data that has been projected into

a higher-dimensional *kernel space*, without incurring the computational expense of explicitly operating in that space. In other words, we can find the *result* of certain operations in kernel space without actually *performing them* in that space.

For example, we may want to operate in a kernel space where each possible interaction of degree 4 or less between the original variables corresponds to an explicit variable in that space. The mapping would be

$$\{x, y\} \longrightarrow \{x^4, x^3y, x^2y^2, xy^3, y^4, x^3, x^2y, xy^2, y^3, x^2, xy, y^2, x, y, 1\}.$$

With this transformation, any polynomial curve of degree 4 or less in data space corresponds to a straight line in kernel space. So if we can be satisfied with only looking at curves in this class, we can limit ourselves to looking at straight lines in kernel space.

Moreover, we don't have to look at *all* of the straight lines, we can satisfy ourselves with the principal components of the data in kernel space. This is because the first principal component lies along the line that covers the greatest variance of the data. The second principal component lies along the line orthogonal to the first that covers the next greatest variance, and so on. We can therefore examine distributions along the principal components, in order, until we've accounted for the amount of cumulative variation in the data that will convince us the distributions are statistically the same.

Now it turns out that finding the principal components in a kernel space is one of the operations that we can do without explicitly operating in that space [101]. To find the kernel principal components, we calculate the kernel matrix K using our chosen kernel function $k(A_i, A_j)$. This kernel function gives the scalar result of performing a dot-product between the records A_i and A_j in the kernel space. The key to kernel methods is that this result can be calculated in terms of the original data, without projecting into the higher-dimensional space. For our 4^{th}-degree example, the kernel function would be

$$k(A_i, A_j) = (A_i^{\mathrm{T}} A_j + 1)^4. \tag{3.2}$$

For any given kernel function $k(A_i, A_j)$, the kernel matrix is calculated with

$$K_{ij} = k(A_i, A_j).$$

The eigenvectors of the kernel matrix are the principal components we're seeking. Calculating the kernel principal components from the original data matrix A has $O(n^3)$ time complexity if done exactly, where n is the number of rows of A. Good approximations can be found in $O(m^2 n)$ time using rank m approximations of K [102].

To assess analytic utility, we first find the kernel principal components w_i of the original data and the distribution of elements within each component. Then we

compare these distributions with those of the projections of the anonymized data onto the same nonlinear features. Amazingly, we can also do this without computing the nonlinear feature. The kernel-space projection $F \cdot \Phi(\alpha)$ of the kernel-space image $\Phi(\alpha)$ of any record α onto the same nonlinear feature F that produced w_i is found by

$$F \cdot \Phi(\alpha) = \sum_{j=1}^{m} w_{ij} k(A_j, \alpha),$$

where w_{ij} is the j^{th} element of w_i and m is the number of rows of A.

The objective utility measure is the set of p-values produced by comparing the distributions of original and anonymized data along the kernel principal components of A. We can include as many components as we want in this measure, until the number of components covers the desired amount of cumulative variation in the data. If we examine enough components to collectively cover 99% of the total variance in the data, then we can be sure that whatever we've missed accounts for only 1% of the total variance. We can set this number to whatever threshold we want, in order to convince ourselves that we haven't missed any significant feature of the data.

Subjectively, we can also look at bivariate plots of the data projected onto two selected kernel principal components to examine even more subtle and complex dependencies in the data. (Figure 3-1) is an example of such a plot. (The figure shows some data from Section 4.2.1 after anonymization with the method of Section 4.1.1, but the particular method is not important at this point). The kernel principal components are plotted along the diagonal, with blue lines indicating the original distribution, and red the anonymized distribution. The magnitude of the eigenvalue (denoted by `lambda`), the cumulative variance explained by all components up to the given one (denoted by `cum`), and the p-value of the Kolmogorov-Smirnov test for equivalency (denoted by `p`) indicated on the plot. Above the diagonal, contour plots are shown, with the plot in row m column n being that of the m^{th} vs. the n^{th} component. Again, blue indicates the original contours and red the anonymized contours. The grey data points in these plots are the anonymized data. Below the diagonal, the anonymized data is again plotted, but in a way that emphasizes the tails of the distributions.

In this example plot, we see that the first kernel principal component is close to the original, although we might want better agreement in some circumstances. The p-value of 0.085 tells us this difference or greater should arise 8.5% of the time if these were distributions of independent samples from the same population. The second principal component is clearly not preserved by the anonymization, with the original tail to the left not present in the anonymized data, and the anonymized data forming a much broader peak. Objectively, the p-value of 10^{-6} tells us the distributions are different. The third principal component seems to be well preserved, with a p-value of 0.75. From the plot of the third principal component, we see that together the top three components account for 75% of the variance in the data.

The bivariate plot in column 3 row 2 is the plot of the third vs. the second principal component. It shows the peaks of the two distributions in different places and the downward tail of the blue distribution missing in the red. Overall, this plot shows us an anonymization that is both objectively (via the p-values) and subjectively (via the visual assessment of the plots) a different multivariate distribution than the original. If we compared this plot to two samples from the same distribution, we would see much closer agreement between the two samples than we see here.

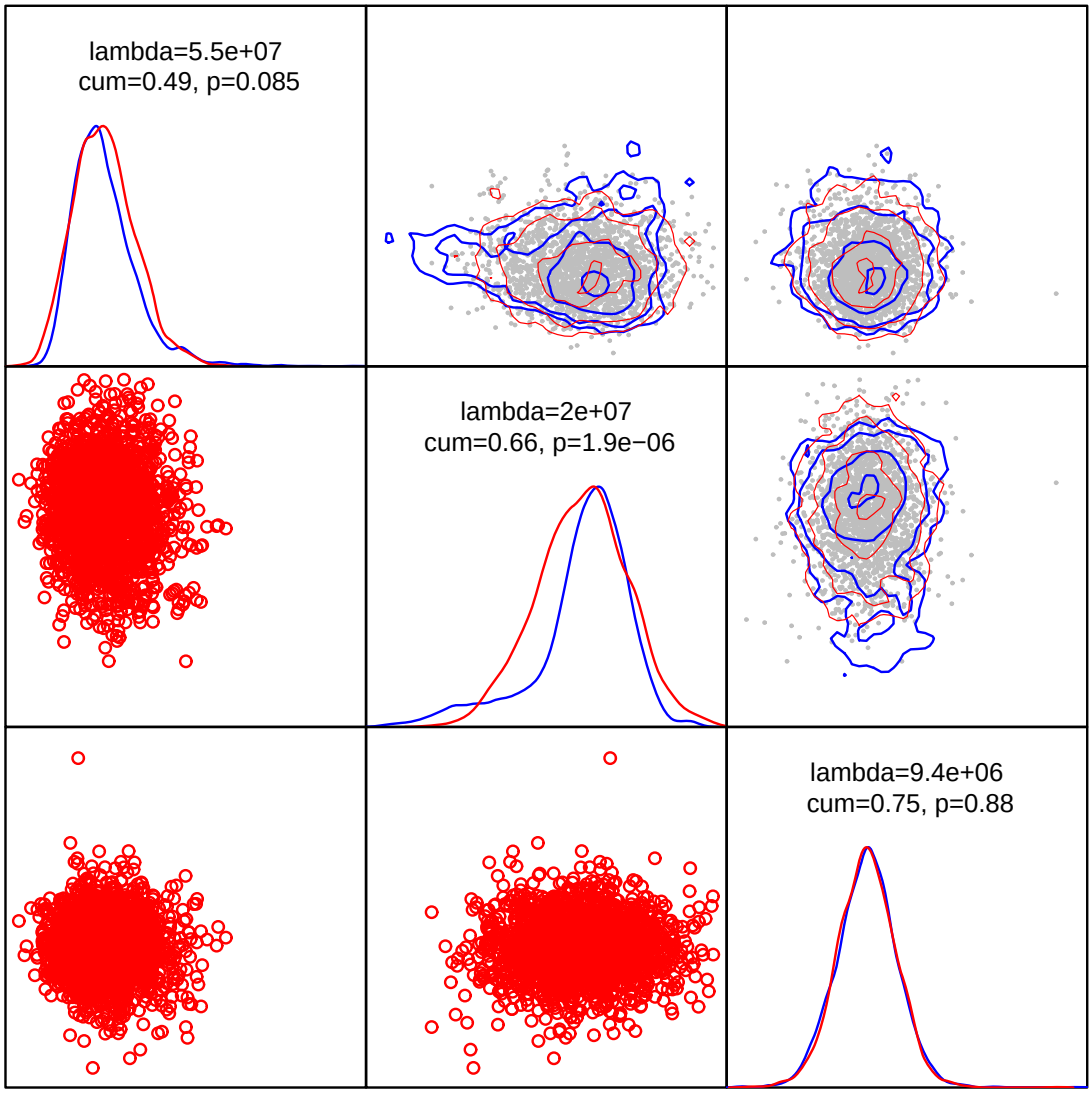


Figure 3-1: Example analysis of analytic utility. See text for details.

Other kernels Amazingly, the computational efficiency of these calculations do not depend on the polynomial degree of the transformation, despite the fact that

the dimension of kernel space grows exponentially with this degree. In fact, we can operate in spaces with infinite numbers of dimensions just as easily as we can operate in the 4^{th}-degree polynomial space of our example.

The particular kernel function in our example allows us to assess whether all 4^{th}-order dependencies between variables were preserved in the anonymization. Looking at the bivariate plots between components we can subjectively assess whether much higher dependencies are also preserved. We can objectively assess these dependencies of up to any positive integer order η by using the polynomial kernel

$$k(A_i, A_j) = (A_i^{\mathrm{T}} A_j + 1)^{\eta}.$$

We also might use an exponential kernel that would include infinitely high-order polynomials (think of the polynomial expansion of the exponential function), or other kernels that preserve features of interest. The question of the best kernel to use for assessment of analytic utility is an open problem.

Chapter 4

Basic Spectral Anonymization

The main contribution of this dissertation is the observation that the anonymizer is not required to operate in the original basis of the dataset, and that by switching to a judiciously chosen alternative basis, we can improve some combination of the privacy protection, the analytic utility, or the computational efficiency of the anonymization. Specifically, I propose that a spectral basis, such as that provided by the data's eigenvectors and eigenvalues, can simplify anonymization methods, improve results, and reduce the curse of dimensionality. I will use the term *spectral anonymization* to refer to the use of a spectral basis in anonymization. The general approach is to project the data onto a spectral basis, apply an anonymization method, and then project back onto the original basis. The intuition is that projecting onto the spectral basis rotates the axes of the dataset to align with the intrinsic structure of the data, and anonymization algorithms can take advantage of that alignment.

This chapter explores basic aspects of this idea in more detail and gives some examples of its use. Chapter 5 will give a more complex example that addresses deficiencies in the simple ones.

4.1 Theory and Examples

Singular Value Decomposition (SVD) [103] provides a useful spectral basis for anonymization. It decomposes a matrix A into $A = UDV^{\mathrm{T}}$, where D is diagonal, and U and V are orthonormal. These matrices have special properties that can facilitate anonymization.

The first useful property is that the columns of V represent a basis that is optimally aligned with the structure of the data in A. Many datasets have internal structure that keeps them from completely filling the space they reside in, filling instead a potentially lower-dimensional manifold within the enclosing space. The matrix V represents axes of the space that are rotated to optimally align with the embedded manifold.

The second useful property is that the elements on the diagonal of D give the

magnitudes of the data variance or manifold thickness in the directions of this new basis, and the product UD gives the projections of the data onto that basis. Knowing the values of D allows us to make engineering decisions about which axes we wish to emphasize in our anonymization, under the reasonable assumption that the thicker dimensions are worth more attention than the smaller ones. The 'optimality' of the basis alignment refers to the fact that the first column of V describes the direction with the greatest data variance, and each remaining column gives the direction of the greatest remaining variance that is perpendicular to all previous columns.

A third useful property of SVD is that the columns of U are uncorrelated. This allows us to anonymize U one column at a time, skirting the curse of dimensionality, without affecting linear correlations among the variables.

4.1.1 Example - Cell Swapping

As described in Section 2.3.3, simple cell swapping anonymizes a dataset by exchanging the values of selected cells within columns of the dataset [63]. This preserves the univariate distributions of the data but swapping indiscriminately tends to destroy relationships between variables. The challenge is to select cells for swapping that will preserve the statistics of interest. Since choosing swaps that exactly preserve particular statistics is NP-hard [88], swaps are sought that only approximately preserve them.

Approximately preserving even the correlations alone between variables is difficult to do, because it implies several statistical constraints that need to be met [88]. Variations of swapping that attempt to preserve statistical properties have turned out to provide little or no privacy protection [70, 69, 71], and variations focusing on privacy protection have difficulty preserving multivariate statistics [63]. There is a recent variation that generates synthetic data in a multivariate normal distribution, and then replaces the values in each column with the equally-ranked values from the original data [71]. This variation, named *data shuffling* by its authors, represents the state of the art of cell swapping. It has been shown to provide reasonable privacy protection and to conserve univariate distributions exactly and rank correlations asymptotically.

Cell swapping is well-suited to a spectral variation (Figure 4-1). Instead of producing the anonymized $\tilde{A}$ directly, spectral swapping applies a uniform random permutation separately to each column of U to produce $\tilde{U}$. We then construct the anonymized $\tilde{A}$ by $\tilde{A} = \tilde{U}DV^{\mathrm{T}}$. The permutations of U do not affect the correlations of $\tilde{A}$ because the correlation matrix of U is the identity matrix, and our permutations preserve this.

Formally, $U^{\mathrm{T}}U = I = \tilde{U}^{\mathrm{T}}\tilde{U}$, so

$$\begin{aligned}\mathrm{cov}(A) &= A^{\mathrm{T}}A \\ &= (UDV^{\mathrm{T}})^{\mathrm{T}}(UDV^{\mathrm{T}}) \\ &= VD^{\mathrm{T}}U^{\mathrm{T}}UDV^{\mathrm{T}} \\ &= VD^{\mathrm{T}}\tilde{U}^{\mathrm{T}}\tilde{U}DV^{\mathrm{T}} \\ &= \tilde{A}^{\mathrm{T}}\tilde{A} \\ &= \mathrm{cov}(\tilde{A})\end{aligned}$$

This assumes that we first subtract the column means of A, anonymize, and replace the means. This method conserves means (exactly), variances, covariances, and linear correlations of the original data. It also conserves the univariate distributions along the principal components of A, which in some cases may be more useful than preserving the univariate distributions of the original variables.

In practice, the permutations won't produce a $\tilde{U}$ with a correlation matrix of exactly I, so the means and distributions are the only statistics that are exactly conserved.

SVD-Swap(A, tol)
1 $(U, D, V) \leftarrow \mathrm{SVD}(A)$
2 **repeat** $\tilde{U} \leftarrow$ Column-Swap(U)
3 **until** $\max(\mathrm{cov}[\tilde{U}]) \leq tol$
4 $\tilde{A} \leftarrow \tilde{U}DV^{\mathrm{T}}$
5 **return** $\tilde{A}$

Figure 4-1: The SVD Swapping algorithm. Column-Swap applies a uniform random permutation to each column.

Under anonymization by spectral swapping, the practical reidentification risk is zero because there is no unique deterministic relationship between a released record and any individual. The attacker could conceivably reverse the applied randomization to reconstruct the original data, but it is not obvious how this might be done, or even if it is indeed possible. At a minimum it is NP-hard to un-swap the cells to match a given set of statistics (such as covariances of the original data that might be known to the attacker) [88]. As the experiments below demonstrate, the protection that spectral swapping provides against predictive disclosure is stronger than both our reference standard and the comparison data-shuffling algorithm.

4.1.2 Example - Microaggregation

My second example uses a microaggregation method. As discussed in Section 2.3.4, microaggregation anonymizes a dataset by collecting similar data points into groups and replacing all k members of the group with k copies of a single representative record. The representative record may be chosen from the members of the group, or it may be some kind of calculated central tendency like the mean.

For this example we'll use the specific microaggregation method of Recursive Histogram Sanitization (RHS)[45]. RHS is one of the few anonymization methods with rigorously proven anonymity properties (although so far only for quite restrictive distributional assumptions), and it demonstrates a large benefit from using a spectral basis. RHS operates by splitting the data in every dimension at the median, forming in one pass a total of 2^n potential groups for a dataset with n dimensions. For a high-dimensional dataset, most of these potential groups will have no members. Of the groups with nonzero membership, if any have membership greater than $2k$ samples, it recurses on those groups.

A major problem with RHS that prevents its practical use is that for higher-dimensional data, the first split produces many groups that contain only one sample, preventing any anonymization for those samples. We will see that this is a problem for our dataset.

Spectral-RHS is a spectral version of RHS that works on the $T = UD$ product matrix instead of the original A (Figure 4-2)). Spectral-RHS makes use of the natural order of singular vectors to prevent the exponential explosion of group formation. In the original RHS, the relative importance of each column of A is unclear, and all columns are necessarily bisected simultaneously. In Spectral-RHS, each successive column of the matrix T spans a smaller range than its predecessor, and we can bisect one at a time based on that ordering. We select the column with the largest range at any particular step, bisect it at the median, and recurse on the two new groups. Upon the algorithm's return, the thinner dimensions will probably not have been partitioned at all for most groups, but that makes little difference to the anonymization.

Since this algorithm replaces a cluster of membership between k and $2k-1$ with copies of a single representative, we expect an empirical reidentification rate of somewhere between $\frac{1}{k}$ and $\frac{1}{(2k-1)}$, because the representative must be nearest to *some* member of the original cluster (barring a tie). However, this does not necessarily mean that the correctly matched records are at higher risk of reidentification. For identities to be at risk, the attacker must be able to distinguish correct from incorrect matches. The ambiguity distribution can help us assess this distinction.

The strong, rigorously proven anonymity guarantees of RHS may or may not be preserved by the spectral transformation - these guarantees depend on distributional assumptions that may be violated by the change of basis. It is an open problem to show whether the rotated distribution would degrade these strong guarantees. But

```
SPECTRAL-RHS(T, k)
1  if rows[T] ≤ 2k
2     then return MASK(T)
3  i ← SELECT-COLUMN(T, k)
4  (A, B) ← PARTITION(T, i)
5  return MERGE(SPECTRAL-RHS(A, k), SPECTRAL-RHS(B, k))
```

Figure 4-2: The spectral adaptation of the Recursive Histogram Sanitization procedure. T is the matrix to be anonymized with anonymization parameter k. SELECT-COLUMN selects the column of T with the largest range. PARTITION(T, i) divides T at the median of its i^{th} column, returning two matrices. MASK(T) performs the desired masking, such as replacing all elements of T with their mean. MERGE concatenates its array arguments in a vertical stack.

SPECTRAL-RHS only relies on the properties afforded by microaggregation in general, and these are not affected by the spectral transformation.

4.2 Experiments

This section presents experimental results of these basic examples. The performance of SVD Swapping and SPECTRAL-RHS was compared to data shuffling, nonspectral RHS, a reference standard, and an additive noise algorithm benchmark. The results show the spectral algorithms matching or exceeding the performance of their nonspectral counterparts in privacy protection and analytic utility. The SVD Swapping results show the simple spectral algorithm providing competitive analytic validity and stronger privacy protection than the complex, nonspectral state of the art. The SPECTRAL-RHS results show the spectral algorithm avoiding the curse of dimensionality that defeated the nonspectral RHS.

4.2.1 Methods

Dataset The dataset used for all experiments in this dissertation was a public dataset obtained from the National Health and Examination Survey (NHANES)[104] (Appendix A). The dataset contained 11763 records of 69 continuous, ordinal, and binary attributes (after converting categorical attributes into binary). The attributes included demographic, clinical, and behavioral variables. Binary attributes were recoded as $\{-1, +1\}$. Continuous variables were all strictly positive and were log transformed and then standardized. From this we randomly sampled $m = 2000$ records and selected a representative $n = 28$ attributes for computational efficiency.

A sample of 2000 records that did not include any records in the above sample were randomly selected from the same original dataset of 11763 records. This was used as the reference standard in measures of disclosure risk and analytic utility.

Perturbation Algorithms SVD swapping and data shuffling [71] were implemented and compared. SPECTRAL-RHS and non-spectral RHS were implemented with design parameter $k = 5$ and compared.

For a baseline comparison, (non-spectral) anonymization by adding zero-mean multivariate normal noise was implemented with a noise covariance matrix $b\Sigma$, where $b = 0.1$ and Σ is the covariance matrix of the original data. The anonymized data was corrected for mean and variance distortion [52, 53]. Noise addition is not an effective method for anonymizing high-dimensional data with many binary attributes, but we include it here as a well-known benchmark.

Privacy Protection Measures Privacy protection was assessed using prediction distance, ambiguity and uncertainty with the distance measure of (3.1) and $k = 5$. All assessments were made with the continuous data in the standardized log form described above, original binary data in $\{-1, +1\}$ encoding, and anonymized binary data thresholded at zero. For statistical comparison, the distribution of each measure was compared against the corresponding distribution of the reference sample using the one-sided Kolmogorov-Smirnov test.

For additive noise and SPECTRAL-RHS, reidentification risk was assessed by matching records with the distance measure of (3.1). The empirical reidentification rate was assessed, and distributions of the three privacy measures were compared between correctly and incorrectly matched records. The area under the receiver operating characteristic curve (AUC) [105] was calculated with the non-parametric empirical method separately for each privacy measure. The AUC measures how accurately each method distinguishes correct from incorrect matches, and is therefore used to indicate whether correct matches are predictably or unpredictably correct.

Analytic Utility Measures Analytic utility was assessed by comparing the median differences in the univariate means, variances, and correlation matrices of the original vs. anonymized datasets. To adjust for differences in variable scale, the differences between means were normalized by the standard deviation of the variable in the original data, and the differences between variances were normalized by the variance of the original.

4.2.2 Results

Privacy Protection The two spectral algorithms improved the privacy protection of their nonspectral counterparts according to all three measures (Figure 4-3). Nonspectral RHS failed to produce any anonymization due to the dataset's high

dimensionality - the first pass partitioned all but two records into their own cell, producing distances, ambiguities, and uncertainties of zero. The baseline additive noise algorithm produced some protection, but that protection was much weaker than the reference standard despite the high amount of noise added. Spectral swapping and data shuffling both produced privacy protection superior to the reference standard in all three measures, with spectral swapping providing the stronger protection in each case. SPECTRAL-RHS provided larger (better) prediction distance than the reference standard; its uncertainty was zero (weaker than the reference standard) and ambiguity was unity (stronger than the reference standard) by design.

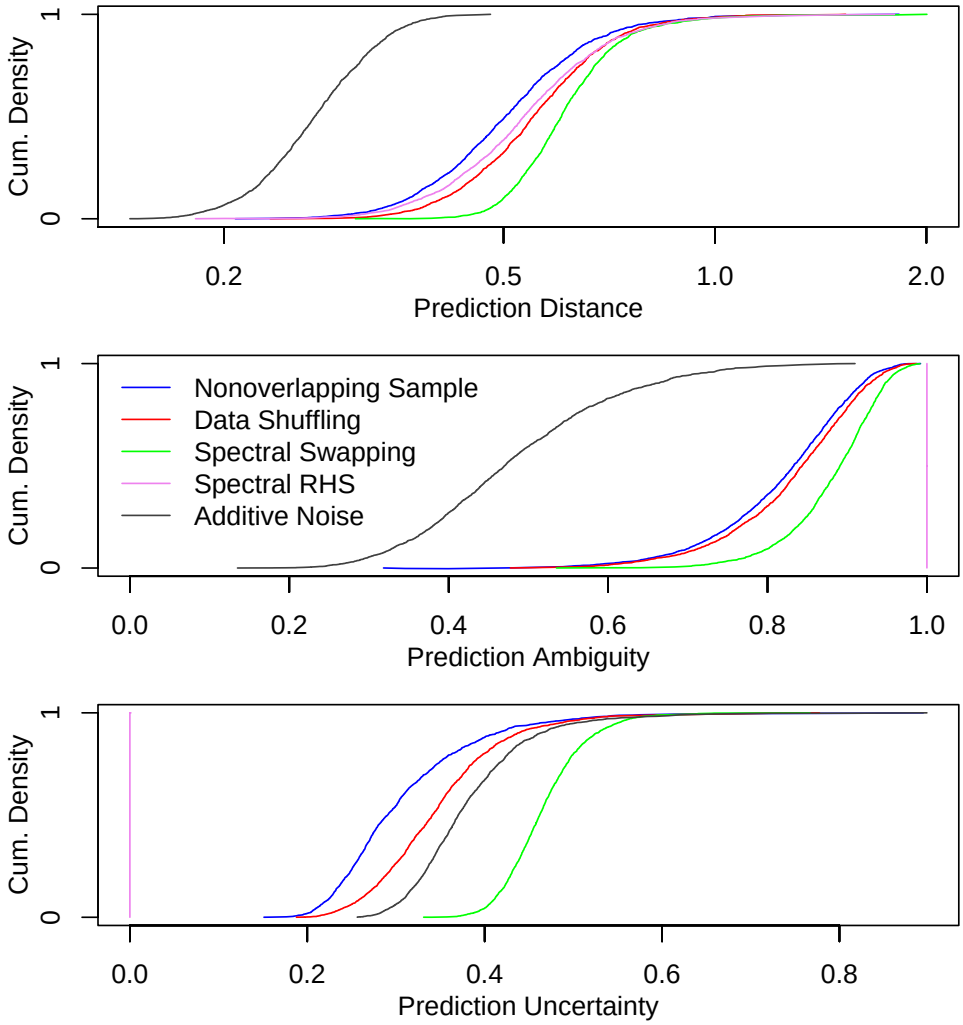
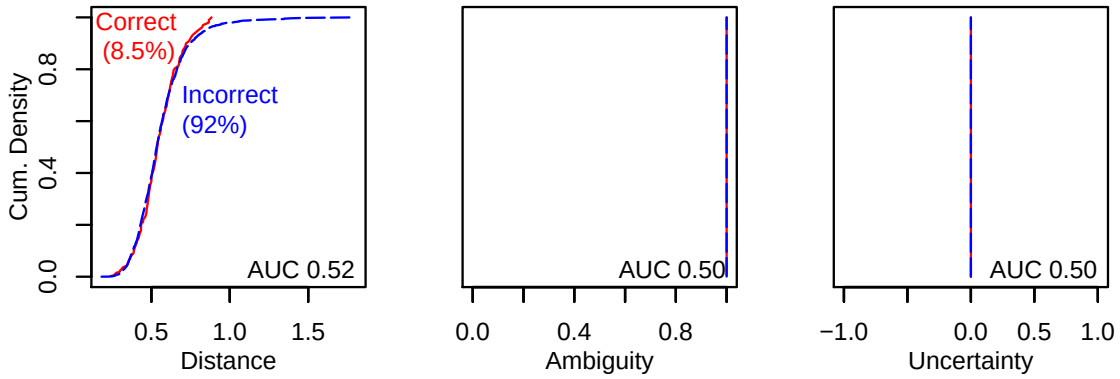


Figure 4-3: Privacy protection of basic spectral anonymization. The spectral algorithms provide improved privacy protection over their nonspectral counterparts. Nonspectral RHS failed to anonymize at all, and is not shown.

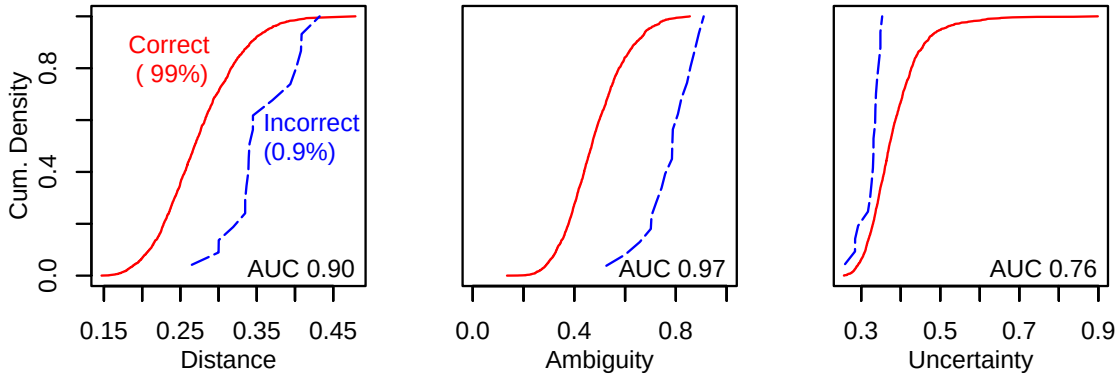
Under empirical matching, the data anonymized by SPECTRAL-RHS allowed 170

(8.5%) correct matches (Figure 4-4). With $k = 5$, we expected slightly more than this, somewhere between 11% and 20% correct. Correct matches were indistinguishable from incorrect matches on the basis of prediction distance (AUC 0.52), ambiguity (AUC 0.50), or uncertainty (AUC 0.50) (Figure 4-4a).

Additive noise allowed 1982 (99%) correct matches. These were almost completely distinguishable from incorrect matches on the basis of prediction distance (AUC 0.90) or ambiguity (AUC 0.97), and to a lesser extent on the basis of uncertainty (AUC 0.76) (Figure 4-4b).



(a) SPECTRAL-RHS



(b) Additive noise

Figure 4-4: Reidentification analysis using the new privacy measures. Correct matches are distinguishable from incorrect under additive noise anonymization, but not under SPECTRAL-RHS. Reidentification risk is therefore high for additive noise, low for SPECTRAL-RHS. See Section 4.2.3 for further discussion of these figures.

Analytic Utility All methods that produced anonymized data approximately preserved all target statistics, with the exception that SPECTRAL-RHS did not preserve the variance of the original data (Table 4.1).

Table 4.1: Target statistics were approximately preserved by all methods used, except variance was not preserved by Spectral RHS. Values are median differences between original and anonymized data. Mean and variance values were normalized as described in the text.

	Median difference in			
	mean	var	cor	rank cor
Spectral Swapping	0.014	0.022	0.020	0.024
Spectral RHS	0.027	0.380	0.066	0.066
Data Shuffling	0.014	0.018	0.031	0.025
Additive Noise	0.015	0.023	0.020	0.020
Reference Standard	0.016	0.013	0.017	0.017

4.2.3 Discussion

These experiments demonstrate basic improvements in anonymization that can be made by operating in a spectral basis. In the cell-swapping example, the spectral form of simple swapping provided competitive analytic validity and stronger privacy protection than data shuffling. The practical effect of the stronger privacy protection may be less important, however, since both algorithms give stronger protection than required by the reference standard, and would both therefore be sufficient by that standard. But the example demonstrates that simply choosing a judicious basis for anonymization allows the original, basic cell swapping method to transform from a weak algorithm of mainly historical interest to one that performs as well as the complex state-of-the-art method.

The experiments also demonstrate how spectral anonymization can help overcome the curse of dimensionality. In the microaggregation example, the nonspectral RHS method was unable to anonymize the high-dimensional dataset at all, whereas Spectral-RHS provided sufficient privacy protection as measured by the reference standard.

Additionally, these examples demonstrate some important added value of the new privacy measures. The empirical reidentification rate allowed by the microaggregation example was 8.5%, which would appear unacceptable. But this 8.5% in fact refers roughly to a situation where each original record is approximately equidistant from 12 anonymized records, with an attacker being forced to choose randomly between the 12 in a matching attack. We would expect the attacker to choose correctly about one time in 12, but the attacker is unable to distinguish when that happens.

The distance, ambiguity, and uncertainty curves for Spectral-RHS are nearly exactly the same for correct vs. incorrect matches (Figure 4-4a). An attacker therefore cannot tell which candidate matches are correct on the basis of how close a candidate match is. The AUC value of 0.52 for prediction distance is an objective demonstra-

tion that for SPECTRAL-RHS, closer match distance does not at all suggest a correct match (Figure 4-4a), and by design of the algorithm neither ambiguity nor uncertainty measurements aid in making that distinction. The privacy protection afforded by SPECTRAL-RHS could therefore be acceptable for many applications — but we wouldn't know that by looking at the empirical reidentification rate alone.

The privacy measures tell a different story about anonymization by additive noise. They confirm what we already expected, that this method would be inadequate for our data. Both prediction distance and ambiguity were weaker (lower) under additive noise than for the reference standard, indicating high disclosure risk. Indeed, the empirical reidentification rate was 99%, and correct matches are easily distinguishable from incorrect matches on the basis of either distance, ambiguity, or to a lesser degree, uncertainty (Figure 4-4b). Prediction distance, for example, is much lower for correct matches than for incorrect matches, and would be a reliable indicator of a successful reidentification — one could accept any match with a distance below 0.3, and this would find 80% of the correct matches, and almost no incorrect matches. We suspect, but did not investigate, that a model built on the combination of the three measures would be even better at predicting correct vs. incorrect matches.

Intuitively, the benefits of spectral anonymization come from aligning the axes of anonymization to better correspond to the inherent structure in the data. For data with simple structure, the realignment can produce optimal results. Spectral swapping on multivariate normal data, for example, would produce perfect anonymization (in the sense that it meets or exceeds our reference standard) and perfect analytic utility (in the sense that all statistics computed on the anonymized data would be equally valid as those computed on the original data). But this type of data is uncommon in the real world. For real-world data with nonlinear structure, the realignment can help, but further improvements need to be made. The next chapter discusses one adjustment we can make that significantly improves the analytic utility of the anonymized data.

Chapter 5

Nonlinear Spectral Anonymization

As described in Chapter 4, SVD Swapping is a spectral anonymization algorithm that provides strong privacy protection and analytic utility comparable to any known method that provides equivalent privacy protection. There is ample room for improvement, however, because SVD Swapping preserves only linear dependencies between variables, and these are usually of only basic interest to analysts. We would like an algorithm that can preserve at least some of the data's nonlinear dependencies, thereby allowing some subgroup analysis or nonlinear model building.

5.1 Theory and Example

We start by observing that n-dimensional data with non-trivial dependencies often exist on a lower-dimensional manifold within the n-dimensional space. If we can identify the geometry of that manifold, we can try to maintain its structure by perturbing the data only along its surface. The Laplacian Eigenmap method seeks to capture the manifold geometry and transform the data to a low-dimensional basis corresponding to the manifold surface in a way that preserves locality information [106]. That is, points near each other in the original space remain so in the transformed space. This property naturally preserves and emphasizes clusters in the original data.

Interestingly, spectral clustering methods [107, 108] transform the data to the Laplacian Eigenmap basis before performing the clustering [106], and there are similar ties to the Locally Linear Embedding technique of dimensionality reduction [109]. We might therefore consider the manifold as patches of low-dimensional clusters hinged together in some pattern, hanging in the n-dimensional space. We can identify these with spectral clustering, anonymize them individually in their low-dimensional space, and expect that this will preserve much of the structure of the manifold.

We transform the original data A (with row j denoted A_j) to the new spectral basis by re-normalizing the eigenvectors of the Laplacian matrix [108]. To do this, we first find the affinity matrix S, where

$$S_{ij} = \begin{cases} e^{\frac{-\|A_i - A_j\|^2}{2\sigma^2}} & \text{if } i \neq j \\ 0 & \text{if } i = j, \end{cases} \tag{5.1}$$

and σ is a parameter that determines the scale of the geometry we seek. We then construct a diagonal matrix D, with

$$D_{ii} = \sum_j S_{ij} \tag{5.2}$$

and zeros off the diagonal. We then construct the Laplacian matrix L by

$$L = D^{-\frac{1}{2}} S D^{-\frac{1}{2}}. \tag{5.3}$$

We now find the q largest eigenvectors $x_1, x_2, \ldots, x_q$ of L, and collect them as columns of a matrix X. The choice of q is a parameter of the method. For now we will leave it as a free parameter, but our results will suggest a method of choosing the optimal q for anonymization. The coordinates of the data in the desired basis are the rows of Y, where

$$Y_{ij} = \frac{X_{ij}}{(\sum_l X_{il}^2)^{\frac{1}{2}}} \tag{5.4}$$

The Y_{ij} are the coordinates of the original points mapped onto the surface of the manifold. For any given q, these points will naturally fall into q groups located in mutually orthogonal directions from the origin, and can be easily partitioned into q clusters with a standard clustering algorithm like k-means.

We now anonymize each cluster individually with spectral swapping using a basis provided by Independent Component Analysis (ICA) [110, 111]. ICA uses an iterative algorithm to find a linear basis V for the data such that the joint data distribution is equal to the product of the marginal distributions in that basis. That is, if U is the projection of the data onto basis V (so $A = UV$), it seeks V such that the multivariate probability distribution $p(u_1, u_2, \ldots, u_n)$ can be factored into univariate distributions, or

$$p(u_1, u_2, \ldots, u_n) = p(u_1)p(u_2)\cdots p(u_n).$$

We can think of this basis as finding "independent directions" within the data.

ICA has historically been used to find independent signals that have been linearly mixed to produce the measured data, such as might be done with two independent audio sources recorded by two microphones, all located in different parts of the room. Each microphone picks up a linear combination of the two sources, but in different proportions. ICA provides a way to estimate the original sources and the mixing matrix given only the measured data. It does this by finding a basis V that implies the most independent sources U, giving the best fit to (5.1).

For most applications, ICA fails when the sources are Gaussian, since any set of

orthogonal directions within a (standardized) multivariate Gaussian are independent. Under these conditions, it is impossible to find any preference for any particular basis, and the search for the original sources comes up empty. For our purposes, however, this is not a problem. We want to find *any* set of directions v_i that give mutually independent projections u_i, so the multivariate Gaussian distribution represents the best of all worlds to us. We can pick any set of orthogonal directions, and they will be perpendicular.

While SVD swapping preserves linear dependencies, ICA swapping preserves *all* dependencies between variables, as long an exact factoring is found. In a large dataset an exact factoring probably does not exist, but in clusters of smaller population and smaller dimension, one or more may well exist. I will call this algorithm *Partitioned ICA Swapping* (Figure 5-1).

```
PICA(A,q)
 1  Calculate S using (5.1)
 2  Calculate D using (5.2)
 3  Calculate L using (5.3)
 4  Calculate Y using (5.4)
 5  C ← CLUSTER(Y, q)                    ▷ C is a set of q clusters
 6  Ã ← NIL
 7  for each cluster c ∈ C
 8        do (U, V) ← FAST-ICA(c)
 9           Ũ ← COLUMN-SWAP(U)
10           Ã ← MERGE(Ã, ŨV)
11  return Ã
```

Figure 5-1: The Partitioned ICA Swapping algorithm. CLUSTER can be any standard clustering algorithm, such as K-MEANS. COLUMN-SWAP applies a uniform random permutation to each column. MERGE merges two arrays by stacking their rows vertically.

5.2 Experiments

This section presents experimental results of this example of nonlinear spectral anonymization. The performance of Partitioned ICA Swapping was compared to SVD swapping and the reference sample. The results show that according to my measures, Partitioned ICA Swapping provides equal or better privacy protection and analytic utility as the reference sample, which is an extremely encouraging result. The analytic

utility of Partitioned ICA was stronger than SVD Swapping, although the privacy protection of SVD swapping was stronger than that of Partitioned ICA. The difference in privacy protection is probably less important, however, because both methods met the standard of the reference sample.

5.2.1 Methods

Dataset The dataset used for these experiments (including the reference sample) was the same as described in Section 4.2.1.

Perturbation Algorithms The dataset was anonymized in turn by SVD Swapping and once each by Partitioned ICA Swapping with target partition sizes $t = 100$, 200 and 300 (that is, with a clustering parameter $q = 2000/t$. The ICA algorithm used was the efficient fastICA algorithm [110], implemented as a package in R [112], using the log cosh approximation to the negentropy cost function. The spectral clustering implementation used was that provided by the kernlab package in R [113]. The scale parameter σ was chosen automatically by the kernlab function, following a built-in heuristic. After anonymization, originally binary variables were thresholded at zero and continuous variables were transformed to their original scale. For brevity, these algorithms will be referred to as PICAt.

Privacy Protection Measures Privacy measures were calculated using the distance function of (3.1) with continuous variables in their log-standardized form and binary variables thresholded at zero. Distributions of the measures were compared using the Kolmogorov-Smirnov test to the reference standard described above. For this experiment, I arbitrarily chose a leftward shift of 5% of the data in the distance measure to be to be of practical significance, and 10% in the other two measures. To align statistical significance with practical significance, a random subset of 1296 data points was taken to produce a power of at least 0.8 (and probably much higher [98]) to detect a cumulative difference of 0.05 in the distance measures at a 0.05 significance level, and 324 points sub-sampled to provide the same power for a difference of 0.1 in the other two measures [98].

Analytic Utility Measures Analytic utility was assessed using the kernel principal components with polynomial kernel function $k(x, y) = (x^{\mathrm{T}}y + 1)^4$ with binary variables thresholded at zero and continuous variables in unstandardized log form. Subjective assessment used the two-sided Kolmogorov-Smirnov test for equal distributions of corresponding kernel principal components. I arbitrarily chose a threshold of 5% of the data shifted in either direction to be of practical significance, and randomly sub-sampled 1296 points for comparison as above [98]. Objective assessment was by pairwise scatterplots of the data along kernel principal components.

5.2.2 Results

Privacy Protection

SVD Swapping provided the strongest privacy protection of all methods tested (Figure 5-2). PICA200 and PICA300 provided equal or stronger protection compared to the reference standard for all three measures (all $p > 0.15$ for PICA200, all $p > 0.7$ for PICA300). PICA100 provided weaker protection in all three measures(all $p < 0.02$).

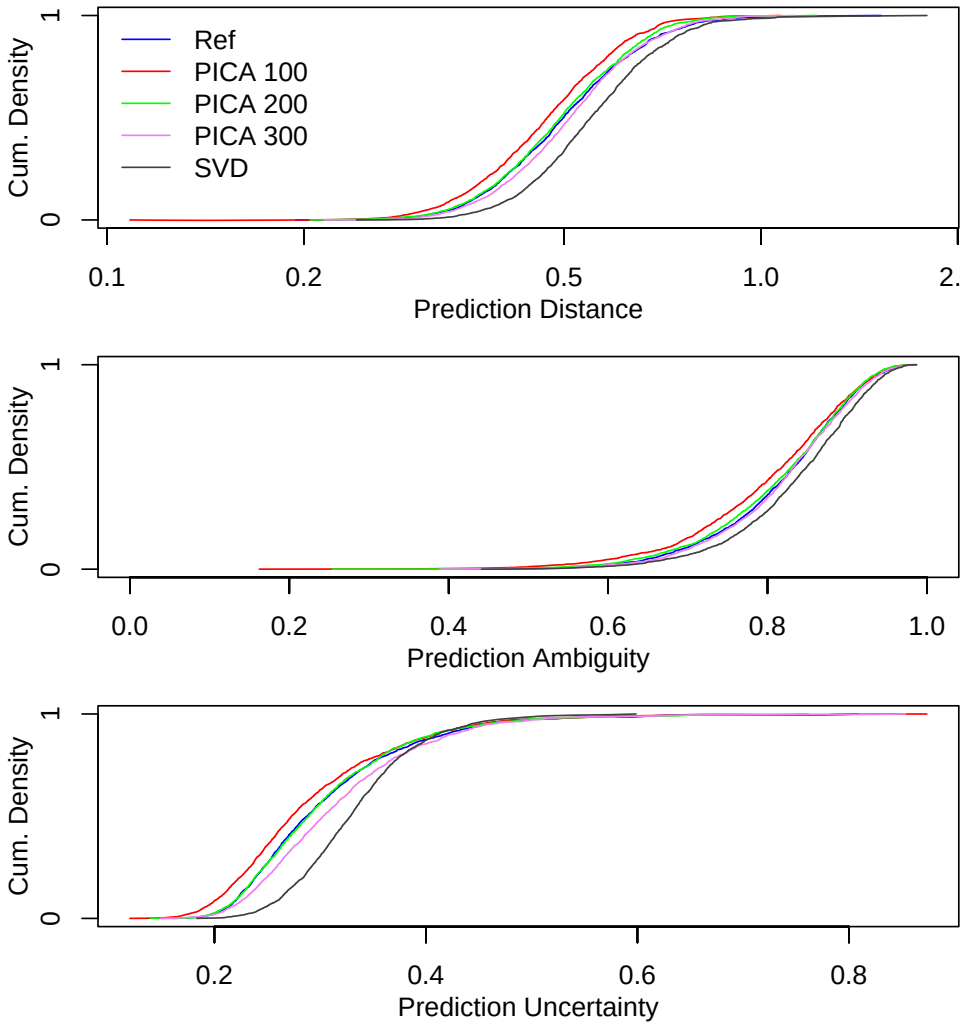


Figure 5-2: Privacy protection of nonlinear anonymization methods. PICA200 provides prediction distance and prediction uncertainty at least as strong as a nonoverlapping sample, and prediction ambiguity roughly the same.

Analytic Utility

Distributions along the top 20 kernel principal components differed significantly from the original for SVD Swapping but not for the other methods (Table 5.1). Objective assessment gives similar results, showing the distributions as well preserved for PICA200 as for the reference standard. (Figure 5-3).

5.2.3 Discussion

In this chapter I have demonstrated a new spectral anonymization algorithm, Partitioned ICA Swapping, that preserves nonlinear structure with high fidelity and reduces computational disclosure risk to that equivalent of non-participants. The practical reidentification risk is reduced to zero since there is no unique deterministic association between any study participant and any anonymized record. The predictive disclosure risk reduced to a level that is statistically no greater than that of the reference sample. The overall computational disclosure risk to study participants due to the released anonymized data is therefore no greater than if they had not participated in the study at all. The analytic utility provided by this algorithm preserves complex interactions of up to fourth order and possibly more. This is improved over the utility provided by SVD Swapping, which preserves linear correlations among variables but no nonlinear structure (Figure 5-3b). I know of no existing anonymization algorithm that provides this level of combined utility and anonymity. Additionally, this algorithm works on datasets of high dimension and on continuous, binary, ordinal, and categorical variables.

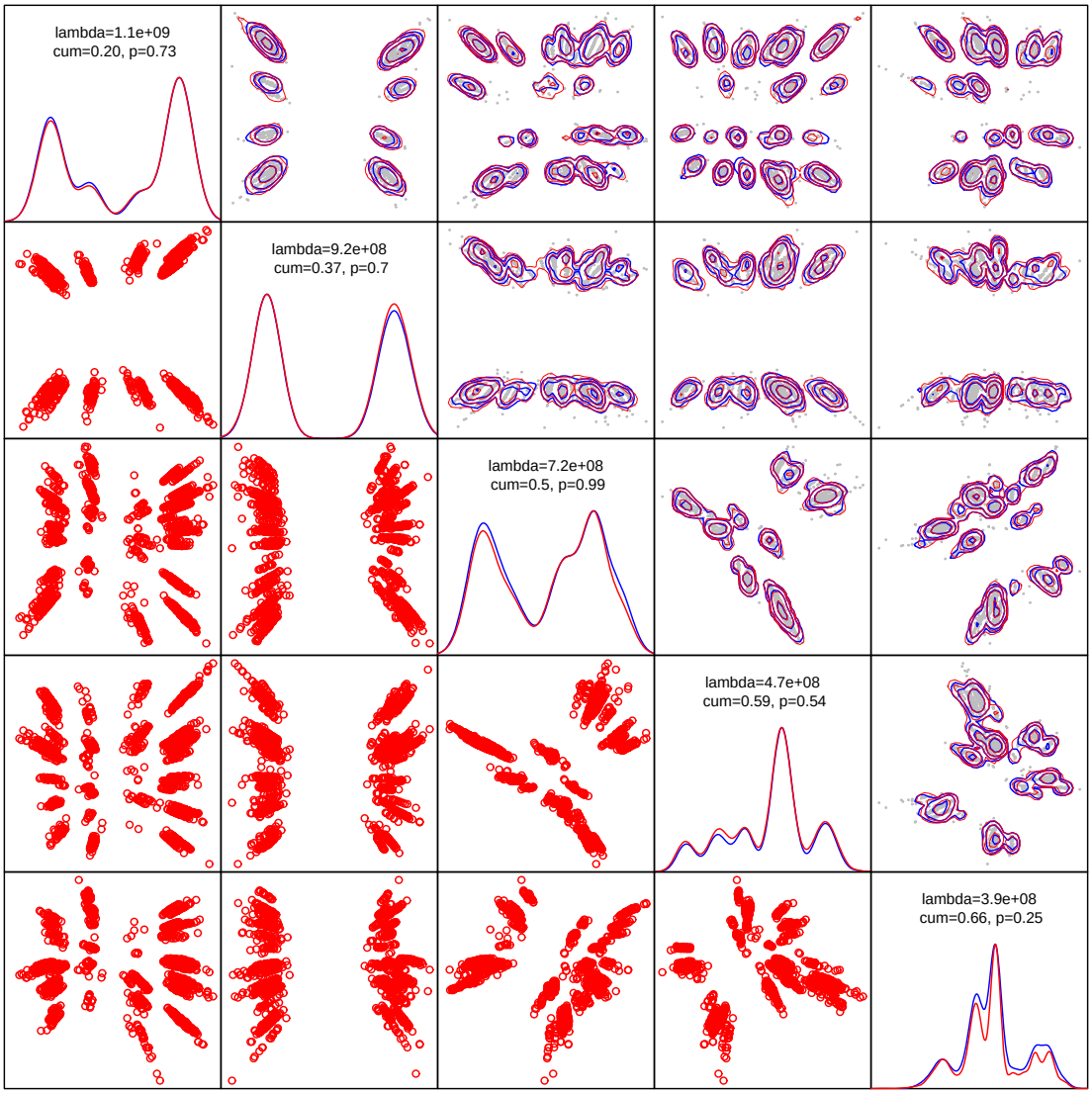
The privacy results suggest a method to choose the cluster parameter q. The prediction distance distribution appears to decrease monotonically with q (Figure 5-2), so we should choose the smallest value that does not drop the privacy distributions below the reference sample. In this experiment, a partition size of 100 produced a statistically significant number of prediction distances below those of the reference sample. Partition sizes of 200 and 300 provided well-anonymized data, so we would choose $q = 2000/200 = 10$ for this anonymization. I suspect, but have not investigated, that the optimum partition size varies with the dimensionality of the data, and probably also with the distribution of eigenvalues.

Partitioned ICA Swapping is similar to synthetic data methods in that it essentially generates a new dataset with the information from a single original record dispersed among many anonymized records. But it differs from most synthetic methods in that it explicitly encodes few assumptions about the data or their distribution. This makes it much less likely that an unexpected or subtle feature of the data will be lost because the synthesis model didn't anticipate it. Since it is not an explicit model, it would be also difficult to release a closed-form description of it suitable for direct analysis.

The most consequential assumption encoded by Partitioned ICA Swapping is that data in small enough clusters lie in a factorable distribution. That is, our model

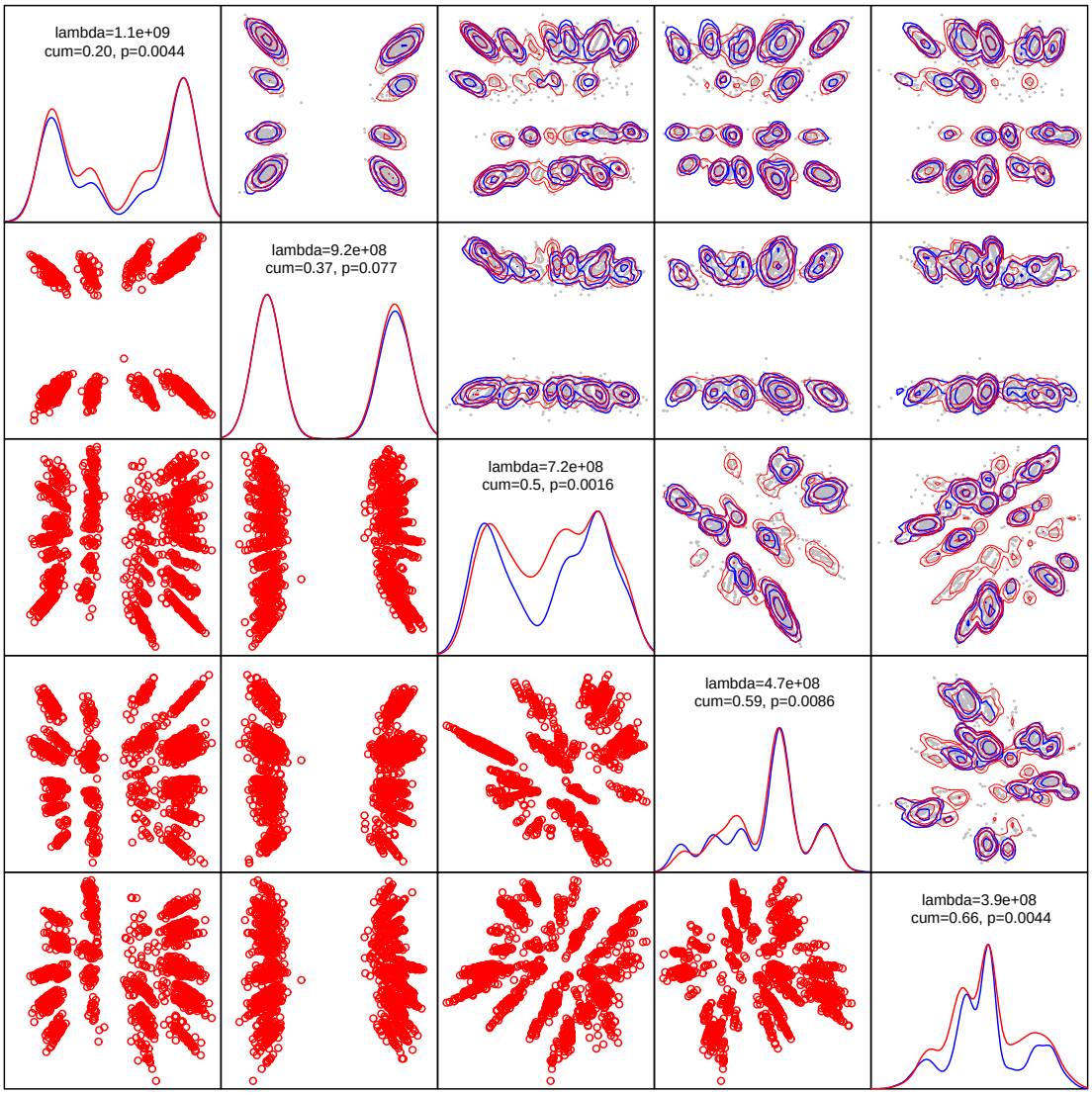
kPC	PICA 100	PICA 200	PICA 300	SVD	Ref
1	0.9	0.11	0.42	0.0098	0.99
2	0.54	0.94	0.9	0.7	0.92
3	0.98	0.54	0.34	0.0025	0.31
4	0.7	0.54	0.27	0.0014	0.15
5	0.39	0.42	0.29	0.037	0.47
6	0.97	0.57	0.94	0.6	0.99
7	0.92	0.76	0.79	0.5	0.92
8	0.85	0.88	0.57	0.16	0.31
9	0.97	0.96	0.31	1.2e-05	0.15
10	0.96	0.79	0.42	0.021	0.47
11	0.9	0.14	0.54	1.4e-05	0.99
12	0.7	0.79	0.39	0.085	0.92
13	0.47	1	0.31	0.47	0.31
14	0.31	0.94	0.57	0.34	0.15
15	0.9	0.57	0.79	0.041	0.47
16	0.7	0.39	0.63	0.085	0.99
17	0.98	0.99	0.23	0.36	0.92
18	0.25	0.92	0.29	0.046	0.31
19	0.67	0.39	0.21	1.2e-05	0.15
20	0.79	0.57	0.7	0.0004	0.47

Table 5.1: Utility of Nonlinear Methods. The first column gives the number of a kernel principal component (kPC) of the original data. Each remaining column represents an anonymization method. Values in these columns report how well the distribution along the principal components are preserved by the anonymization. Higher values mean better preservation of the distribution. Specifically, row n gives the p-values of the two-sided Kolmogorov-Smirnov test over the distribution of the n^{th} kPC, indicating the probability that a difference at least as large as that observed between the original data and the anonymized data would arise in two independent samples from the same population. Partitioned ICA swapping (PICA) preserved the distributions of kPCs as faithfully as the reference standard (Ref) while SVD swapping (SVD) did not.



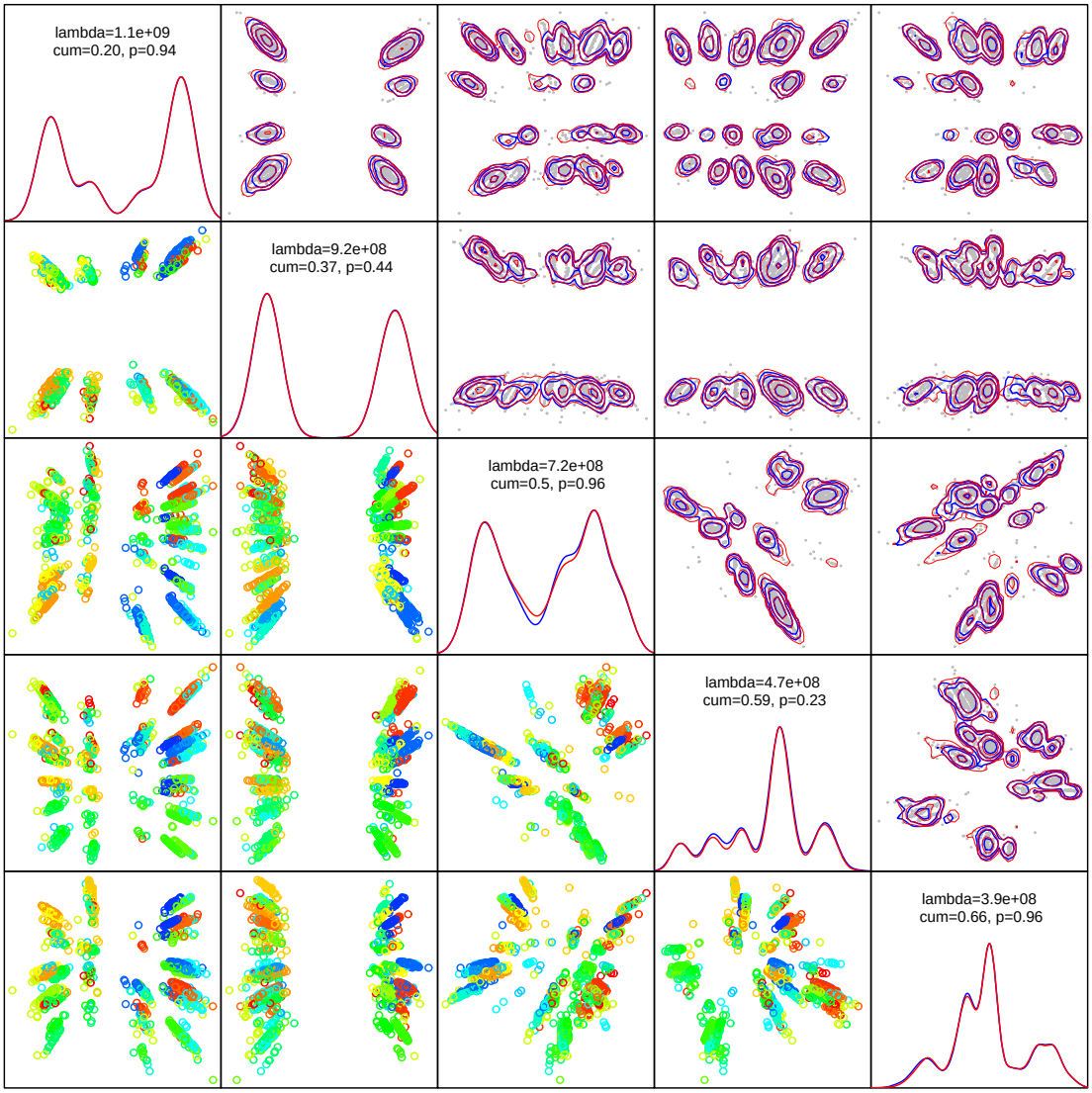
(a) Reference sample. Plots are projections of the data onto the top five fourth-degree kernel principal components. Blue lines represent curves of the original data, red indicate anonymized data (in this case, the reference sample). `cum` represents the cumulative variance explained by the components up to the given component, p values are for the Kolmogorov-Smirnov test for equality of the distribution along the given component. Further explanation of these figures is given in Section 3.2.

Figure 5-3: Analytic utility of nonlinear anonymization methods. Subjective evaluation shows no substantial difference in utility between PICA200 and the reference sample (continued on following pages).



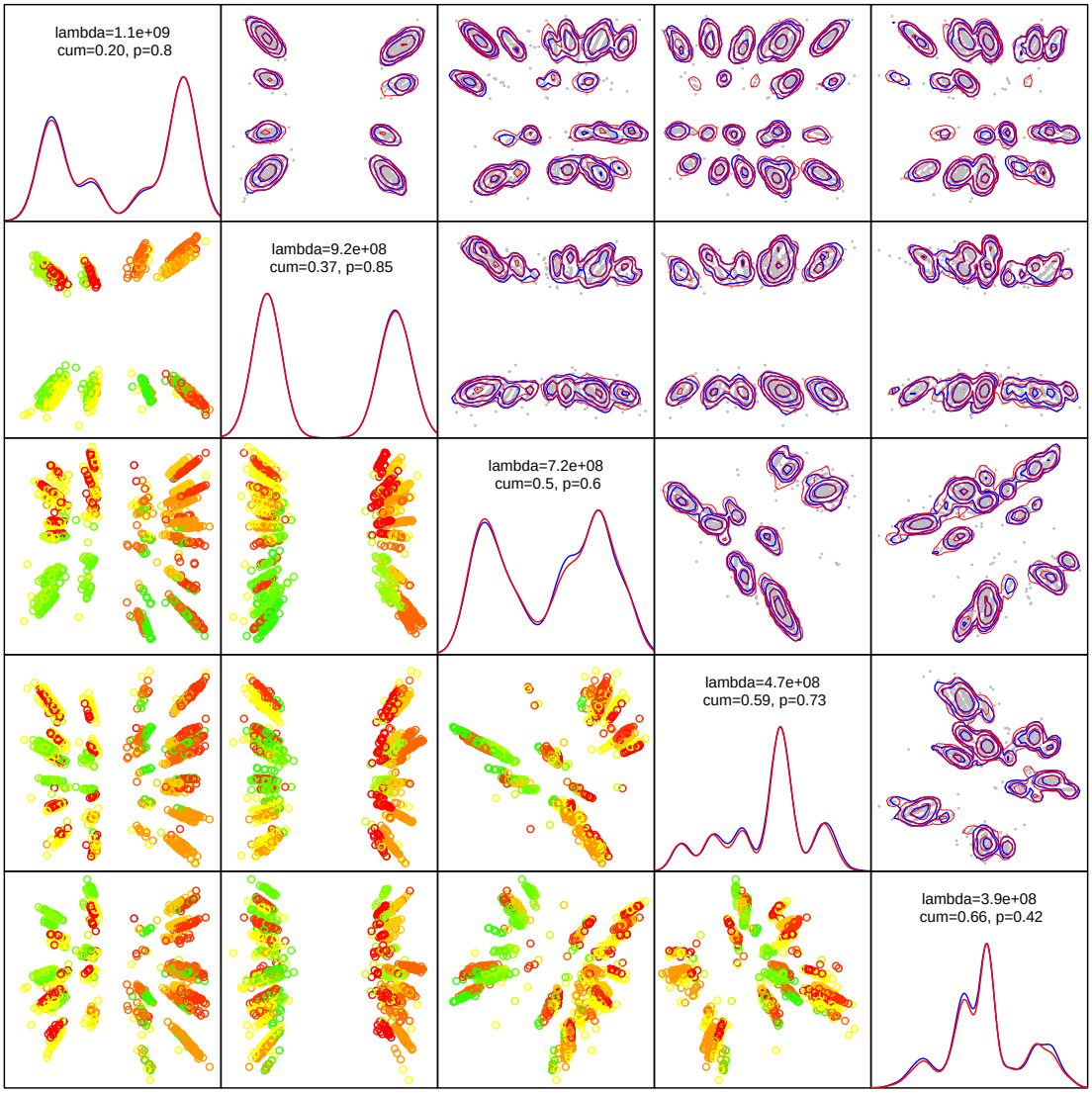
(b) SVD Swapping. Differences in these plots appear subjectively much greater than those of the reference sample (Figure 5-3a). Notation as in Figure 5-3a

Figure 5-3: Analytic utility (continued).



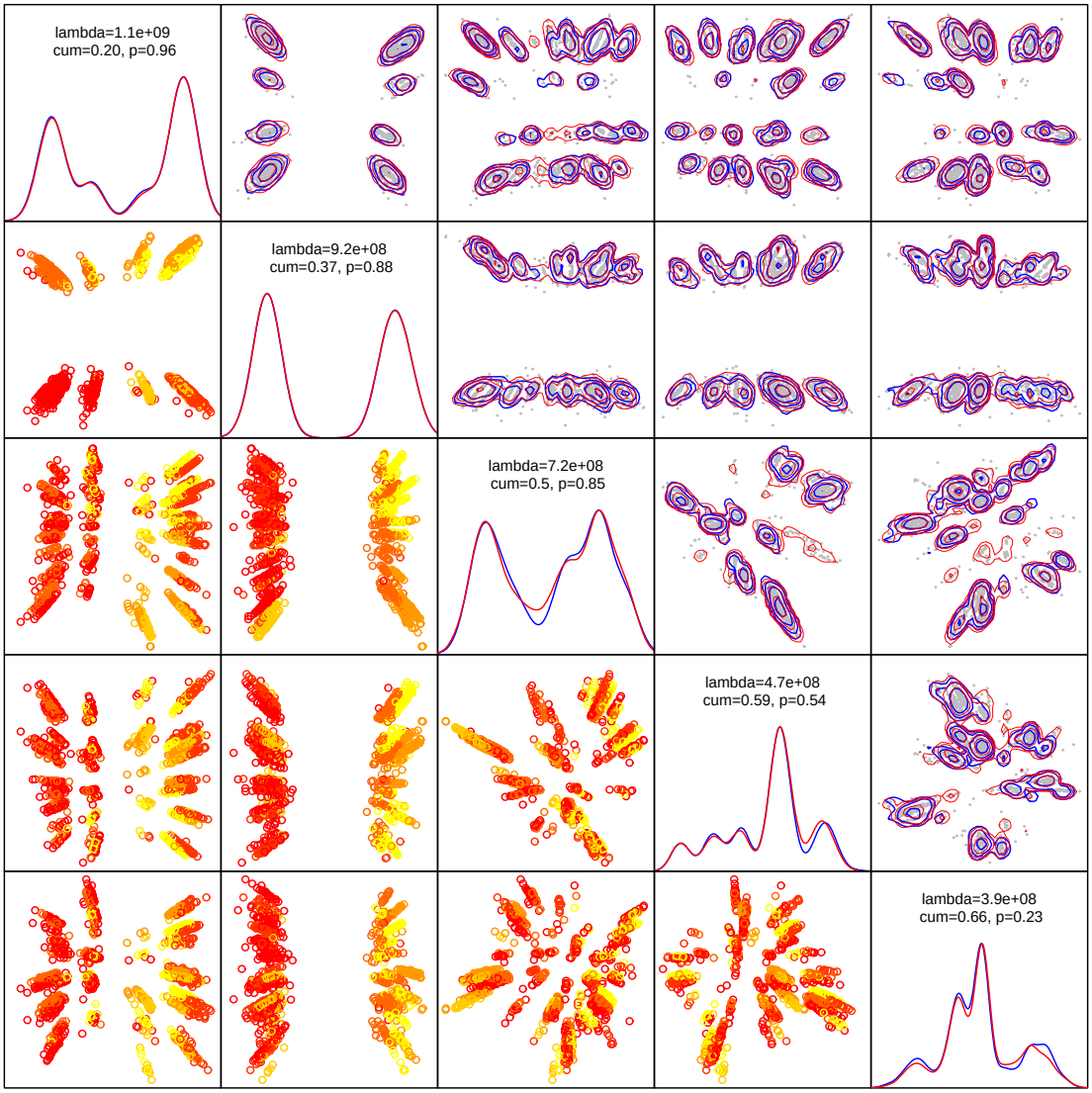
(c) PICA100. Differences in these plots appear subjectively similar to those of the reference sample (Figure 5-3a). Colors in plots below the diagonal represent assigned clusters, other notation as in Figure 5-3a

Figure 5-3: Analytic utility (continued).



(d) PICA200. Differences in these plots appear subjectively similar to those of the reference sample. (Figure 5-3a). Colors in plots below the diagonal represent assigned clusters, other notation as in Figure 5-3a

Figure 5-3: Analytic utility (continued).



(e) PICA300. Differences in these plots appear subjectively greater than in the reference sample (Figure 5-3a). Colors in plots below the diagonal represent assigned clusters, other notation as in Figure 5-3a

Figure 5-3: Analytic utility (continued.)

finds patches on the surface of the data manifold and seeks directions within each patch that factorize the distribution over the patch. To the extent the distribution is truly independent along these directions, this model should capture all statistics of interest. Patches that don't meet this assumption will be imperfectly reproduced by the anonymization.

Chapter 6

Conclusions and Open Problems

6.1 Summary

The great challenge for anonymization is to simultaneously protect the privacy of individuals and allow accurate scientific analysis of the data. In this dissertation I have made the new observation that the anonymization does not need to be carried out in the original basis of the data, and that simply choosing a judicious basis can improve some combination of privacy protection, analytic utility, or computational efficiency.

I have given examples of this observation in practice. First, I showed how projecting onto the spectral basis provided by singular value decomposition changes the original cell-swapping algorithm from a weak algorithm of only historical interest to one that provides competitive analytic utility and stronger privacy protection than the state of the art swapping algorithm. Second, I showed how switching to a spectral basis allows the RHS algorithm to overcome the curse of dimensionality that otherwise prohibits its use on high-dimensional data. Third, I showed how clustering in the lower-dimensional, nonlinear basis describing the surface of the manifold on which the data lie allows us to factor the data's distribution and produce an unprecedented combination of high analytic utility and strong privacy protection.

Additionally, I have proposed new measures for privacy protection and analytic utility that are both more general and more informative than existing measures. The measures of prediction distance, prediction ambiguity, and prediction uncertainty quantify how well an attacker can predict the values in a particular original record. They also allow us to gauge the vulnerability of anonymized records to a reidentification attack. I have also proposed the use of kernel principal components to efficiently assess analytic utility, including the preservation of higher-order, nonlinear dependencies in the original data.

And finally, I have proposed the nonoverlapping sample as the reference standard by which we can judge privacy protection and data utility. If our anonymization provides privacy to the original participants that is at least as strong as if we released

data from an entirely different sample of people, I propose that this is sufficient privacy for the original participants. Similarly, if the measures of data utility on the anonymized data are at least as good as on the reference sample, I propose that this is sufficient utility.

This work has application in medical research and survey statistics as noted, but it also may have relevance to commerce. Especially with electronic commerce, retailers are collecting huge databases of customer behavior. These retailers have privacy policies which (sometimes) pledge not to disclose individual customer information. Since individual information is irreversibly removed from anonymized data, releasing it may turn out not to violate these policies, allowing a new source of revenue for many companies. The valuable information of exactly who bought what would not available in anonymized data, but perhaps equally valuable and detailed information on aggregate purchasing behavior, trends, associations, and dependencies could be sold without violating customer privacy.

6.2 Limitations

This method only applies to numeric data, whether continuous, ordinal, or categorical. It is not obvious if it could be applied to non-categorical textual data.

As with many anonymization methods, the usefulness of spectral anonymization is limited to analysis of relationships between variables of a monolithic dataset, although those relationships may be complex and nonlinear. Fortunately, this is a large domain — much medical research, for example, lies therein. But it is not useful for analyses that depend on details of individuals, because those details are deliberately suppressed. We could anonymize supermarket loyalty-card purchases, for example, and still be able to analyze shopping trends, discover promising bundling possibilities, and assess shelf placement effects. But we could not use the anonymized data to generate targeted marketing offers, because it doesn't contain the shopping history of any real person.

Also in common with all methods, my results depend completely on the adequacy of my assessment measures. If an attacker is able to find a measure that distinguishes correct matches or predictions from incorrect ones, and this measure does not depend on distance, ambiguity, or uncertainty between candidates, he may be able to discover some sensitive information. What that measure would be is difficult to imagine, but I have not proven it does not exist.

My privacy measures are also not foolproof. An anonymization algorithm that simply adds a large offset to each element of data would show large prediction distance, low ambiguity, and unchanged uncertainty. By anonymizing and assessing the data in standardized form, we avoid this pitfall, but it shows that these measures can be fooled by deliberate cheating. There may be other operations that clearly provide no anonymity but assess well under my measures.

6.3 Open Problems

Anonymizing in kernel space Partitioned ICA Swapping maps the data to a lower-dimensional, nonlinear space corresponding to the surface of the embedded data manifold in order to find clusters within the data. Once the data are assigned to clusters, the anonymization proceeds by cluster, but in the original space of the data. (Actually, the data are transformed to a spectral basis for anonymization, but the point is this is not the same as the lower-dimensional space found for the clustering.) While I did not emphasize the point earlier, the transformation to the lower dimensional space is actually a kernel transformation similar to the polynomial transformation used to evaluate analytic utility in Chapter 5. It would be a tempting modification to anonymize the data directly in this kernel feature space and then return it to the original space. This would correspond to sliding the data around on the surface of this manifold, which would preserve much, perhaps all, of the data's important structure.

A similarly tempting modification would be, instead of projecting to a lower dimensional space, to project to a *higher*-dimensional space that inherently preserves specific nonlinear dependencies. For example, if we have a dataset with the dimensions $\{x, y\}$, we might operate in the higher-dimensional feature-space basis represented by $\{x^2, xy, y^2, x, y, 1\}$. The covariance matrix in this space would include up to 4^{th}-order moments, and we could anonymize along the eigenvectors of this matrix and preserve those moments. The number of feature-space dimensions can quickly become intractably large for higher-degree interactions, but we have seen how kernel methods can elegantly handle this, since they find the the principal components in feature space without computing actual features [101]. Thus, we might in theory preserve arbitrarily high-order interactions between variables in a clean and principled way. The polynomial degree of the kernel transform would determine the maximum order of preserved interaction, and the number of principal components used would determine how closely those interactions are conserved, trading off with computational complexity.

The difficulty with both of these ideas arises when we try to map the data back into its original space. While we have preserved all desired information up to this point, it is easy to distort it in the reverse transformation. This happens because in the anonymization we are likely to have created points in feature space that do not correspond to any possible point in the original data space, and thus an exact pre-image of the anonymized data does not exist. For example, we could construct a point in feature space where $x = 2$ and $x^2 = 5$. This is a perfectly acceptable point in feature space, but it lies outside the manifold of points that come from data space, and we must make an approximation in order to place it in that space. Identifying the optimal reverse map to data space has come to be known as *the pre-image problem*, and there are several approaches to solving it [114, 115, 116, 117]. These all work well for certain applications, but in my experiments (not reported here) they all

unacceptably distort the information we'd like to preserve in anonymization. In fact, the distortion imposed by the reverse transformation turns out to be the dominant factor in the quality of the anonymization. Identifying a usable reverse map, perhaps in combination with a synergistic anonymization method, would be an intriguing direction for future research.

This direction is perhaps more promising for the problem of returning from the lower-dimensional manifold surface than for returning from the higher-dimensional polynomial kernel space. One may be able to find a way to anonymize on the surface of the manifold while maintaining the constraints of that space, such that projection back into the original space does not suffer from the preimage problem.

Time-series data It is not obvious to me how to extend these methods to anonymize time-series data. The first hurdle is in representing time-series data as a matrix. Instead of one row per subject, we now have a series of rows, with one row per time point. One could concatenate these rows as one big row, but that is not quite satisfying. One could extract features from the time series of each column, and form those features into a row for analysis, but that may have problems as well. This is an intriguing problem that would make for interesting research.

Hierarchical data Some categorical variables are arranged in a hierarchy, for example {`vehicle`, `car`, `Ford`, `Taurus`, ... }. Since the hierarchy is implicit in the semantics of the data, rather than explicit in its structure, it may or may not make sense when encoded as binary variables and interpreted probabilistically as described in Chapter 2. It would be interesting to see if we could continue the probabilistic interpretation with hierarchical variables, or if some extra pre-processing of the dataset needs to be done to preserve the hierarchical semantics.

Very high dimensional data One of the benefits of Partitioned ICA swapping compared to existing methods is that it handles high dimensional data with no greater difficulty than low dimensional data. But it does assume that there are more rows than columns in the data matrix. Both Singular Value Decomposition and Independent Component Analysis can operate on matrices with more columns than rows, but they will never find more components than the smaller of the two. One hot problem these days is anonymizing genetic data, which tends to produce wide, short matrices. Three billion base pairs per human genome with four choices per base pair leads to a very wide matrix. This is reduced if we only look at single nucleotide polymorphisms (also known as SNPs or 'snips'), but even SNPs produce very wide matrices. On top of the theoretical implications of very high dimension, there is also the practical issue of computational efficiency. There are existing ways of improving the efficiency of spectral clustering and kernel principal component analysis [102], but as far as I know, no theoretical efficiency limits have yet been reached.

Tree-dependent Component Analysis A generalization of ICA that relaxes the requirements for a complete factorization of the data is *Tree-dependent Component Analysis* (TCA). TCA seeks to find a set of basis vectors in which the dependencies in the data can be described by a set of trees such as are used in graphical models [118]. Components within the same tree are dependent, and components in different trees are independent. This would allow us to capture more of the structure of the data than pure ICA, if a complete factorization does not exist. Replacing ICA with TCA in partitioned spectral swapping seems a very promising direction for future research.

Assessment measures The assessment measures I propose here capture more detailed information about the quality of the anonymization than previous measures, but I do not claim that they are optimal. (In fact I have given an example above of how they can be cheated.) There is plenty of room for improvement, even by such simple steps as investigating different distance measures or kernels. An exponential kernel, for example, is a generalization of the polynomial kernel (consider the expansion of e^x as powers of x), and may better capture all pertinent information about the analytic utility of the anonymized data. Similarly, research into more useful distance or variance functions for the privacy measures is likely to bear fruit. An attacker will ultimately want to compare candidate matches or predictions in probabilistic terms, and a distance function that captures this would be certainly useful. As it is, I used a Mahalonobis distance, modified for use with mixed continuous/binary data, as a proxy for a probabilistic distance function.

6.4 Conclusion

This dissertation demonstrates that it is possible to provide strong protection against computational disclosure and still faithfully preserve even subtle features of the data. This is a combination that some had guessed was impossible, but it was accomplished in part by re-examining what is really required of an anonymization. Identifying the fundamental requirements of analytic utility and privacy protection reveals that the two do not actually conflict. The requirement for analytic utility is the preservation of the multivariate distribution. The requirement for privacy protection is the conditional independence of variable values in the anonymized dataset from those in the original dataset, given the multivariate distribution. These can coexist quite peacefully, as they do any time one draws two independent samples from the same population. The methods I have proposed here take advantage of this peaceful coexistence to simultaneously satisfy demanding standards of both privacy protection and analytic utility.

Appendix A

Dataset Details

The dataset used in these experiments contained 15 binary variables, representing the categorical variables of Gender, Race, Marital Status, HIV Status, and Previous Drug Use. The categorical variables of Gender, Race, and HIV Status are represented by one binary variable for each category. Categorical variables of Race and Previous Drug Use are represented by selected categories only (Table A.1).

Variable	Count
Male	996
Female	1004
Mexican American	581
Other Hispanic	103
Non-Hispanic White	846
Non-Hispanic Black	407
Other Race	63
Never Married	650
HIV Positive	1
HIV Negative	880
HIV Indeterminate	1
HIV Not Answered	1118
Ever Used Drugs	146
Drugs Unanswered	1167
Ever Used Needle	11

Table A.1: Binary Variables. 'Count' is the number of records with a positive value for that variable.

The dataset contained 13 continuous variables representing laboratory values (Figure A).

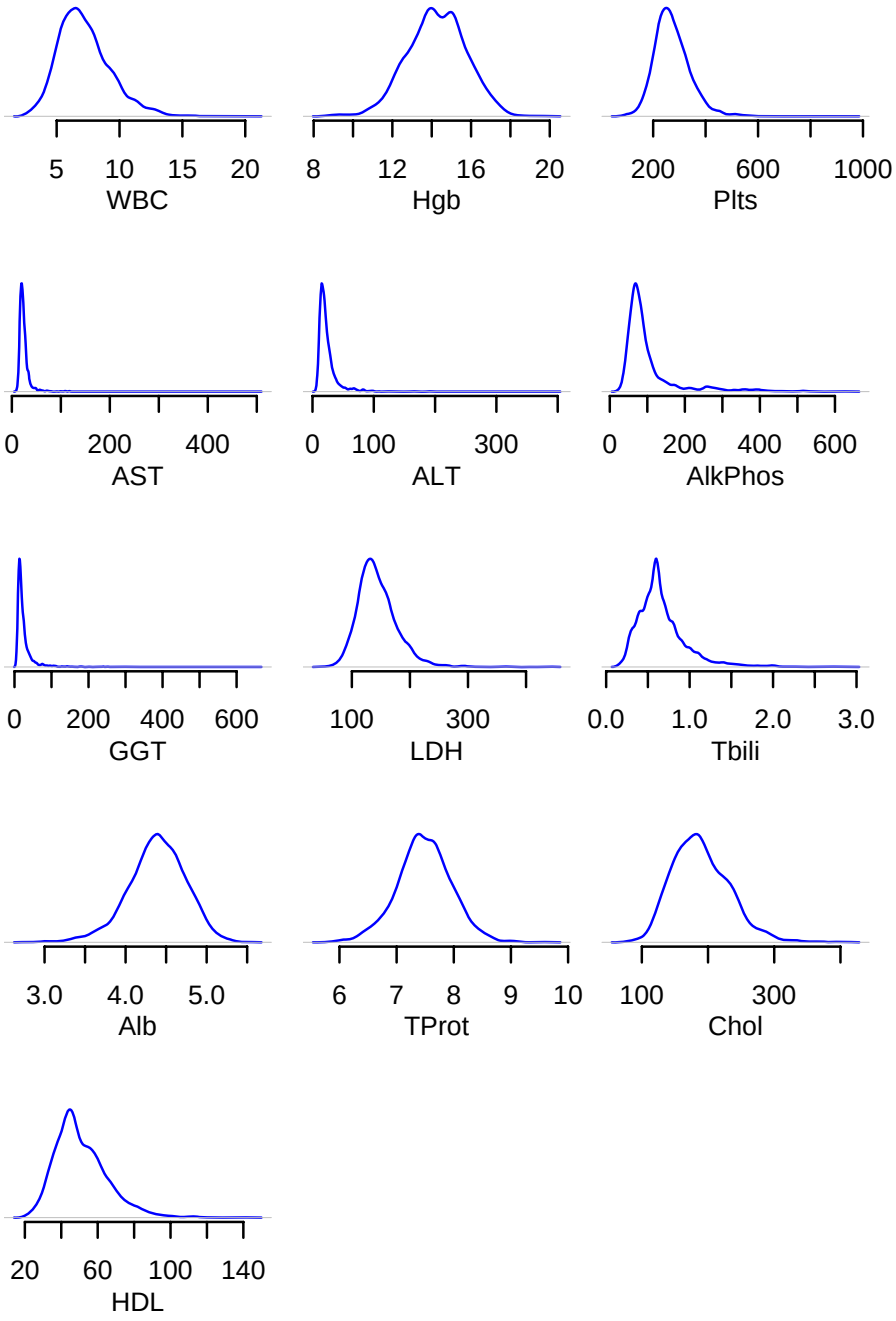


Figure A-1: Continuous Variables. Names are the standard laboratory abbreviations.

Bibliography

[1] Jelke G. Bethlehem, Wouter J. Keller, and Jeroen Pannekoek. Disclosure control of microdata. *J Amer Statist Assoc*, 85(409):38–45, Mar 1990.

[2] Julie R Ingelfinger and Jeffrey M Drazen. Registry research and medical privacy. *N Engl J Med*, 350(14):1452–1453, Apr 2004.

[3] C. A. Welch. Sacred secrets–the privacy of medical records. *N Engl J Med*, 345(5):371–372, Aug 2001.

[4] Jack V. Tu, Donald J. Willison, Frank L. Silver, Jiming Fang, Janice A. Richards, Andreas Laupacis, and Moira K. Kapral. Impracticability of informed consent in the registry of the canadian stroke network. *N Engl J Med*, 350:1414 – 1421, 2004.

[5] L. H. Cox, S. McDonald, and D. Nelson. Confidentiality issues at the united states bureau of the census. *J Off Stat*, 2(2):135–160, 1986.

[6] US Department of Health Office for Civil Rights and Human Services. Medical privacy - national standards to protect the privacy of personal health information.

[7] Roberto J. Bayardo and Rakesh Agrawal. Data privacy through optimal k-anonymization. In *ICDE '05: Proceedings of the 21st International Conference on Data Engineering (ICDE'05)*, pages 217–228, Washington, DC, USA, 2005. IEEE Computer Society.

[8] L. Sweeney. Guaranteeing anonymity when sharing medical data, the Datafly System. In *Proc AMIA Annu Fall Symp*, pages 51–55, 1997.

[9] N. Spruill. The confidentiality and analytic usefulness of masked business microdata. In *Proceedings of the Section on Survey Research Methods*, pages 602–607. American Statistical Association, 1983.

[10] W. A. Fuller. Masking procedures for microdata disclosure limitation. *J Off Stat*, 9(2):383 – 406, 1993.

[11] Charu C. Aggarwal. On k-anonymity and the curse of dimensionality. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 901–909. VLDB Endowment, 2005.

[12] San Jose Medical Group. Alert!! computer theft - information for patients. `http://www.sanjosemed.com/sjmg/Documents/PtInfoSheets.asp`, April 2005.

[13] David Becker. Ucla laptop theft exposes id info. `http://news.zdnet.com/2100-1009_22-5230662.html`, March 2005.

[14] Paul Festa. Id theft fears linger after laptop returned. `http://news.zdnet.com/2100-1009_22-5501200.html`, Dec 2004.

[15] CNW Group. Stolen laptop sparks order by Commissioner Cavoukian requiring encryption of identifiable data: Identity must be protected.

[16] A. Rubin. Records no longer for doctors' eye only. Los Angeles Times, Sep 1 1998. p. A1.

[17] Houston Chronicle. Selling singer's files gets man 6 months. December 2, 2000, p. A2.

[18] Jessica Heslam. Brigham sent bank new moms' records.

[19] J. Constanzo. Ihc sues over misplaced records. The Deseret News, December 2 1998.

[20] Associated Press. Medical records found scattered in downtown Lewiston. `http://www.oregonlive.com/printer/printer.ssf?/base/news/13/1114757604289930.xml\&storylist=orwashington`, April 2005.

[21] Associated Press. Medical records fall out of vehicle, blown through street. Associated Press, May 26 2000.

[22] Tore Dalenius. Towards a methodology for statistical disclosure control. *Statistisk Tidskrift*, 5:429–444, 1977.

[23] Leon Willenborg and Ton de Waal. *Statistical Disclosure Control in Practice*, volume 111 of *Lecture Notes in Statistics*. Springer, Voorburg, The Netherlands, 1996.

[24] William E. Winkler. Masking and re-identification methods for public-use microdata: Overview and research problems. Research Report Series, Statistics 2004-06, Statistical Research Division, U.S. Bureau of the Census, Washington DC, October 2004.

[25] Charu C. Aggarwal. A condensation approach to privacy preserving data mining. *Lecture Notes in Comput Sci*, 2992:183–199, Jan 2004.

[26] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *PODS '05: Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138, New York, NY, USA, 2005. ACM Press.

[27] Shuchi Chawla, Cynthia Dwork, Frank McSherry, Adam Smith, and Hoetek Wee. Toward privacy in public databases. In *Theory of Cryptography Conference TCC 2005*, Cambridge, MA, Feb 2005.

[28] Leon Willenborg and Ton de Waal. *Elements of Statistical Disclosure Control*. Number 155 in Lecture Notes in Statistics. Springer, 2001.

[29] J. Domingo-Ferrer, editor. *Inference Control in Statistical Databases*. Springer, New York, 2002.

[30] Matteo Fischetti and Juan Jose Salazar. Modeling and solving the cell suppression problem for linearly-constrained tabular data. In *Proc Statistical Disclosure Protection '98*, March 1998.

[31] Dorothy E. Denning. Secure statistical databases with random sample queries. *ACM Trans Database Syst*, 5(3):291–315, 1980.

[32] Leon Willenborg and Ton de Waal. *Elements of Statistical Disclosure Control*, chapter 2.1 Predictive Disclosure, pages 42 – 46. In *Lecture Notes in Statistics* [28], 2001.

[33] Leon Willenborg and Ton de Waal. *Elements of Statistical Disclosure Control*, chapter 2.5 Reidentification Risk, pages 46 – 51. In *Lecture Notes in Statistics* [28], 2001.

[34] W. E. Winkler. Re-identification methods for evaluating the confidentiality of analytically valid microdata. *Research in Official Statistics*, 1:87 – 104, 1998.

[35] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, May 1998.

[36] Ivan P. Fellegi and Alan B. Sunter. A theory for record linkage. *J Amer Statist Assoc*, 64(328):1183–1210, December 1969.

[37] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J Roy Stat Soc B Met*, 39:1 – 38, 1977.

[38] Winkler. Advanced methods for record linkage. In *Proceedings of the Section on Survey Research Methods*, pages 467 – 472. American Statistical Association, 1994.

[39] William E. Yancey, William E. Winkler, and Robert H. Creecy. Disclosure risk assessment in perturbative microdata protection. In *Inference Control in Statistical Databases, From Theory to Practice*, pages 135–152, London, UK, 2002. Springer-Verlag.

[40] Matthew A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *J Amer Statist Assoc*, 84(406):414 – 420, Jun. 1989.

[41] J. Domingo-Ferrer, J. Mateo-Sanz, and V. Torra. Comparing sdc methods for microdata on the basis of information loss and disclosure risk. In *Proceedings of NTTS and ETK*, 2001.

[42] S. E. Fienberg. Confidentiality and disclosure limitation methodology: Challenges for national statistics and statistical research. Working Paper 668, Carnegie Mellon Department of Statistics, Pittsburgh, PA, 1997.

[43] Staal A. Vinterbo. Privacy: A machine learning view. *IEEE T Knowl Data En*, 16(8):939–948, 2004.

[44] Josep Domingo-Ferrer and Vicen Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Min Knowl Discov*, 11(2):195–212, Sep 2005.

[45] Shuchi Chawla, Cynthia Dwork, Frank McSherry, Adam Smith, and Hoeteck Wee. Towards privacy in public databases. In *Second Theory of Cryptography Conference, TCC 2005*, Cambridge, MA, February 2005.

[46] Josep M. Mateo-Sanz, Josep Domingo-Ferrer, and Francesc Seb. Probabilistic information loss measures in confidentiality protection of continuous microdata. *Data Min Knowl Discov*, 11(2):181–193, Sep 2005.

[47] S.V. Gomatam and A. Karr. On data swapping of categorical data. Technical Report 131, National Institute of Statistical Sciences, Research Triangle Park, NC, Jan 2003.

[48] R. Brand. *Inference Control in Statistical Databases*, chapter Microdata protection through noise addition, pages 97 – 116. Springer, 2002.

[49] Dakshi Agrawal and Charu C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the 20th Symposium on Principles of Database Systems*, 2001.

[50] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 439–450, New York, NY, USA, 2000. ACM Press.

[51] Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar. Random-data perturbation techniques and privacy-preserving data mining. *Knowl Inf Syst*, 7(4):387 – 414, May 2005.

[52] J. Kim. A method for limiting disclosure in microdata based on random noise and transformation. In *Proceedings of the Section on Survey Research Methods*, pages 370–374. American Statistical Association, 1986.

[53] Patrick Tendick and Norman Matloff. A modified random perturbation method for database security. *ACM Trans Database Syst*, 19(1):47–63, 1994.

[54] G. Sullivan and W. A. Fuller. The use of measurement error to avoid disclosure. In *Proceedings of the Survey Research Methods Section*, pages 802 – 807. American Statistical Association, 1989.

[55] Gary Sullivan and Wayne A. Fuller. Construction of masking error for categorical variables. In *Proceedings of the Survey Research Methods Section*. American Statistical Association, 1990.

[56] Gina Roque. Application and analysis of the mixture-of-normals approach to masking public-use microdata. Manuscript obtained from `http://theory.stanford.edu/~nmishra/cs369-2004.html`, 2003.

[57] Roderick J. A. Little. Statistical analysis of masked data. *J Off Stat*, 9(2):407 – 426, 1993.

[58] A. G. de Waal, A. J. Hundepool, and L. C. R. J. Willenborg. Argus: software for statistical disclosure control of microdata. In *Proceedings of the 1996 Annual Research Conference*. U. S. Bureau of the Census, 1996.

[59] Vijay S. Iyengar. Transforming data to satisfy privacy constraints. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 279–288, New York, NY, USA, 2002. ACM Press.

[60] Leon Willenborg and Ton de Waal. *Elements of Statistical Disclosure Control*, page 26. In *Lecture Notes in Statistics* [28], 2001.

[61] A. G. de Waal and L. C. R. J. Willenborg. Optimal local suppression in microdata. *J Off Stat*, 14:421 – 435, 1998.

[62] A. Ohrn and L. Ohno-Machado. Using Boolean reasoning to anonymize databases. *Artif Intell Med*, 15(3):235–254, Mar 1999.

[63] Stephen E. Fienberg and Julie McIntyre. Data swapping: variations on a theme by dalenius and reiss. *Lecture Notes in Comput Sci*, 3050:14 – 29, 2004.

[64] J.P. Reiss. Practical data swapping: the first steps. *ACM Trans Database Syst*, 9(1):20 – 37, 1984.

[65] William E. Winkler. Re-identification methods for masked microdata. Research Report Series, Statistics 2004-03, Statistical Research Division, U.S. Bureau of the Census, Washington DC, April 2004.

[66] R. Griffin, A. Navarrow, and L. Flores-Baez. Disclosure avoidance for the 1990 census. In *Proceedings of the Section on Survey Research Methods*, pages 516 – 521. American Statistical Association, 1989.

[67] Stephn E. Fienberg, Russell J. Steele, and Udi E. Makov. Statistical notions of data disclosure avoidance and their relationship to traditional statistical methodology: Data swapping and loglinear models. In *Proceedings of Bureau of the Census 1996 Annual Research Conferrence*, pages 87 – 105. US Bureau of the Census, March 1996.

[68] Laura Zayatz. *Inference Control in Statistical Databases*, chapter SDC in the 2000 U.S. Decennial census, pages 183 – 202. Springer, Berlin, 2002.

[69] R. Moore. Controlled data swapping techniques for masking public use data sets. Report rr96/04, U.S. Bureau of the Census, Statistical Research Division, 1996.

[70] Michael Carlson and Mickael Salabasis. A data-swapping technique using ranks — a method for disclosure control. *Research in Official Statistics*, 6(2):35 – 64, 2002.

[71] Krishnamurty Muralidhar and Rathindra Sarathy. Data shuffling — a new masking approach for numerical data. *Manage Sci*, 52(5):658 – 670, May 2006.

[72] Akimichi Takemura. Local recoding and record swapping by maximum weight matching for disclosure control of microdata sets. *J Off Stat*, 18(2):275 – 289, 2002.

[73] J. Domingo-Ferrer and J. M. Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *IEEE T Knowl Data En*, 14(1):189–201, 2002.

[74] D. Defays and M. N. Anwar. Masking microdata using micro-aggregation. *J Off Stat*, 14(4):449 – 461, 1998.

[75] Anna Oganian and Josep Dommingo-Ferrer. On the complexity of optimal microaggregation for statistical disclosure control. *Statistical Journal of the United Nations Economic Commission for Europe*, 18(4):345–354, Dec 2001.

[76] Stephen Lee Hansen. A polynomial algorithm for optimal univariate microaggregation. *IEEE T Knowl Data En*, 15:1043 – 1044, 2003.

[77] Josep Domingo-Ferrer and Vincen Torra. *Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies*, chapter A quantitative comparison of disclosure control methods for microdata, pages 111 – 134. North-Holland, 2001.

[78] J. H. Ward. Hierarchical grouping to optimize an objective function. *J Amer Statist Assoc*, 58:236 – 244, 1963.

[79] Josep A. Sanchez, Julia Urrutia, and Enric Ripoll. Trade-off between disclosure risk and information loss using multivariate microaggregation: a case study on business data. *Lecture Notes in Comput Sci*, 3050:307–322, 2004.

[80] Vicen Torra and Sadaaki Miyamoto. Evaluating fuzzy clustering algorithms for microdata protection. *Lecture Notes in Comput Sci*, 3050:175–186, 2004.

[81] M. Laszlo and S. Mukherjee. Minimum spanning tree partitioning algorithm for microaggregation. *IEEE T Knowl Data En*, 17(7):902–911, 2005.

[82] Gordon Sande. Exact and approximate methods for data directed microaggregation in one or more dimensions. *Int J Uncertain Fuzz*, 10(5):459–476, Oct 2002.

[83] Harold N. Gabow. An efficient implementation of edmonds' algorithm for maximum matching on graphs. *J ACM*, 23(2):221–234, 1976.

[84] G. Ausiello, P. Crescenzi, G. Gambosi, B. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation: Combinatorial Optimazation Problems and Their Approximability Properties*, page 375. Springer, 2003.

[85] George T. Duncan, Sallie Keller-McNulty, and S. Lynne Stokes. Disclosure risk vs. data unility through the r-u confidentiality map in multivariate settings. Working Paper 2005-16, H. John Heinz III School of Public Policy & Management, Carnegie Mellon University, 2005. Available from `http://www.heinz.cmu.edu/wpapers/author.jsp?id=gd17`.

[86] C. K. Liew, U. J. Choi, and C. J. Liew. A data distortion by probability distribution. *ACM Trans Database Syst*, 10(3):395 – 411, 1985.

[87] Peter Kooiman. Comment. *J Off Stat*, 14(4):503 – 508, 1998.

[88] Steven P. Reiss, Mark J. Post, and Tore Dalenius. Non-reversible privacy transformations. In *PODS '82: Proceedings of the 1st ACM SIGACT-SIGMOD symposium on Principles of database systems*, pages 139–146, New York, NY, USA, 1982. ACM Press.

[89] Ramesh A. Dandekar, Michael Cohen, and Nancy Kirkendall. Sensitive micro data protection using latin hypercube sampling technique. In *Inference Control in Statistical Databases, From Theory to Practice*, pages 117–125, London, UK, 2002. Springer-Verlag.

[90] Robert T. Clemen and Terence Reilly. Correlations and copulas for decision and risk analysis. *Manage Sci*, 45(2):208–224, 1999.

[91] S. Polettini. Maximum entropy simulation for microdata protection. *Statist Comput*, 13(4):307 – 320., 2003.

[92] Donald B. Rubin. Discussion: Statistical disclosure limitation. *J Off Stat*, 9(2):461–468, 1993. Also cited as "Satisfying confidentiality constraints through the use of synthetic multiply-imputed microdata".

[93] T.E. Raghunathan, J.P. Reiter, and D.B. Rubin. Multiple imputation for statistical disclosure limitation. *J Off Stat*, 19(1):1–16, 2003.

[94] J. L. Schafer. Multiple imputation: a primer. *Statistical Methods Med Res*, 8(1):3 – 15, 1999.

[95] J.P. Reiter. Satisfying disclosure restrictions with synthetic data sets. *J Off Stat*, 18(4):531–543, 2002.

[96] Sallie Keller-McNulty and George T. Duncan. Bayesian insights on disclosure limitation: Mask or impute. Technical Report LA-UR-00-3771, Los Alamos National Laboratory, 2000.

[97] F. Liu and R.J.A. Little. Selective multiple imputation of keys for statistical disclosure-control in microdata. In *Proceedings of the Section on Survey Research Methods*. American Statistical Association, 2002.

[98] Frank J. Massey, Jr. The Kolmogorov-Smirnov test for goodness of fit. *J Amer Statist Assoc*, 46(253):68 – 78, Mar 1951.

[99] W. J. Conover, editor. *Practical Nonparametric statistics*, chapter 6.3 Tests on two independent samples, pages 368 – 376. Wiley, 2nd edition, 1980.

[100] B. Scholkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[101] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput*, 10:1299–1319, 1998.

[102] Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. *J Mach Learn Res*, 3:1–48, 2002.

[103] Gilbert Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge, Wellesley, MA, third edition, 2003.

[104] Centers for Disease Control and Prevention (CDC). National Center for Health Statistics (NCHS). National Health and Examination Survey (nhanes).

[105] Thomas A Lasko, Jui G Bhagwat, Kelly H Zou, and Lucila Ohno-Machado. The use of receiver operating characteristic curves in biomedical informatics. *J Biomed Inform*, 38(5):404–415, Oct 2005.

[106] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensional reduction and data representation. *Neural Comput*, 15:1373 – 1396, 2003.

[107] Yair Weiss. Segmentation using eigenvectors: A unifying view. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, pages 975 – 982, Washington, DC, USA, 1999. IEEE Computer Society.

[108] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[109] Lawrence K. Saul and Sam T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *J Mach Learn Res*, 4:119–155, 2003.

[110] A Hyvarinen and E Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13:411–430, 2000.

[111] Aapo Hyvarinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. Wiley, New York, 2001.

[112] J. L. Marchini, C. Heaton, and B. D. Ripley. *fastICA: FastICA algorithms to perform ICA and Projection Pursuit*, 2006. R package version 1.1-8.

[113] Alexandros Karatzoglou, Alex Smola, Kurt Hornik, and Achim Zeileis. kernlab – an S4 package for kernel methods in R. *J Stat Softw*, 11(9):1–20, 2004.

[114] Bernhard Scholkopf, Sebastian Mika, Chris J. C. Burges, Phiulipp Knirsch, Klaus-Robert Muller, Gunnar Ratsch, and Alex J. Smola. Input space vs. feature space in kernel-based methods. *IEEE transactions on neural networks*, 10(5):1000 – 1017, Sep 1999.

[115] James T. Kwok and Ivor W. Tsang. The pre-image problem in kernel methods. *IEEE transactions on neural networks*, 15(6):1517 – 1525, Nov 2004.

[116] G. H. Bakir, J. Weston, and B. Scholkopf. Learning to find pre-images. In S. Thrun and B. Saul, L. Scholkopf, editors, *Advances in Neural information processing systems 16*, pages 449 – 456, 2004.

[117] Wei-Shi Zheng and Jian-huang Lai. Regularized locality preserving learning of pre-image problem in kernel principal component analysis. In *The 18th International Conference on Pattern Recognition (ICPR'06)*, 2006.

[118] Francis R. Bach and Michael I. Jordan. Beyond independent components: trees and clusters. *J Mach Learn Res*, 4:1205–1233, 2003.