The Dissertation Committee for Yubin Park
certifies that this is the approved version of the following dissertation:

Privacy-aware Publication and Utilization of Healthcare Data

Committee:

Joydeep Ghosh, Supervisor

Haris Vikalo

Sriram Vishwanath

Mia K. Markey

Mallikarjun Shankar

Privacy-aware Publication and Utilization of Healthcare Data

by

Yubin Park, B.S.; M.S.E.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN
August 2014

Dedicated to my family.

Acknowledgments

I am deeply grateful to my advisor, Professor Joydeep Ghosh for his guidance and help. I also thank my colleagues, Sreangsu Acharyya, Sanmi Koyejo, Ayan Acharya, Cheng H. Lee, Joyce Ho, Suriya Gunasekar, Rajiv Khanna, Shalmali Joshi, and Avradeep Bhowmik for their helpful advice and support.

I would like to express my gratitude to my committee members, Haris Vikalo, Sriram Vishwanath, Mia K. Markey, and Mallikarjun Shankar.

Finally, I would like to thank my family for their constant support.

Privacy-aware Publication and Utilization of Healthcare Data

Publication No. _____

Yubin Park, Ph.D.
The University of Texas at Austin, 2014

Supervisor: Joydeep Ghosh

Open access to health data can bring enormous social and economical benefits. However, such access can also lead to privacy breaches, which may result in discrimination in insurance and employment markets. Privacy is a subjective and contextual concept, thus it should be interpreted from both systemic and information perspectives to clearly understand potential breaches and consequences. This dissertation investigates three popular use cases of healthcare data: specifically, 1) synthetic data publication, 2) aggregate data utilization, and 3) privacy-aware API implementation. For each case, we develop statistical models that improve the privacy-utility Pareto frontier by leveraging a variety of machine learning techniques such as information theoretic privacy measures, Bayesian graphical models, non-parametric modeling, and low-rank factorization techniques. It shows that much utility can be extracted from health records while maintaining strong privacy guarantees and protection of sensitive health information.

Table of Contents

Chapter 1

Introduction

In 2011, the World Health Organization reported that the total healthcare spending of the United States recorded the highest in the world [1]. The Department of Health and Human Services expects that such spending will continue to increase [74]. It is increasingly believed that "data-driven" approaches can help reduce current healthcare expenditure growth in the United States [93]. The Institute of Medicine mentioned that data science is a critical component for continuous healthcare quality improvement [69] (see Figure 1.1). Effective data-driven solutions can prevent cost leakage and waste in healthcare systems [10] as well as can provide "best care at lower cost" [125].

1.1 Trade-off between Utility and Privacy

McKinsey & Company, Inc. reported that "open data" has the potential to generate more than $300 billion a year in the healthcare domain alone [88]. The United States government has launched several open data initiatives, for example data.gov, healthdata.gov, and data.cms.gov. The Health Data Consortium, a collaboration among government, non-profit, and private sector organizations, has been formed to foster the availability and innovative

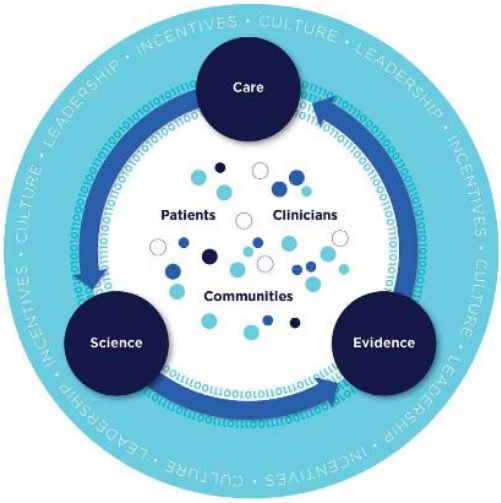


Figure 1.1: Data science is a key component in improving healthcare quality and lowering costs. [Source]: Institute of Medicine, Best Care at Lower Cost: The Path to Continuously Learning Health Care in America.

use of open health data. U.S. Open Data Action Plan [98], a recent article from the Office of Science and Technology Office, starts by stating that

> Throughout his Administration, President Obama has articulated a vision of the U.S. Government managing information as a national asset and opening up its data, where possible, as a public good to advance government efficiency, improve accountability, and fuel private sector innovation, scientific discovery, and economic growth. ...

Sharing personal health information can bring enormous economical benefits. In fact, several legal theorists have argued that privacy is overrated

compared to the social gains of sharing health information. Judge Posner viewed that privacy can be used "to manipulate the world around them by selective disclosure of facts about themselves" [105]. Professor Epstein stated that regulations on data privacy may create "an elaborate set of cross-subsidies that reduces the total level of social wealth as it transfers wealth between parties" [48]. In a complex healthcare system, however, the negative consequences for open access of health information overwhelm the idealistic economical benefits [124]. For example, insurance companies and employers can maliciously utilize such data to increase their revenues, discriminating out unhealthy subpopulations. Thus, there exists a delicate equilibrium point between utility and privacy, and an extreme point cannot be a viable solution.

Privacy is a subjective and contextual concept, and it conveys different connotations and interpretations in different fields; e.g. banking and healthcare sectors focus on different privacy aspects [46]. In the healthcare sector, the definition of privacy is commonly accepted as "a person's right and desire to control the disclosure of their personal health information" [116], where the type of health information ranges from a person's identity to disease/medication history. The concept of healthcare data privacy sometimes extends to cover organizational information such as hospitals and insurance companies, not just patient information [5].

Contrary to the systemic views on medical privacy, in the computer science and statistics literature, privacy is often approached from an information theoretic perspective in an attempt to quantify the level of privacy

[135, 4]. Popular privacy metrics include k-anonymity [128], l-diversity [87], and ϵ-differential privacy [44]. Different privacy measures assume different access settings and attack scenarios; for example, k-anonymity and l-diversity fit in a data publication setting, and ϵ-differential privacy is motivated from the statistical database literature. Privacy, especially in healthcare, should be interpreted from both systemic and information perspectives to clearly understand potential breaches and consequences [75].

1.2 Current Practices and Limitations

In recent years, both public and private sectors have released and published an incredible amount of data that were previously not accessible. For example, data.gov, the open data initiative of the U.S. government, has curated more than 80,000 datasets from over 200 organizations. There are multiple non-profit organizations catalyzing and promoting the open data movement, such as Open Data Institute founded by Sir Tim Berners-Lee and Professor Nigel Shadbolt, and Open Knowledge founded by Rufus Pollock. Several state governments, e.g. Texas, New York, and California, have published patient discharge records and billing information for research purposes.

Public use data typically need to to satisfy two competing objectives: maintaining relevant statistical properties of the original data and protecting privacy of individuals. To address these two goals, various statistical disclosure limitation techniques have been developed [135]. Some popular disclosure techniques are data swapping [31, 50], top-coding, feature generalization, and

additive random noise with measurement error models [59]. Each method has distinct utility and risk aspects. In practice, however, it is extremely difficult to balance out utility and privacy. Overly privacy-protecting data would provide not so much useful information, and easily accessible data are vulnerable to privacy attacks. We visit two real examples for both cases.

Case Study 1.2.1 (EHR and HIE)**.** In 2009, the United States government enacted the Health Information Technology for Economic and Clinical Health Act (HITECH) that includes an incentive program totaling up to $27 billion for the adoption and meaningful use of Electronic Health Records (EHRs). Health information exchanges (HIE) have emerged to facilitate the meaningful use of health information by sharing and exchanging somewhat disparate and distributed EHRs. According to HITECH, the meaningful use of EHRs can help "improve care coordination, reduce disparities, engage patients and their families, and improve population and public health" [24]. Such meaningful use can only be achieved through carefully controlled sharing and exchanging of personal health information and complying with existing regulations such as Health Insurance Portability and Accountability Act (HIPAA), otherwise the privacy of patients may be severely damaged. *In the US, 75% of patients have expressed concerns about un-informed sharing of their health information [110, 35], possibly due to the frequent data breaches in medical institutions [65].*

Case Study 1.2.2 (Claims and Billing data)**.** One of the most notable and immediate impacts of the Patient Protection and Affordable Care Act (PPACA,

also called Obamacare) has been a surge of interest in modeling and predicting hospital costs in the USA. In particular, to set up and run an Affordable Care Organization (ACO), a type of health care organization greatly encouraged by PPACA, being able to predict such costs is critical since ACOs are reimbursed for a fixed amount per patient, which is dramatically different from the fee-for-service model that has ruled healthcare in the USA all these years. Centers for Medicare & Medicaid Services recently published Medicare Provider Utilization and Payment Data [23] that contain average price information per procedure code. Recent studies have revealed that hospital bills in the United States vary greatly across regions and hospitals [92, 12]. Although such comparisons may help bringing transparency to the current healthcare landscape, comparing hospital charges based on procedure billing codes overlook the true nature of the cost. The procedure codes used in billing systems are abstracted forms of actually performed procedures, and such codes do not capture the full information about a patient's conditions, complications, and types of facilities. Regarding the cost variation across hospitals, teaching hospitals tend to treat more critical patients. In essence, estimating hospital charges is a multi-faceted problem, and it is also important to discriminate costs that are legitimate versus those that are primarily due to bad practices/management, fraud, etc. *To effectively address the cost problems in healthcare, we need individual-level public use files with a comprehensive set of relevant variables.*

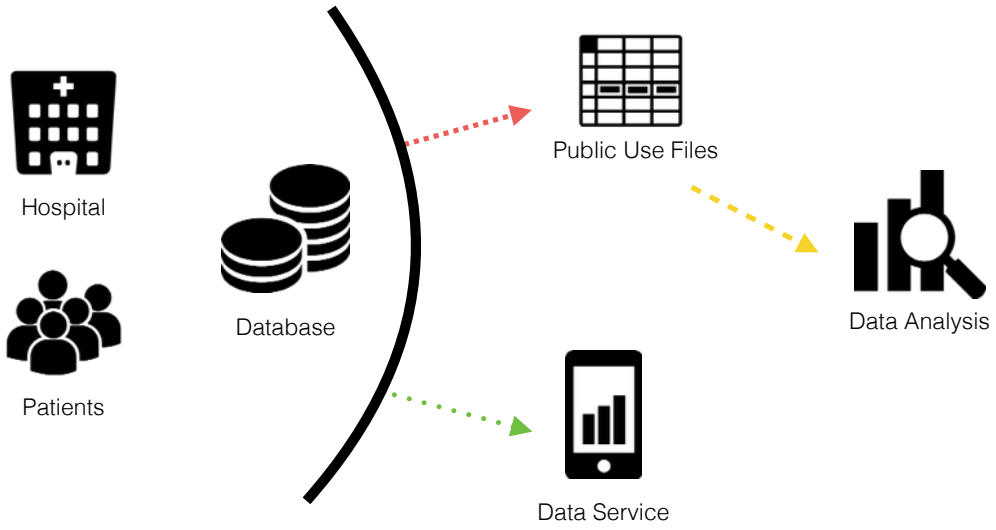


Figure 1.2: Data privacy from three different perspectives: data publisher (red), data analyst (yellow), and data API (green).

1.3 Our Approaches

The pursuits for utility and privacy in sharing healthcare data are almost Pareto-efficient; the utility of data cannot be increased without sacrificing the privacy of data. In Case Study 1, Centers for Medicare & Medicaid Services released only a few variables, because providing other variables may increase the risk for correlation attacks and linking attacks. On the other hand, in Case Study 2, patients are worried about un-informed sharing of health information, since their electronic health records can be readily accessible by multiple physicians. These two cases are only a subset of multiple use case scenarios. Problems with privacy and utility are, in fact, even more complicated, because

each use case exhibits distinct utility and risk perspectives.

In this dissertation, we investigate three popular use cases of healthcare data (see Figure 1.2), and propose statistical models that can improve the privacy-utility Pareto frontier for each case. Specifically, we focus on:

1. **Data Publication**: Privacy-preserving mechanisms for releasing public use files have been extensively studied in the statistics and computer science literature. Even if all personal identifiable information is removed, a publicly released dataset can be linked using some other variables revealing identities of a group of people. For example, in 2006, Netflix, an online DVD-rental and streaming service, released an "anonymized" training dataset with random user IDs. The dataset was designed for a competition for improving the company's recommender system. Narayanan and Shmatikov [96], however, linked the Internet Movie Database (IMDB) dataset with the Netflix dataset using movie rating information, then de-anonymized a subset of the Netflix competition dataset.

   The generation of synthetic data [119] is a popular public data release methodology. For example, multiple imputation, which was originally developed to impute missing values in survey responses [118], can also be used to generate either partially or fully synthetic data. As synthetic data preserves the structure and resolution of the original data, preprocessing steps and analytical procedures on synthetic data can be effortlessly transferred to the original data. This aspect has contributed to adoption of synthetic data in diverse research areas.

We propose a categorical data synthesizer algorithm that guarantees a quantifiable disclosure risk. Our algorithm, named Perturbed Gibbs Sampler (PeGS), can handle high-dimensional categorical data that are intractable if represented as contingency tables. PeGS involves three intuitive steps: 1) disintegration, 2) noise injection, and 3) synthesis. We first disintegrate the original data into building blocks that (approximately) capture essential statistical characteristics of the original data. This process is efficiently implemented using feature hashing and nonparametric distribution approximation. In the next step, an optimal amount of noise is injected to the estimated statistical building blocks to guarantee differential privacy or l-diversity. Finally, synthetic samples are drawn using a customized Gibbs sampler.

2. **Utilization of Public Use File**: In the past few years, the government and other agencies have publicly released a prodigious amount of data that can be potentially mined to benefit the society at large. However, data such as health records are typically only provided at aggregated levels (e.g. per State, per Hospital Referral Region, etc.) to protect privacy. Unfortunately aggregation can severely diminish the utility of such data when modeling or analysis is desired at a per-individual basis. The use of aggregate data is typically limited to group-level studies, often referred to as ecological studies for historic reasons.

   Applying the result from aggregate data to individual-level inference often results in the classic problem of ecological fallacy [117]. Ecolog-

ical fallacy occurs when aggregate-level statistics are misinterpreted as individual-level inferences. For example, the high correlation between "per capita consumption of dietary fat" and "breast cancer" in different countries [21] does not necessarily imply that dietary fat causes breast cancer.

So, not surprisingly, despite the increasing abundance of aggregate data, there have been very few successful attempts in exploiting them for individual-level analyses. We introduce LUDIA, a novel low-rank approximation algorithm that utilizes aggregation constraints in addition to auxiliary information in order to estimate or "reconstruct" the original individual-level values from aggregate data. If the reconstructed data are statistically similar to the original individual-level data, off-the-shelf individual-level models can be readily and reliably applied for subsequent predictive or descriptive analytics. LUDIA is more robust to nonlinear estimates and random effects than other reconstruction algorithms. It solves a Sylvester equation and leverages multi-level (also known as hierarchical or mixed-effect) modeling approaches efficiently. A novel graphical model is also introduced to provide a probabilistic viewpoint of LUDIA. Experimental results using a Texas inpatient dataset show that individual-level data can be reasonably reconstructed from county-, hospital-, and zip code-level aggregate data. Several factors affecting the reconstruction quality are discussed, along with the implications of this work for current aggregation guidelines.

3. **Implementation of Privacy-preserving APIs**: In recent years, a large portion of enterprise databases have been migrated to cloud computing environments, and internal databases are commonly accessed through Application Programming Interfaces (APIs). The applications of APIs include a wide range of Software as a Service (SaaS) applications such as recommender systems, prediction algorithms, and data management services. A data access through API is substantially different from traditional publication/utilization scenarios; API outputs are released, not data. Thus, a novel approach to address the privacy risk for an API call is required.

   There have been several privacy breaches reported in data APIs. Calandrino et al. [19] showed that passive observations of Amazon.com's collaborative filtering outputs, which can be viewed as a data-mining API, can reveal customers' transaction records. Narayanan and Shmatikov [97] showed that topological information, which cab be obtained through Facebook and Twitter APIs, can be used to de-anonymize social network data. Preventing privacy breaches from data APIs is critical as the web is becoming more social and personalized.

   In this dissertation, we demonstrate a proper implementation of privacy-preserving risk score APIs for patients. Risk scores are computed using an ensemble of Differentially Private α-Trees (DPaT). α-Tree is a generalization of C4.5 that uses α-divergence as its splitting criterion. We develop an algorithm that constructs 0-differentially private decision tree

structure, and then ϵ-differential Laplace noise is added to an API output. We show that an ensemble of DPaTs can increase the accuracy of the API outputs while adhering to ϵ-differential privacy.

In Chapter 2, we overview traditional privacy-preserving approaches for public use files. Texas inpatient discharge datasets are used to demonstrate variable reduction, generalization and suppression, and synthesizing techniques. Chapter 3 introduces our privacy-preserving synthesizing mechanism (PeGS), which is a promising alternative for preparing public use files, and then Chapter 4 explains a novel aggregate data utilization technique (LUDIA). We discuss an implementation method for privacy-preserving decision tree APIs (DPaT) in Chapter 5, and summarize and discuss future work in Chapter 6.

Chapter 2

Background

This chapter describes a variety of methods that have been developed and deployed for modifying health-related data sets before data release. We use the Texas Inpatient Public Use Data File from the Texas Department of State Health Services (DSHS, [130]) to concretely illustrate existing approaches for preparing public use files. Hospital billing records collected from 1999 to 2007 are publicly available through their website. Each yearly dataset contains about 2.8 millions events with more than 250 features. Except for a few exempt hospitals, all the hospitals in Texas reported inpatient discharge events to DSHS. This chapter uses the inpatient records from the fourth quarter of 2006, and we specifically focus on the natural delivery events from Parkland Memorial Hospital. The dataset is already anonymized, and does not contain any identifiable information such as name, social security number, and driver license number.

2.1 Variable Reduction and Coarsening

Let us assume that a group of researchers submitted a pilot study proposal about modeling the relationship between demographic factors (sex,

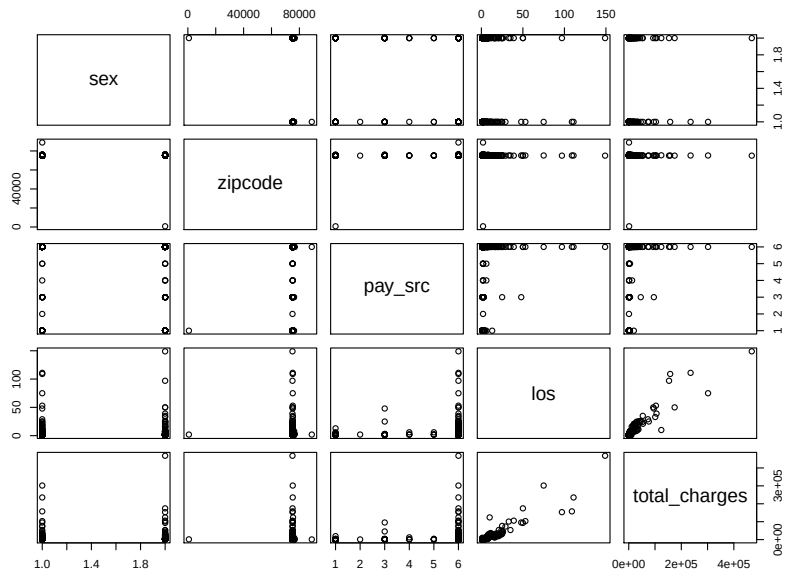


Figure 2.1: Cross-Scatter Plots of the original Texas Inpatient Data.

address), insurance, and hospital charges. Our objective is to publish this dataset for the specified research objective while protecting patients' privacy. We first remove irrelevant variables for the research objective except for five research-related variables: sex (of an infant), zip code, payment source (primary insurance), length of stay, and total charges. Before applying actual privacy-preserving algorithms, the first step is to check the characteristics of the data. Figure 2.1 shows the cross-scatter plots of the original data. As can be seen, there is one missing zip code record (`zipcode=0`), and very few patients paid more than \$10K. Such outliers and rare events can be vulnerable to linking attack (see Section 2.2), thus we filter out these records. The "total charges" variable contains the original numeric scale dollar values. Such nu-

meric variables tend to have many unique entries, which can be easily utilized in linking attack as well. Therefore, we bin the original numeric values of total charges into 20 ranges: $[0, 500), [500, 1000), \ldots [9500, 10000)$.

Table 2.1: Summary Statistics of Texas Inpatient Data.

sex	zipcode	pay_src	los	total_charges
F:648	75217 : 82	09: 155	Min. :0.000	Min. : 0
M:844	75211 : 79	11: 1	1st Qu.:1.000	1st Qu.:1000
	75220 : 79	12: 30	Median :1.000	Median :1000
	75061 : 68	15: 3	Mean :1.068	Mean :1160
	75228 : 54	HM: 5	3rd Qu.:1.000	3rd Qu.:1000
	75231 : 54	MC:1298	Max. :6.000	Max. :9500
	(Other):1076			

Table 2.1 illustrates overall summary statistics of this preprocessed dataset. From the total 1432 patients, 1298 patients were paid by the Medicaid program (`pay_src=MC`), and 155 patients self-paid (`pay_src=09`). On average, patients stayed 1.068 days (`Mean los=1.068`), and paid 1160 dollars (`Mean total_charges=1160`). Figure 2.2 shows the cross-scatter plots of the preprocessed data. As can be seen, we effectively removed easily identifiable data points by coarsening and truncating data.

These simple procedures are, however, not sufficient for comprehensive privacy protection. For example, we can observe that only one patient is paid by a non-federal program (`pay_src=11`). If an attacker has a list of beneficiaries from this non-federal program, then the patient identity of the record can be easily hacked. Figure 2.3 shows the histogram of duplicate records from the dataset. With the full combination of five variables, 134 (about 10%) records

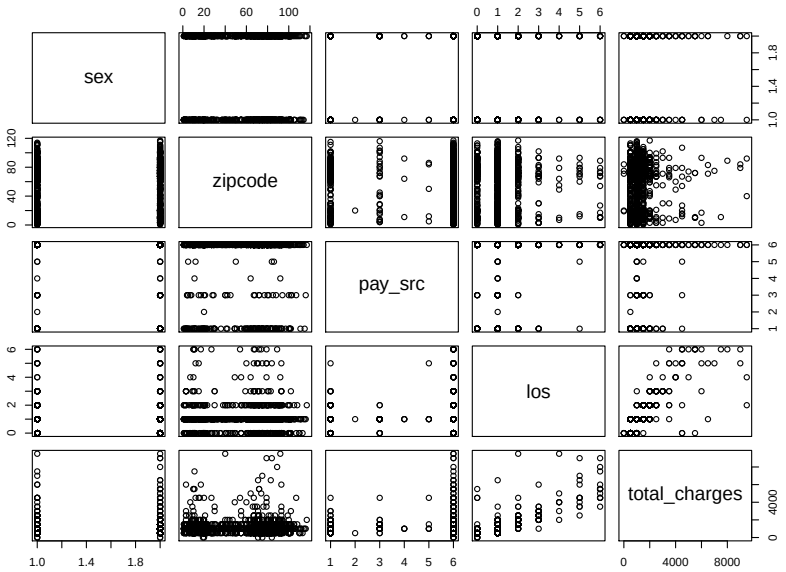


Figure 2.2: Scatter Plots of the preprocessed Texas Inpatient Data.

are unique. Population uniqueness is a very important concept in privacy-preserving algorithms. In Section 2.2, we will illustrate the potential threats for unique records, and algorithms to prevent such attacks.

2.2 Generalization and Suppression

The most basic step before publishing sensitive data is to remove any personal identifiable variables such as name, telephone number, social security number, and driving license number. For the Texas inpatient data, the Texas Department of State Health Services has already removed these explicit identifiers, and assigned an arbitrary record number to each row. As another example, the Synthetic Data from Centers for Medicare and Medicaid Ser-

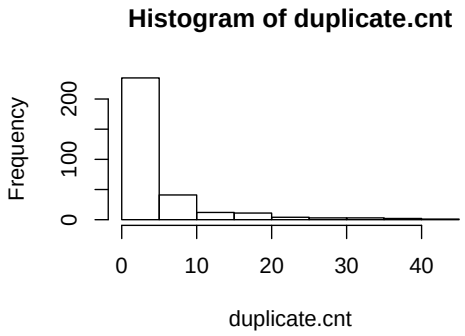
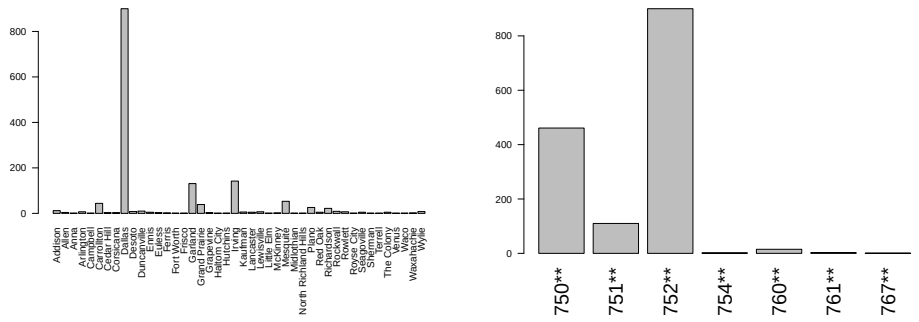


Figure 2.3: Histogram of Duplicate Records.

vices replaced explicit identifiers with random hash codes, so that users can link and match the records from the same patient, but not with external data sources. This seemingly intuitive process, however, is not sufficient to protect the patient identities from "linking attack".

Sweeney [128] provided a simple example by linking two datasets: a dataset from the Group Insurance Commission (GIC) in Massachusetts and the voter registration list for Cambridge. The GIC dataset does not include explicit identifiers, but the voter registration list does; it contains name and address. These two datasets have three common variables: ZIP, birth date, and sex. By linking the two datasets using these three variables, she demonstrated that the governor of Massachusetts can be identified from the GIC data. This type of privacy attack is called a linking attack. A linking attack is difficult to foresee and prevent, since it is almost impossible to check all the external linking datasets before publishing.

Generalization and suppression techniques alleviate the disclosure risks



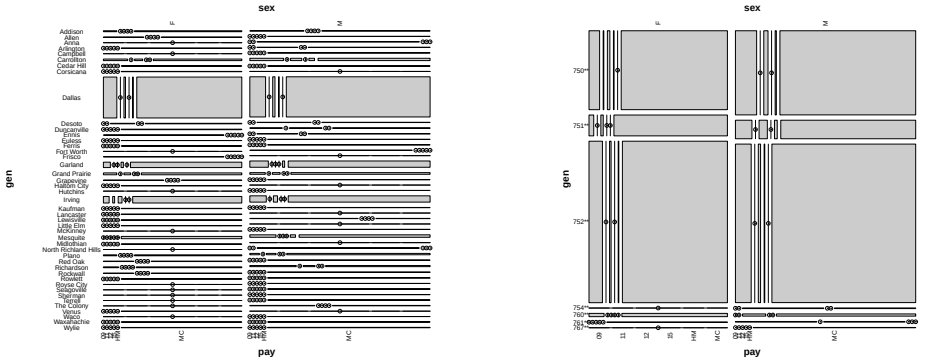
(a) Generalization (city-level) (b) Generalization (three-digit ZIP)

Figure 2.4: Generalization of the ZIP code from the Texas inpatient dataset.

that may arise from linking attacks [122, 127]. Generalization replaces a value with a less specific but semantically consistent value. For example, a ZIP code can be generalized into city or county. On the other hand, suppression replaces a value to a non-informative value e.g. $70512 \rightarrow *$. In Figure 2.4, we demonstrated two types of generalization: city-level generalization, and three-digit ZIP code generalization by removing the last two digits. Parkland Memorial Hospital is located in Dallas, and of course, most of the patients live in Dallas (the zip codes of Dallas start with 752). As can be seen, the Dallas population is shown as the peaks on both bar charts. We are, however, more interested in low-count categories such as Waco and 761** (high-risk values).

Sweeney [128] proposed k-anonymity to formalize and quantify the disclosure risk of unique populations. The definition of k-anonymity is as follows:

Definition 2.2.1 (k-anonymity)**.** A table is said to satisfy k-anonymity if and only if each set of quasi-identifiers from the table appears with at least k



(a) City-level codes (b) Three-digit ZIP codes

Figure 2.5: Mosaic Plots of the Quasi Identifiers (sex, payment source, address)

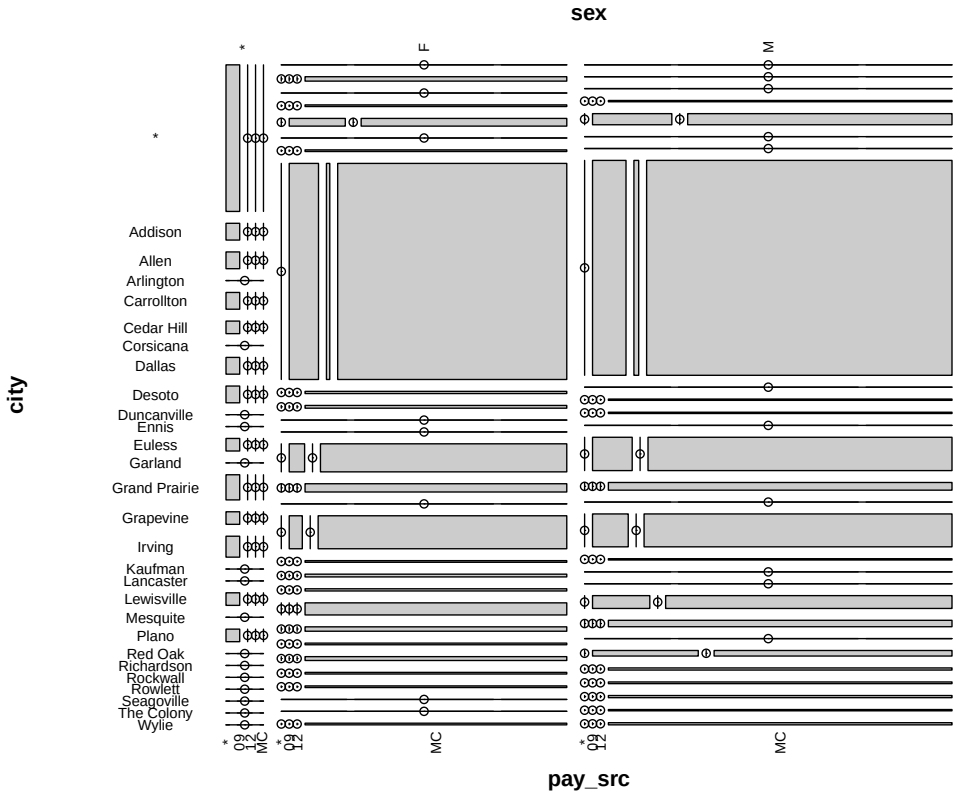


Figure 2.6: Mosaic Plot of the Generalized and Suppressed Quasi Identifiers.

occurrences.

In other words, to adhere to the k-anonymity principle, each row in a dataset should be indistinguishable with at least $k-1$ other rows. Quasi-identifiers are the attributes that can be used for linking. For this example, we assume that sex, payment source, and address are the quasi-identifiers of the dataset.

Suppose that we want to publish a dataset that satisfies 3-anonymity. Figure 2.5 shows mosaic plots of the quasi-identifiers for two different ZIP code generalization methods. Mosaic plots visualize multi-way contingency tables. The vertical axis (y-axis) represents the generalized ZIP codes, and the horizontal axis (x-axis) shows the cross-tabulation of sex (top) and payment source (bottom). The areas of the rectangles specify the number of entries with the corresponding attribute values; bigger rectangles mean more data points. For example, the Medicare population in Dallas is shown as two big boxes in Figure 2.5 (a), and the HMO populations are shown as tiny boxes ($\odot$ represents no entry). Mosaic plots are useful not only in categorical variable analysis, but also in visual diagnostics for data privacy assessment. For more information about mosaic plots, see [57, 58]. As can be seen, there exist many unique data points even if we generalize the address variable into either city or three-digit ZIP code. The three-digit ZIP code generalization has less unique data points i.e. small-sized rectangles, but the address resolution became overly coarsened.

To achieve 3-anonymity ($k = 3$), we combine generalization and suppression. Using the city-level generalized data, we collect the data points that are unique or appearing only two times. We suppress these rare data points to *. Figure 2.6 shows the mosaic plot of the generalized and suppressed quasi-identifiers. As can be seen, there are no more unique data points, and at the same time, the original data properties are reasonably preserved. Note that generalization and suppression should not be abused, otherwise the utility of data can be seriously damaged. As an illustrative example, k-anonymity with higher k values can be easily obtained by generalizing the address variable to a state-level variable, or suppressing all the rows; the address variable does not contain any information. Therefore, generalization and suppression should be minimally applied to the extent that the transformed data satisfy k-anonymity.

Achieving the optimal k-anonymity is, in fact, NP-hard, and there are several heuristic and greedy algorithms developed. Sweeney [127] proposed the Preferred Minimal Generalization (MinGen) algorithm. However, the proposed algorithm was computationally inefficient, and numerous practical algorithms were later proposed e.g., Incognito [80] and Mondrian [81]. Satisfying k-anonymity is not a perfect protective solution, and there exist several failure modes. Machanavajjhala [87] discovered two attack scenarios in which k-anonymity can fail: homogeneity attack and background knowledge attack, and suggested an extended privacy metric, l-diversity. Xiao and Tao [138] proposed a linear-time algorithm that satisfies l-diversity. Li et al. [82] proposed a privacy metric, t-closeness, that overcomes limitations of k-anonymity

and l-diversity, and Xiao and Tao [139] suggested a generalization principle, m-invariance, that caters to re-publication issues of microdata.

2.3 Synthetic Data using Multiple Imputation

The generation of synthetic data [119] is an alternative (and sometimes complementary) approach to data transforming disclosure techniques. Multiple imputation, which was originally developed to impute missing values in survey responses [118], can also be used to generate either partially or fully synthetic data. Abowd and Woodcock [3] synthesized a French longitudinal linked database, and Raghunathan et al. [108] provided general methods for obtaining valid inferences using multiply imputed data. Markov Chain Monte Carlo simulation methods and generalized linear models are typically used for sampling. Decision trees models, such as CART and Random forests, can also be used as imputation models in multiple imputation [112, 18]. Some illustrative empirical studies have used U.S. census data [37], German business database [113], and U.S. American Community Survey [121].

Let us start from the missing value imputation setting. Consider a survey with two variables x and z, $\mathcal{D} = \{(x, z)\}$, where some of the x responses are missing. Let x_{obs} be the observed subset of x. In the multiple imputation approach, the unobserved responses are imputed using samples from a predictive posterior model as follows:

$$x \sim \Pr(x \mid x_{\text{obs}}, z)$$

Note that the predictive posterior can be modeled using the observed subset, and often obtained using generalized linear models, Bayesian Bootstrapping methods, or Markov Chain Monte Carlo simulations [123, 112]. For example, an R package for multivariate imputation for chained equation [133] provides nine different imputation models including predictive mean matching, Bayesian linear regression, Linear regression, Unconditional mean imputation, etc. Generating fully synthetic data is straightforward from this framework[1]. First, z is drawn from $\Pr(z)$, then x is drawn from the predictive posterior distribution. Typically, this entire process is repeated independently K times to obtain K different synthetic datasets.

Raghunathan et al. [108] showed that valid inferences can be obtained from multiply imputed synthetic data. Let Q be a function of (x, z). For example, Q may represent the population mean of (x, z) or the population regression coefficients of x on z. Let q_i and v_i be the estimate of Q and its variance obtained from the ith synthetic dataset. Then, valid inferences on Q can be obtained as follows:

$$\bar{q}_K = \sum_{i=1}^{K} q_i/K$$
$$T_s = (1 + \frac{1}{K})b_K - \bar{v}_K$$

where $b_K = \sum_{i=1}^{K}(q_i - \bar{q}_K)^2/(K-1)$ and $\bar{v}_K = \sum_{i=1}^{K} v_i/K$. These two quantities $\bar{q}_K$ and T_s estimate the original Q and the variance from sampling.

[1] To see the difference between partially and fully synthetic datasets, see [36]

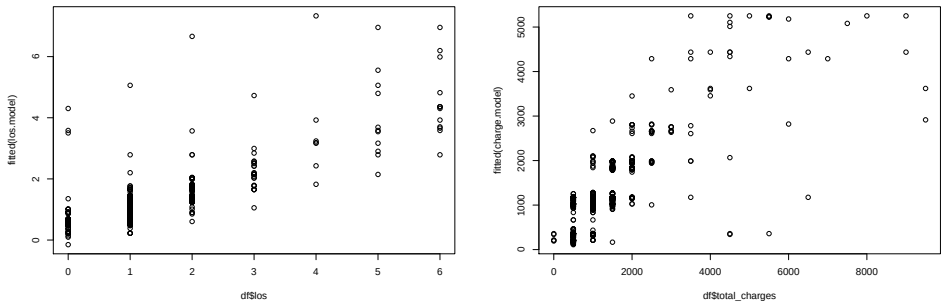
For our Texas inpatient example, we fit simple imputation models based on linear regression. Note that, in theory, multiple imputation for fully/partially synthetic data requires sampling from predictive posterior distributions, and our approach can be viewed as a pseudo multiple imputation implementation[2]. We build two regression models for length of stay and total charges, respectively. The other three variables, sex, payment source, and address, are categorical variables, and can also be modeled using generalized linear models, but we skip the process since 1) the goodness of fit of the fitted generalized linear models are poorly measured, and 2) these three variables are already k-anonymized. Specifically, for the two numeric variables, we build regression models as follows:

$$\text{length of stay} \sim \text{sex} + \text{payment source} + \text{city} + \text{total charges}$$

$$\text{total charges} \sim \text{sex} + \text{payment source} + \text{city} + \text{length of stay (los)}$$

We estimate regression coefficients and residual variances shown in Table 2.2 (length of stay) and Table 2.3 (total charges). As can be seen, the length of stay variable is primarily determined by the total charges variable, and vice versa. Interestingly, the total charges are slightly affected by the city variable e.g. see the coefficients of cityDesoto and cityPlano. Male infants generally cost more, since many of them receive circumcision. Figure 2.7 shows the goodness of fit of the regression models: the fitted values (y-axis) and the

[2]Multiple imputation is a sophisticated Bayesian methodology, and there are several different aspects from the example we presented. Our example is designed to convey the overall idea of multiple imputation. For more information, see [36].



(a) Fitted vs original length of stay (b) Fitted vs original total charges

Figure 2.7: Scatter plots of the original and fitted data for synthetic data.

original values (x-axis). From the fitted data, synthetic data can be obtained by adding Gaussian noise with the estimated residual variances.

2.4 Measuring Utility and Risk

As privacy is a subjective and contextual concept, so is "utility". Utility can be measured in multiple ways, depending on the research objectives. For our Texas inpatient example, if we want to find out relevant variables and their coefficients that affect hospital charges, we can measure the utility as follows:

$$\text{Utility} = \exp(-\|\boldsymbol{\beta}_{\text{original}} - \boldsymbol{\beta}_{\text{synthetic}}\|^2)$$

where $\boldsymbol{\beta}_{\text{original}}$ and $\boldsymbol{\beta}_{\text{synthetic}}$ are regression coefficients from the original and synthetic data, respectively. This utility metric is maximized when $\boldsymbol{\beta}_{\text{original}} = \boldsymbol{\beta}_{\text{synthetic}}$. Note that this utility metric is one of many other utility metrics that can measure similar quantities. As an another example, if aggregate statistics

Table 2.2: Regression coefficients for the "length of stay" model.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.1759	0.0889	1.98	0.0481
sexF	-0.0049	0.1002	-0.05	0.9610
sexM	-0.1198	0.1007	-1.19	0.2342
pay_src09	0.0101	0.0470	0.21	0.8302
pay_src12	-0.1005	0.1489	-0.67	0.4999
cityAddison	0.1015	0.1839	0.55	0.5813
cityAllen	0.0666	0.2703	0.25	0.8053
cityArlington	0.2729	0.2471	1.10	0.2696
cityCarrollton	0.1263	0.1473	0.86	0.3913
cityCedar Hill	0.1475	0.3080	0.48	0.6321
cityCorsicana	0.0715	0.3239	0.22	0.8253
cityDallas	0.0812	0.1327	0.61	0.5405
cityDesoto	-0.7054	0.2072	-3.40	0.0007
cityDuncanville	-0.1657	0.2241	-0.74	0.4597
cityEnnis	0.2673	0.3240	0.82	0.4096
cityEuless	-0.3476	0.3082	-1.13	0.2596
cityGarland	0.1374	0.1394	0.99	0.3245
cityGrand Prairie	0.1048	0.1459	0.72	0.4727
cityGrapevine	0.1929	0.3081	0.63	0.5314
cityIrving	0.0394	0.1363	0.29	0.7725
cityKaufman	0.1290	0.2472	0.52	0.6019
cityLancaster	0.0443	0.2645	0.17	0.8670
cityLewisville	0.0153	0.2196	0.07	0.9444
cityMesquite	0.0999	0.1498	0.67	0.5052
cityPlano	0.1227	0.1593	0.77	0.4414
cityRed Oak	0.3240	0.3241	1.00	0.3176
cityRichardson	0.1738	0.1748	0.99	0.3203
cityRockwall	-0.0763	0.2240	-0.34	0.7333
cityRowlett	0.1025	0.2343	0.44	0.6618
citySeagoville	0.1107	0.2647	0.42	0.6760
cityThe Colony	-0.1469	0.3241	-0.45	0.6504
cityWylie	0.0960	0.2241	0.43	0.6684
total_charges	0.0008	0.0000	48.54	0.0000

are the primary concerns, the utility can be measured as follows:

$$\text{Utiliy} = \exp(-\|E[x_{\text{original}}] - E[x_{\text{synthetic}}]\|^2)$$

where x is the variable of interest. It is recommended to try several different utility metrics before publishing transformed data.

Even if there exist theoretical privacy guarantees for transformed datasets, rigorous risk analyses should be performed before actual publishing. Researchers need to consider possible and worst-case attack scenarios, and try

Table 2.3: Regression coefficients for the "total charges" model.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	359.1373	91.9207	3.91	0.0001
sexF	27.3817	103.9231	0.26	0.7922
sexM	174.8796	104.3848	1.68	0.0941
pay_src09	-48.5822	48.7065	-1.00	0.3187
pay_src12	95.4214	154.4706	0.62	0.5368
cityAddison	-234.4290	190.7568	-1.23	0.2193
cityAllen	-174.5079	280.3842	-0.62	0.5338
cityArlington	-297.4287	256.3879	-1.16	0.2462
cityCarrollton	-159.3104	152.7674	-1.04	0.2972
cityCedar Hill	-112.9648	319.5768	-0.35	0.7238
cityCorsicana	-201.8896	336.0570	-0.60	0.5481
cityDallas	-175.0181	137.5925	-1.27	0.2036
cityDesoto	897.3799	214.5596	4.18	0.0000
cityDuncanville	-76.4373	232.5305	-0.33	0.7424
cityEnnis	-287.8444	336.1882	-0.86	0.3920
cityEuless	35.7392	319.8752	0.11	0.9111
cityGarland	-196.9284	144.5687	-1.36	0.1734
cityGrand Prairie	-98.2934	151.3952	-0.65	0.5163
cityGrapevine	-341.1746	319.5782	-1.07	0.2859
cityIrving	-192.8540	141.3302	-1.36	0.1726
cityKaufman	-275.6385	256.4033	-1.08	0.2825
cityLancaster	-64.9637	274.4042	-0.24	0.8129
cityLewisville	-118.7260	227.8391	-0.52	0.6024
cityMesquite	-204.3538	155.3656	-1.32	0.1886
cityPlano	-264.5943	165.1621	-1.60	0.1094
cityRed Oak	-535.2229	336.0570	-1.59	0.1115
cityRichardson	-226.1209	181.3205	-1.25	0.2126
cityRockwall	-67.1545	232.4419	-0.29	0.7727
cityRowlett	-312.5497	243.0179	-1.29	0.1986
citySeagoville	-249.3875	274.6011	-0.91	0.3639
cityThe Colony	-77.5973	336.3026	-0.23	0.8176
cityWylie	-231.5758	232.4009	-1.00	0.3192
los	815.3706	16.7967	48.54	0.0000

simulating such attacks. Matching internal databases, and searching already published external databases are good practices as well. By doing so, data publishers can estimate the potential consequences of privacy breaches. Privacy breaches may result in significant amount of legal and social costs, and data publishers should to be aware of the worst case scenarios.

Privacy must be interpreted both contextually and through information theoretic ways. In healthcare systems, some variables may be more sensitive

than others. For example, address or name can be less sensitive than disease or medication history. Domain knowledge and data exploration steps are exceptionally important because of the complex healthcare ecosystem. Furthermore, perceptions on privacy also changes over time with new technologies: e.g. social network services. Therefore, for successful privacy-preserving data publishing in healthcare, one needs to understand social infrastructures as well as information-theoretic or statistical privacy concepts.

Chapter 3

Privacy-aware Synthetic Data Publication

The two competing requirements for public use data similarly apply to synthetic data disclosure. Synthetic data need to be accurate enough to answer relevant statistical queries without revealing private information to third parties. On the other hand, synthetic data from overly accurate models may leak private information [2].

The balance between accuracy and privacy can be addressed by using cryptographic privacy measures such as ϵ-differential privacy [44]. However, several attempts to achieve such strong privacy guarantees have shown to be impractical to implement. For example, Barak et al. [8] showed that it is possible to release contingency tables under the differential privacy regime using Fourier transform and additive Laplace noise. However, this proposed release mechanism was later criticized for being too conservative and disrupting statistical properties of the original data [140, 25]. On the other hand, Soria-Cormas and Drechsler [126] claimed that ϵ-differential privacy can be a useful privacy measure when disclosing a large size of data with a limited number of variables. For example, differentially private synthetic data have been demonstrated using the Census Bureau's OnTheMap data that consists

Table 3.1: Synthesizer algorithms discussed in this chapter.

Name	Abbreviated Model Eq.	Parameters
Contingency table	$\Pr_{\mathcal{D}}(\mathbf{x})$	non-parametric
Marginal Bayesian Bootstrap	$\prod_i^M \Pr_{\mathcal{D}}(x_i)$	non-parametric
Multiple imputation	$\prod_i^M \Pr_{\hat{\mathbf{w}}}(x_i \mid \mathbf{x}_{-i})$	$\hat{\mathbf{w}}$: model parameter
Perturbed Gibbs Sampler	$\prod_i^M \Pr_{\mathcal{D},\alpha}(x_i \mid h(\mathbf{x}_{-i}))$	α: privacy parameter
Block PeGS with Reset	$\prod_b^B \prod_i^M \Pr_{\mathcal{D},\alpha}(x_i \mid h(\mathbf{x}_{-i}))$	B: sample block size

of approximately one million records but with only two variables [86].

In this chapter, we propose a *practical* multi-dimensional categorical data synthesizer that satisfies ϵ-differential privacy. The proposed synthesizer can handle multi-dimensional data that are not practical to be represented as contingency tables. We demonstrate our algorithm using a subset of California Patient Discharge data, and generate multiple synthetic discharge datasets. Although ϵ-differential privacy is extensively used in our algorithm analyses, we note that ϵ-differential privacy is one of many descriptive measures for disclosure risks. Differential privacy is a measure for functions, not for data [51], and this measure can be overly pessimistic for data-specific applications. Thus, we also evaluate disclosure risks of the proposed algorithms using the population uniqueness of synthetic records [30] and indirect-matching probabilistic disclosure risks [41]. To measure the statistical similarities between synthetic and the original data, we compare marginal and conditional distributions, and regression coefficients from the synthesized data to those from the original data.

There are two brute-force approaches to generating synthetic categorical data. As statistical properties of categorical data are fully captured in contingency tables, in theory, a synthetic sample $\mathbf{x}$ can be drawn directly from an M-way full contingency table $\Pr_{\mathcal{D}}(\mathbf{x})$, where M is the total number of features. For data with a small number of features, this contingency table can be estimated by either direct counting or log-linear models [136, 137]. However, this strategy does not scale for high-dimensional datasets. As we will see in Section 3.3, our experiment dataset has 13 features and their possible feature combinations are approximately 2.6 trillion. More importantly, sampling from an exact distribution may reveal too much detail about the original data, thus this is not a privacy-safe disclosure method. On the other extreme, one may model the joint distribution as a product of univariate marginal distributions. Although this approach can easily achieve differential privacy [89], the synthetic data loses critical joint distributional information about the original data.

The proposed algorithm generates *realistic but not real* synthetic samples by calibrating a privacy parameter α. In addition, the exponentially number of cells in a contingency table is avoided by using chained equations and feature hashing [134] as follows:

$$\begin{aligned} &\text{for } i \text{ in } 1:M \\ &\quad x_i \sim \Pr_{\mathcal{D},\alpha}(x_i \mid h(\mathbf{x}_{-i})) \end{aligned} \tag{3.1}$$

where $\mathbf{x}_{-i}$ is a feature vector except for the ith feature and $h(\mathbf{x}_{-i})$ represents a

hashed feature vector. $\Pr_{\mathcal{D},\alpha}(x_i \mid h(\mathbf{x}_{-i}))$ is the compressed and perturbed conditional distribution of the ith feature and M is the total number of features. The joint probability distribution is represented as M conditional distributions. Note that the conditional distribution in Equation (3.1) is not exact. The full condition $\mathbf{x}_{-i}$ is compressed using a hash function $h(\mathbf{x}_{-i})$ and perturbed by a privacy parameter α. Ignoring these two additional components i.e. $h(\mathbf{x}_{-i})$ and α, if the probability is modeled using generalized linear models, then the proposed algorithm is the same as a multiple imputation algorithm for fully synthetic data. The proposed synthesizer is named as Perturbed Gibbs Sampler (PeGS). This process is somewhat analogous to multivariate imputation by chained equations (also known as sequential regression multiple imputation) [109, 133]. In Section 3.2.4, we will show that this synthesis cycle can also be recursively applied multiple times i.e. $\mathbf{s} = \text{PeGS}(\text{PeGS}(\ldots \text{PeGS}(\mathbf{x})))$. This recursive synthesis will be shown to be very effective in our block sampling algorithm.

Table 3.1 summarizes the synthesizer models that are described in this chapter. More details on this list and privacy guarantees for both one iteration and multiple iterations are described in Section 3.2.

The objective of PeGS is to generate a single realistic synthetic dataset that adheres to rigorous privacy metrics, balancing the trade-off between utility and risk in a flexible and effective manner. This is substantially different from the goal of multiple imputation, which primarily focuses on improving the analytical validity of missing data imputation and not on privacy vs. util-

ity trade-off. The name *multiple imputation* refers to the fact that multiple imputed datasets are released to alleviate sampling uncertainty. Disclosing multiple datasets can be helpful for numerous statistical analyses, but at the same time, the improved accuracy may lead to an unexpected privacy breach. Consider a statistical database that provides a synthetic data row per query. To obtain N rows, intuitively, we need N queries. According to the sequential composition rule of differential privacy [90], the privacy risk for N queries is N times greater than the risk of one query. Similarly, releasing K imputed datasets can be K times more risky than releasing a single dataset.

The target use case of our synthetic data is also quite different from traditional uses of synthetic data. Our algorithms are primarily designed to protect the privacy of the original data, and then, within the privacy constraint, to maximize the statistical validity and utility of synthetic data. Such synthetic data can be useful when providing a single "realistic" dataset to third party data scientists so that they can explore and develop innovative data applications. As an illustrative example, Centers for Medicare & Medicaid Services recently released synthetic public use files[1], saying that:

> *... Although the DE-SynPUF has very limited inferential research value to draw conclusions about Medicare beneficiaries due to the synthetic processes used to create the file, the Medicare DE-SynPUF does increase access to a realistic Medicare claims data file in a*

[1] `http://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/SynPUFs/DE_Syn_PUF.html`

timely and less expensive manner to spur the innovation necessary to achieve the goals of better care for beneficiaries and improve the health of the population. ...

The users of our synthetic data can be from a wide range of disciplines such as statistics, computer science, and healthcare policy studies. Providing synthetic data is much more than just providing the data schema. Users can write scripts and codes for exploring and extracting information using the "realistic" synthetic data, and deliver their applications to the data owner to check the validity of their claims.

The rest of this chapter is organized as follows: In Section 3.1, we cover the basics of privacy measures and synthetic data. In Section 3.2, the details of the PeGS algorithms are illustrated, and the privacy guarantees of the proposed algorithms are derived. We demonstrate our algorithms using California Patient Discharge dataset in Section 3.3.

3.1 Preliminaries & Related Work

Privacy is an abstract concept, and it can be defined and quantified in many different ways. We describe two privacy measures that are popular in computer science, ϵ-differential privacy and l-diversity. These two measures will be also used in our algorithm to quantify the privacy risks of synthetic data.

Differential privacy [44] is a mathematical measure of privacy that

quantifies disclosure risks of statistical functions. To satisfy ϵ-differential privacy, the inclusion or exclusion of any particular record in data cannot affect the outcome of functions by much. Specifically, a randomized function $f : \mathcal{D} \to f(\mathcal{D})$ provides ϵ-differential privacy, if it satisfies:

$$\frac{\Pr(f(\mathcal{D}_1) \in \mathcal{S})}{\Pr(f(\mathcal{D}_2) \in \mathcal{S})} \leq \exp(\epsilon)$$

for all possible $\mathcal{D}_1, \mathcal{D}_2 \in \mathcal{D}$ where $\mathcal{D}_1$ and $\mathcal{D}_2$ differ by at most one element, and $\forall \mathcal{S} \in \text{Range}(f(\mathcal{D}))$. For a synthetic sample, this definition can be interpreted as follows [89]:

$$\frac{\Pr_{\mathcal{D}_1}(\mathbf{x})}{\Pr_{\mathcal{D}_2}(\mathbf{x})} \leq \exp(\epsilon) \tag{3.2}$$

where $\mathbf{x}$ represents a random sample from synthesizers. In other words, a data synthesizer $\Pr_{\mathcal{D}}(\mathbf{x})$ is ϵ-differentially private, if the probabilities of generating $\mathbf{x}$ from $\mathcal{D}_1$ and $\mathcal{D}_2$ are indistinguishable to the extent of $\exp(\epsilon)$.

Several mechanisms have been developed to achieve differential privacy. For numeric outputs, the most popular technique is to add Laplace noise with mean 0 and scale $\Delta f/\epsilon$ where Δf is the L_1 sensitivity of function f. Exponential mechanism [91] is a general differential privacy mechanism that can be applied to non-numeric outputs. For categorical data, Dirichlet prior can be used as a noise mechanism to achieve differential privacy [86, 89].

***l*-diversity**. When publishing a public use file, a certain combination of features can identify an individual from an anonymized dataset, even if personal identifiers, such as driver license number and social security number, are

removed from a dataset. Such threats are commonly prevented by generalizing or suppressing features; for example, ZIP codes with small population are replaced by corresponding county names (generalization), or can be replaced by * (suppression). Sweeney [128] proposed a privacy definition for measuring the degree of such feature generalization and suppression, k-anonymity. To adhere the k-anonymity principle, each row in a dataset should be indistinguishable with at least $k-1$ other rows.

The definition of k-anonymity, however, does not include two important aspects of data privacy: feature diversity and attackers' background knowledge. Machanavajjhala [87] illustrated two potential threats to a k-anonymized dataset, then proposed a new privacy criterion, l-diversity. The definition of l-diversity states that the diversity of sensitive features should be kept within a block of samples. There are several ways of achieving l-diversity; in this chapter, we use Entropy l-diversity. A dataset is Entropy l-diverse if

$$-\sum_{x_i} \Pr(x_i \mid \mathbf{x}_{-i}) \log \Pr(x_i \mid \mathbf{x}_{-i}) \geq \log l \tag{3.3}$$

where $1 \leq l$. This definition originally applies to a dataset with feature generalization or suppression. For a synthetic sample, Park et al. [103] suggested an analogous definition of l-diversity: A synthetic dataset is synthetically l-diverse if a synthetic sample x_i is drawn from a distribution that satisfies l-diversity.

Synthetic data generation typically involves two steps: 1) statistical modeling of the original data, and 2) sampling from the obtained model. In the

modeling step, one can apply a wide range of statistical models, from a simple linear regression model to advanced Markov Chain Monte Carlo sampling methods. Disclosure risks of synthetic samples are traditionally analyzed after the sampling step, but recently several researchers have attempted to merge privacy metrics in the modeling and sampling steps. Depending on the application, synthetic data can replace either the entire original data [119], or specific columns or values that bear high disclosure risks i.e. partially synthetic data [83]. The notion of fully synthetic data in the multiple imputation literature is slightly different from our notion. Figure 3.1 shows various categories of synthetic data. Note that, in this chapter, fully synthetic data refers to completely synthetic data with no original records.

The quality of a synthetic dataset is mainly determined by the quality of the statistical model used. Lombardo and Moniz [85] proposed generating synthetic medical records for outbreak studies. They suggested using both domain knowledge and actual data. The underlying dynamics of the original data is modeled through a set of sub-models such as exposure model, infection model, disease model, and behavior model. Buczak et al. [17] demonstrated a pilot study for generating synthetic medical records. In their pilot study, the synthetic data were generated through three steps: 1) patient information generation, 2) similar patients clustering, and 3) adapting care models to synthesized patients. As can be seen, these approaches require a considerable amount of domain knowledge and non-automated processes. Furthermore, no privacy measure was incorporated in the synthesizing processes.

Machanavajjhala et al. [86] generated a differentially private synthetic dataset for commuting pattern studies. They derived an appropriate amount of Laplace smoothing is derived to guarantee ϵ-differential privacy. A subset from the Census Bureau's OnTheMap microdata was used in their study. However, the demonstrated dataset had only three columns (id, origin block, destination block), and the suggested algorithm was specific to the application. Barak et al. [8] suggested a differentially private release mechanism for contingency tables. Note that releasing a contingency table is different from releasing a synthetic dataset, but one always can sample a synthetic sample from the released contingency table. Their approach used Fourier transform and Linear programming to guarantee differential privacy. Although the theoretical results are solid, experimental results using real datasets show that the suggested differential privacy mechanism is not practical for data mining purposes [140].

Synthetic data generation has also been used for privacy-aware distributed data mining scenarios. In the prototypical approach [94, 95], parametric models are separately learnt on individual (local) databases. Then, only the model parameters are transmitted to a trusted central site from each local database, instead of the raw data, to address privacy concerns. At the central site, the parameters received are used to generate synthetic data that (approximately) represents the union of the different databases. Finally, a global model is learnt using such data. However, privacy constraints such as l-diversity are not directly built into the data generation or modeling process,

as it is in this chapter. Also, the goal is to attain a global statistical model under sharing constraints, rather than create a synthetic dataset for public release.

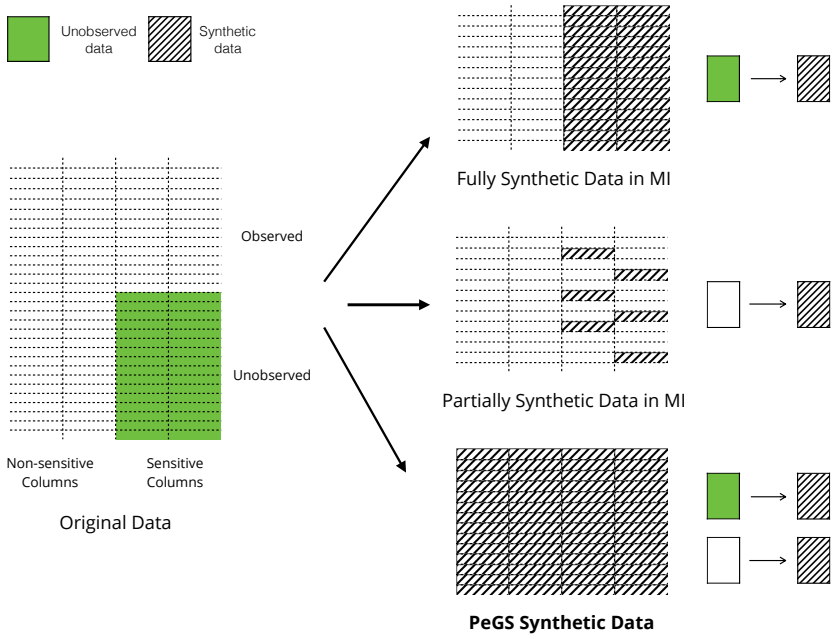


Figure 3.1: Three different notions of synthetic data. In multiple imputation, only sensitive columns are synthesized. To create a fully synthetic dataset, one replaces the unobserved sensitive values with synthetic values [36]. A partially synthetic dataset only replaces the observed sensitive values that bear high disclosure risks. Our definition of fully synthetic data refers to "completely" synthetic data with no original records regardless of sensitive or non-sensitive columns.

The disclosure risks of (multiply imputed) synthetic datasets are typically measured after synthetic datasets are generated i.e. using *post hoc* risk analysis. This flexibility makes it difficult to apply and analyze rigorous pri-

vacy measures, such as differential privacy and l-diversity, in a unified framework. In contrast, we derive the relationship between the amount of Laplace smoothing and privacy measures (ϵ in differential privacy and l in l-diversity) by using a simple non-parametric model. Our algorithm directly incorporates these privacy measures in the synthesizing process, guaranteeing the desired level of privacy for synthetic data.

3.2 Perturbed Gibbs Sampler

In this section, we propose the Perturbed Gibbs Sampler (PeGS) for categorical synthetic data. We first overview the algorithm, then describe its three main components: feature hashing, statistical building blocks, and noise mechanism. Next, we illustrate how the PeGS algorithm can be efficiently extended to draw a block of random samples. Finally, we show that multiple imputation can be similarly extended to satisfy differential privacy, which will be used as our baseline model in Section 3.3.

3.2.1 Algorithm Overview

Perturbed Gibbs Sampler (PeGS) is a categorical data synthesizer that consists of three main steps:

1. *Disintegrate*: In this step, the original data $\mathcal{D}$ is disintegrated into statistical building blocks i.e. $\Pr_{\mathcal{D}}(x_i \mid h(\mathbf{x}_{-i}))$ where h is a suitable hash function. These compressed conditional distributions are estimated by counting the corresponding occurrences in the original data.

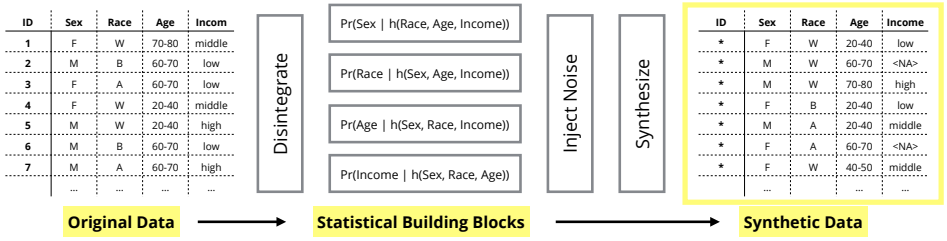


Figure 3.2: PeGS Process Diagram for a four feature dataset: {(Sex, Age, Race, Income)}. Four types of conditional distributions are estimated from the original data, then uniform Dirichlet priors are used to perturb the conditional distributions. Synthetic samples are drawn by iterating over the statistical building blocks.

2. *Inject Noise*: For a specified privacy parameter α, the statistical building blocks are modified to satisfy differential privacy or l-diversity, $\Pr_{\mathcal{D}}(x_i \mid h(\mathbf{x}_{-i})) \rightarrow \Pr_{\mathcal{D},\alpha}(x_i \mid h(\mathbf{x}_{-i}))$.

3. *Synthesize*: We first pick a random seed from a predefined pool; this can be regarded as a query to our model. The seed sample is transformed to a synthetic sample by iteratively sampling each feature from the statistical building blocks, $x_i \sim \Pr_{\mathcal{D},\alpha}(x_i \mid h(\mathbf{x}_{-i}))$.

Figure 3.2 visualizes the overall sequential steps of the PeGS algorithm. Figure 3.3 illustrates the synthesis step. Three components are essential in the PeGS algorithm: feature hashing, statistical building blocks, and perturbation. The number of possible conditions is exponential with respect to the number of features, Therefore, feature hashing is used to compress the number of the possible conditions $\mathbf{x}_{-i}$. Statistical building blocks are built based on this feature hashing, which are essentially multiple hash-tables describing

compressed conditional distributions. They serve a key role when we try to sample a block of synthetic examples. Perturbation is required to guarantee the differential privacy. Without perturbation, synthetic samples may reveal too much about the original data.

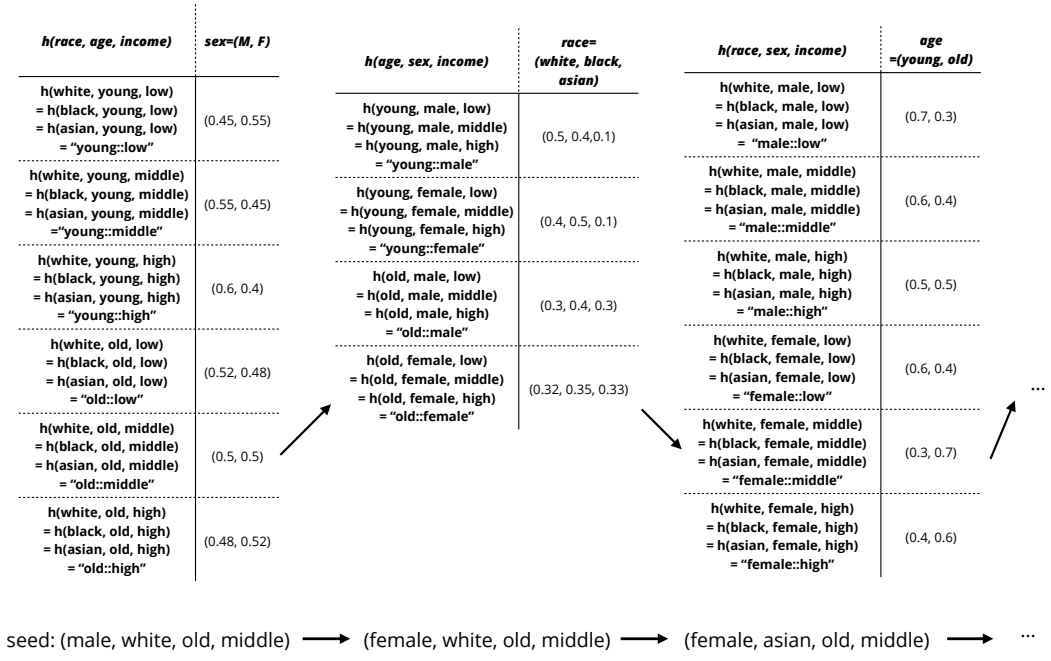


Figure 3.3: Synthesis Steps in PeGS. Three tables represent the statistical building blocks of the example in Figure 3.2. In the disintegration step, these three statistical building blocks are stored. In the noise injection step, the probability vectors of the tables are perturbed. In the synthesis step, a new sample is generated by iteratively sampling over the tables.

3.2.2 Feature Hashing

The hash function $h(\mathbf{x}_{-i})$ in PeGS maps a feature vector to an integer key, where the range of the hash key is much smaller than 2^M (exponential in

the number of features). In essence, our purpose is to design a hash function that exhibits good compression while maintaining the statistical properties of data. Such a hash function has been deeply investigated in the machine learning literature for compressing high-dimensional feature spaces. This technique is sometimes known as the hashing trick [134]. For extremely high-dimensional, sparse, and unstructured data such as natural language texts, Locality Sensitive Hashing [68] and min-hashing [61] can be good candidates for the PeGS hash function.

We use a simpler approach to compress the feature space, as we are using lower dimensional data. We select the m variables that have the most mutual information with x_i to form the hash key, and ignore the other variables. Thus:

$$\underbrace{x_{o(1)} :: \ldots :: x_{o(m)}}_{\text{trivial hash}} \longrightarrow \underbrace{1 \ldots H}_{\text{Hash key}}$$

where $H \ll 2^M$ and $x_{o(j)}$ represents the feature with the jth highest mutual information with x_i. Let C_i be the number of categories for x_i, and $C_{\max} = \max_i C_i$. The key space of this simple hash function is upper bounded by $(C_{\max})^m \ll \prod_i C_i$.

Figure 3.3 illustrates the basic idea of feature hashing. The left and right columns of the conditional tables represent hashed features $h(\mathbf{x}_{-i})$ and smoothed probability estimates $\Pr_{\mathcal{D},\alpha}(x_i \mid h(\mathbf{x}_{-i}))$, respectively. The first table uses `race`, `age`, `income` as conditional variables, and `sex` as a target variable. For illustrative purposes, we use $m = 2$, and assume that the

target variable is closely related to the `age` and `income` variables. In other words, the `race` variable is ignored while constructing the conditional table i.e. $h(\text{white}, \text{young}, \text{low})$, $h(\text{black}, \text{young}, \text{low})$, and $h(\text{asian}, \text{young}, \text{low})$ belong to the same bin. The smoothed probability distributions are estimated based on a subset of samples that have the same hash key. Synthetic samples are sequentially drawn from these based and smoothed estimates, as Figure 3.3 shows. The details about smoothing and sampling processes will be discussed in Section 3.2.3.

The compressed conditional distribution $\Pr(x_i \mid h(\mathbf{x}_{-i}))$, which is basically a occurrence count hash-table for a given hash key, can now be stored in either memory or disk. There are several advantages of using this compressed conditional distribution over parametric modeling. First, the process of building statistical building blocks does not involve complicated statistical procedures such as parameter estimation and model selection. Second, the resulting statistical building blocks are robust to overfitting. Overfitting may occur when there are not enough samples in a table entry. Hashing reduces the number of table cells and smoothes out the estimated probability vector. Finally, this simple table representation is intuitive, and the process is easily extensible. This aspect is critical in our efficient block sampling scheme, which will be illustrated in Section 3.2.4.

Note that our feature hashing is different from multinomial models in which certain main effects and interactions are set to zero. The key difference is that our process is iterative. As an illustrative example, suppose that we

have three features x_1, x_2, and x_3, and we use $m = 1$. Let us assume that x_1 depends on x_2, x_2 depends on x_3, and x_3 depends on x_1. Then x_1 and x_3 get coupled and are not independent. Thus the synthetic process does not translate to simple multinomial models.

We now provide a brief guideline for determining the value of m. As a rule of thumb, we suggest that each cell approximately contains at least 30 data points to estimate probabilities. There are other physical constraints on the value of m such as the size of memory and hard disk. For example, if we want to minimize the access to hard disk, m should also satisfy $2^m <$ Memory Size. But, if m is too small, then this hash function effectively imposes an unrealistic conditional independence assumption. Therefore, the value of m should be carefully determined considering these listed aspects.

3.2.3 Perturbed Conditional Distribution

To satisfy the differential privacy, a certain amount of noise should be injected to the compressed conditional distributions. The form of noise may depend on applications and privacy measures. For example, noise can be added to maximize entropy [104] or to satisfy l-diversity [102, 103]. We use the Dirichlet prior perturbation to smooth out raw count based estimators to satisfy differential privacy and l-diversity. Specifically, α virtual samples are added to each category of the variable x_i, when the conditional distribution $\Pr_\alpha(x_i \mid h(\mathbf{x}_{-i}))$ is estimated. The amount α is a privacy parameter that controls the degrees of differential privacy and l-diversity. To be more precise,

our differentially private perturbation requires a single value of α, while our l-diverse perturbation needs different α values for each hashed condition $h(\mathbf{x}_{-i})$ i.e. $\alpha_{h(\mathbf{x}_{-i})}$. For analytical simplicity, we assume α virtual samples, $\alpha_{h(\mathbf{x}_{-i})}$ virtual samples for l-diversity, are uniformly added to all the categories of the variable x_i (see Equation 3.6). In practice, different amounts of virtual samples can be added to different categories of the variable x_i; for example, α can be proportional to the corresponding marginal distribution i.e. $\alpha_j \propto \Pr(x_i = j)$.

We first derive the probability of sampling $\mathbf{x}$ from the PeGS algorithm. From a random seed sample $\mathbf{s}$ (or a query), the probability of synthesizing $\mathbf{x}$ is factorized as follows:

$$\Pr_{\mathcal{D}_1,\alpha}(\mathbf{x} \mid \mathbf{s}) = \prod_{i=1}^{M} \Pr_{\mathcal{D}_1,\alpha}(x_i \mid h(x_{1:(i-1)}, s_{(i+1):M})) \tag{3.4}$$

where $x_{1:0}$ and $s_{(M+1):M}$ are just null values. For another dataset $\mathcal{D}_2$ that differs by at most one element, the probability of sampling $\mathbf{x}$ can be similarly derived.

For differential privacy (see Equation 3.2), the ratio between two quantities should satisfy the following relation:

$$\frac{\Pr_{\mathcal{D}_1,\alpha}(\mathbf{x} \mid \mathbf{s})}{\Pr_{\mathcal{D}_2,\alpha}(\mathbf{x} \mid \mathbf{s})} = \frac{\prod_{i=1}^{M} \Pr_{\mathcal{D}_1,\alpha}(x_i \mid h(x_{1:(i-1)}, s_{(i+1):M}))}{\prod_{i=1}^{M} \Pr_{\mathcal{D}_2,\alpha}(x_i \mid h(x_{1:(i-1)}, s_{(i+1):M}))} \leq \exp(\epsilon) \tag{3.5}$$

Let us focus on the ith component as follows:

$$\Pr_{\mathcal{D}_1,\alpha}(x_i = j \mid h(\mathbf{x}_{-i})) = \frac{n_{ij} + \alpha}{N_{h(\mathbf{x}_{-i})} + C_i\alpha} \tag{3.6}$$

$$N_{h(\mathbf{x}_{-i})} = \sum_{\mathbf{x}'_{-i}} \mathbb{1}(h(\mathbf{x}'_{-i}) = h(\mathbf{x}_{-i})) \tag{3.7}$$

where $N_{h(\mathbf{x}_{-i})}$ is the total number of rows that have the same hash key as $h(\mathbf{x}_{-i})$ and n_{ij} is the count of the jth category i.e. $x_i = j$ within the $N_{h(\mathbf{x}_{-i})}$ samples. In other words, the probability of sampling the jth category is proportional to the number of the original samples that have the jth category. The privacy parameter α acts as a uniform Dirichlet prior on this raw multinomial count estimate.

The value of α depends on the privacy criterion. We study two cases: differential privacy and l-diversity.

A. Differential Privacy. The two datasets for defining differential privacy $\mathcal{D}_1$ and $\mathcal{D}_2$ have at most one different row. Without loss of generality, let us assume that $\mathcal{D}_1$ has one more row than $\mathcal{D}_2$ i.e. $\mathcal{D}_1 = \mathcal{D}_2 \cup \mathbf{x}^d$. Except for the entry with hash keys $\{h(\mathbf{x}^d_{-i})\}_{i=1}^M$, the other entries of the two hash tables from $\mathcal{D}_1$ and $\mathcal{D}_2$ are identical; only one entry of the hash table is different. For the different entries of the hash tables, there are two possibilities:

$$\begin{aligned} \text{if} \quad & x_i^d \neq j, \qquad \Pr_{\mathcal{D}_1,\alpha}(x_i^d = j \mid h(\mathbf{x}_{-i}^d)) = \frac{n_{ij} + \alpha}{N_{h(\mathbf{x}_{-i})} + 1 + C_i\alpha} \\ \text{else if} \quad & x_i^d = j, \qquad \Pr_{\mathcal{D}_1,\alpha}(x_i^d = j \mid h(\mathbf{x}_{-i}^d)) = \frac{n_{ij} + 1 + \alpha}{N_{h(\mathbf{x}_{-i})} + 1 + C_i\alpha} \end{aligned}$$

Given $\alpha > 0$, we obtain the upper-bound for the ith component as follows:

$$\max_{\mathcal{D}_1,\mathcal{D}_2} \frac{\Pr_{\mathcal{D}_1,\alpha}(x_i = j \mid h(\mathbf{x}_{-i}))}{\Pr_{\mathcal{D}_2,\alpha}(x_i = j \mid h(\mathbf{x}_{-i}))} \leq \max_{\mathcal{D}_1,\mathcal{D}_2} \frac{\frac{n_{ij}+1+\alpha}{N_{h(\mathbf{x}_{-i})}+1+C_i\alpha}}{\frac{n_{ij}+\alpha}{N_{h(\mathbf{x}_{-i})}+C_i\alpha}} \leq 1 + \frac{1}{\alpha}$$

where the first inequality is because the two datasets only differ by at most one element. The second inequality comes from the fact that $\frac{N_{h(\mathbf{x}_{-i})}+C_i\alpha}{N_{h(\mathbf{x}_{-i})}+1+C_i\alpha} < 1$

and that the equation is maximized when $n_{ij} = 0$. As we iterate this process for the M variables, the differential probability is upper-bounded by:

$$\frac{\prod_{i=1}^{M} \Pr_{\mathcal{D}_1,\alpha}(x_i \mid h(x_{1:(i-1)}, s_{(i+1):M}))}{\prod_{i=1}^{M} \Pr_{\mathcal{D}_2,\alpha}(x_i \mid h(x_{1:(i-1)}, s_{(i+1):M}))} \leq \prod_{i}^{M}(1 + \frac{1}{\alpha})$$

Therefore, we obtain the relation between α and ϵ as follows:

$$M \log(1 + \frac{1}{\alpha}) \leq \epsilon$$

Rearranging the terms, we have:

$$\alpha \geq \frac{1}{\exp(\epsilon/M) - 1} \tag{3.8}$$

Note that for univariate binary synthetic data, [89] showed the relationship between α and ϵ as $\alpha = \frac{1}{\exp(\epsilon)-1}$. Equation (3.8) says that a higher level of privacy (low ϵ) needs a high value of α. Intuitively, high values of α mean stronger priors, thus the synthetic data are more strongly masked by the priors (or virtual samples).

B. l-Diversity. For l-diversity (See Equation 3.3), perturbed conditional distributions need to satisfy the synthetic l-diversity criterion:

$$H_\alpha(x_i \mid \mathbf{x}_{-i}) = -\sum_{j} \Pr_{\mathcal{D},\alpha} \log \Pr_{\mathcal{D},\alpha} \geq \log l$$

where $H_\alpha(x_i \mid \mathbf{x}_{-i})$ is the Shannon entropy of the perturbed distribution, $\Pr_{\mathcal{D},\alpha}$. The entropy H_α is a monotonically increasing function with respect to α. To satisfy the synthetic l-diversity criterion with minimal perturbation, we set α as follows:

$$\alpha = \begin{cases} \alpha^* \quad \text{s.t.} \quad -\sum_j \Pr_{\mathcal{D},\alpha} \log \Pr_{\mathcal{D},\alpha} = \log l, & \text{if} \quad H_\alpha < \log l \\ 0, & \text{otherwise} \end{cases}$$

where α is set to zero when H_α already satisfies the l-diversity criterion. Unlike the single α for differential privacy, the α values for l-diversity vary depending on conditional distributions. This is because l-diversity applies to a dataset, whereas differential privacy applies to a function. l-diversity is data-aware, but may not provide rigorous guarantees for privacy. This is also noted in [28] who observed that syntactic methods such as k-anonymity and l-diversity are designed for privacy-preserving data publishing, while differential privacy is typically applicable for privacy-preserving data mining. Thus these two approaches are not directly competing, and indeed can be used side-by-side. This chapter also provides a detailed assessment of both the limitations and promise of both types of approaches.

3.2.4 Removing Sampling Footprints

This section illustrates an effective block sampling extension of PeGS, and is specific to differential privacy. PeGS generates one synthetic sample for one seed sample. In other words, one synthetic sample costs ϵ in the differential privacy regime. We modify the PeGS algorithm to sample a block of samples from one seed sample, while achieving the same ϵ-differential privacy. One sampling iteration of PeGS is now repeated many times, but each time, the visited conditional distributions are reset. The procedure of Block PeGS with Reset (PeGS.rs) is as follows:

1. Pick a random seed $\mathbf{s}$ from a predefined pool.

2. For b in $1 : B$,

(a) Sample $\mathbf{x}^{(b)}$ using PeGS seeded by the previous sample $\mathbf{x}^{(b-1)}$, where $\mathbf{x}^{(0)} = \mathbf{s}$

(b) Reset all visited conditional distributions $\Pr(x_i \mid h(\mathbf{x}_{-i}))$ to uniform distributions

This algorithm produces a block of synthetic samples $(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(B)})$ with the same privacy cost ϵ. Figure 3.4 illustrates the process of PeGS with Reset. The synthesizing process of PeGS.rs is exactly the same as the process of PeGS Figure 3.3 except for the resetting step. After sampling from conditional tables, the probability distribution of the visited bin is set to a uniform distribution. The red lines in Figure 3.4 illustrate this resetting step. In our block sampling scheme, there is a chance of re-visiting the bins that are already visited in the previous sampling steps. For such cases, new samples are drawn from uniform distributions, since probability estimates are reset to uniform.

To analyze the privacy aspect of this modified PeGS algorithm, we first need to calculate the probability of synthesizing a block of samples:

$$\Pr^{B}_{\mathcal{D}_1,\alpha}(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(B)} \mid \mathbf{s}) = \Pr^{(1)}_{\mathcal{D}_1,\alpha}(\mathbf{x}^{(1)} \mid \mathbf{s}) \prod_{b=2}^{B} \Pr^{(b)}_{\mathcal{D}_1,\alpha}(\mathbf{x}^{(b)} \mid \mathbf{x}^{(b-1)})$$

where $\Pr^{(b)}_{\mathcal{D}_1,\alpha}(\mathbf{x}^{(b)} \mid \mathbf{x}^{(b-1)})$ is the transition probability from $\mathbf{x}^{(b-1)}$ to $\mathbf{x}^{(b)}$. Note that $\Pr^{(b)}$ and $\Pr^{(b+1)}$ are different conditional distributions, as M components of $\Pr^{(b)}$ are reset to the initial states. The ratio between two probabilities is written as follows:

$$\frac{\Pr^{(1)}_{\mathcal{D}_1,\alpha}(\mathbf{x}^{(1)} \mid \mathbf{s}) \prod_{b=2}^{B} \Pr^{(b)}_{\mathcal{D}_1,\alpha}(\mathbf{x}^{(b)} \mid \mathbf{x}^{(b-1)})}{\Pr^{(1)}_{\mathcal{D}_2,\alpha}(\mathbf{x}^{(1)} \mid \mathbf{s}) \prod_{b=2}^{B} \Pr^{(b)}_{\mathcal{D}_2,\alpha}(\mathbf{x}^{(b)} \mid \mathbf{x}^{(b-1)})} \leq \exp(\epsilon)$$

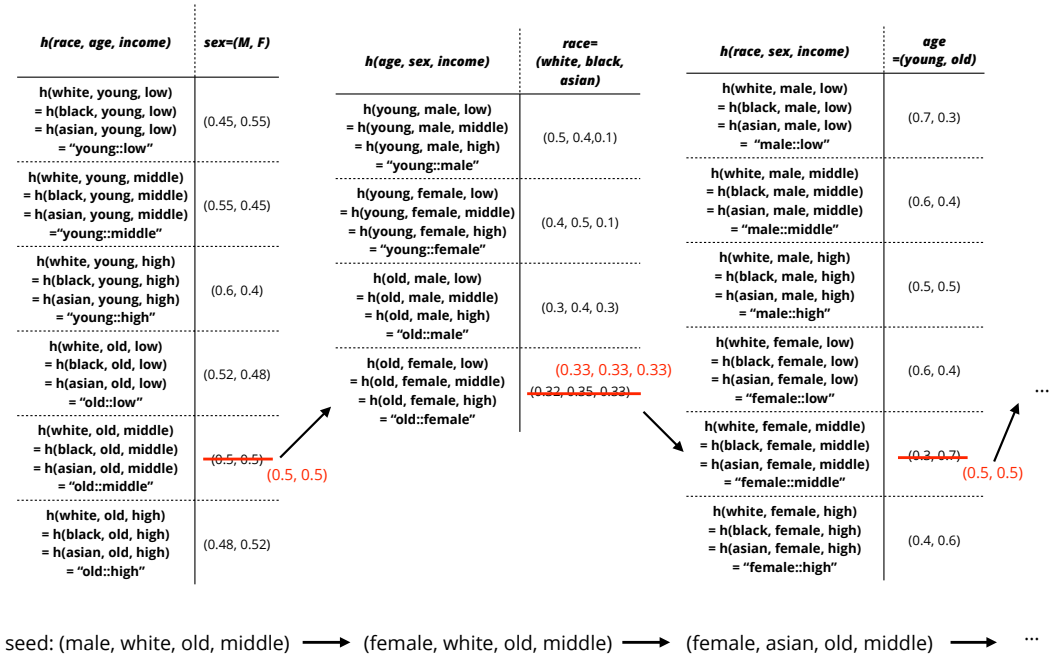


Figure 3.4: Synthesis Steps in PeGS with Reset. Visited rows in statistical building blocks are reset to the initial state. In this example, the initial states are uniform distributions over categories.

Recall that the statistical building blocks from both datasets differ at most M components, as the two datasets differ at most one element. We provide a sketch of the proof that this algorithm satisfies ϵ-differential privacy as follows:

1. To generate the same block of samples, the sequences of statistical building blocks need to be the same as well. In other words, as the two samples, $\mathbf{x}^{(b)} \mid \mathcal{D}_1$ and $\mathbf{x}^{(b)} \mid \mathcal{D}_2$, are the same, $\mathbf{x}^{(b)}_{-i} \mid \mathcal{D}_1$ and $\mathbf{x}^{(b)}_{-i} \mid \mathcal{D}_2$ will also be the same. Thus, they use the building blocks from the same location for sampling x_i at the bth iteration, $\Pr^{(b)}_{\mathcal{D}_1,\alpha}(x_i \mid \mathbf{x}_{-i})$ and $\Pr^{(b)}_{\mathcal{D}_2,\alpha}(x_i \mid \mathbf{x}_{-i})$.

2. There are at most M different components between $\Pr^{(1)}_{\mathcal{D}_1,\alpha}$ and $\Pr^{(1)}_{\mathcal{D}_2,\alpha}$, and let $\mathcal{M}$ be the set of different components. This is because $\mathcal{D}_1$ and $\mathcal{D}_2$ differ by at most one row.

3. If $\Pr^{(1)}_{\mathcal{D}_1,\alpha}$ touched $(M-d)$ components in $\mathcal{M}$, then $\frac{(M-d)}{M}\epsilon$ privacy cost is spent in the process (see Section 3.2.3).

4. If $\Pr^{(1)}_{\mathcal{D}_1,\alpha}$ touched $(M-d)$ components in $\mathcal{M}$, then the rest of the sequences can differ at most d components. This is because those $(M-d)$ components are reset to uniform distributions, and they became indistinguishable i.e. the visited components from $\mathcal{D}_1$ and $\mathcal{D}_2$ became the same uniform distribution. Every visit of an element in $\mathcal{M}$ decreases the number of different elements.

5. Therefore, the whole sequence can differ at most M components (upper-bound), thus the proposed block sampling algorithm satisfies the same ϵ-differential privacy for generating a block of B samples.

As we have more samples for the same cost, the privacy cost per sample can be written as:

$$\alpha \geq \frac{1}{\exp(\epsilon' B/M) - 1} \tag{3.9}$$

where $\epsilon/B = \epsilon'$. The privacy cost is smaller by a factor of B. As an illustrative example, suppose that we need 10 synthetic samples that satisfy ϵ-differential privacy. To obtain 10 samples from PeGS, we perturb the statistical building block by $\alpha_{\text{PeGS}} = \frac{1}{\exp(\epsilon/M)-1}$. On the other hand, if we use $B = 10$, the amount

of perturbation for PeGS.rs is given as $\alpha_{\text{PeGS.rs}} = \frac{1}{\exp(10\epsilon/M)-1}$, which can be much smaller than α_{PeGS}. However, the block size B cannot be arbitrarily large. As every visited statistical building block is reset, the synthetic samples tend to be more noisy as we increase the size of the block.

The relationship between α and ϵ represents the trade-off between utility and risk. Low α values can generate more realistic synthetic data, and low ϵ values can provide higher levels of privacy protection. Note the difference between Equation 3.8 and Equation 3.9. PeGS.rs has the additional parameter B that can fine-tune the relationship between α and ϵ. A smart choice of B can improve the trade-off curve depending on the characteristics of a dataset. This property will be illustrated using a real dataset in Section 3.3.

3.2.5 Perturbed Multiple Imputation

The Dirichlet perturbation can similarly be applied to multiple imputation, specifically the multiple imputation using sequential regressions. Perturbed Multiple Imputation is a naive extension of multiple imputation that satisfies ϵ-differential privacy. A multiple imputation with generalized linear models can be written as follows:

$$\Pr_{\hat{\mathbf{w}}(\mathbf{x})}(\mathbf{x}) = \prod_{i=1}^{M} g_{x_i}(\hat{\mathbf{w}}_i(\mathcal{D}_1)^\top \mathbf{x}_{-i}))$$

where $g_{x_i}(\hat{\mathbf{w}}_i(\mathcal{D}_1)^\top \mathbf{x}_{-i}))$ is the estimated response probability of x_i using a generalized linear model. We assume that the response is a *normalized* probability measure, thus $g_{x_i} \in [0, 1]$. We propose perturbed multiple imputation

as follows:

$$\Pr_{\hat{\mathbf{w}}(\mathbf{x}),\alpha}(\mathbf{x}) = \prod_{i=1}^{M} g_{x_i}^{\alpha}(\hat{\mathbf{w}}_i(\mathcal{D}_1)^{\top}\mathbf{x}_{-i}))$$

Perturbed multiple imputation satisfies ϵ-differential privacy, if the output is perturbed as

$$g_{x_i}^{\alpha}(\hat{\mathbf{w}}_i(\mathcal{D}_1)^{\top}\mathbf{x}_{-i}) = \frac{g_{x_i}(\hat{\mathbf{w}}_i(\mathcal{D}_1)^{\top}\mathbf{x}_{-i}) + \alpha}{\sum_{x_i} g_{x_i \in X_i}(\hat{\mathbf{w}}_i(\mathcal{D}_1)^{\top}\mathbf{x}_{-i}) + C_i\alpha} = \frac{g_{x_i}(\hat{\mathbf{w}}_i(\mathcal{D}_1)^{\top}\mathbf{x}_{-i}) + \alpha}{1 + C_i\alpha}$$

where $\alpha = 1/(\exp(\epsilon/M) - 1)$. The proof is analogous to the proof for the PeGS algorithm. With $\alpha = 0$, this algorithm is the same as a multiple imputation with generalized linear model.

3.3 Empirical Study

In this section, we evaluate the PeGS algorithm using a real dataset from two perspectives: utility and risk of the PeGS-synthesized data. The utility is measured by comparing marginal, conditional distributions and regression coefficients with those from the original data. The risk is first measured by the differential privacy parameter ϵ. As the differential privacy parameter can be too conservative for a real dataset, we also measure population uniqueness and indirect probabilistic disclosure risks. The presented experiments are mainly for the differentially private perturbation, and the experiment with the l-diversity perturbation can be found in [102, 103].

3.3.1 Dataset Overview

We use public Patient Discharge Data from California Office of Statewide Health Planning and Development[2]. This dataset contains inpatient, emergency care, and ambulatory surgery data collected from licensed California hospitals. Each row of the data represents either one discharge event of a patient or one outpatient encounter. The data are already processed with several disclosure limitation techniques. Feature generalization and masking rules are applied to the data based on population uniqueness.

For our experiment, we use 2011 Los Angeles data. Although there are almost 40 variables in the provided data, we use 13 important variables. The selected variables are listed in Table 3.2. For the numeric variables such as age and charge, we transformed the variables into categorical variables by grouping. We subset the data to focus on populous zip code areas, and use this preprocessed dataset to be our ground-truth original data. As can be seen, the possible combinations of the categories are approximately 2 trillion: $2 \times 10^{12} \approx 6 \times 18 \times 3 \times 4 \times 7 \times 16 \times 16 \times 13 \times 9 \times 25 \times 25 \times 3 \times 2$. A table of this size cannot be stored in a personal computer.

Diagnostic and procedural codes are not included in this experiment. In the original data, diagnoses and procedures are coded following the rules of International Classification of Diseases (ICD-9). Both codes can specify very fine levels of diagnoses and procedures; for example, the ICD-9 codes

[2] `http://www.oshpd.ca.gov/HID/Data_Request_Center/Manuals_Guides.html`

include information about a underlying disease and a manifestation in a particular organ. These diagnostic and procedural codes can be grouped into a smaller number of categories. Major Diagnostic Categories (MDC) and Medicare Severity Diagnosis-Related Group (MSDRG) are two examples of coarser diagnostic codes. In this example, we only include higher level abstractions of the detailed features. To keep the semantics of the data, we recommend a two step procedure: first generating a higher level feature, then synthesizing detailed features based on the higher level feature.

Three numeric variables, `age`, length-of-stay (`los`), and `charge`, are grouped and transformed into categorical features. The age variable is equipartitioned to have 5 years gap between consecutive categories. The `los` and `charge` variables are grouped based on their marginal distributions. For example, almost half of the population stayed less than 10 days in a hospital. Thus, the `los` variable is grouped to have 1 day gap before 10 days threshold, and 20 days gap after 10 days. The `charge` variable exhibited a similar marginal distribution; almost a half of the population pay less than 20K dollars, and we binned this variable to have almost equal sizes of population. The grouping rules are illustrated in Table 3.2.

3.3.2 Sampling Demonstration

PeGS transforms each feature one by one conditioned on the rest of the features. This approach differs from a multiple imputation strategy in two aspects. First, PeGS estimates compressed conditional distributions rather

Table 3.2: California discharge data. Los Angeles.

Variable	Description	Category Values
typ	Type of care	Acute Care, Skilled Nursing, etc. (6 levels)
age.yrs	Age of the patient (5 years bin)	0, 5, 10, 15, ..., 80, NA (18 levels)
sex	Gender of the patient	Male, Female, NA (3 levels)
ethncty	Ethnicity of the patient	Hispanic, Non-Hispanic, etc. (4 levels)
race	Race of the patient	White, Black, Asian, etc. (7 levels)
patzip	Patient ZIP code (in LA)	900xx, 902xx, ... , 935xx (16 levels)
los	Length of stay (in days)	0, 1, 2, ... , 9, 50-70, 90+, NA (16 levels)
disp	The consequent arrangement	Routine, Acute Care, etc. (13 levels)
pay	Payment category	Meicare, Medi-Cal, Private, etc. (9 levels)
charge	Total hospital charges	0, 2K, 6K, 8K, 10K, ..., 100K+ (25 levels)
MDC	Major diagnostic category	Nervous sys., Eye, ENMT, etc. (25 levels)
sev	Severity code	0, 1, 2 (3 levels)
cat	Category code	Medical, Surgical (2 levels)

than parameterized approximations e.g., generalized linear models. Second, the compressed conditional distributions can be further perturbed by calibrating the privacy parameter, which makes synthetic data ϵ-differentially private. Table 3.3 shows how PeGS transforms a random seed into a private synthetic sample. The first row of the table is a random seed, and each consecutive row shows the corresponding sampling step. Note that some features change their values, whereas other features maintain the original values. The final sample is shown in the last row. As can be seen, the final transformed sample is different from the seed; for example, it has a different age, zip code, and disposition code.

PeGS can be iterated many times, however, without the reset option, there is no gain for the privacy cost. The reset option in PeGS.rs removes sampling footsteps, but the synthetic samples after many iterations may not be useful for representing the original data. Figure 3.5 shows histograms from the

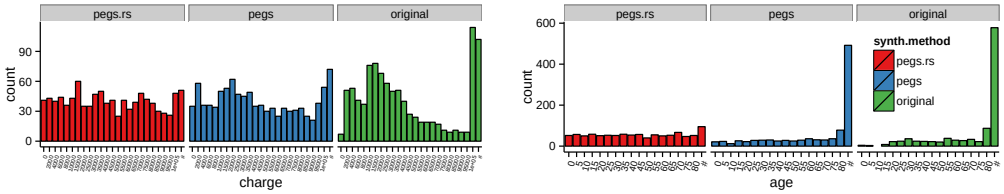
Table 3.3: Detailed Sampling Steps in the PeGS synthesis step. Four variables, sex, race, payment category (pay), category code (cat), are not changed in the final transformed example.

sequence	typ	age	sex	eth	race	zip	los	disp	pay	chg	MDC	sev	cat
seed	4	55	2	1	1	917	8	1	3	40K	25	1	M
$X_1 \mid X_{-1}$	*5*	55	2	1	1	917	8	1	3	40K	25	1	M
$X_2 \mid X_{-2}$	5	*75*	2	1	1	917	8	1	3	40K	25	1	M
$X_3 \mid X_{-3}$	5	75	*2*	1	1	917	8	1	3	40K	25	1	M
$X_4 \mid X_{-4}$	5	75	2	*2*	1	917	8	1	3	40K	25	1	M
$X_5 \mid X_{-5}$	5	75	2	2	*1*	917	8	1	3	40K	25	1	M
$X_6 \mid X_{-6}$	5	75	2	2	1	*913*	8	1	3	40K	25	1	M
$X_7 \mid X_{-7}$	5	75	2	2	1	913	*9*	1	3	40K	25	1	M
$X_8 \mid X_{-8}$	5	75	2	2	1	913	9	*5*	3	40K	25	1	M
$X_9 \mid X_{-9}$	5	75	2	2	1	913	9	5	*3*	40K	25	1	M
$X_{10} \mid X_{-10}$	5	75	2	2	1	913	9	5	3	*65K*	25	1	M
$X_{11} \mid X_{-11}$	5	75	2	2	1	913	9	5	3	65K	*7*	1	M
$X_{12} \mid X_{-12}$	5	75	2	2	1	913	9	5	3	65K	7	*0*	M
$X_{13} \mid X_{-13}$	5	75	2	2	1	913	9	5	3	65K	7	0	*M*

generated samples. As can be seen, the block samples from PeGS.rs are more uniformly distributed than those from PeGS. The distributions from PeGS are actually closer to the distribution of the original data than those from PeGS.rs. It is important to note that the PeGS and PeGS.rs in this experiment have different privacy cost; PeGS.rs only used ϵ, while PeGS requires $\epsilon \times$ Iterations. The goal of this experiment is to show the limitation of PeGS.rs. Although PeGS.rs provides more number of samples given the same privacy cost, an arbitrarily large size of block may not be useful in practice.

3.3.3 Risk (ϵ) vs. Utility

Reducing disclosure risk and improving data utility are two competing objectives when publishing privacy-safe synthetic data. As these two goals



(a) Histogram for the charge variable

(b) Histogram for the age variable

Figure 3.5: Dataset comparisons with PeGS.rs (PeGS with Reset), PeGS, and the original.

cannot be satisfied at the same time, a certain trade-off is necessary for preparing public use data. This trade-off has been traditionally represented using a graphical measure, called R-U confidentiality map [40]. The R-U confidentiality map consists of two axis: typically a risk measure on the x-axis and a utility measure on the y-axis. Note that risk and utility measures can be domain and application specific. In this chapter, we first show R-U maps where the risk is measured using differential privacy. The utility is primarily measured by comparing statistics from the original data and synthetic data.

We use three different algorithms and seven different privacy parameters for each algorithm as follows:

- **PeGS**: Perturbed Gibbs Sampler
- **PeGS.rs**: Perturbed Gibbs Block Sampler with Reset. Block size = 10.
- **PMI**: Perturbed Multiple Imputation (baseline algorithm). With higher values of ϵ, this is the same as a multiple imputation strategy for fully

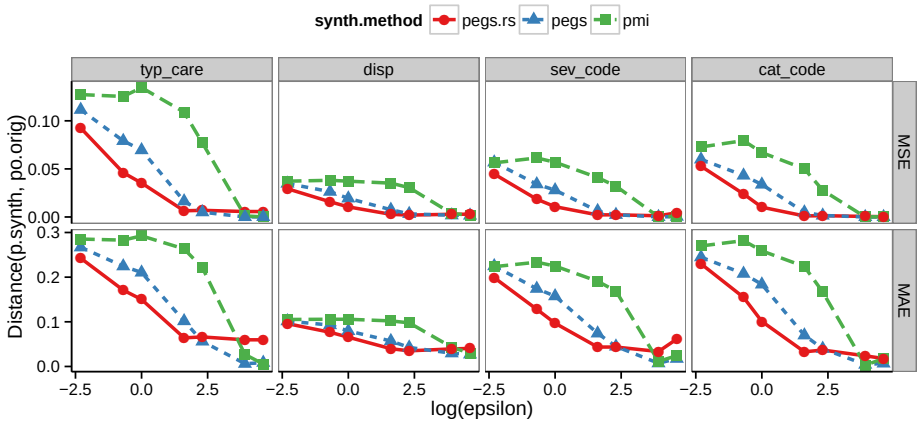


Figure 3.6: R-U maps where the negative utility is measured as the difference in marginal distributions; smaller distances imply greater utilities.

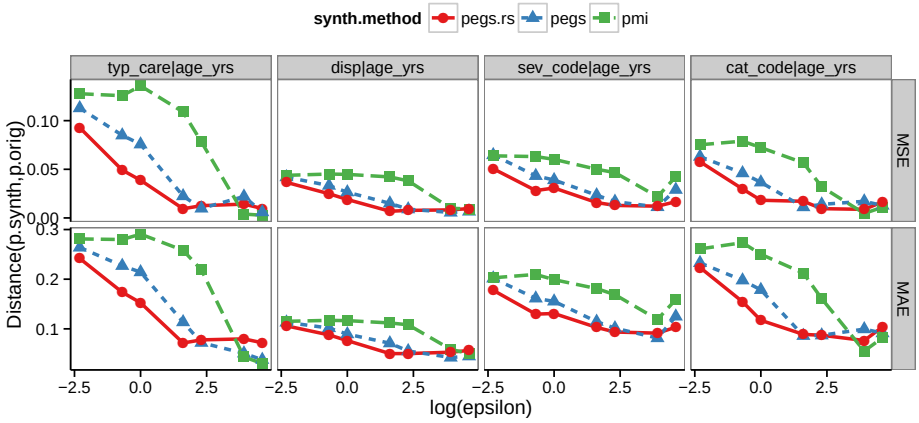
synthetic data. In PMI, the conditional distributions are modeled using the elastic-net regularized multinomial logistic regression, specifically `glmnet` package in `R` 2.15.3 [56]. The variable x_i is regressed on the rest of the variables $\mathbf{x}_{-i}$, and the regularization parameter λ was tuned based-on cross-validation:

$$\Pr(x_i = j \mid \mathbf{x}_{-i}) \propto \exp(c_{ij} + \boldsymbol{\beta}_{ij}^{\top}\mathbf{x}_{-i})$$

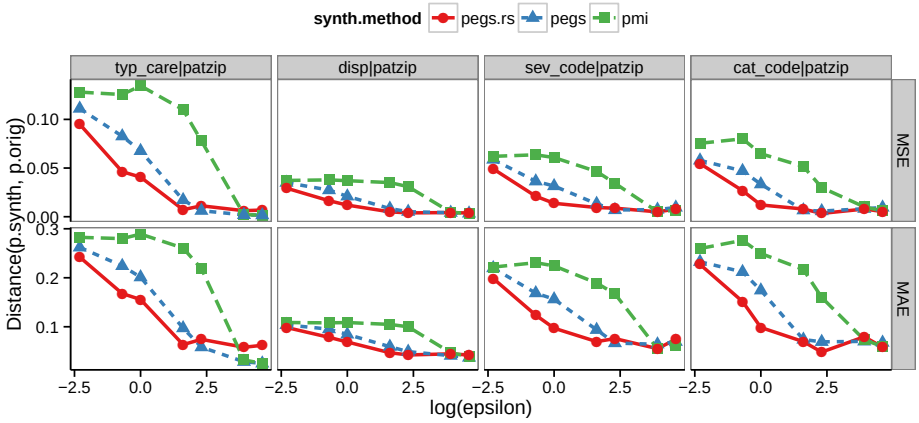
where c_{ij} and $\boldsymbol{\beta}_{ij}$ are estimated from the data.

where the privacy parameters are given as $\epsilon \in \{0.1, 0.5, 1, 5, 10, 50, 100\}$ per synthetic sample. We generated 1000 samples for each case. As a result, we have $21 = 7 \times 3$ synthetic datasets and one original dataset.

The negative utility is first measured using the distance between marginal



(a) Distributional distances conditioned on the age variable, $X_i \mid$ age.yrs



(b) Distributional distances conditioned on the ZIP code variable, $X_i \mid$ patzip

Figure 3.7: R-U maps where the negative utility is measured as the difference in conditional distributions; smaller distances imply greater utilities.

and conditional distributions. The term "negative" implies that smaller the distances, greater the utilities. Marginal and conditional distributions are measured from the original and synthetic datasets, then distances are calculated as follows:

$$\begin{aligned}
\text{Marginal MSE} &= \text{Avg}_{x_i \in X_i}(\hat{\Pr}_{\text{synth},\epsilon}(x_i) - \hat{\Pr}_{\text{orig}}(x_i))^2 \\
\text{Marginal MAE} &= \text{Avg}_{x_i \in X_i}|\hat{\Pr}_{\text{synth},\epsilon}(x_i) - \hat{\Pr}_{\text{orig}}(x_i)| \\
\text{Conditional MSE} &= \text{Avg}_{x_j \in X_j}\text{Avg}_{x_i \in X_i}(\hat{\Pr}_{\text{synth},\epsilon}(x_i \mid x_j) - \hat{\Pr}_{\text{orig}}(x_i \mid x_j))^2 \\
\text{Conditional MAE } &= \text{Avg}_{x_j \in X_j}\text{Avg}_{x_i \in X_i}|\hat{\Pr}_{\text{synth},\epsilon}(x_i \mid x_j) - \hat{\Pr}_{\text{orig}}(x_i \mid x_j)|
\end{aligned}$$

where these distances are inverse surrogates for the utility. Figure 3.6 and Figure 3.7 show the R-U maps where the utility is measured as the difference in marginal and conditional distributions, respectively. As can be seen, all synthetic datasets become similar to the original data with higher values of ϵ. However, for smaller values of ϵ, the synthetic data from PeGS.rs are much more similar to the original than the others. The distributional distances of PeGS are slightly smaller than those of PeGS.rs for higher values of ϵ. Since α values are very small for these privacy parameters, the reset operation of PeGS.rs becomes more noticeable, and it pushes synthetic samples away from the original distributions.

The utility can be measured in many different ways. In this example, we examine whether synthetic samples preserve the ordering of marginal and conditional distributions of the original data. Marginal and conditional distributions are first ranked based on frequencies. We then compare the ranks from

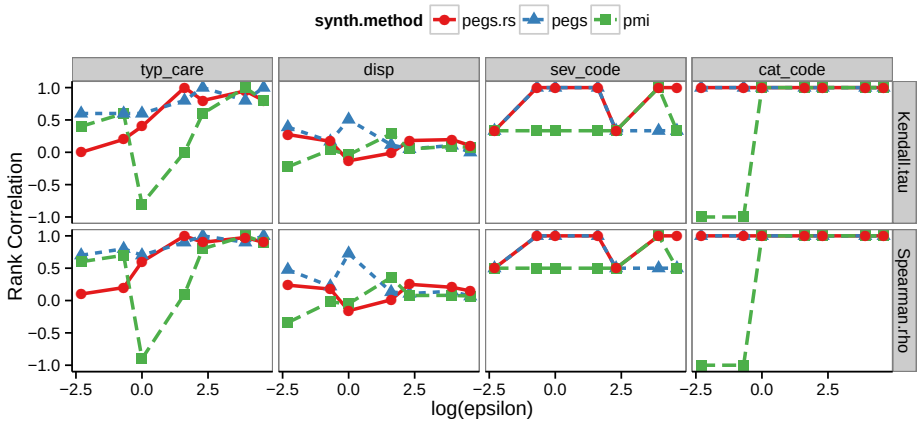


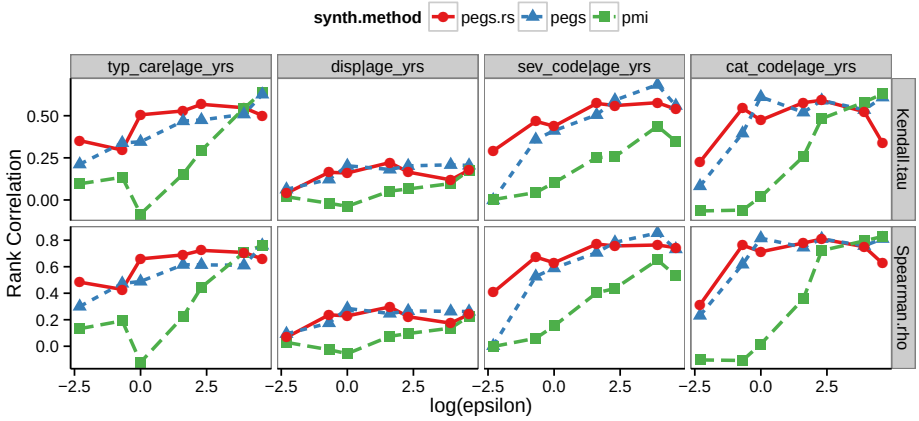
Figure 3.8: Rank correlation between marginal distributions vs. privacy parameter ϵ. As ϵ increases, the ordering of a marginal distribution remains the same as the original ordering.

the original and synthetic distributions using Kendall's τ and Spearman's ρ:

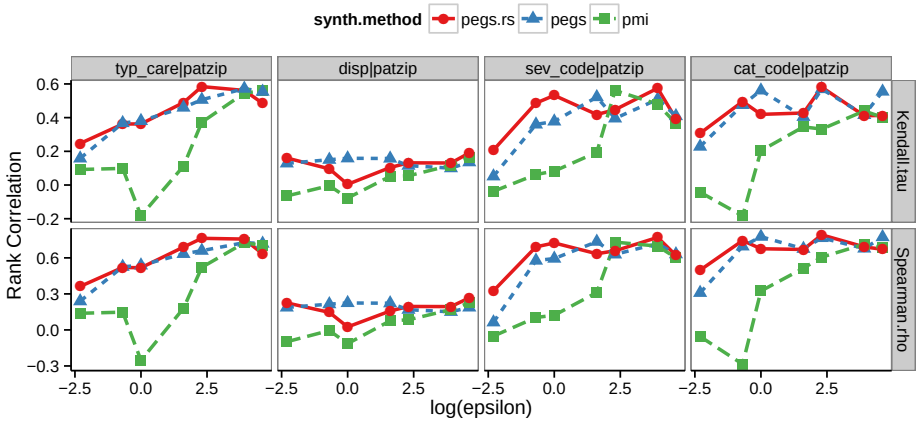
$$\text{Kendall's } \tau = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{\frac{1}{2}C_i(1 - C_i)}$$

$$\text{Spearman's } \rho = \frac{\sum_j (r_j^o - C_i/2)(r_j^s - C_i/2)}{\sqrt{\sum_j (r_j^o - C_i/2)^2 \sum_j (r_j^s - C_i/2)^2}}$$

where C_i represents the number of categories for x_i, and r_j^o and r_j^s are ranks of the j category from the original and synthetic data, respectively. Both τ and ρ lie between -1 (strong negative correlation) and 1 (strong positive correlation). These rank correlation statistics are visualized in Figure 3.8 and 3.9 with respect to the privacy parameter ϵ. As can be seen, less perturbed synthetic datasets better preserve the ordering of distributions. Overall, PeGS.rs best preserves the frequency order of the original distributions.



(a) Rank correlations conditioned on the age variable, $X_i \mid$ age.yrs



(b) Rank correlations conditioned on the ZIP code variable, $X_i \mid$ patzip

Figure 3.9: Rank correlation between conditional distributions vs. privacy parameter ϵ. As ϵ increases, the ordering of a marginal distribution remains the same as the original ordering.

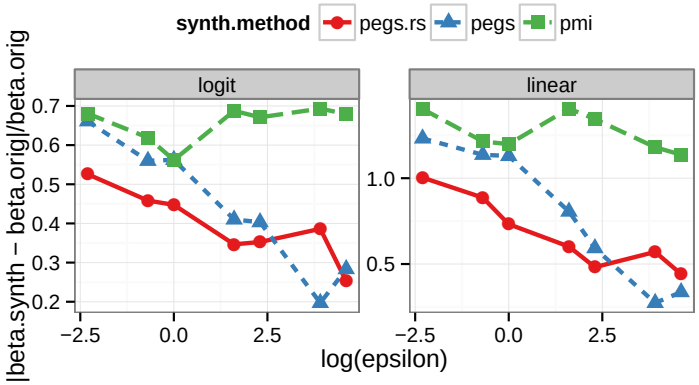
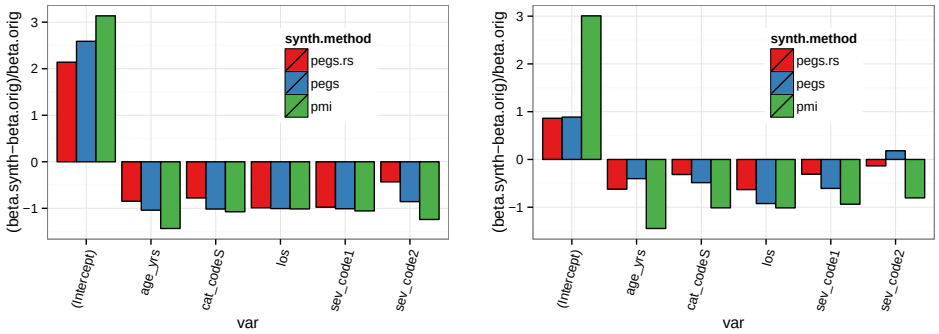


Figure 3.10: R-U maps where the utility is measured as the difference in regression coefficients.



(a) $\epsilon = 0.1$ (b) $\epsilon = 10$

Figure 3.11: Estimated logistic regression coefficients for (a) $\epsilon = 0.1$ and (b) $\epsilon = 1$. The coefficients from $\epsilon = 10$ (lower level of privacy) are closer to the original coefficients than those from $\epsilon = 0.1$ (higher level of privacy).

Next, we compare the coefficients from regression models learned on the datasets. We learned logistic and linear models as follows:

$$I(\text{charge} > 25K) \sim \text{as.numeric(age.yrs)} + \text{sev} + \text{cat} + \text{as.numeric(los)}$$

$$\text{as.numeric(charge)} \sim \text{as.numeric(age.yrs)} + \text{sev} + \text{cat} + \text{as.numeric(los)}$$

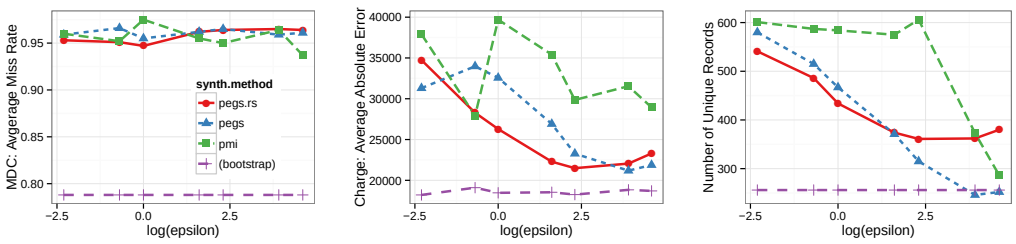
where some of the features are changed to numeric features based on their actual meaning. The choice of the target variable was arbitrary, as the goal of this illustrative experiment is to show the applicability of synthetic data in predictive modeling tasks. After learning the coefficients of each model, the distance between the coefficients is measured as follows:

$$\text{Regression Distance} = \sum_i |\frac{\beta_{i,\text{synth}} - \beta_{i,\text{orig}}}{\beta_{i,\text{orig}}}|$$

Figure 3.10 shows the R-U map from the regression experiment. As can be seen, the synthetic samples from PeGS.rs provide the most similar coefficients to those from the original data. Figure 3.11 shows each coefficient deviation from the linear regression example. Notice that the intercept coefficients from the synthetic datasets tend to overshoot the actual value, while the other feature coefficients tend to undershoot. This is because the perturbation decreases all feature correlations including the correlation between the target and independent variables.

3.3.4 Estimating Re-identification Risk

Although differential privacy provides a theoretically sound framework for measuring disclosure risks, the measure is originally designed for functions,



(a) Attack on the MDC variable (b) Attack on the charge variable (c) Population uniqueness

Figure 3.12: Simulated attack scenarios on the MDC and charge variables (left and center), and population uniqueness (right).

not data [32]. For many cases, the measures can be overly conservative or strict for a real dataset. In the statistical disclosure limitation literature, there have been many attempts to measure disclosure risks for synthetic data. Franconi and Stander [52] proposed a method to quantify disclosure risks for model-based synthetic data. Their proposed approach checks whether it is possible to recognize a unit in the released data assuming the original data are given to an intruder. This provides a somewhat conservative measure, but is still useful to compare the risks from different release mechanisms. Reiter [111] later formalized measuring probabilistic disclosure risk scores for partially or fully synthetic data. Probabilistic disclosure risks are used to asses the risks of the fully synthetic data using Random Forests in Caiola and Reiter [18].

We measure the disclosure risks from two different angles: recoverability of feature values and population uniqueness. First, we examine whether it is possible to infer the values of sensitive feature given demographic information. Specifically, if the intruder knows someone's `age`, `sex`, `los`, and `zip`, we would

like to measure the likelihood of getting the correct values as follows:

$$\mathrm{E}[\mathbb{1}(\text{inferred MDC} \neq \text{correct MDC}) \mid \text{age}, \text{sex}, \text{zip}]$$

$$\mathrm{E}[|\text{inferred charge} - \text{correct charge}| \mid \text{age}, \text{los}, \text{zip}]$$

where the inferred values are (1) the most frequent MDC categories and (2) sample means from conditioned synthetic samples. We also measure the population uniqueness based on age, sex, and zip code information. Figure 3.12 shows the results from this simulated intruder experiment. Private records are more difficult to reconstruct if misclassification rates and absolute errors are high. The probability of recovering MDC is significantly lower than using a simple bootstrap method, but no one method is distinctly better than the other. The absolute distance of hospital charges shows that synthetic data has comparable predictive power with the bootstrap method. Noticeably, the absolute errors are higher when the differential privacy parameters are low, and this finding partially supports our use of differential privacy as a disclosure risk measure. As can be seen in Figure 3.12 (right), the perturbed synthetic datasets have more unique samples. This is the most distinct characteristics of PeGS compared to other statistical disclosure techniques. Privacy preserving algorithms, such as k-anonymity and l-diversity, try to reduce population uniqueness, while PeGS increases the diversity of samples. The former algorithms apply privacy-preserving transforms on the original data, while the latter algorithm synthesizes a diversified dataset.

3.4 Summary

We proposed a categorical data synthesizer that guarantees prescribed differential privacy or l-diversity levels. The use of a hash function allows the Perturbed Gibbs Sampler to handle high-dimensional categorical data. The non-parametric modeling of categorial data provides a flexible alternative to traditional (GLM-based) Multiple Imputation techniques. Additionally, this simple representation of conditional distributions is a crucial component of our block sampling algorithm that enhances the utility of synthetic data given a fixed privacy budget.

The California Patient Discharge dataset was used to demonstrate the analytical validity and utility of the proposed synthetic methodologies. Marginal and conditional distributions, as well as regression coefficients of predictive models learned from the synthesized data were compared to those from the original data to quantify the amount of distortion introduced by the synthesization process. Simulated intruder scenarios were studied to show the confidentiality of the synthesized data. The empirical studies showed that the proposed mechanisms can provide useful risk-calibrated synthetic data.

Currently, PeGS only deals with categorical variables. Numeric variables need to be binned to form categorical variables. Although this approach may be adequate enough for some applications, brute-force binning ignores numeric similarity or ordering information. For example, two consecutive values from an ordinal variable are more similar than separated values. Consider a size variable with three values: small, medium, and large. The ordering in-

formation states that similarity(small, medium) > similarity(small, big), but this information is lost if we bin the size variable into three (non-ordered) categories. Such semantic correlation cannot be captured in the current synthetic and perturbation model.

Although it was originally designed for computational efficiency, the hashing step of PeGS also provides an added degree of privacy protection. When building the PeGS statistical building blocks, each row $\mathbf{x}$ of the original data is hashed based on $h(\mathbf{x}_{-i})$, and aggregated with other rows with the same hash key, $\{\mathbf{z} \mid h(\mathbf{z}_{-i}) = h(\mathbf{x}_{-i})\}$. This aggregation (or hashing) step should be also incorporated for a tighter guarantee of privacy. The privacy guarantee of PeGS will be affected by different hash resolutions and mechanisms, and this topic needs to be covered in future work.

Although the proposed algorithms show substantially better performance on ϵ-differential privacy and l-diversity[3] measures, they were only marginally better than PMI in other probabilistic disclosure risk measures. The differential privacy measure may be too conservative for real data, and the probabilistic measure may not exhaustively capture all the attack scenarios. This is why we provided multiple risk measures. The connection between the differential privacy and disclosure risks should be further addressed to better evaluate the validity and utility of the synthetic data.

In practice, multiple disclosure techniques are sequentially mixed to

[3]Experimental results on l-diversified synthetic data are presented in [102].

achieve better protection of the records. For example, PeGS can be applied on top of feature generalization or masking techniques. Furthermore, some features can be modeled using generalized linear models; for example, numeric features. It would be worthwhile to investigate novel cocktails of different statistical disclosure limitation techniques.

Chapter 4

Privacy-aware Aggregate Data Utilization

Individual-level datasets that contain one or more records per person are rich sources for data mining applications. In the healthcare domain, the application of advanced data mining methods on individual level records across large populations can enable major breakthroughs in both personalized and population-level healthcare, leading to much improved, more cost-effective and timely diagnoses and interventions [106]. However, such data often contain a substantial amount of privacy-sensitive attributes. In practice, as previously described in Chapter 2, privacy concerns are typically addressed through multiple Statistical Disclosure Limitation (SDL) techniques [39], such as data aggregation [6], data swapping [31, 50], top-coding, feature generalization such as k-anonymity [128] or l-diversity [87], and additive random noise with measurement error [59]. Each method has distinct utility and risk aspects. Often an appropriate mix of disclosure limitation techniques is carefully chosen by domain experts and statisticians. For example, Centers for Medicare and Medicaid Services applied six different SDL techniques when publishing synthetic public use files[1]: variable reduction, suppression, substitution, imputation,

[1] http://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/SynPUFs/DE_Syn_PUF.html

data perturbation, and coarsening.

Among various SDL approaches, data aggregation is currently the most widely used. Data aggregation is a process of summarizing individual-level data into a small set of representative values such as mean and median statistics computed over groups that are typically geographically or administratively defined (such as county, hospital group, state, etc). This process is straightforward to apply on diverse datasets: wireless sensor networks [66], regional healthcare statistics [22], and government data [33]. Moreover, such aggregate data can be efficiently and effortlessly generated in RDBMS [99] and statistical programming languages [129]. Data collecting agencies publish various aggregate datasets at different levels of aggregation (including individual-level for non-sensitive information). In particular, the U.S. government's open data project, data.gov has recently released a substantial amount of regional and topic-based aggregate data regarding agriculture, education, and energy. Centers for Disease Control and Prevention annually publishes various regional statistics related to aging, cancer, and diabetes. Other notable sources of aggregated health data are dartmouthatlas.org and healthdata.gov.

The use of aggregate data is typically limited to group-level studies, often referred to as ecological studies for historic reasons. Applying the result from aggregate data to individual-level inference often results in the classic problem of ecological fallacy [117]. Ecological fallacy occurs when aggregate-level statistics are misinterpreted as individual-level inferences. For example, the high correlation between "per capita consumption of dietary fat" and

"breast cancer" in different countries [21] does not imply that dietary fat causes breast cancer.

There have been many attempts to circumvent the ecological fallacy while analyzing aggregate data. This is because individual-level data acquisition is usually expensive, and it is sometimes legally and ethically implausible. Duncan [42] developed the method of bounds that uses the constraints of contingency tables, but the bounds are often uninformative in real applications [54]. The constancy assumption, suggested by Goodman [64], allows an individual-level interpretation of ecological regression. Suppose that we want to check the relationship between Length of Stay (LoS) and Hospital Charge (HC) variables from state-level aggregate data:

$$\mathrm{HC}_{\mathrm{state}} \sim c_{\mathrm{state}} + \beta_{\mathrm{state}} \mathrm{LoS}_{\mathrm{state}}$$

The constancy assumption states that daily hospital charge rates are the same across different states i.e. $\beta_{\mathrm{state}} = \beta$ and $c_{\mathrm{state}} = c$. Of course, this assumption is rarely true in real datasets; for this example, it is more natural to assume that each state has a different daily charge rate, thereby indicating that multi-level modeling can be used [62]. Such an approach, however, is under-identified and can't be solved using aggregate data, since we have more parameters than observations. King [76, 77] proposed a Bayesian prior-based multi-level approach to overcome the limitation of Goodman's assumption, but Freedman [53] criticized that King's method cannot be validated on the basis of aggregate data.

Table 4.1: Illustrative health data files: artificial individual-level data (left) and aggregate-level summary (right) [73].

ID	Age	Length	State
1	19	1 day	TX
2	35	2 days	CA
3	3	10 days	FL
6	68	100 days	FL
⋮	⋮	⋮	⋮

State	Avg. Hospital Charge
CA	$ 2,706
FL	$ 1,809
NY	$ 1,954
TX	$ 2,001
⋮	⋮

We provide a novel approach for addressing the ecological fallacy dilemma by leveraging available sources of individual-level data for which the values of the partitioning or aggregation variable is known. For example, an aggregation variable can indicate state, county, or zip codes, that can be used to link to aggregate-level dataset that is aggregated along such geographical regions. In practice, it is not difficult to collect multiple datasets with different levels of aggregations from multiple agencies, so little added data-collection expense is involved.

Table 4.1 shows a simple, illustrative example of two health datasets. Non-sensitive fields are published at individual-level, while a sensitive field (hospital charge) is aggregated over the partition variable "state". Our approach is substantially different from previous ecological fallacy solutions where only aggregate data were considered.

We use a two-stage approach to avoid the ecological fallacy. We first reconstruct the masked individual-level variables from aggregate data, then apply multi-level regression models to the reconstructed data. In other words, we first synthesize "pseudo individual-level" data that are statistically similar

to the original (unseen) individual-level data. Not only multi-level regression models, but also numerous off-the-shelf data mining algorithms can be easily applied to such pseudo individual-level data. Our reconstruction algorithm is based on two key observations:

- Aggregation is a linear transformation, thus it preserves several algebraic properties including the associative property.
- Using a proper data model, additional individual-level data can provide statistical clues for the reconstruction of the masked columns. From the previous hospital charge example, if we know *a priori* that hospital charge (aggregate-level) is a function of length of stay (individual-level), we roughly expect that a person with a longer stay may have paid more than a person who stayed only a day. We demonstrate that such clues can be captured using a low rank model.

We demonstrate our reconstruction algorithm on both simulated and real datasets. Many factors contribute to the reconstruction quality, for example, the number of data points per aggregation and correlation strength with other columns. These factors will be illustrated in Section 4.5 using Texas Inpatient Public Use Files. The main contributions are:

- We formulate a data model, LUDIA, that reconstructs individual values from aggregate values.

- We derive efficient algorithms for solving optimization problems associated with LUDIA.
- We show that our reconstructed data can capture aggregate-level random effects, thus the reconstructed data can be used for multi-level analyses as well as more sophisticated data mining applications.

The first two contributions will be illustrated in Section 4.3, the last contribution will be explained in Section 4.4. Experimental results are provided in Section 4.5.

4.1 Preliminaries & Related Work

This section starts by setting up the notation of this chapter, and visiting two key existing approaches for tackling aggregate data. We extend these approaches to reconstruct the original individual-level data, and briefly discuss their modeling assumptions and limitations.

Aggregation is a compressive linear transformation, which we denote as $\mathbf{A}$. For example, suppose that there are five individuals from two different groups: the first two from Group A and the last three from Group B. Individual-level observations, say $\mathbf{y} = \begin{bmatrix}1 & 2 & 3 & 4 & 5\end{bmatrix}^\top$, can be aggregated into two groups by multiplying an aggregation matrix defined as follows:

$$\mathbf{s} = \begin{bmatrix}1.5 \\ 4\end{bmatrix} = \mathbf{A}\mathbf{y} = \begin{bmatrix}1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 1/3 & 1/3\end{bmatrix} \mathbf{y}$$

Table 4.2 summarizes the notation of this chapter.

Table 4.2: Notation. For simplicity but without loss of generality, we use $d = 1$ in this chapter.

Symbol	Explanation
$\mathbf{X}$	$n \times m$, individual-level matrix
$\mathbf{x}_i$	$1 \times m$, ith row of $\mathbf{X}$
$\mathbf{y}$	$n \times d$, masked individual-level vector
$\mathbf{A}$	$p \times n$, aggregation matrix
$\mathbf{s}$	$p \times d$, aggregate-level vector i.e. $\mathbf{Ay}$
$\mathbf{U}, \mathbf{V}$	$n \times r$, $m \times r$ low-rank matrices

The processes of aggregation and reconstruction can be illustrated as follows:

$$\text{(Compression)} \qquad \mathbf{Ay} \xrightarrow{\text{compressive linear}} \mathbf{s}$$

$$\text{(Reconstruction)} \qquad \hat{\mathbf{y}} \xleftarrow{\text{low-rank modeling}} \text{Recon}(\mathbf{s}, \mathbf{A}, \mathbf{X})$$

where $\mathbf{X}$ represents individual-level data, and Recon is a reconstruction algorithm. To give a brief overview, our reconstruction algorithm, LUDIA (**L**ow-rank factorization **U**sing **Di**fferent levels of **A**ggregation), is a constrained low-rank factorization algorithm that can capture multi-level effects. Figure 4.1 illustrates the overall idea of LUDIA and other reconstruction algorithms. We have two sources of errors that construct our reconstruction triangle: aggregation and modeling errors. LUDIA reduces the aggregation error using a low-rank model, but the LUDIA error is lower-bounded by the modeling error.

To illustrate existing approaches for aggregate data, let us consider the previous "hospital charge vs. length of stay" example. When using aggregate data, three approaches have been popular:

- The neighborhood model [55], proposed by Freedman, will imply that hospital charges are more influenced by geographical attributes rather

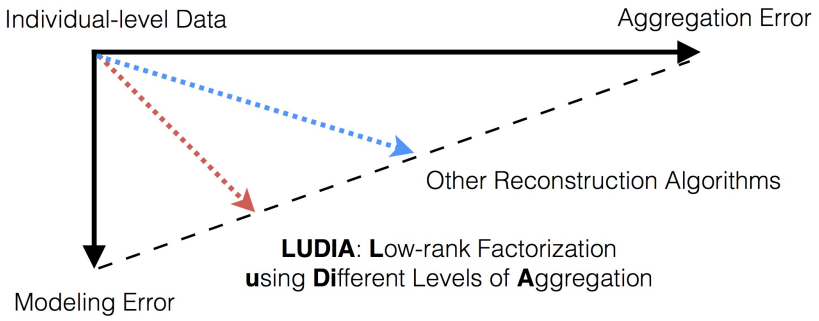


Figure 4.1: Reconstruction triangle and LUDIA.

than the length of stay variable, since each geographical partition is assumed to contain a homogeneous population group.

- Ecological regression [63], suggested by Goodman, will assume that the effect size of length of stay is the same across different states, based on the constancy assumption. According to the constancy assumption, geographical partitions are treated as different batches of i.i.d. experiments.

- Ecological inference, also known as King's method [77], combines the method of bounds and Goodman's ecological regression. King's method is a multi-level approach that models different effect sizes for different states. The multi-level parameters are first characterized by their acceptable regions using the method of bounds, then their joint distributions are modeled under three assumptions [78]: uni-modal joint distribution, absence of spatial correlation, and independence between multi-level coefficients and dependent variables. However, these assumptions are not verifiable on the basis of aggregate-level data [53], and this method re-

quires manual tuning of the parameter distributions. In short, ecological inference is a method with many knobs and unverifiable assumptions, and we do not include this method in our baseline methods.

These previous approaches have been developed to tackle aggregate data, and need to be slightly modified to synthesize individual-level data. Imagine that we now obtained individual-level length of stay data[2] $\mathbf{X}$ and each individual's location information $\mathbf{A}$. To reconstruct the masked hospital charge data $\mathbf{y}$, two direct extensions from the previous approaches can be considered:

- Moore-Penrose (MP) solution is an extension of the neighborhood model. As the neighborhood model only focuses on the aggregation matrix $\mathbf{A}$, the reconstructed values are obtained by applying the Moore-Penrose pseudo-inverse of $\mathbf{A}$ to the aggregate data:

  $$\hat{\mathbf{y}}_{\text{MP}} = \mathbf{A}^{+}\mathbf{s}$$

  where $\mathbf{A}^{+} = \mathbf{A}(\mathbf{A}\mathbf{A}^{\top})^{-1}$.

- Ecological Regression (ER) solution is an extension of Goodman's ecological regression. Assuming that the effect sizes are the same across

[2]**Note**: This simple example has only one individual level (LoS) and one aggregated (HC) feature, and one level of aggregation, called "State", so as to convey the concepts most easily. Our approach readily generalizes to multiple individual and aggregate variables as well as multiple levels of aggregations, as will be seen later

different states, we obtain the regression parameter $\boldsymbol{\beta}$ from the aggregate data, then apply to the individual-level covariate:

$$\hat{\mathbf{y}}_{\text{ER}} = \boldsymbol{\beta}_{\text{ER}}\mathbf{X}$$

where $\boldsymbol{\beta}_{\text{ER}} = (\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top\mathbf{s}$ and $\mathbf{Z}$ is the aggregate-level representation of $\mathbf{X}$ i.e. $\mathbf{AX}$.

MP and ER exhibit different failure modes. MP ignores the effects of individual-level covariates, which may substantially leverage the utility of aggregate data. On the other hand, ER relies on the constancy assumption, which is rarely true in real settings.

For our hospital charge example, daily charge rates are significantly different across city and rural areas (see Section 4.5). This geographical variation on daily charge rates can be expressed as follows:

$$\begin{aligned} y_i &= \mathbf{x}_i\boldsymbol{\beta}_{\text{state}} + \zeta_{\text{state}} + e_i & e_i &\sim \text{N}(0, \sigma_e^2) \\ \boldsymbol{\beta}_{\text{state}} &= \boldsymbol{\beta}_{\text{global}} + \boldsymbol{\eta}_{\text{state}} & \boldsymbol{\eta}_{\text{state}} &\sim \text{N}(0, \sigma_\eta^2\mathbf{D}) \\ \zeta_{\text{state}} &\sim \text{N}(0, \sigma_\zeta^2) \end{aligned}$$

where ζ_{state} and $\boldsymbol{\eta}_{\text{state}}$ represent state-level biases for the intercept and slope; they are called random intercept and random slope, respectively. Assuming that we have two states A and B, and individuals listed by state, this multi-

level approach [60] can be written in a matrix form:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_{n-1} \\ \mathbf{x}_n \end{bmatrix} \boldsymbol{\beta}_{\text{global}} + \begin{bmatrix} \mathbf{x}_1 & \mathbf{0} & 1 & 0 \\ \mathbf{x}_2 & \mathbf{0} & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{x}_{n-1} & 0 & 1 \\ \mathbf{0} & \mathbf{x}_n & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_A \\ \boldsymbol{\eta}_B \\ \zeta_A \\ \zeta_B \end{bmatrix} + \mathbf{E}$$

We define new matrices $\boldsymbol{\gamma}$ (random effects) and $\mathbf{G}$ (covariates for random effects) to obtain a compact form:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}_{\text{global}} + \mathbf{G}\boldsymbol{\gamma} + \mathbf{E} \tag{4.1}$$

Aggregate data are obtained through the aggregation operation as follows:

$$\mathbf{A}\mathbf{y} = \mathbf{A}\mathbf{X}\boldsymbol{\beta}_{\text{global}} + \mathbf{A}\mathbf{G}\boldsymbol{\gamma} + \mathbf{A}\mathbf{E}$$

$$\Rightarrow \qquad \mathbf{s} = \mathbf{Z}\boldsymbol{\beta}_{\text{global}} + \mathbf{A}\mathbf{G}\boldsymbol{\gamma} + \mathbf{A}\mathbf{E}$$

As can be seen, the ER solution is valid only if

- $\boldsymbol{\gamma} = 0$ (no random effect)
- $((\mathbf{A}\mathbf{X})^\top \mathbf{A}\mathbf{X})^{-1}(\mathbf{A}\mathbf{X})^\top \mathbf{A}\mathbf{y} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}$

These two conditions are rarely realistic in real applications.

MP and ER are formulated based on two orthogonal assumptions. MP assumes that only geographical partitions affect the dependent variable, while ER posits that geographical partitions are merely random groupings. These assumptions are necessary to obtain some meaningful results from aggregate data, as the ecological fallacy is, in fact, the problem of statistical under-identification [115]. However, the direct extensions from the previous approaches do not utilize the full potential of auxiliary individual-level data.

4.2 CUDIA

CUDIA [100], which is our previous work as well as the motivation of this chapter, is a probabilistic clustering algorithm that utilizes both aggregated and individual-level data. The parameters of CUDIA are estimated using a Monte-Carlo Expectation Maximization (MCEM) algorithm. Although CUDIA can reasonably reconstruct the data based on the estimated cluster centers, the primary objective of CUDIA is still clustering rather than reconstruction. Furthermore, the presented MCEM algorithm is not scalable to large-scale data. We show that CUDIA is, in fact, a special case of LUDIA with a non-negative constraint on $\mathbf{U}$ (see Section 4.4.3). LUDIA generalizes CUDIA with a more flexible representation of $\mathbf{U}$. This generalization provides an efficient optimization algorithm that is suitable for large-scale data.

4.2.1 Problem Formulation

Suppose two datasets from possibly multiple sources are available for research where their aggregation levels are also different. We refer to the dataset with a finer granularity as the individual-level dataset, and the other dataset as the aggregate-level dataset. Assuming that the dataset of interest is generated by a mixture model that represents underlying heterogeneous groups, we introduce a novel generative process that captures the underlying distributions using a Bayesian directed graphical model and the Central Limit Theorem. Figure 4.2 illustrates the overall flow of the algorithm. Many datasets in the healthcare domain are divided into multiple tables contain-

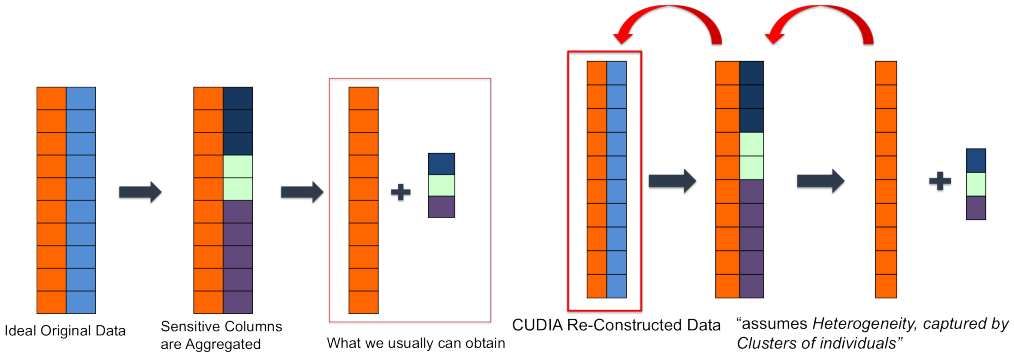


Figure 4.2: The orange and blue columns represent non-sensitive and sensitive features, respectively. The sensitive column is aggregated, and we typically observe the summary statistics. With the heterogeneity assumption, our algorithm re-constructs the individual-level data.

ing different levels of aggregation (sometimes obtained from different sources), and the suggested methodology in this section can be useful in increasing the utility in such scenarios.

Suppose a "complete" microdata, $\mathcal{D} = \{(x, y)_i\}_{i=1}^{N}$ where x and y are two random variables. We assume that x contains non-sensitive information, while y comprises of a sensitive field such as a disease record. The privacy-sensitive variable is aggregated over partitions that are defined as: $\mathcal{P} = \{\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^P\}$, where $\bigcup_{p=1}^{P} \mathcal{D}^p = \mathcal{D}$ and $\mathcal{D}^p \bigcap \mathcal{D}^q = \emptyset$ for any distinct p, q. The aggregate statistics of y are obtained as $\mathcal{D}_s = \{s_p\}_{p=1}^{P}$, where $s_p = \text{Average}(y_i \mid i \in \mathcal{D}^p)$. In other words, $\mathbf{s} = \mathbf{A}\mathbf{y}$ following the notation in Section 4.1.

We postulate the sample population is heterogeneous, thus there exist K distinct clusters denoted by a discrete random variable z. For a technical

reason, we additionally assumes that two feature vectors are conditionally independent given their cluster labels: $p(x, y \mid z) = p(x \mid z)p(y \mid z)$. Let $\boldsymbol{\pi}_p = (p(z = 1 \mid \mathcal{D}^p), p(z = 2 \mid \mathcal{D}^p), \ldots, p(z = K \mid \mathcal{D}^p)) = (\pi_{p1}, \pi_{p2}, ..., \pi_{pK})$, which represents the mixing coefficients of partition p. Let ξ_k and θ_k be the sufficient statistics for the probability densities $p(x \mid z = k)$ and $p(y \mid z = k)$ respectively. If all data features are observed at the individual level, an LDA-like clustering model can be built based on the conditional independence assumption.

The complexity of the model can be reduced by removing the unobserved variable y. Assuming that $N_p = |\mathcal{D}^p|$ is large enough, let η_k and T_k^2 be the mean and covariance of the distribution, $p(y \mid z = k)$. Using the *linearity* of mean statistics and the *Central Limit Theorem* (CLT), s_p can be approximated as being generated from a normal distribution as follows:

$$s_p \sim \text{Normal}(\mu_p, \Sigma_p^2) \tag{4.2}$$

$$\mu_p = \boldsymbol{\pi}_p \cdot \boldsymbol{\eta} \tag{4.3}$$

$$\Sigma_p^2 = \frac{1}{N_p}[\boldsymbol{\pi}_p \cdot (\boldsymbol{\eta}^2 + \mathbf{T}^2) - \mu_p^2] \tag{4.4}$$

where $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_K)^\top$, $\mathbf{T}_k^2 = (T_1^2, T_2^2, \ldots, T_K^2)^\top$. We now remove y resulting in the **C**lustering **U**sing features with **DI**fferent levels of **A**ggregation (CUDIA) model. The full generative process for CUDIA is as follows:

For s_p in $\mathcal{D}_s$

Sample $\boldsymbol{\pi}_p \sim \text{Dirichlet}(\boldsymbol{\alpha})$

Sample $s_p \sim \mathrm{N}(\mu_p, \Sigma_p^2)$

For x in $\mathcal{D}^p$

Sample $z \sim \mathrm{Multinomial}(\boldsymbol{\pi}_p)$

Sample $x \sim p(x \mid \theta_z)$

where π is sampled from a Dirichlet distribution parametrized by $\boldsymbol{\alpha}$, and the observed sample mean statistics s is generated from a Normal distribution parametrized by a mixture of true means η's and a covariance Σ^2. The clustering index z in each partition is sampled from a Multinomial distribution parametrized by $\boldsymbol{\pi}$, which is specific to the partition, and corresponding x is sampled from a distribution $p(x \mid \theta_z)$, where the suitable form of $p(x \mid \theta_z)$ depends on the properties of the variable x.

Algorithm 1: CUDIA MCEM algorithm

Input: $\mathbf{x}, \mathbf{s}$
Output: $\boldsymbol{\eta}, \boldsymbol{\theta}, \boldsymbol{\alpha}$
repeat
  (E-Step) Algorithm 2 in [100];
  (M-Step) Learn $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$;
  $\mathbf{H}^*_{\cdot i} = (\hat{\mathbf{\Pi}}^T \mathbf{W} \hat{\mathbf{\Pi}} + \lambda \mathbf{I})^{-1} \hat{\mathbf{\Pi}}^T \mathbf{W} \mathbf{S}_i$;
until *Convergence*;

4.2.2 Parameter Estimation

From the generative process, the likelihood function of the CUDIA model is as follows:

$$
\begin{aligned}
& p(\mathbf{x}, \mathbf{s}) \\
&= \sum_{z} \int_{\boldsymbol{\pi}} p(\mathbf{x}, \mathbf{s} \mid \mathbf{z}, \boldsymbol{\pi}) p(\mathbf{z}, \boldsymbol{\pi}) d\boldsymbol{\pi} \\
&= \sum_{z} \int_{\boldsymbol{\pi}} p(\mathbf{x} \mid \mathbf{z}) p(\mathbf{s} \mid \boldsymbol{\pi}) p(\mathbf{z} \mid \boldsymbol{\pi}) p(\boldsymbol{\pi}) d\boldsymbol{\pi} \\
&= \sum_{z} \int_{\boldsymbol{\pi}} \prod_{i} p(x_i \mid z_i) p(z_i \mid \boldsymbol{\pi}_p) \prod_{p} p(s_p \mid \boldsymbol{\pi}_p) p(\boldsymbol{\pi}_p) d\boldsymbol{\pi}
\end{aligned}
$$

A generic EM algorithm [34] cannot estimate this posterior distribution, since the normalization constant of its posterior distribution is intractable. Collapsed Gibbs sampling [84] also cannot be applied because $\boldsymbol{\pi}$ cannot be integrated out due to non-conjugacy between $\mathbf{s}$ and $\boldsymbol{\pi}$. In this case, the model can be learned using either variational methods or Gibbs sampling approaches, and CUDIA follows the latter alternative. Naïve Gibbs sampling approaches are computationally inefficient. We employ an approximated Gibbs sampling approach, which can be applied when the dimension of $\mathbf{x}$ is small. The model parameter estimation follows the MCEM algorithm [14] using this approximation technique. Algorithm 1 describes the overall idea (for detail, see [100])

4.3 LUDIA

LUDIA is a low-rank factorization algorithm using aggregate data. We first describe the underlying data model of LUDIA, then formulate LUDIA's

objective function. Because of the non-trivial aggregation constraint, we derive a customized minimization approach that uses the Sylvester equation.

4.3.1 Low-rank Data Model

LUDIA employs a bottom-up approach starting from individual level data. We first design a data model for a complete matrix $\mathbf{D} = \begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix}$, then formulate an objective function when $\mathbf{y}$ is masked and only $\mathbf{s} = \mathbf{A}\mathbf{y}$ is provided. The data model for LUDIA is based on the low-rank approximation theory as follows:

$$\begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix} = \mathbf{U}\mathbf{V}^\top + \mathbf{E} = \mathbf{U}\begin{bmatrix}\mathbf{V}_x^\top & \mathbf{v}_y^\top\end{bmatrix} + \mathbf{E} \tag{4.5}$$

where $\mathbf{U} \in \mathbb{R}^{n\times r}$, $\mathbf{V} \in \mathbb{R}^{m\times r}$, $\mathbf{E} \in \mathbb{R}^{n\times m}$, and $r \leq \min(n, m)$. Note that we divided $\mathbf{V}$ into two block matrices: $\mathbf{V}_x$ and $\mathbf{v}_y$, so that $\mathbf{X} \approx \mathbf{U}\mathbf{V}_x$ and $\mathbf{y} \approx \mathbf{U}\mathbf{v}_y$.

The main objective of our approach is to reconstruct the masked values, $\mathbf{y}$. In theory, under certain assumptions such as an underlying low-rank structure and a uniform missing mechanism, missing values in a matrix can be reconstructed. Candes and Recht [20] showed that, for matrix entries that are missing at random, they can be exactly recovered if the number of observations exceed a certain threshold value. However, the settings for the matrix completion problem are not suitable for our problem, since we consider a situation wherein one or more columns of a matrix is entirely missing, but its aggregated statistics are given.

We approximate the original matrix using two low-rank matrices. This problem is different from the matrix completion problem [71]. Low-rank approximation is typically posed as a minimization problem as follows:

$$\min \|\hat{\mathbf{D}} - \mathbf{D}\|_F^2 \quad \text{s.t.} \quad \text{rank}(\hat{\mathbf{D}}) \leq r$$

where $\mathbf{D}$ and $\hat{\mathbf{D}}$ are both $n \times m$ matrices, and $r \leq \min(n, m)$. The Eckart-Young-Mirsky theorem [47] says that rank r approximation of the data matrix $\mathbf{D}$ is given as follows:

$$\hat{\mathbf{D}} = \mathbf{U}\mathbf{\Gamma}\mathbf{V}^\top = (\mathbf{U}\mathbf{\Gamma}^{1/2})(\mathbf{\Gamma}^{1/2}\mathbf{V}^\top) = \tilde{\mathbf{U}}\tilde{\mathbf{V}}^\top$$

where $\mathbf{U}$, $\mathbf{\Gamma}$, $\mathbf{V}$ are $n \times r$, $r \times r$, $m \times r$ truncated Singular Vector Decomposition matrices, respectively. The data model in Equation 4.5 is, however, inapplicable to our reconstruction application. The model should instead reflect the constraint that $\mathbf{y}$ is masked and only $\mathbf{s} = \mathbf{A}\mathbf{y}$ is provided.

4.3.2 Aggregation Constraint

A novel optimization problem for three latent matrices $\mathbf{y}$, $\mathbf{U}$, and $\mathbf{V}$ is proposed as follows:

$$\begin{aligned} \min_{\mathbf{y},\mathbf{U},\mathbf{V}} \quad & \| \begin{bmatrix} \mathbf{X} & \mathbf{y} \end{bmatrix} - \mathbf{U}\mathbf{V}^\top \|_F^2 \\ \text{subject to} \quad & \mathbf{A}\mathbf{y} = \mathbf{s} \end{aligned} \tag{4.6}$$

A simultaneous minimization over $\mathbf{y}$, $\mathbf{U}$, and $\mathbf{V}$ is a difficult non-convex optimization problem. However, minimization over one set of variables alone is a convex problem.

We tackle this problem by removing the equality constraint. The equality constraint on $\mathbf{y}$ can be eliminated if we fix the other two variables. Given that $\mathbf{U}$ and $\mathbf{V}$ are fixed, the optimality condition [15] is given as:

$$\mathbf{A}\mathbf{y}^* = \mathbf{s} \quad \text{and} \quad \nabla f(\mathbf{y}^*) + \mathbf{A}^\top \mathbf{\Psi}^* = 0$$

where $f(\mathbf{Y}) = \|\mathbf{X} - \mathbf{U}\mathbf{V}_x^\top\|_F^2 + \|\mathbf{y} - \mathbf{U}\mathbf{v}_y^\top\|_2^2$ and $\mathbf{\Psi}^* \in \mathbb{R}^p$ is a dual variable. $\mathbf{Y}^*$ is optimal if and only if there exists $\mathbf{\Psi}^*$ satisfying the optimality conditions. It turns out that, for this system, $\mathbf{y}^*$ can be solved in a closed form.

To eliminate the constraint, we solve Karush-Kuhn-Tucker (KKT) equations as follows:

$$\nabla f(\mathbf{y}^*) + \mathbf{A}^\top \mathbf{\Psi}^* = 2\mathbf{y}^* - 2\mathbf{U}\mathbf{v}_y^\top + \mathbf{A}^\top \mathbf{\Psi}^* = 0$$

We multiply $\mathbf{A}$ on both sides of the second KKT equation, and solve for $\mathbf{\Psi}^*$:

$$2\mathbf{A}\mathbf{y}^* - 2\mathbf{A}\mathbf{U}\mathbf{v}_y^\top + \mathbf{A}\mathbf{A}^\top \mathbf{\Psi}^* = 0$$

$$\mathbf{\Psi}^* = -2(\mathbf{A}\mathbf{A}^\top)^{-1}(\mathbf{s} - \mathbf{A}\mathbf{U}\mathbf{v}_y)$$

Thus, the optimal $\mathbf{y}^*$ is:

$$\mathbf{y}^* = \mathbf{U}\mathbf{v}_y^\top + \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}(\mathbf{s} - \mathbf{A}\mathbf{U}\mathbf{v}_y^\top) \tag{4.7}$$

We plug the optimal $\mathbf{y}^*$ into the original objective function to obtain:

$$\min_{\mathbf{U},\mathbf{V}} \ \|\mathbf{X} - \mathbf{U}\mathbf{V}_x\|_F^2 + (\mathbf{s} - \mathbf{A}\mathbf{U}\mathbf{v}_y)^\top(\mathbf{A}\mathbf{A}^\top)^{-1}(\mathbf{s} - \mathbf{A}\mathbf{U}\mathbf{v}_y) \tag{4.8}$$

We have thus transformed the original objective function with three variables and an equality constraint into a simpler unconstrained objective function with two variables.

4.3.3 Objective Function

Although we simplified the constrained optimization problem to the non-constrained optimization problem in Equation 4.8, solving the objective function poses another challenge. Intuitively, one can approach the problem using an alternating minimization approach over $\mathbf{U}$ and $\mathbf{V}$. Solving for $\mathbf{U}$, however, does not have a closed form solution, because the low rank matrix $\mathbf{U}$ is surrounded by $\mathbf{A}$ and $\mathbf{v}_y$. Using a divide-and-conquer approach, we can solve for one row $\mathbf{u}_i$ of $\mathbf{U}$, and iterate over the entire rows. This divide-and-conquer approach is, however, susceptible to the sequence of rows, and cannot be generalized to an arbitrary aggregation matrix.

We propose a simple and efficient optimization solution by introducing an auxiliary variable $\mathbf{\Pi} = \mathbf{AU}$ where we treat $\mathbf{\Pi}$ as an independent variable. We also relax the hard relationship between $\mathbf{\Pi}$ and $\mathbf{U}$ as a penalty term.

Combining these two tricks, our new objective function is written as follows:

$$\min_{\mathbf{U},\mathbf{V},\mathbf{\Pi}} \|\mathbf{X} - \mathbf{U}\mathbf{V}_x^\top\|_F^2 + \|\sqrt{\mathbf{W}}(\mathbf{s} - \mathbf{\Pi}\mathbf{v}_y^\top)\|_2^2 + \|\mathbf{AU} - \mathbf{\Pi}\|_F^2 \tag{4.9}$$

where $\mathbf{W} = (\mathbf{A}\mathbf{A}^\top)^{-1}$. This objective function is LUDIA's objective function, and denote as $\mathcal{L}(\mathbf{U}, \mathbf{V}, \mathbf{\Pi})$. We now apply our alternating minimization technique to Equation 4.9.

Solving for U. First, we derive the partial derivative of the LUDIA

objective function with respect to $\mathbf{U}$:

$$\frac{\partial \mathcal{L}(\mathbf{U}, \mathbf{V}, \mathbf{\Pi})}{\partial \mathbf{U}} = -\mathbf{X}\mathbf{V}_x + \mathbf{U}\mathbf{V}_x^\top \mathbf{V}_x + \mathbf{A}^\top \mathbf{A}\mathbf{U} - \mathbf{A}^\top \mathbf{\Pi} = 0$$

Rearranging the terms, we obtain:

$$\mathbf{U}\mathbf{V}_x^\top \mathbf{V}_x + \mathbf{A}^\top \mathbf{A}\mathbf{U} = \mathbf{X}\mathbf{V}_x + \mathbf{A}^\top \mathbf{\Pi}$$

This is a type of a Sylvester equation [9]. This form of equation widely appears in the field of control theory [11], and the continuous Lyapanov equation is a special case of the Sylvester equation. If $\mathbf{V}_x^\top \mathbf{V}_x$ and $\mathbf{A}^\top \mathbf{A}$ have no common eigenvalues, a unique solution exists and it is given as:

$$\text{vec}(\mathbf{U}) = (\mathbf{V}_x^\top \mathbf{V}_x \otimes \mathbf{I}_n + \mathbf{I}_r \otimes \mathbf{A}^\top \mathbf{A})^{-1} \text{vec}(\mathbf{X}\mathbf{V}_x + \mathbf{A}^\top \mathbf{\Pi})$$

where vec is a vectorization operator, and $\otimes$ represents the Kronecker product. For example, vec($\mathbf{U}$) is defined as:

$$\text{vec}(\mathbf{U}) = \begin{bmatrix} u_{1,1} & \dots & u_{n,1} & u_{1,2} & \dots & u_{1,r}, \dots u_{n,r} \end{bmatrix}^\top$$

Solving for $\mathbf{\Pi}$. Next, we derive a partial derivative of the LUDIA objective function with respect to $\mathbf{\Pi}$:

$$\frac{\partial \mathcal{L}(\mathbf{U}, \mathbf{V}, \mathbf{\Pi})}{\partial \mathbf{\Pi}} = -\mathbf{W}(\mathbf{s} - \mathbf{\Pi}\mathbf{v}_y^\top)\mathbf{v}_y - \mathbf{A}\mathbf{U} + \mathbf{\Pi} = 0$$

Rearranging the terms, we obtain another Sylvester equation:

$$\mathbf{W}\mathbf{\Pi}\mathbf{v}_y^\top \mathbf{v}_y + \mathbf{\Pi} = \mathbf{W}\mathbf{s}\mathbf{v}_y + \mathbf{A}\mathbf{U}$$

The solution is given as:

$$\text{vec}(\mathbf{\Pi}) = (\mathbf{I}_r \otimes \mathbf{I}_p + \mathbf{v}_y^\top \mathbf{v}_y \otimes \mathbf{W})^{-1} \text{vec}(\mathbf{W}\mathbf{s}\mathbf{v}_y + \mathbf{A}\mathbf{U})$$

Solving for V. Finally, we derive closed form update equations for two block matrices $\mathbf{V}_x$ and $\mathbf{v}_y$. The partial derivative with respect to $\mathbf{V}_x$ is given as:

$$\frac{\partial \mathcal{L}(\mathbf{U}, \mathbf{V}, \mathbf{\Pi})}{\partial \mathbf{V}_x} = -\mathbf{U}^\top (\mathbf{X} - \mathbf{U}\mathbf{V}_x) = 0$$

Rearranging the terms, we obtain:

$$\mathbf{V}_x = (\mathbf{U}^\top \mathbf{U})^{-1} \mathbf{U}^\top \mathbf{X}$$

Similarly, the partial derivative with respect to $\mathbf{v}_y$ is:

$$\frac{\partial \mathcal{L}(\mathbf{U}, \mathbf{V}, \mathbf{\Pi})}{\partial \mathbf{v}_y} = -\mathbf{\Pi}^\top \mathbf{W} (\mathbf{s} - \mathbf{\Pi}\mathbf{v}_y^\top) = 0$$

Thus, the update form is:

$$\mathbf{v}_y^\top = (\mathbf{\Pi}^\top \mathbf{W} \mathbf{\Pi})^{-1} \mathbf{\Pi}^\top \mathbf{W}\mathbf{s}$$

4.3.4 Algorithm

Algorithm 2 summarizes our alternating minimization approach combining three different minimization equations for $\mathbf{U}$, $\mathbf{\Pi}$, and $\mathbf{V}$. The algorithm takes three input matrices: individual-level matrix $\mathbf{X}$, aggregation matrix $\mathbf{A}$, and aggregate-level matrix $\mathbf{s}$. The output of the algorithm is the reconstructed individual-level data $\hat{\mathbf{y}}$. The algorithm does not require any other parameters.

Algorithm 2: LUDIA Estimation Algorithm

Data: $\mathbf{X}, \mathbf{A}, \mathbf{s}$
Result: $\hat{\mathbf{y}}$
$r = \text{rank}(\mathbf{X}))$;
$\bar{\mathbf{y}} = \mathbf{A}^{+}\mathbf{s}$;
$\mathbf{U}, \mathbf{V} = \text{SVD}(\begin{bmatrix}\mathbf{X} & \bar{\mathbf{y}}\end{bmatrix}, \text{rank} = r)$;
$\mathbf{\Pi} = \mathbf{AU}$;
while *not converged* **do**
  $\text{vec}(\mathbf{U}) = (\mathbf{V}_x^{\top}\mathbf{V}_x \otimes \mathbf{I}_n + \mathbf{I}_r \otimes \mathbf{A}^{\top}\mathbf{A})^{-1}\text{vec}(\mathbf{X}\mathbf{V}_x + \mathbf{A}^{\top}\mathbf{\Pi})$;
  $\text{vec}(\mathbf{\Pi}) = (\mathbf{I}_r \otimes \mathbf{I}_p + \mathbf{v}_y^{\top}\mathbf{v}_y \otimes \mathbf{W})^{-1}\text{vec}(\mathbf{W}\mathbf{s}\mathbf{v}_y + \mathbf{AU})$;
  $\mathbf{V}_x = (\mathbf{U}^{\top}\mathbf{U})^{-1}\mathbf{U}^{\top}\mathbf{X}$;
  $\mathbf{v}_y^{\top} = (\mathbf{\Pi}^{\top}\mathbf{A}\mathbf{A}^{\top}\mathbf{\Pi})^{-1}\mathbf{\Pi}^{\top}\mathbf{A}\mathbf{A}^{\top}\mathbf{s}$;
end
$\hat{\mathbf{y}} = \mathbf{U}\mathbf{v}_y^{\top}$;
// correction equation;
$\hat{\mathbf{y}} = \hat{\mathbf{y}} + \mathbf{A}^{+}(\mathbf{s} - \mathbf{A}\hat{\mathbf{y}})$

The initialization of $\mathbf{U}$ and $\mathbf{V}$ is based on the MP solution. We first pseudo-reconstruct the masked individual-level data using MP, then run SVD on the pseudo-complete matrix. The rank parameter of the SVD algorithm is given as the rank of $\mathbf{X}$. This setting captures both our low-rank data model and a linear model defined as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}$. If this linear model is the true underlying data model for the data, then the rank of the complete matrix is the same as the rank of $\mathbf{X}$.

The last line of the algorithm calibrates the final output. Recall that the optimal $\mathbf{y}^*$ was given in Equation 4.7. This correction equation ensures that the aggregation of the reconstructed values are the same as the given aggregate data i.e. $\mathbf{s} = \mathbf{A}\hat{\mathbf{y}}$. However, if the aggregate values do not necessarily need to match the reconstructed values (possibly from noise or sub-sampling), we can

ignore the last line of the algorithm.

4.4 Extensions

We illustrate two extensions of the LUDIA algorithm. The first extension shows that LUDIA can directly incorporate multi-level data models. This extended reconstruction method can capture group-level effects, which were not possible in classical frameworks. The second extension explores whether we can improve the reconstruction quality if we have multiple levels of aggregate data.

4.4.1 Multi-level Modeling

The ecological fallacy problem is essentially *statistical under-identification* [115]. For aggregate data analyses, the maximum degrees of freedom are limited by the number of partitions. Individual-level analyses, such as multi-level models [62], often require more parameters than the number of partitions. This under-identification problem is traditionally approached by more assumptions; Goodman's and King's assumptions are two extreme cases. These assumptions are usually unrealistic, and they are almost impossible to verify on the basis of aggregate data.

Smartly utilizing auxiliary individual-level data can provide higher degrees of freedom than the number of partitions. The key observation comes from the connection between the degrees of freedom and the rank of a full matrix. Suppose that a target $\mathbf{y}$ is a function of r degrees of freedom. Then

the rank of the full matrix $\begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix}$ is r, since $\mathbf{y}$ can be expressed by a linear combination of $\mathbf{X}$. Analogously, if a target is a multi-level function of r variables and p levels, then the degrees of freedom for this model is given by $(r \times p)$. To capture the variability of the target, the corresponding full matrix needs to have the rank of $(r \times p)$. In this section, we show that this rank augmentation can be seamlessly integrated with the LUDIA framework.

As illustrated in Equation 4.1, a multi-level model can be compactly written as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{G}\boldsymbol{\gamma} + \mathbf{E} \approx \begin{bmatrix}\mathbf{X} & \mathbf{G}\end{bmatrix} \begin{bmatrix}\boldsymbol{\beta} \\ \boldsymbol{\gamma}\end{bmatrix}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^{l \times 1}$ is a random effect vector, and $\mathbf{G} \in \mathbb{R}^{n \times l}$ represents encoded covariates according to $\boldsymbol{\gamma}$. For this model, the degrees of freedom are given as $(r + l)$ where $r = \text{rank}(\mathbf{X})$. The full matrix has $(r + l + 1)$ columns, and this matrix can be written as a product of two rank $(r + l)$ matrices.

To fully reconstruct the masked individual-level data, the rank of our low-rank model should be at least $(r+l)$. This can be achieved by augmenting the data by l:

$$\begin{bmatrix}\mathbf{X} & \mathbf{G} & \mathbf{y}\end{bmatrix} \approx \begin{bmatrix}\mathbf{U} & \tilde{\mathbf{U}}\end{bmatrix} \begin{bmatrix}\mathbf{V}_x^\top & \tilde{\mathbf{V}}_{x1} & \mathbf{v}_y^\top \\ \tilde{\mathbf{V}}_{x2} & \tilde{\mathbf{V}}_{x3} & \mathbf{v}_a^\top\end{bmatrix}$$

where $\tilde{\mathbf{U}} \in \mathbb{R}^{n \times l}$, $\tilde{\mathbf{V}}_{x\cdot} \in \mathbb{R}^{l \times l}$, and $\mathbf{v}_a \in \mathbb{R}^{1 \times l}$.

Although one can run LUDIA with these augmented terms, we show that a simple post-processing approach can mimic the result from this augmentation. The block matrix $\tilde{\mathbf{V}}_x$ can be treated as a nuance parameter, since

it does not directly affect the reconstruction of $\mathbf{y}$. The trick is to specify our low-rank matrices to be of a specific form as follows:

$$\begin{bmatrix}\mathbf{X} & \mathbf{G} & \mathbf{y}\end{bmatrix} \approx \begin{bmatrix}\mathbf{U} & \mathbf{G}\end{bmatrix} \begin{bmatrix}\mathbf{V}_x^\top & \mathbf{0} & \mathbf{v}_y^\top \\ \mathbf{0} & \mathbf{I}_l & \mathbf{v}_a^\top\end{bmatrix}$$

Then we do not need to estimate $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}_x$, but only $\mathbf{v}_a$. The augmented term $\mathbf{v}_a$ needs to minimize the second term of Equation 4.9:

$$\min_{\mathbf{v}_a} \|\sqrt{\mathbf{W}}(\mathbf{s} - \mathbf{A}\begin{bmatrix}\mathbf{U} & \mathbf{G}\end{bmatrix}\begin{bmatrix}\mathbf{v}_y^\top \\ \mathbf{v}_a^\top\end{bmatrix})\|_2^2$$

The solution for this minimization problem is given as follows:

$$\hat{\mathbf{v}}_a^\top = ((\mathbf{AG})^\top \mathbf{WAG})^{-1}(\mathbf{AG})^\top \mathbf{W}(\mathbf{s} - \mathbf{AU}\mathbf{v}_y^\top)$$

Using this $\mathbf{v}_a$, we calibrate the reconstruction of $\mathbf{y}$:

$$\hat{\mathbf{y}} = \mathbf{U}\mathbf{v}_y^\top + \mathbf{G}\hat{\mathbf{v}}_a^\top$$

This adjustment equation mimics the original augmentation.

This data augmentation technique for multi-level modeling is not suitable for the MP and ER frameworks. MP only focuses on the aggregation matrix, and does not involve individual-level covariates. Adding the augmented block matrix $\mathbf{G}$ requires a different approach. The number of covariates in ER is upper-bounded by the number of partitions. The simplest multi-level model, a random intercept model, requires the number of covariates to be the same as the number of partitions. LUDIA utilizes the full potential of individual-level covariates, and thus it can be easily extended to more complex models.

4.4.2 Aggregation Stacking

Thus far, we have considered only one source of aggregate data. There can be many levels of groupings based on geography, administration, or other factors. This section answers how one can further improve the reconstruction quality with additional aggregate data.

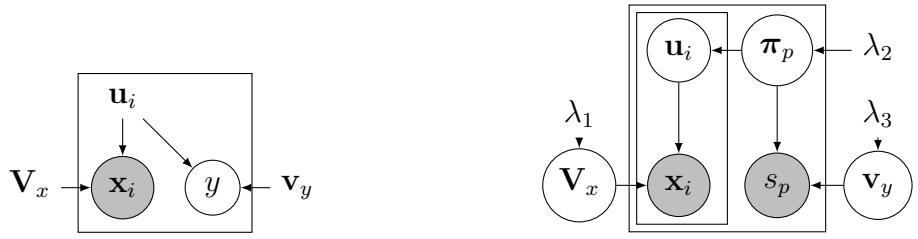
The key trick is to stack two aggregate-level datasets and create a new aggregate dataset. Algorithm 3 illustrates this approach. In the algorithm, we have two sources of aggregate data: $(\mathbf{A}_1, \mathbf{s}_1)$ and $(\mathbf{A}_2, \mathbf{s}_2)$. For example, there can be county-level and state-level aggregate data, respectively. This kind of augmentation can further improve the reconstruction accuracy. This is because we have more constraints on $\mathbf{y}$, and the degrees of freedom for $\mathbf{y}$ decrease accordingly.

Algorithm 3: LUDIA with Aggregation Stacking

Data: $\mathbf{X}, \mathbf{A}_1, \mathbf{s}_1, \mathbf{A}_2, \mathbf{s}_2$
Result: $\hat{\mathbf{y}}$
$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}$ and $\mathbf{s} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}$;
$\hat{\mathbf{y}} = \text{LUDIA}(\mathbf{X}, \mathbf{A}, \mathbf{s})$;

4.4.3 Probabilistic Interpretation

This section presents a probabilistic interpretation of the proposed LUDIA objective function. Figure 4.3a shows our low-rank model for the complete data. Note that the node for y is not shaded, since the variable is masked. To incorporate the aggregation constraint, we draw another plate that represents



(a) Low-rank model for complete data.

(b) Low-rank model with the aggregation constraint.

Figure 4.3: Probabilistic Models for LUDIA.

groupings. Figure 4.3b illustrates the graphical model for LUDIA. Each $\mathbf{u}_i$ in a group is assumed to be drawn from a multivariate Gaussian centered at $\boldsymbol{\pi}_p$. Thus, the log-likelihood $\log p(\mathbf{U}, \mathbf{V}, \boldsymbol{\Pi} \mid \mathbf{X}, \mathbf{s})$ of LUDIA is written as:

$$
\begin{aligned}
&- (\mathbf{X} - \mathbf{U}\mathbf{V}_x^\top)^\top \boldsymbol{\Sigma}_x^{-1} (\mathbf{X} - \mathbf{U}\mathbf{V}_x^\top) \\
&- (\mathbf{s} - \boldsymbol{\Pi}\mathbf{v}_y^\top)^\top \boldsymbol{\Sigma}_y^{-1} (\mathbf{s} - \boldsymbol{\Pi}\mathbf{v}_y^\top) \\
&- (\boldsymbol{\Pi} - \mathbf{A}\mathbf{U})^\top \boldsymbol{\Sigma}_\pi^{-1} (\boldsymbol{\Pi} - \mathbf{A}\mathbf{U}) + \text{const.}
\end{aligned}
$$

In our setting, each row of $\mathbf{X}$ is i.i.d., thus $\boldsymbol{\Sigma}_x$ can be modeled as an identity matrix $\mathbf{I}_n$. Before characterizing $\boldsymbol{\Sigma}_y$, we first show that $\mathbf{A}\mathbf{A}^\top$ is invertible and positive-semidefinite. This property can be shown from the fact that $\text{rank}(\mathbf{A}) = p$ and $\mathbf{A}\mathbf{A}^\top \in \mathbb{R}^{p \times p}$. Moreover, the (p, p)th diagonal component of $(\mathbf{A}\mathbf{A}^\top)^{-1}$ is the same as n_p, the number of data points in group p. Thus, $\mathbf{A}\mathbf{A}^\top$ can replace $\boldsymbol{\Sigma}_y$. Finally, if we assume that $\boldsymbol{\Sigma}_\pi = \mathbf{I}_p$, then this log-likehood is actually a negative of the LUDIA objective function.

To show the connection to CUDIA, let us assume that we restrict the

shape of $\mathbf{U}$ to be as follows:

$$\mathbf{U}_C \quad \text{s.t.} \quad u_{ij} \in \{0, 1\} \quad \text{and} \quad \sum_j u_{ij} = 1$$

In other words, each column of $\mathbf{U}$ becomes an indicator column for clusters. The rank parameter r of LUDIA is now interpreted as r different clusters, and $\mathbf{V}$ represents cluster centers. If we plug in this constraint to the LUDIA's log-likelihood function, we obtain the log-likelihood of CUDIA. Although this formulation may provide a different perspective on combining multiple sources of data, the minimization of the CUDIA objective function is more complicated to solve because of the non-negative constraint. Thus, CUDIA requires a computationally heavy MCEM algorithm, or greedy deterministic algorithm [100]. As the non-negative case is a special case of $\mathbf{U}$, we also have:

$$\|\mathbf{D} - \mathbf{U}\mathbf{V}^\top\|_F^2 \leq \|\mathbf{D} - \mathbf{U}_C\mathbf{V}^\top\|_F^2$$

This is why the CUDIA imputation is not so suitable for complex modeling such as multi-level modeling and non-linear estimates, while the LUDIA reconstruction provides valid inferences in such situations (see Section 4.5).

4.5 Empirical Study

We provide experimental results using simulated data and Texas Inpatient Discharge data. A simulated dataset is used to illustrate the differences between ER, MP, and LUDIA. Next, we illustrate reconstruction tasks using actual health data. In this set of experiments, we mask sensitive columns, then

show how well LUDIA can reconstruct the masked original values for different analytical tasks including non-linear estimates and multi-level modeling.

4.5.1 Simulated Data

We generate four different simulated datasets as follows:

- Low-Rank (LR) model emulates the model assumption of LUDIA. The parameters are given as $r = 2$ and $m = 4$. The equation for simulated data is as follows:

$$\begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix} = \mathbf{U}\mathbf{V}^\top + \mathbf{E}$$

  where $\mathbf{U}$ and $\mathbf{V}$ are drawn from the standard normal distribution, and the noise matrix $\mathbf{E}$ is drawn from a normal distribution with 0.4 standard deviation.

- Fixed Effect (FE) model emulates the model assumption of ER. We generate individual-level matrices with $m = 2$ from the standard normal distribution. The model equation is:

$$\mathbf{y} = \mathbf{c} + \mathbf{X}\boldsymbol{\beta} + \mathbf{E}$$

  where $\mathbf{E}$ is drawn from a normal distribution with 0.2 standard deviation.

- Random Intercept (RE1) and Random Slope (RE2) model check whether the LUDIA's multi-level argument is valid. The model equation is:

$$\mathbf{y} = \mathbf{c} + \mathbf{X}\boldsymbol{\beta} + \mathbf{G}\boldsymbol{\gamma} + \mathbf{E}$$

  where $\boldsymbol{\gamma}$ is drawn from a normal distribution with 0.2 standard deviation.

We fix the number of partitions to be five, and vary the number of total data points. Aggregation matrices are generated using random assignment of partitions.

Figure 4.4 shows the reconstruction errors for different simulated data and reconstruction methods. Each cell represents a different simulated dataset, and the horizontal axes represent the number of data points per partition. The lower the curve is, the better the reconstruction quality is. MP is not affected by the number of data points per partition, but its performance is the worst from the experiments. The performance of ER is comparable with that of LUDIA for the FE dataset, but it does not capture the low-rank structure and random effects. For the random effect datasets, ER is largely affected by the number of data points per partition. LUDIA shows robust and stable performances over different datasets.

Figure 4.5 shows the reconstructed values compared to the original values from the RE1 dataset. The leftmost first two cells show the reconstructed values from MP and ER, respectively. In this figure, we show three different initialization methods for LUDIA: MP, ER, and random initialization methods. The alternating minimization approach of LUDIA does not guarantee the convergence to the global optimum, and the algorithm is susceptible to initial points. All three initialization methods provide comparable performances, and it would be worthwhile to investigate the better choice of initialization methods. The rest of the experiments use the MP initialization to maintain the consistency of our algorithm.

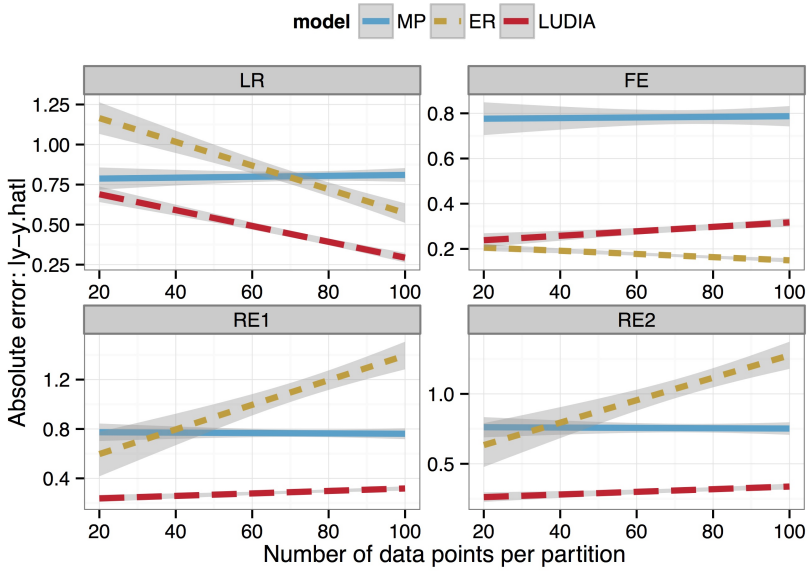


Figure 4.4: Reconstruction error vs. number of data points per partition. Except the FE case, the LUDIA reconstruction shows the least absolute errors.

4.5.2 Texas Inpatient Data

We use Texas Inpatient Public Use Data File [131] from the Texas Department of State Health Services (DSHS). Hospital billing records collected from 1999 to 2007 are publicly available through their website. Each yearly dataset contains about 2.8 millions events with more than 250 features including hospital name, county, patient ZIP codes, etc. Specifically, we use the inpatient records from Central Texas in the fourth quarter of 2006. Except for a few exempt hospitals, all the hospitals in Texas reported inpatient discharge events to DSHS. The public use data file we use is a subset of the DSHS's hospital discharge database. Our primary interest is the hospital charge for

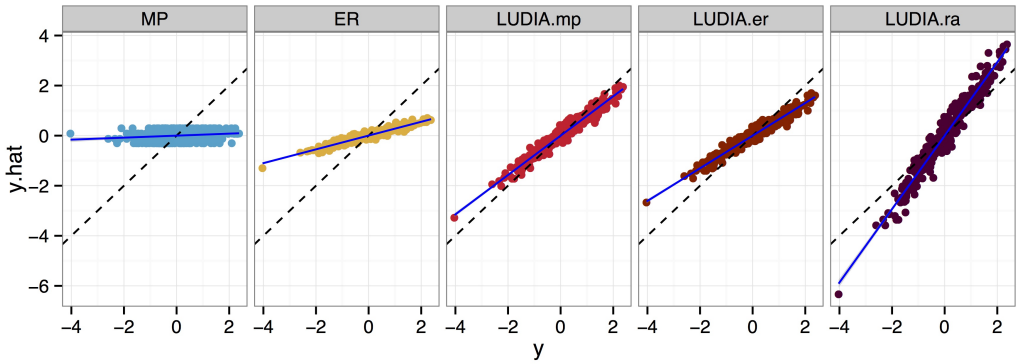


Figure 4.5: Reconstructed vs. original. We show three different initializations for LUDIA: MP, ER, and random initializations. All these three LUDIA reconstructions are closer to the original values, and the MP initialization performs the best.

normal delivery. We aggregate the individual-level hospital charges at county-, hospital-, and ZIP code-levels. We assume that some of the individual-level covariates are available such race, specialty unit, length of stay variables.

Hospital charge is primarily a function of length of stay, but it is substantially different across regions and is also affected by many other factors:

$$\text{HC} = \beta_{\text{hospital}} \text{LoS}^{\alpha} + \text{unit} + \text{severity} + \ldots + \text{error}$$

where HC and LoS represent Hospital Charge and Length of Stay, respectively. Note that the coefficient for LoS is indexed by hospital, since daily charge rate is a function of hospital. The distribution of HC is, in fact, similar to a log-normal distribution. It is a better practice to log-transform the data, before applying a linear model:

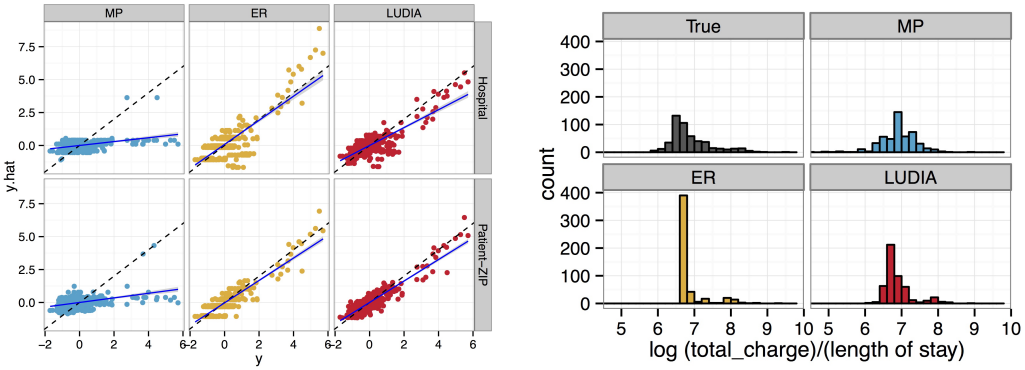
$$\log \text{HC} = \log \beta_{\text{hospital}} + \alpha \log \text{LoS} + \ldots + \text{Error}'$$

Table 4.3: Reconstruction Accuracy of the Texas dataset

Level	Model	MAE	MSE
County	MR	0.648 (± 0.75)	0.976 (± 3.28)
	ER	**0.466** (± 0.45)	**0.422** (± 0.87)
	LUDIA	0.514 (± 0.48)	0.497 (± 1.14)
Hospital	MR	0.609 (± 0.69)	0.851 (± 2.92)
	ER	0.513 (± 0.49)	0.501 (± 1.10)
	LUDIA	**0.435** (± 0.40)	**0.348** (± 0.68)
Patient ZIP	MR	0.589 (± 0.69)	0.824 (± 2.92)
	ER	0.319 (± 0.28)	0.184 (± 0.38)
	LUDIA	**0.289** (± 0.26)	**0.152** (± 0.34)

This log-transformed linear model turns out to be a simple random intercept model.

Table 4.3 shows the reconstruction errors from three different levels of aggregation. Except for the county-level case, the LUDIA-reconstructed values are the closest to the original values with smallest variances. ER performs slightly better than LUDIA for the county-level aggregate data. This is because the multi-level effects at county-level are not distinctive enough i.e. the constancy assumption can be applied. Figure 4.6a illustrates the reconstructed values compared to the original values. If reconstruction is perfect, points should lie on the dotted diagonal lines. As can be seen, the MP reconstructions do not capture the tails. This is because, when the HC values are averaged, those tail values are typically cancelled out, and MP cannot infer beyond the provided average statistics. The ER reconstructions perform reasonably well, but does not capture the multi-level bias. LUDIA provides better estimates for the original values in terms of Mean Absolute Error (MAE).



(a) (b)

Figure 4.6: (a) Reconstructed vs. original for the 3 models. (b) Estimated histograms of daily hospital charges. LUDIA histogram is the closest to the original.

The advantages of LUDIA are even more highlighted when calculating non-linear estimates. As an illustrative example, suppose that we want to estimate average daily charges. To calculate this value, we first need to reconstruct individual-level hospital charges, and then divide the reconstructed charges by the individual-level length of stay variable. In other words, average daily charges are calculated as follows:

$$\text{Average Daily Charge} = \frac{1}{n}\sum_i \frac{\hat{\text{HC}}_i}{\text{LoS}_i}$$

Figure 4.6b show the histograms of the estimated average daily charges. As can be seen, the histogram from LUDIA captures the asymmetrical shape of the original histogram.

As shown in Section 4.4, multi-level modeling can be directly integrated with LUDIA. We extract rural counties of Central Texas, and compare the

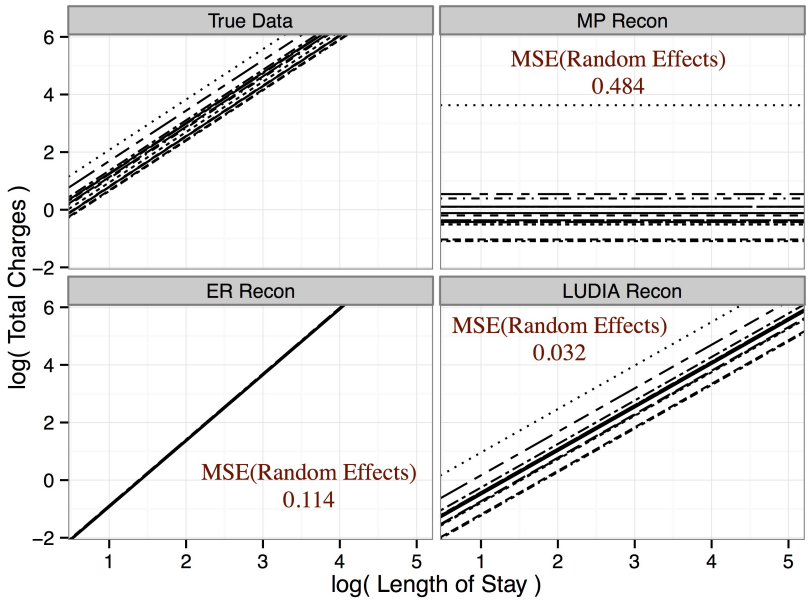


Figure 4.7: Multi-level modeling and the mean squared errors, shown as "MSE(Random Effects)", between the original random effects and estimated random effects. LUDIA's random effects are almost the same as the original.

hospital charges by applying a random intercept model. Figure 4.7 shows the fitted lines from the multi-level models. As can be seen, the original data clearly show the random intercept terms. It was impossible to estimate the slope term from the MP reconstructed values. For the ER reconstructed values, although the global model was similar to the original data, we cannot visually check the random intercepts. This is because ER ignores the information from the aggregation matrix. On the other hand, LUDIA provides almost the exact same random effect coefficients.

Reconstructed values from aggregate data can be used in various data mining applications. We show a simple predictive analysis when a target

column is provided in an aggregate form. By reconstructing the individual-level target values from the aggregate data, we can train a model, and then apply the model to test data as follows:

1. Combine the aggregate and individual-level data, then reconstruct the masked column
2. Train a predictive model using the pseudo complete data
3. For new data points, predict the target values using the trained model

We first divided the Texas inpatient dataset into a training (80%) and a hold-out (test, 20%) set. Assuming the total charges (target) are provided in only an aggregate form, we reconstruct the target using three different algorithms. We trained a Lasso regression model, and then measured the predictive accuracies of the target. Figure 4.8 shows the results from the test set. As can be seen, the LUDIA-reconstructed training dataset provides the best Lasso model in terms of MAEs. In this example, we included the performance of a model that is trained on CUDIA-reconstructed data. The CUDIA-reconstructed dataset provides better predictive accuracies than the MP- and ER-reconstructed training datasets. However, CUDIA is still a clustering algorithm, and the reconstruction from CUDIA is based on estimated cluster centers. Although CUDIA provides homogenous cluster centers, it does not generate fine-grained reconstruction like LUDIA. The predictive Lasso model trained on the CUDIA-reconstructed dataset exhibits higher MAE and variances than the model trained on the LUDIA-reconstructed dataset.

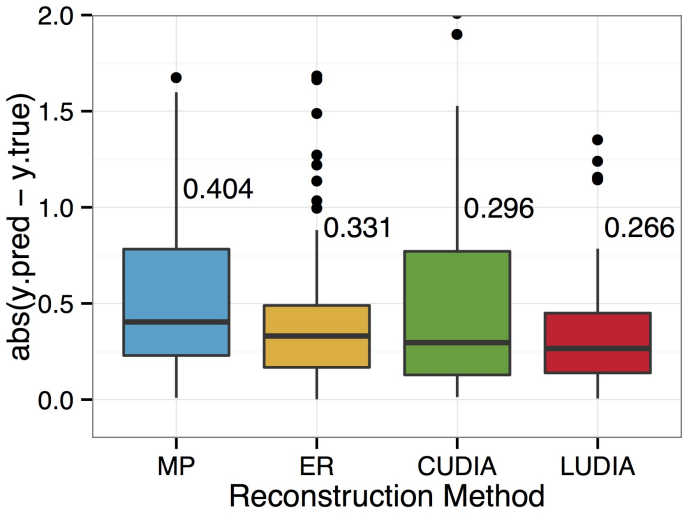


Figure 4.8: Predictive performance of the Lasso (`glmnet`) models trained on the reconstructed data. Absolute errors are measured using a hold-out dataset.

4.6 Summary

The implication of LUDIA can be viewed from two perspectives.

Utility perspective. Our method allows aggregated data to be effectively utilized in individual-level inferential tasks. This is particularly important since standard imputation techniques do not make use of the summary statistics provided by aggregated data that are widely available for social good. Many machine learning algorithms that require completely observed data can now be directly applied to the LUDIA-reconstructed data.

Privacy perspective. Although the reconstructed values are not guaranteed to be identical to the true values, it is clear that the estimated values are correlated with the actual values. If additional theoretical guar-

antees are developed, data aggregation may be no longer perfectly safe from privacy attacks. With enough auxiliary information, it is possible that private information gets revealed using techniques similar to LUDIA. This implies that, in the future, reconstruction performance will need to be considered prior to data aggregation, to guarantee that privacy requirements are met.

The proposed LUDIA framework can be extended to more complex data models. It is also worthwhile to investigate more efficient solutions for the Sylvester equation. One can also explore theoretical reconstruction guarantees that depend on the characteristics of the datasets and of the aggregation matrices.

Chapter 5

Privacy-aware Decision Tree API

Open API (Application Programming Interface) is a set of end point protocols between websites. The technology has been rapidly adopted in Web 2.0 applications; in 2011, Twitter Inc. processed 13 billion API calls per day on average [43]. Through APIs, developers can quickly build websites that interact with external computational servers and databases. In recent years, several government agencies, such as the United States Census Bureau[1] and the Bureau of Labor Statistics[2], have opened access to their databases through APIs. Healthdata.gov[3] and Data.gov[4] also have released APIs that allow users to access their healthcare datasets.

In this chapter, we are primarily interested in implementing data-mining APIs that access healthcare databases. Such APIs can facilitate and catalyze developments of mobile and web applications for patient engagement and compliance. As an illustrative example, consider two APIs that provide risk scores for diabetes and septic shock:

[1] `http://www.census.gov/developers/`
[2] `http://www.bls.gov/developers/`
[3] `http://healthdata.gov/data-api`
[4] `https://www.data.gov/developers/apis`

- **Diabetic Risk Score API**

```
POST /diabeteScore
  Request
    Content-type: application/json
    { "glucoseLevel": "130", "bodyMassIndex": "25", ... }
  Response
    Content-type: application/json
     { "riskScore": "80%" }
```

- **Septic Shock Risk Score API**

```
POST /septicShockScore
  Request
    Content-type: application/json
    { "systolicBP": "105", "heartRate": "61", ... }
  Response
    Content-type: application/json
     { "riskScore": "60%" }
```

where users send their health information, and receive risk scores using the RESTful (Representational State Transfer) framework [49]. Figure 5.1 conceptually illustrates our API call framework.

Data access scenarios through APIs are substantially different from traditional publication and utilization scenarios, so are privacy breaches. Although individual records are not directly revealed, an attacker can combine

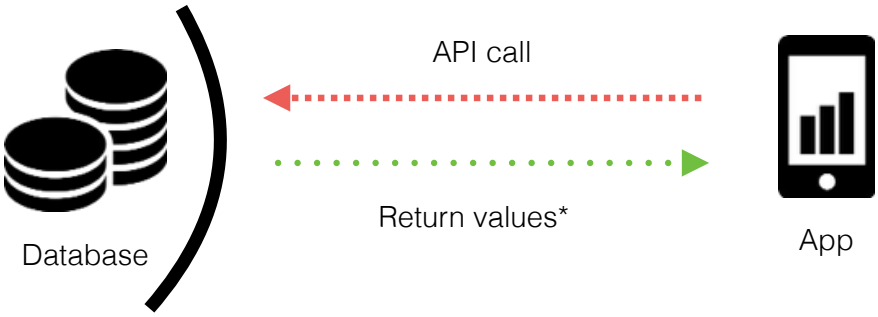


Figure 5.1: Privacy-preserving API call.

a certain set of API calls, and identify the structures, algorithms, parameters, and sometimes individual records of external databases. Calandrino et al. [19] showed that passive observations of Amazon.com's collaborative filtering outputs, which can be viewed as a data-mining API, can reveal customers' transaction records. By observing temporal changes of recommendation outputs, the authors designed an algorithm that achieves 80% accuracy on estimating transactions for a subset of customers. Privacy breaches from APIs are not limited to collaborative filtering applications. Narayanan and Shmatikov [97] showed that topological information, which can be obtained through Facebook and Twitter APIs, can be used to de-anonymize social network data. Preventing privacy breaches from data APIs is critical as the web is becoming more social and personalized.

Even our straightforward risk score APIs can be vulnerable from privacy attacks:

Case Study 5.0.1. Alice was showing Bob her new diabetic risk score mobile application, and Bob happened to see her diabetic risk score by chance. Bob knows Alice's basic health information such as age and gender, but he does not know whether she has other chronic conditions e.g. hypertension. He downloads the same mobile application, and simulates chronic conditions until he gets exactly the same score that he saw from Alice's phone. *Now, Bob knows which other chronic conditions Alice has.*

Case Study 5.0.2. This time, Bob discovered that the training dataset of the risk score application is, in fact, derived from his hospital. He also found out that the app is using a decision tree as its risk score algorithm. Bob knows that Dave is the only one who has similar demographic and health conditions in the hospital. Bob wants to know whether Dave has diabetes, and finds that the structures and estimates of decision trees change depending on Dave's diabetic conditions $\{\text{True}, \text{False}\}$. *After testing some simulated inputs, Bob now knows that Dave has diabetes.*

These two cases are only a subset of potential privacy attack cases. In practice, privacy attacks can accompany more sophisticated and malicious techniques such as eavesdropping and wire-tapping. Therefore, the API outputs and internal algorithms should be properly randomized to alleviate the risks for privacy breaches.

To address the privacy breaches in data-mining APIs, this chapter proposes differentially private decision tree, Differentially Private α-Tree (DPaT).

In the proposed framework, each API call costs ϵ in the differential privacy regime. Note that, in practice, ϵ can be associated with a dollar amount in proprietary API engines e.g. Google Map charges \$0.5 per 1000 map loadings. DPaT is a two stage algorithm:

1. **Model-layer:** We construct a 0-differentially private tree structure i.e. a decision tree structure that does not change regardless of inclusion or deletion of any record in the training dataset.

2. **Output-layer:** We calculate the sensitivity of the output, and add Laplace noise to achieve ϵ-differential privacy.

The performance of DPaT is, of course, inversely correlated with the level of privacy protection; higher levels of privacy guarantees result in lower accuracies of the results. This Pareto-frontier can be improved through a classifier ensemble technique. We show that an ensemble of DPaTs can enhance the risk score performance at a given privacy level. Our empirical study using Pima Indian and MIMIC-II datasets shows that accurate risk score APIs can be implemented without sacrificing data privacy.

In this chapter, $\mathcal{D}$ represents a dataset that consists of feature variables $\{X_i\}_1^M$ and class label Y. We focus on binary classification problems, thus $Y \in \{0, 1\}$. Without loss of generality, we assume that X_i is a binary feature. Such binary features can be obtained by 1) dummy-coding for categorical variables, and 2) thresholding for numerical variables. For example, for a numeric variable $Z \in \mathbb{R}$, we set up K number of thresholds $(t_1, t_2, \ldots, t_K)$,

and then create $K+1$ number of binary variables by $I(t_{k-1} < Z \leq t_k)$ where $k \in 1 : K+1$, $t_0 = -\infty$, and $t_{K+1} = \infty$. The binary feature variables X in this chapter is obtained from the original feature variable Z that can be multi-category and numeric variables i.e. $X = \text{Binarize}(Z)$ where Z refers to the original feature variable. Since all the features are binary variables, selecting a splitting "variable" is equivalent to partitioning the original dataset into two groups, $\{(X, Y) \mid X_{i^*} = 0\}$ (left) and $\{(X, Y) \mid X_{i^*} = 1\}$ (right).

5.1 Preliminaries

Decision trees are rule-based classification algorithms that can be obtained through:

1. selecting a splitting feature based on a certain criterion,
2. partitioning input data based on the selected splitting feature, and then
3. recursively repeating this process until certain stopping criteria are met.

Decision trees use different splitting criteria, e.g. C4.5 and ID3 use Information Gain and Information Gain Ratio [107], CHAID uses the Chi-squared test, and CART [16] uses the Gini impurity measure. There are numerous other impurity measures such as misclassification rate and Hellinger distance. It is generally believed that no single splitting criterion is guaranteed to outperform over the other criteria [38, p. 161].

A divergence, $D_\alpha(\mathrm{P}\|\mathrm{Q})$, is a function that measures the distance between two distributions: P and Q. There exists many different kinds of divergences such as f-divergence [29] and β- and γ-divergences [27]. α-Tree uses α-divergence [70, 141], defined as follows:

$$D_\alpha(\mathrm{P}\|\mathrm{Q}) = \frac{\int_x \alpha\mathrm{P}(x) + (1-\alpha)\mathrm{Q}(x) - \mathrm{P}(x)^\alpha \mathrm{Q}(x)^{1-\alpha} dx}{\alpha(1-\alpha)} \tag{5.1}$$

where P and Q are two probability distributions, and α is a real number. The α-divergence was introduced by Chernoff [26] to upper-bound the theoretical error probability of classification tasks. The mathematical form of α-divergence is closely related to those of Renyi entropy [114], Tsallis entropy [132], and generalized diversity index [72]; all four share the exponent term α.

If both P and Q are proper probability density functions (i.e. $\int_x \mathrm{P}(x)dx = \int_x \mathrm{Q}(x)dx = 1$), then Equation (5.1) simplifies to:

$$D_\alpha(\mathrm{P}\|\mathrm{Q}) = \frac{1 - \int_x \mathrm{P}(x)^\alpha \mathrm{Q}(x)^{1-\alpha} dx}{\alpha(1-\alpha)}. \tag{5.2}$$

Some special cases are:

$$D_{-1}(\mathrm{P}\|\mathrm{Q}) = \frac{1}{2}\int_x \frac{(\mathrm{Q}(x) - \mathrm{P}(x))^2}{\mathrm{P}(x)} dx \tag{5.3}$$

$$\lim_{\alpha\to 0} D_\alpha(\mathrm{P}\|\mathrm{Q}) = KL(\mathrm{Q}\|\mathrm{P}) \tag{5.4}$$

$$D_{\frac{1}{2}}(\mathrm{P}\|\mathrm{Q}) = 2\int_x (\sqrt{\mathrm{P}(x)} - \sqrt{\mathrm{Q}(x)})^2 dx \tag{5.5}$$

$$\lim_{\alpha\to 1} D_\alpha(\mathrm{P}\|\mathrm{Q}) = KL(\mathrm{P}\|\mathrm{Q}) \tag{5.6}$$

$$D_2(\mathrm{P}\|\mathrm{Q}) = \frac{1}{2}\int_x \frac{(\mathrm{P}(x) - \mathrm{Q}(x))^2}{\mathrm{Q}(x)} dx \tag{5.7}$$

Equation (5.5) is Hellinger distance, and Equations (5.4) and (5.6) are KL-divergences. Note that α-Divergence is always positive and is zero if and only if $\mathrm{P} = \mathrm{Q}$. Hence, α-divergence can be used as a (dis)similarity measure between two distributions.

α-Tree is a generalization of several decision trees, such as C4.5 and CART. The impurity reduction criterion in C4.5 can be written as a divergence maximization criterion as follows:

$$\begin{aligned}
&\min \quad \sum_{x} \mathrm{P}(x) H(Y \mid x) \\
&= \max \quad H(Y) - H(Y \mid X) \\
&= \max \quad \sum_{x,y} \mathrm{P}(x,y) \log \frac{\mathrm{P}(x,y)}{\mathrm{P}(x)\mathrm{P}(y)} \\
&= \max \quad KL(\mathrm{P}(X,Y) \| \mathrm{P}(X)\mathrm{P}(Y)) \\
&= \max \quad \lim_{\alpha \to 1} D_{\alpha}(\mathrm{P}(X,Y) \| \mathrm{P}(X)\mathrm{P}(Y))
\end{aligned}$$

Replacing the KL divergence with the α-divergence yields the α-Tree algorithm, outlined in Algorithm 4. Thus, the **$\boldsymbol{\alpha}$-divergence criterion** selects a splitting feature which gives the maximum α-divergence between $\mathrm{P}(X,Y)$ and $\mathrm{P}(X)\mathrm{P}(Y)$. The C4.5 splitting criterion can be obtained using $\alpha = 1$.

The α value determines the selection of splitting features, and different values can yield distinct splitting features. This property has been used to increase the diversity of base trees in an ensemble framework [101].

Algorithm 4: α-Tree

Data: $\mathcal{S} = \{(X, Y)\}, \alpha$
Result: $\mathcal{T}$
$X_{i^*} = \arg\max_{X_i} D_\alpha(\mathrm{P}(X_i, Y) \| \mathrm{P}(X_i)\mathrm{P}(Y))$;
if *stopping_criteria($\mathcal{S}$)=True* **then**
  $E[Y \mid \mathcal{S}]$;
else
  $\mathcal{T}_{\text{left}} = \alpha\text{-Tree}(\{(X, Y) \mid X_{i^*} = 0\}, \alpha)$;
  $\mathcal{T}_{\text{right}} = \alpha\text{-Tree}(\{(X, Y) \mid X_{i^*} = 1\}, \alpha)$;
end
$\mathcal{T} = \{\mathcal{T}_{\text{left}}, \mathcal{T}_{\text{right}}\}$;

5.2 Differentially Private Decision Tree

Differentially private decision trees are built in two phases: building a 0-differentially private decision tree structure, and adding noise to the leaf nodes. We anatomize a decision tree into two parts: structure and nodes. Figure 5.2 illustrates the concept. In short, structure refers to decision rules, and nodes indicate probability estimates.

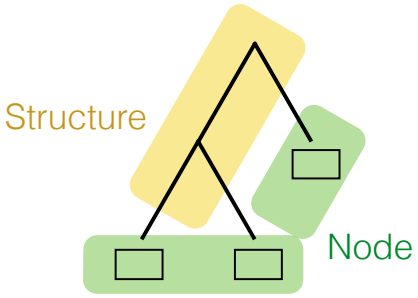


Figure 5.2: Structure and Node of Decision Tree.

5.2.1 Model-layer: obtaining 0-DiffStructure

In this section, we show that it is "possible" and "easy" to obtain a 0-differentially private decision tree structure. In the differential privacy regime, 0-differential privacy is achieved when a single data point does not change the output of a function. Therefore, 0-differentially private decision tree structure (henceforth, 0-DiffStructure) refers to a decision tree structure that does not change regardless of any single data point being removed or added. An algorithm for obtaining a 0-DiffStructure can be formally described as follows:

$$\frac{\Pr(\mathcal{T} = \text{0-DiffStructure}(\mathcal{D}'))}{\Pr(\mathcal{T} = \text{0-DiffStructure}(\mathcal{D}))} \leq \exp(0) = 1 \tag{5.8}$$

where $\mathcal{T}$ is a tree structure, and $\mathcal{D}$ and $\mathcal{D}'$ are two datasets that differ at most one element. For a deterministic 0-DiffStructure algorithm, Equation 5.8 can be re-written as:

$$\begin{aligned} \mathcal{T} &= \text{0-DiffStructure}(\mathcal{D}) \\ &= \text{0-DiffStructure}(\mathcal{D}') \\ \forall \mathcal{D}' \quad &\text{such that} \quad |\mathcal{D} \ominus \mathcal{D}'| \leq 1 \end{aligned}$$

since $\Pr(\mathcal{T} = \text{0-DiffStructure}(\mathcal{D}')) = 1$.

A decision tree structure is essentially a ordered list of splitting variables. If the list of splitting variables do not change regardless of training datasets, then a decision tree structure can be said as 0-differentially private.

Table 5.1: Summary of all possible 8 cases.

	$\mathrm{P}(X=1,Y=1)$	$\mathrm{P}(X=1,Y=0)$	$\mathrm{P}(X=0,Y=1)$	$\mathrm{P}(X=0,Y=0)$
Case 1	$n_{11}+1$	n_{10}	n_{01}	n_{00}
Case 2	n_{11}	$n_{10}+1$	n_{01}	n_{00}
Case 3	n_{11}	n_{10}	$n_{01}+1$	n_{00}
Case 4	n_{11}	n_{10}	n_{01}	$n_{00}+1$
Case 5	$n_{11}-1$	n_{10}	n_{01}	n_{00}
Case 6	n_{11}	$n_{10}-1$	n_{01}	n_{00}
Case 7	n_{11}	n_{10}	$n_{01}-1$	n_{00}
Case 8	n_{11}	n_{10}	n_{01}	$n_{00}-1$

0-DiffStructure admissible splitting variable is a variable that is consistently chosen as a splitting variable across all the datasets that differ by at most one element. Thus, a splitting variable is 0-DiffStructure admissible if:

$$X_{i^*} \mid \mathcal{D} = X_{i^*} \mid \mathcal{D}' \qquad \forall \mathcal{D}' \quad \text{s.t.} \quad |\mathcal{D} \ominus \mathcal{D}'| \leq 1$$

where $X_{i^*} \mid \mathcal{D}$ represents the splitting variable from dataset $\mathcal{D}$. Recall that, in α-Tree, a splitting variable is a variable that provide the highest α-gain.

$$X_{i^*} \mid \mathcal{D} = \arg\max_{X_i} D_\alpha(\mathrm{P}_{\mathcal{D}}(X_i, Y) \| \mathrm{P}_{\mathcal{D}}(X_i)\mathrm{P}(Y))$$

where $\mathrm{P}_{\mathcal{D}}(X_i, Y)$ and $\mathrm{P}_{\mathcal{D}}(X_i)\mathrm{P}_{\mathcal{D}}(Y)$ are estimated from data.

Obtaining a 0-DiffStructure admissible variable is relatively straightforward. When calculating α-gain values for each feature, the first step is to construct a contingency table as follows:

	$Y=1$	$Y=0$	
$X=1$	n_{11}	n_{10}	$n_{1\cdot}$
$X=0$	n_{01}	n_{00}	$n_{0\cdot}$
	$n_{\cdot 1}$	$n_{\cdot 0}$	$n_{\cdot\cdot}$

where n_{ij} specifies the count of examples that are $(X = i, Y = j)$. The relationships between n_{ij}'s are given as:

$$n_{\cdot j} = n_{0j} + n_{1j}$$

$$n_{i\cdot} = n_{i0} + n_{i1}$$

$$n_{\cdot\cdot} = n_{1\cdot} + n_{0\cdot} = n_{\cdot 1} + n_{\cdot 0}$$

For this contingency table, α-gain is obtained as follows:

$$\alpha\text{-gain} = \frac{1 - \sum_{i,j} \mathrm{P}(X = i, Y = j)^{\alpha} \mathrm{P}(X = i)^{1-\alpha} \mathrm{P}(Y = j)^{1-\alpha}}{\alpha(1-\alpha)}$$

$$\mathrm{P}(X = i, Y = j) = \frac{n_{ij}}{n_{\cdot\cdot}}$$

$$\mathrm{P}(X = i)\mathrm{P}(Y = j) = \frac{n_{i\cdot}}{n_{\cdot\cdot}} \frac{n_{\cdot j}}{n_{\cdot\cdot}}$$

Note that addition of a data point changes n_{ij} to $n_{ij}+1$, and deletion of a data point transforms n_{ij} to $n_{ij} - 1$. Since i and j can be either 0 or 1, there are 8 possible cases. Thus, the extensive permutation of all possible datasets can be summarized into only 8 possible cases as shown in Table 5.1. From these 8 cases, we can derive the minimum and the maximum of α-gains across all the possible datasets that differ at most one element from the original dataset $\mathcal{D}$ i.e. $\{\mathcal{D}' \mid |\mathcal{D} \ominus \mathcal{D}'| \leq 1\}$.

If X_i is 0-DiffStructure admissible, then the minimum of α-gain should be greater than the maximums of the other variables' α-gains. Thus, we have a simple sufficient condition for filtering 0-DiffStructure admissible variables:

$$\min_{\mathcal{D}} D_{\alpha}(\mathrm{P}_{\mathcal{D}}(X_{i^*}, Y) \| \mathrm{P}_{\mathcal{D}}(X_{i^*})\mathrm{P}_{\mathcal{D}}(Y)) \geq \max_{\mathcal{D}} D_{\alpha}(\mathrm{P}_{\mathcal{D}}(X_j, Y) \| \mathrm{P}_{\mathcal{D}}(X_j)\mathrm{P}_{\mathcal{D}}(Y)) \tag{5.9}$$

where X_{i^*} is the 0-DiffStructure admissible variable, and X_j represents any variable in the dataset that is not X_{i^*}.

An algorithm for obtaining a 0-DiffStructure is illustrated in Algorithm 5. For each variable X_i, the minimum and maximum of α-gain values

Algorithm 5: 0-DiffStructure

Data: $\mathcal{S} = \{(X, Y)\}, \alpha$
Result: $\mathcal{T}$
Initialize $\boldsymbol{\kappa}, \boldsymbol{\lambda} = []$;
for *i in 1:M* **do**
  Initialize $\mathbf{A} = []$;
  for *case j in 1:8* **do**
    $A_j = D_\alpha(\mathrm{P}_{\text{case } j}(X_i, Y) \| \mathrm{P}_{\text{case } j}(X_i)\mathrm{P}_{\text{case } j}(Y))$;
  end
  $\kappa_i = \min \mathbf{A}$;
  $\lambda_i = \max \mathbf{A}$;
end
$\kappa_{i*} = \max(\kappa_1, \kappa_2, \ldots \kappa_M)$;
if $\kappa_{i^*} \geq \lambda_j \quad \forall\ j \neq i^*$ **then**
  $\mathcal{T}_{\text{left}} = \text{0-DiffStructure}(\{(X, Y) \mid X_{i^*} = 0\}, \alpha)$;
  $\mathcal{T}_{\text{right}} = \text{0-DiffStructure}(\{(X, Y) \mid X_{i^*} = 1\}, \alpha)$;
else
  return {Number of Pos. : $n_{\cdot 1}$, Node Size : $n_{\cdot\cdot}$};
end
$\mathcal{T} = \{\mathcal{T}_{\text{left}}, \mathcal{T}_{\text{right}}\}$;

for the 8 cases are stored in κ_i and λ_i, respectively. We now use the sufficient condition for 0-DiffStructure admissible variables, described in Equation (5.9). The maximum of the minimums, i.e. $\max(\kappa_1, \kappa_2, \ldots, \kappa_M)$, is stored in κ_{i^*}. If κ_{i^*} is greater than the maximum α-gains λ_j of the other variables, then the i^*th variable is 0-DiffStructure admissible. If there exists a 0-DiffStructure

admissible variable, then the algorithm is recursively applied to two disjoint subsets of the input dataset that are partitioned based on X_{i^*}. Otherwise, the 0-DiffStructure algorithm stops and returns two values: the number of positive class examples and the total number of examples.

Not all splitting variables are 0-DiffStructure admissible variables, so the resultant tree from the 0-DiffStructure algorithm is typically smaller than regular decision trees. Note that an obtained 0-DiffStructure is 0-differentially private only for its decision tree structure, not the leaf nodes. To achieve ϵ-differential privacy, the leaf nodes need to be properly noised.

5.2.2 Output-layer: ϵ-DiffPerturbation

Differentially Private α-Tree (DPaT) is a two-layered algorithm: 0-DiffStructure (model-layer) and ϵ-DiffPerturbation (output-layer). In this section, we explain the latter part of the algorithm. Recall our API use case scenario in Figure 5.1. When a query arrives, the query traverses a 0-DiffStructure, and then eventually reaches a leaf node. The leaf node contains two numbers: the number of positive class examples $n_{\cdot 1}$, and the total number of examples $n_{\cdot\cdot}$. Without privacy concerns, the maximum likelihood estimator for the positive class is as follows:

$$\text{API output} = \frac{n_{\cdot 1}}{n_{\cdot\cdot}}$$

However, returning this naïve estimate may result in privacy breaches such as the kinds described in Case Study 5.0.1 and Case Study 5.0.2.

For numeric outputs, differential privacy can be achieved by adding calibrated noise to the true outcome of a query. Before adding calibrated noise, the global sensitivity of a function needs to be defined.

Definition 5.2.1 (Sensitivity [45]). For $f: \mathcal{D} \rightarrow \mathbb{R}$, the sensitivity of f is:

$$\Delta f = \max_{\mathcal{D},\mathcal{D}'} \parallel f(\mathcal{D}) - f(\mathcal{D}') \parallel_1 \tag{5.10}$$

where $|\mathcal{D} \ominus \mathcal{D}'| \leq 1$.

Given the sensitivity of a function f, Theorem 5.2.1 provides the simplest implementation of ϵ-differential privacy.

Theorem 5.2.1 (Additive Laplace Noise [45]). *For $f: \mathcal{D} \rightarrow \mathbb{R}$, a mechanism that adds independently generated noise with distribution Laplace$(\Delta f/\epsilon)$ to an output term enjoys ϵ-differential privacy.*

The sensitivity of our output is straightforward:

$$\begin{aligned}
\phi &= \max(\frac{n_{\cdot 1}+1}{n_{\cdot\cdot}}, \frac{n_{\cdot 1}}{n_{\cdot\cdot}-1}) \\
\chi &= \min(\frac{n_{\cdot 1}+1}{n_{\cdot\cdot}+1}, \frac{n_{\cdot 1}-1}{n_{\cdot\cdot}-1}) \\
\Delta &= \phi - \chi
\end{aligned}$$

Thus, we add Laplace noise as follows:

$$\text{API output} = \frac{n_{\cdot 1}}{n_{\cdot\cdot}} + \text{Laplace}(0, \frac{\Delta}{\epsilon})$$

This mechanism provides ϵ-differential privacy.

The full algorithm for Differentially Private α-Tree (DPaT) is as follows:

1. Construct a 0-DiffStructure from the training dataset
2. For each API call,
   (a) Traverse the constructed 0-DiffStructure
   (b) Extract the corresponding leaf node
   (c) Calculate the sensitivity of the node
   (d) Add ϵ-calibrated Laplace noise, then return

The predictive performance of DPaT is mainly determined by the level of privacy imposed ϵ. Higher ϵ values (low level of privacy) provide more accurate predictions, and lower ϵ values tend to be more noisy providing higher level of privacy protection. In DPaT, the privacy parameter ϵ only affects the output layer, as the model layer already achieves the highest level of privacy protection, 0-differential privacy. Two extreme cases are 0- and ∞-differentially private α-trees. To implement a 0-DPaT, we add random noise sampled from Unif$(-\infty, \infty)$. On the other hand, an ∞-DPaT outputs the true output as it is, since Laplace$(0, 0)$ is a deterministic distribution.

5.2.3 Extension: Ensemble of Differentially Private α-Trees

The predictive performance of a single DPaT is generally not so good as regular decision trees. There are two disjoint reasons that explain the performance behavior:

- **Model-layer**: 0-DiffStructure is generally more restrictive than regular decision tree structures.
- **Output-layer**: Additive Laplace noise perturbs the true outputs.

Although 0-DiffStructure provides nice privacy guarantees, the structure prefers only strong signals ignoring weakly related variables.

We show that we can enhance the predictive performance of DPaT by leveraging the classifier ensemble theory. An ensemble of a diverse set of classifiers can yield better results [79]. Recall that, in α-tree, the α value determines the selection of splitting features, and different values of α can yield distinct splitting features. Similarly, we can obtain a diverse set of 0-DiffStructures by varying α in α-gain.

The difference from traditional ensemble methods is that we need to combine the sensitivities of the base DPaTs as well. For each base 0-DiffStructure i, the maximum and minimum of the output values are noted as ϕ_i and χ_i. Recall that the maximum and minimum are defined with respect to the extensive 8 cases:

$$\phi_i = \max(\frac{n_{\cdot 1} + 1}{n_{\cdot\cdot}}, \frac{n_{\cdot 1}}{n_{\cdot\cdot} - 1}) \mid X_i$$
$$\chi_i = \min(\frac{n_{\cdot 1} + 1}{n_{\cdot\cdot} + 1}, \frac{n_{\cdot 1} - 1}{n_{\cdot\cdot} - 1}) \mid X_i$$

In theory, the maximum of the average output is achieved when all the base 0-DiffStructures' outputs hit their maximum, and the minimum is achieved when

all the base 0-DiffStructures record their minimum. Thus, the sensitivity of the average output can be obtained as follows:

$$\begin{aligned}\Delta &= \text{Average}(\phi_1, \phi_2, \ldots, \phi_E) - \text{Average}(\chi_1, \chi_2, \ldots, \chi_E) \\ &= \text{Average}(\phi_1 - \chi_1, \phi_2 - \chi_2, \ldots, \phi_E - \chi_E) \\ &= \text{Average}(\Delta_1, \Delta_2, \ldots, \Delta_E)\end{aligned}$$

where E represents the total number of base 0-DiffStructures.

The algorithm for Ensemble of Differentially Private α-Trees (EDPaT) is illustrated as follows:

1. Construct E number of 0-DiffStructures from the training dataset by varying α
2. For each API call,
   (a) For each base 0-DiffStructure,
      i. Traverse the constructed 0-DiffStructure
      ii. Extract the corresponding leaf node
      iii. Calculate the sensitivity of the node
   (b) Combine the sensitivities of the base 0-DiffStructure
   (c) Add ϵ-calibrated Laplace noise, then return

The values of α can be any real number. In practice, one can try many α values and pick the values that give distinct 0-DiffStructures.

5.3 Empirical Study

We provide two sets of experiments: diabetic risk scores using Pima Indian dataset, and septic shock risk score using MIMIC-II dataset.

PIMA Indian. The Pima Indian Diabetes dataset is a well-studied public machine learning dataset. The dataset contains 8 feature records of 768 female patients with Pima Indian heritage. The available features in the dataset include:

- **Glucose**: Plasma glucose concentration in an oral glucose tolerance test
- **BMI**: Body Mass Index
- Other features: number of times pregnant, diastolic blood pressure, triceps skin fold thickness, etc.

The class label in the dataset indicates the diagnosis of diabetes. Although there are missing values in the dataset (noted in [7]), we do not specifically treat such missing values.

We first compare the regular tree structure (C4.5) and 0-DiffStructure that are from the same dataset. Figures 5.3 and 5.4 show the regular tree structure and 0-DiffStructure, respectively. As can be seen, both structures have the same splitting variables until the second level splits. However, except for the "$bmi > 26.84$" node, all the second stage nodes of the 0-DiffStructure stopped growing, since there were no more 0-DiffStructure admissible variables. Figure 5.5 shows the values of α-gain measured from the root node and

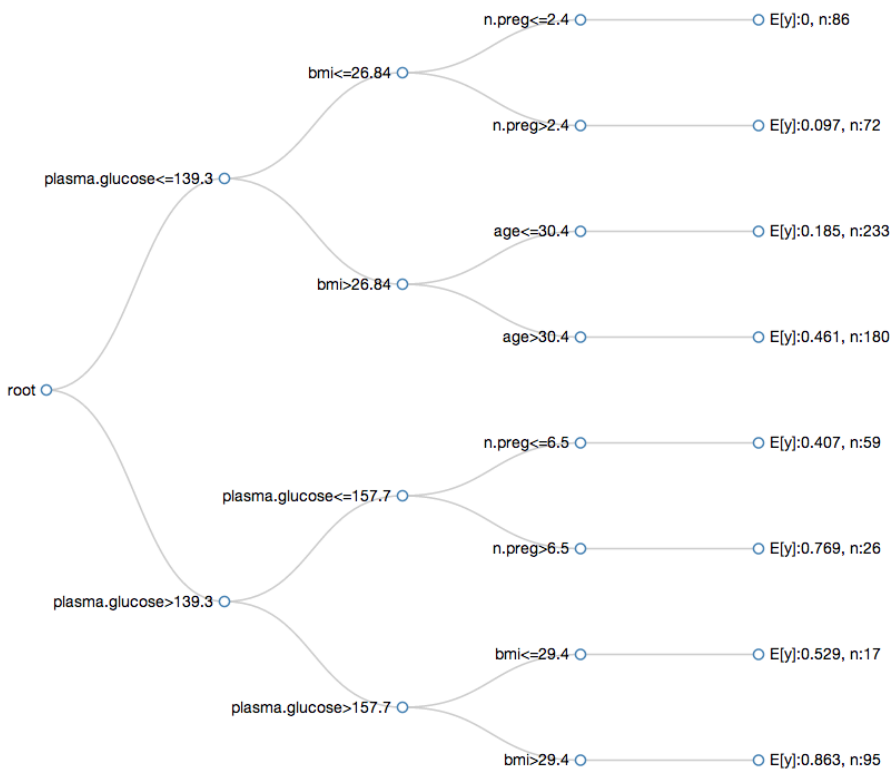


Figure 5.3: Regular α-Tree ($\alpha = 1$, C4.5) trained on the Pima Indian Dataset.

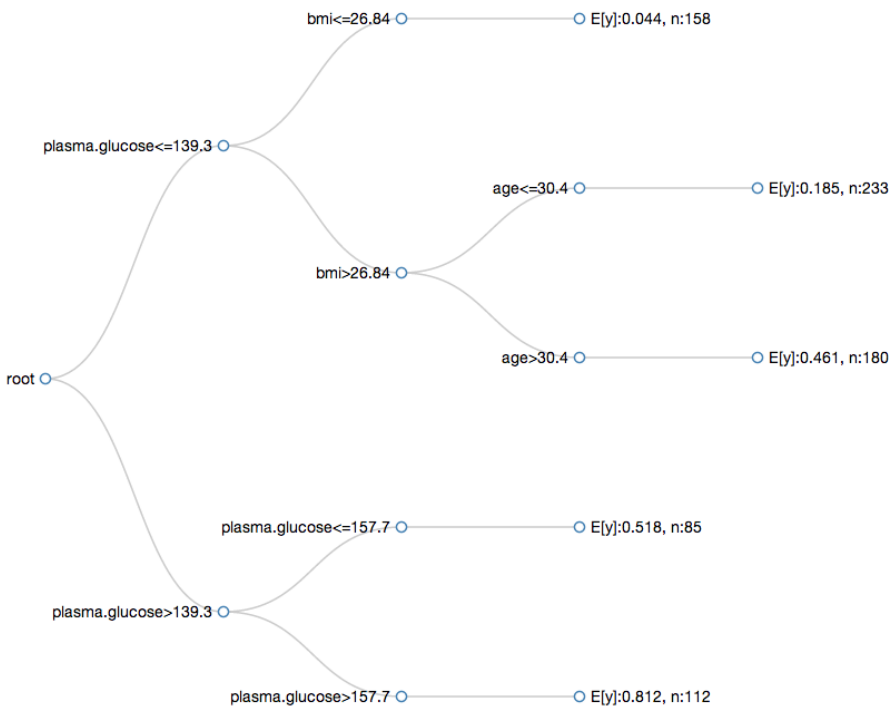
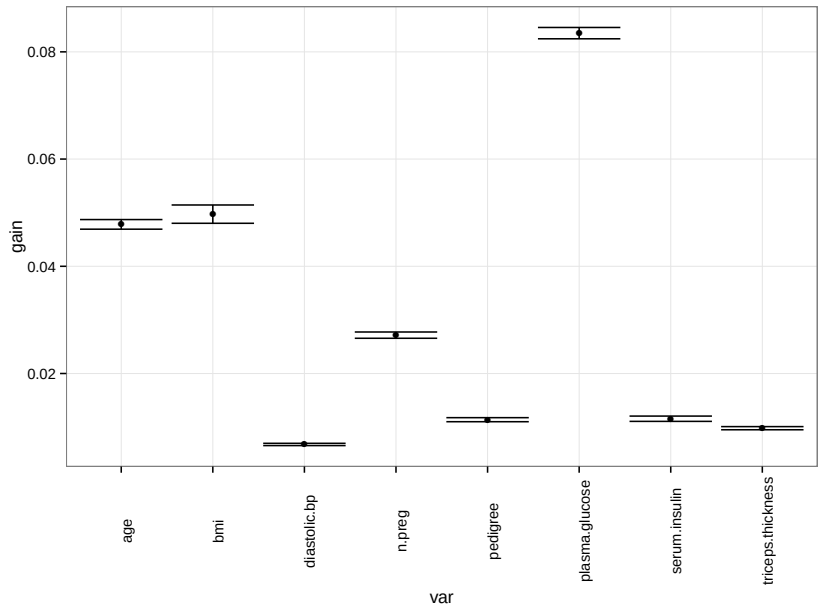
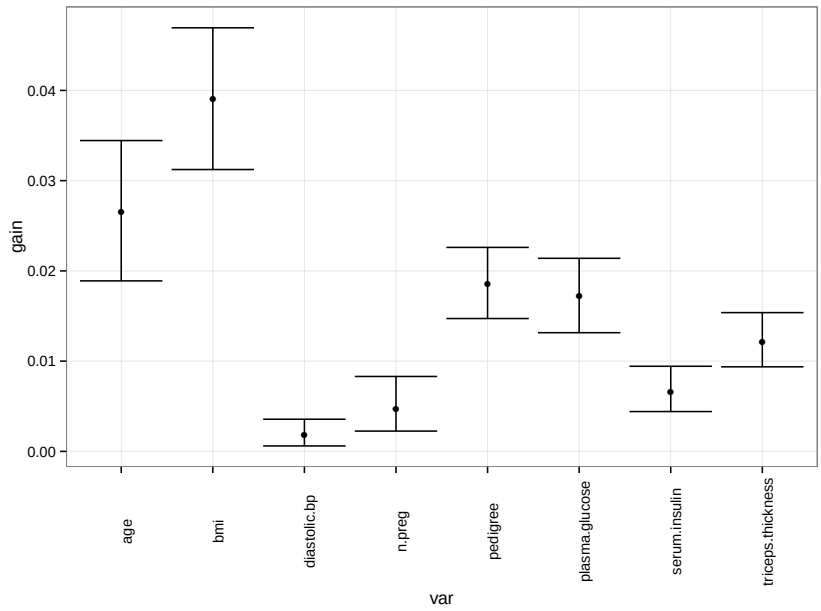


Figure 5.4: 0-differentially private tree structure trained on the Pima Indian Dataset.

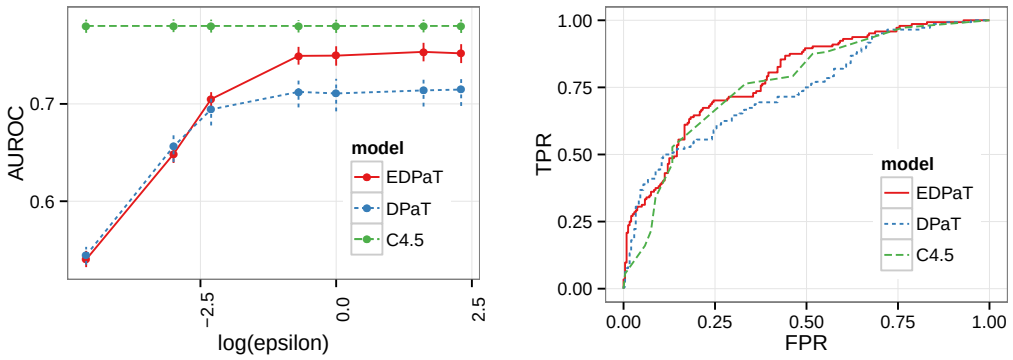


(a) Split



(b) No split

Figure 5.5: 0-DiffStructure admissible split and non-admissible split.



(a) AUROC vs. ϵ (b) A sample ROC curve ($\epsilon = 1$)

Figure 5.6: Results from the Pima Indian dataset.

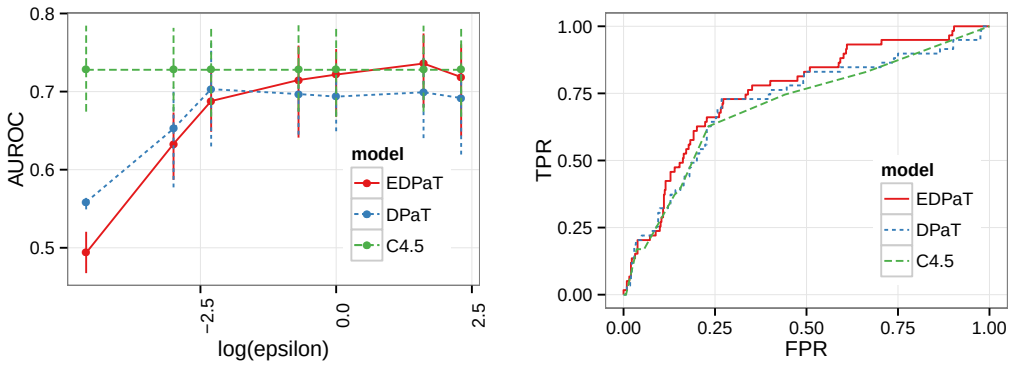
the "$plasma.glucose > 157.7$" node (the last node) of the 0-DiffStructure. The error bars in Figure 5.5 represent the minimum and maximum values computed from the exhaustive 8 cases (see Table 5.1). In the root node, we can observe that the minimum α-gain of "plasma glucose" is significantly higher than the maximums of the other variables, hence "plasma glucose" is chosen as the 0-DiffStructure admissible variable. On the other hand, at the '$bmi \leq 26.84$" node, although "bmi" exhibits the highest α-gain value, but the minimum of "bmi" is smaller than the maximum of "age". Without the 0-DiffStructure constraint, the "bmi" variable would have been chosen as a splitting variable. In fact, the "bmi" variable is the next splitting variable in the regular decision tree structure as shown in Figure 5.3.

Figure 5.6 shows the predictive performance of DPaT and EDPaT over different privacy level constraints. The Pima Indian dataset is randomly partitioned into a training and test sets (50:50), and we measured the predictive

performance (Area Under Receiver Operating Characteristics, AUROC) on the test set. This experiment is repeated 30 times, and the error bars represent the standard deviations. As a baseline, we also measured the performance of C4.5 without privacy constraints (shown as a green line). As can be seen, EDPaT exhibits higher AUROC values than DPaT for the same privacy level ϵ. Noticeably, the performance of EDPaT is comparable to that of a regular decision tree in the region of $\epsilon > 1$. Figure 5.6 (b) shows the ROC curve at $\epsilon = 1$. As the outputs of DPaT and EDPaT are perturbed using Laplace noise, the resultant ROC curves are not so smooth as compared to the curve of C4.5.

MIMIC-II. The MIMIC-II database [120] is one of the largest publicly available clinical databases. The database contains more than 30K patients and 40K ICU admission records. Among many other conditions, in this chapter, we focus on patients with systemic inflammatory response syndrome (SIRS) for septic shock prediction. The features are derived primarily from non-invasive clinical measurements and include blood pressure (systolic and diastolic measurements), body temperature, heart rate, respiratory rate, and pulse oximetry. For each measurement, we use the last observed measurement and three additional sets of derived features: max, min, average values within the last 12 hours.

Septic shock is defined as "sepsis-induced hypotension, persisting despite adequate fluid resuscitation, along with the presence of hypo perfusion abnormalities or organ dysfunction" [13]. The time of septic shock onset was defined using the criteria outlined in a recent work on septic shock prediction



(a) AUROC vs. ϵ (b) A sample ROC curve ($\epsilon = 1$)

Figure 5.7: Results from the MIMIC-II dataset.

[67]. For this subset, there is a total of 1359 patients with 213 transitioning to septic shock.

Figure 5.7 illustrates the results from the septic shock dataset. The septic shock dataset is randomly partitioned into a training and test sets (50:50), and we measured the AUROCs on the test set. Similar to the Pima Indian experiment, this experiment is also repeated 30 times, and the error bars represent the standard deviations. EDPaT again records higher AUROC values than DPaT for the same privacy level ϵ. Surprisingly, the performance of EDPaT is quite comparable to that of C4.5 in the region of $\epsilon > 1$. This implies that it is possible that the predictive performance can be maintained while providing rigorous privacy guarantees.

5.4 Summary

This chapter developed an algorithm for privacy-preserving decision tree APIs. DPaT and its ensemble extension EDPaT provide reasonable predictive performance while protecting the privacy of users. Although we presented only risk score API examples, numerous other APIs can be similarly implemented. For example, hospital quality score estimation, readmission rate prediction, and medical fraud detection problems are possible candidate applications for the DPaT framework. We believe that this kind of privacy-preserving APIs can facilitate creating a safe and accessible environment for healthcare mobile and web applications.

Chapter 6

Conclusions and Future Directions

Open access to health data can unleash a myriad of research and industry opportunities, which will bring enormous social and economic benefits. However, such access can also lead to privacy breaches, which may result in discrimination in insurance and employment markets among others. Unfortunately, the utility of data cannot be increased without sacrificing some privacy. Calibrating the trade-off between utility and privacy will be the key for successful adoption of open health data.

Privacy is a subjective and contextual concept, and it needs to be addressed from both systemic and information perspectives to precisely understand privacy breaches and consequences. This dissertation specifically addressed three popular use cases of health data: 1) synthetic data publication, 2) aggregate data utilization, and 3) privacy-aware API implementation.

PeGS is a categorical data synthesizer algorithm that guarantees a quantifiable disclosure risk. PeGS can handle high-dimensional categorical data that are intractable if represented as contingency tables. California Patient Discharge data were used to demonstrate statistical properties of the proposed synthetic methodology. In practice, PeGS can be applied to already

k-anonymized (generalized and suppressed) data, yielding higher protection on data privacy.

LUDIA and **CUDIA** are novel low-rank approximation algorithms that utilize aggregation constraints in addition to auxiliary information in order to estimate or "reconstruct" the original individual-level values from aggregate data. Experimental results using a Texas inpatient dataset showed that individual-level data can be reasonably reconstructed from county-, hospital-, and zip code-level aggregate data. Several factors affecting the reconstruction quality were also discussed.

DPaT is a privacy-preserving decision tree API framework. The proposed API framework is a general framework that can be adapted to various applications such as hospital quality score estimation, readmission rate prediction, and medical fraud detection problems. Pima Indian and MIMIC-II datasets were used to demonstrate diabetic and septic shock risk score APIs.

The landscape of open health data is rapidly evolving with new data sources, collection methods, and formats. Mobile devices collect behavioral and mobility tracking data, and social network services present graph data. Unstructured data, such as doctors and nurses' notes, are significant in the healthcare domain. Perception of privacy also has changed dynamically over time, and moreover, different cultures exhibit different levels of privacy awareness. These are the open questions for future research to fully and precisely understand the actual privacy risks and benefits of sharing data.

Bibliography

[1] *World Health Statistics*. World Health Organization, 2011.

[2] John M. Abowd and Lars Vilhuber. How protective are synthetic data? *Privacy in Statistical Databases*, 5262:239–246, 2008.

[3] John M. Abowd and Simon D. Woodcock. Disclosure limitation in longitudinal linked data. *Confidentiality Diclosure and Data Access: Theory and Practical Applications for Statistical Agencies*, pages 215–277, 2001.

[4] Charu C. Aggarwal and Philip S. Yu. A general survey of privacy-preserving data mining models and algorithms. In Charu C. Aggarwal and Philip S. Yu, editors, *Privacy-Preserving Data Mining*, volume 34 of *Advances in Database Systems*, pages 11–52. Springer US, 2008.

[5] Ajit Appari and M. Eric Johnson. Information security and privacy in healthcare: current state of research. *International Journal of Internet and Enterprise Management*, 6(4):279–314, 2010.

[6] Marc P. Armstrong, Gerard Rushton, and Dale L. Zimmerman. Geographically masking health data to preserve confidentiality. *Statistics in Medicine*, 18:497–525, 1999.

[7] K. Bache and M. Lichman. UCI machine learning repository.

[8] Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank McSherry, and Kunal Talwar. Privacy, accuracy, and consistency too: A holistic solution to contingency table release. In *The 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2007.

[9] R. H. Bartels and G. W. Stewart. Solution of the matrix equation $ax + xb = c$. *Communications of the ACM*, 15(9):820–826, 1972.

[10] Donald M. Berwick and Andrew D. Hackbarth. Eliminating Waste in US Health Care. *The Journal of the American Medical Association*, 2012.

[11] Rajendra Bhatia and Peter Rosenthal. How and why to solve the operater equation $ax - xb = y$. *Bulletin of the London Mathematical Society*, 29(1):1–21, 1997.

[12] Matthew Block, Amanda Cox, Jo Craven McGinty, and Matthew Ericson. How much hospitals charge for the same procedures, May 2013.

[13] R. C. Bone, R. A. Balk, F. B. Cerra, R. P. Dellinger, A. M. Fein, W. A. Knaus, R. M. Schein, and W. J. Sibbald. Definitions for sepsis and organ failure and guidelines for th euse of innovative therapies in sepsis. *The ACCP/SCCM Consensus Conference Comittee*, 1992.

[14] James G. Booth and James P. Hovert. Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. *Journal of the Royal Statistical Society: Series B*, 61:265–285, 1999.

[15] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[16] Leo Breiman. *Classification and regression trees*. Wadsworth International Group, 1984.

[17] Anna L Buczak, Steven Babin, and Linda J. Moniz. Data-driven approach for creating synthetic electronic medical records. *BMC Medical Informatics and Decision Making*, 10(59), 2010.

[18] Gregory Caiola and Jerome P. Reiter. Random Forests for Generating Partially Synthetic, Categorical Data. *Transactions on Data Privacy*, 3:27–42, 2010.

[19] Joseph A. Calandrino, Arvind Narayanan, Edward W. Felten, and Vitaly Shmatikov. Don't review that book: Privacy risks of collaborative filtering. In *Manuscript*, 2009.

[20] Emmanuel J. Candes and Benjamin Recht. Exact Matrix Completion via Convex Optimization. *Foundations of Computational Mathematics*, 2008.

[21] K. Carroll. Experimental evidence of dietary factors and hormone-dependent cancers. *Cancer Research*, 35:3374–3383, 1975.

[22] Centers for Disease Control and Prevension (CDC). Data and statistics. http://www.cdc.gov/datastatistics/, 2014.

[23] Centers for Medicare and Medicaid Services. Medicare provider charge data. online, January 2014.

[24] Centers for Medicare and Medicaid Services. CMS EHR Meaningful Use Overview. *EHR Incentive Programs*, Octorber 2011.

[25] Anne-Sophie Charest. Empirical evaluation of statistical inference from differentially-private contingency tables. *Privacy in Statistical Databases*, 7556:257–272, 2012.

[26] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 1952.

[27] Andrzej Cichocki and Shun ishi Amari. Families of alpha- beta- and gamma- divergences: Flexible and robust measures of similarities. *Entropy*, 2010.

[28] Chris Clifton and Tamir Tassa. On syntactic anonymity and differential privacy. *Transactions on Data Privacy*, 6(2):161–183, 2013.

[29] Imre Csiszar. Information-type measures of difference of probability distributions and indirect observation. *Studia Scientiarum Mathematicarum Hungarica*, 1967.

[30] Angela Dale and Mark Elliot. Proposals for 2001 samples of anonymized records: an assessment of disclosure risk. *Journal of Royal Statistical Society: Series A*, 2001.

[31] Tore Dalenius and Steven P. Reiss. Data-swapping: A technique for disclosure control (exteded abstract). In *Proceedings of the Section on Survey Research Methods*, 1978.

[32] Fida Kamal Dankar and Khaled El Emam. The application of differential privacy to health data. In *Proceedings of Privacy and Anonymity in the Information Society (PAIS)*, 2012.

[33] Data.CMS.gov. Inpatient prospective payment system. `https://data.cms.gov/Medicare/Inpatient-Prospective-Payment-System-IPPS-Provider/97k6-zzx3`, 2014.

[34] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society: Series B*, 39, 1976.

[35] Rina V. Dhopeshwarker, Lisa M. Kern, Heather C. O'Donnell, Alison M. Edwards, and Rainu Kaushal. Health care consumers' preferences around health information exchange. *Annals of Family Medicine*, 10(5):428–434, 2012.

[36] Jorg Drechsler, Stefan Bender, and Susanne Rassler. Comparing fully and partially synthetic datasets for statistical disclosure control in the

german IAB establishment panel. *Transactions on Data Privacy*, 1:105–130, 2008.

[37] Jorg Drechsler and Jerome P. Reiter. Sampling with synthesis: A new approach for releasing public use census microdata. *Journal of the American Statistical Association*, 105(492):1347–1357, 2010.

[38] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, 2001.

[39] George T. Duncan, Mark Elliot, and Juan-Jose Salazar-Gonzalez. *Statistical Confidentiality: Principles and Practice*. Springer, 2011.

[40] George T. Duncan, Sallie A. Keller-McNulty, and S. Lynne Stokes. Disclosure risk vs. data utility: The R-U confidentiality map. Technical report, National Institute of Statistical Sciences, 2001.

[41] George T. Duncan and Diane Lambert. Disclosure-limited data dissemination. *Journal of the American Statistical Association*, 1986.

[42] O. D. Duncan and B. Davis. An alternative to ecological correlation. *American Sociological Review*, 18:665–666, 1953.

[43] Adam DuVander. Who belongs to the API billionaires club?, 2011.

[44] Cynthia Dwork. Differential privacy. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, volume 4052, pages 1–12, 2006.

[45] Cynthia Dwork. Differential privacy: a survey of results. In *Proceedings of TAMC*, pages 1–19, Berlin, Heidelberg, 2008. Springer-Verlag.

[46] Julia Brande Earp and Fay Cobb Payton. Information privacy in the service sector: An exploratory study of health care and banking professionals. *Journal of Organizational Computing and Electronic Commerce*, 16(2):105–122, 2006.

[47] Carl Eckart and Gale Young. The approximation of one matrix by another lower rank. *Psychometrika*, 1936.

[48] Richard A. Epstein. The legal regulation of genetic discrimination: Old responses to new technology. *Boston University Law Review*, 74(1):1–24, 1994.

[49] Roy T. Fielding and Richard N. Taylor. Principled design of the modern web architecture. *ACM Transactions on Intelligent Systems and Technology*, 2(2):115–150, 2002.

[50] Stephen E. Fienberg and Julie McIntyre. Data swapping: Variations on a theme by Dalenius and Reiss. *Journal of Official Statistics*, 21:309–323, 2005.

[51] Stephen E. Fienberg, Alessandro Rinaldo, and Xiaolin Yang. Differential privacy and the risk-utility tradeoff for multi-dimensional contingency tables. In *The 2010 International Conference on Privacy in Statistical Databases*, pages 187–199, 2010.

[52] Luisa Franconi and Julian Stander. A model-based method for disclosure limitation of business microdata. *Journal of the Royal Statistical Society, Series D*, 51(1):51–61, 2002.

[53] D. A. Freedman, S. P. Klein, M. Ostland, and M. Roberts. On "solutions" to the ecological inference probelm. *Journal of the American Statistical Association*, 93:1518–22, 1999.

[54] David A Freedman. Ecological inference and the ecological fallacy. Technical Report 549, Department of Statistics, University of California Berkeley, CA 94720, October 1999.

[55] David A. Freedman, Stephen P. Klein, Jerome Sacks, Charles A. Smyth, and Charles G. Everett. Ecological regression and voting rights. *Evaluation Review*, (673-816), 15.

[56] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 2010.

[57] Michael Friendly. Mosaic displays for multi-way contingency tables. *Journal of the American Statistical Association*, 89(425):190–200, 1994.

[58] Michael Friendly. Extending mosaic displays: Marginal, conditional, and partial views of categorical data. *Journal of Computational and Graphical Statistics*, 8(3):373–395, 1999.

[59] Wayne A. Fuller. Masking procedures for microdata disclosure limitation. *Journal of Official Statistics*, 9(2):383–406, 1993.

[60] Andrew Gelman and Jennifer Hill. *Data Analysis using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, 2007.

[61] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th Very Large Database*, 1999.

[62] Harvey Goldstein. *Multilevel Statistical Models*. Wiley, 4th edition, 2010.

[63] L Goodman. Ecological regression and the behavior of individuals. *American Sociological Review*, 18:663–664, 1953.

[64] L Goodman. Some alternatives to ecological correlation. *American Journal of Socialogy*, 64:610–625, 1959.

[65] Ragib Hassan and William Yurcik. A statistical analysis of disclosed storage security breaches. In *Proceedings of the second ACM workshop on Storage security and survivability*, pages 1–8, 2006.

[66] Wenbo He, Xue Liu, Hoang Nguyen, Klara Nahrstedt, and Tarek Abdelzaher. PDA: Privacy-preserving data aggregation in wireless sensor netoworks. *IEEE International Conference on Computer Communications*, pages 2045–2053, 2007.

[67] Joyce C Ho, Cheng H Lee, and Joydeep Ghosh. Imputation-Enhanced Prediction of Septic Shock in ICU Patients. In *ACM SIGKDD Workshop on Health Informatics (HI-KDD 2012)*, 2012.

[68] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of 30th Symposium on Theory of Computing*, 1998.

[69] Institute of Medicine. Transformation of health system needed to improve care and reduce costs. Press Release, 9 2012.

[70] Shun ishi Amari. Integration of stochastic models by minimizing α-divergence. *Neural Computation*, 2007.

[71] Charles R. Johnson. Matrix completion problems: a survey. *Proceedings of Symposia in Applied Mathematics*, 1990.

[72] Lou Jost. Entropy and diversity. *Oikos*, 2006.

[73] Kaiser Family Foundation. Hospital adjusted expenses per inpatient day. `http://kff.org/other/state-indicator/expenses-per-inpatient-day/`, 2011.

[74] Sean Keehan, Andrea Sisko, Christopher Truffer, Sheila Smith, Cathy Cowan, John Poisal, M. Kent Clemens, and National Health Expenditure Accounts Proejctions Team. Health Spending Projections Through 2017: The Baby-Boom Generation Is Coming To Medicare. *Health Affairs*, 2008.

[75] Rashid Hussain Khokhar, Rui Chen, Benjamin C.M. Fung, and Siu Man Lui. Quantifying the costs and benefits of privacy-preserving health data publishing. *Journal of Biomedical Informatics*, (0):–, 2014.

[76] Gary King. *A Solution to the ecological inference problem: reconstructing individual behavior from aggregate data*. Princeton University Press, 1997.

[77] Gary King, Ori Rosen, and Martin A. Tanner. Binomial-beta hierarchical models for ecological inference. *Sociological Methods and Research*, 28:61–90, 1999.

[78] Gary King, Ori Rosen, and Martin A. Tanner. *Ecological Inference*. Cambridge University Press, 2004.

[79] Ludmila I. Kuncheva and Christopher J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51:181–207, 2003.

[80] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: efficient full-domain k-anonymity. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 46–60, 2005.

[81] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *Data Engineering, 2006. ICDE '06. Proceedings of the 22nd International Conference on*, 2006.

[82] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proceedings of International Conference on Data Engineering*, 2007.

[83] Roderick J. A. Little. Statistical analysis of masked data. *Journal of Official statistics*, 9(2):407–426, 1993.

[84] Jun S. Liu. The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427):958–966, 1994.

[85] Joseph S. Lombardo and Linda J. Moniz. A method for generation and distribution of synthetic medical record data for evaluation of disease-monitoring systems. *Johns Hopkins APL Technical Digest*, 27(4):356–365, 2008.

[86] Ashwin Machanavajjhala, Daniel Kifer, John Abowd, Johannes Gehrke, and Lars Vilhuber. Privacy: Theory meets Practice on the Map. In *Proceedings of the 24th International Conference on Data Engineering*, 2008.

[87] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramanian. l-diversity: Privacy beyond k-anonymity. *Transactions on Knowledge Discovery from Data*, 1, 2007.

[88] James Manyika, Michael Chui, Peter Groves, Diana Farrell, Steve Van Kuiken, and Elizabeth Almasi Doshi. Open data: Unlocking innovation

and performance with liquid information. *McKinsey Global Institute*, October 2013.

[89] David McClure and Jerome P. Reiter. Differential privacy and statistical disclosure risk measures: An investigation with binary synthetic data. *Transactions on Data Privacy*, 5:535–552, 2012.

[90] Frank McSherry. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2009.

[91] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual Symposium of Foundations of Computer Science*, 2007.

[92] Barry Meier, Jo Craven McGinty, and Julie Creswell. Hospital billing varies wildly, government data shows, May 2013.

[93] Carlos Meier. A role for data: An observation on empowering stakeholders. *American Journal of Preventive Medicine*, 2013.

[94] Srujana Merugu and Joydeep Ghosh. Privacy preserving distributed clustering using generative models. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 211–218, 2003.

[95] Srujana Merugu and Joydeep Ghosh. A distributed learning framework for heterogeneous data sources. In *Proceedings of the 11th ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 208–217, 2005.

[96] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 111–125, 2008.

[97] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy*, pages 173–187, 2009.

[98] Office of Science and Technology Policy. U.S. Open Data Action Plan. *http://www.whitehouse.gov/sites/default/files/microsites/ostp/us_open_data_action_plan.pdf*, 2014.

[99] Carlos Ordonez and Zhibo Chen. Horizontal aggregations in sql to prepare data sets for data mining analysis. *IEEE Transactions on Knowledge and Data Engineering*, pages 678–691, 2012.

[100] Yubin Park and Joydeep Ghosh. Cudia: Probabilistic cross-level imputation using individual auxiliary information. *ACM Transactions on Intelligent Systems and Technology*, 2012.

[101] Yubin Park and Joydeep Ghosh. Ensembles of α-Trees for Imbalanced Classification Problems. *IEEE Transactions on Knowledge and Data Engineering*, 26(1):131–143, 2014.

[102] Yubin Park, Joydeep Ghosh, and Mallikarjun Shankar. Lugs: A scalable non-parametric data synthesizer for privacy preserving big health data publication. In *International Conference on Machine Learning WHEALTH 2013*, 2013.

[103] Yubin Park, Joydeep Ghosh, and Mallikarjun Shankar. Perturbed gibbs samplers for generating large-scale privacy-safe synthetic health data. In *Healthcare Informatics (ICHI), 2013 IEEE International Conference on*, pages 493–498, 2013.

[104] Silvia Polettini. Maximum entropy simulation for microdata protection. *Statistics and Computing*, 13:307–320, 2003.

[105] Richard A. Posner. The right of privacy. *Georgia Law Review*, 4(1):393–422, 1978.

[106] President's Concil of Advisors on Science and Technology. Report to the president realizing the full potential of health information technology to improve healthcare for americans: the path forward. Technical report, Office of Science and Technology Policy, the White House, December 2010.

[107] John Ross Quinlan. *C4.5: prgrams for machine learning*. Morgan kaufmann, 1993.

[108] T. E. Raghunathan, Jerome P. Reiter, and Donald B. Rubin. Multiple imputation for statistical disclosure limitation. *Journal of Official*

Statistics, 19(1):1–16, 2003.

[109] Trivellore E. Raghunathan, James M. Lepkowski, John Van Hoewyk, and Peter Solenberger. A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology*, 27(1):85–95, 2001.

[110] A. Raman. Enforcing privacy through security in remote patient monitoring ecosystems. In *6th International Special Topic Conference on Information Technology Applications in Biomedicine*, pages 298–301, 2007.

[111] Jerome P. Reiter. Estimating risks of identification disclosure in microdata. *Journal of American Statistical Association*, 100(472):1103–1112, 2005.

[112] Jerome P. Reiter. Releasing multiply imputed, synthetic public use microdata: an illustration and empirical study. *Journal of the Royal Statistical Society, Series A*, 168:185–205, 2005.

[113] Jerome P. Reiter and Jorg Drechsler. Releasing multiply imputed synthetic data generated in two stages to protect confidentiality. *Statistica Sinica*, 20:405–421, 2010.

[114] Alfred Renyi. On measures of information and entropy. *Proceedings of the fourth Berkeley Symposium on Mathematics*, 1961.

[115] J. Richmond. Aggregation and identification. *International Economic Review*, 17:47–56, 1976.

[116] Thomas C. Rindfieisch. Privacy, information technology, and health care. *Communications of the ACM*, 40(8):92–100, 1997.

[117] W. S. Robinson. Ecological correlations and the behavior of individuals. *American Sociological Review*, 15:351–357, 1950.

[118] Donald B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. Wiley, 1987.

[119] Donald B. Rubin. Discussion: Statistical disclosure limitation. *Journal of Official Statistics*, 9:462–468, 1993.

[120] Mohammed Saeed, Mauricio Villarroel, Andrew T. Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H. Kyaw, benjamin Moody, and Roger G. Mark. Multiparameter intelligent monitoring in intensive care ii (MIMIC-II): A public access intensive care unit database. *Critical Care Medicine*, 2011.

[121] Joseph W. Sakshaug and Trivellore E. Raghunathan. Synthetic data for small area estimation. *Privacy in Statistical Databases*, 6344:162–173, 2011.

[122] Pierangela Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 1998.

[123] J. L. Schafer. *Analysis of Incomplete Multivariate Data*. Chapman and Hall, 1997.

[124] Paul M. Schwartz. Privacy and the economics of personal health care information. *Texas Law Review*, 76(1):1–76, 1997.

[125] Mark Smith, Robert Saunders, Leigh Stuckhardt, and J. Michael McGinnis. *Best Care at Lower Cost: The Path to Continuously Learning Health Care in America*. The National Academies Press, 2012.

[126] Jordi Soria-Cormas and Jorg Drechsler. Evaluating the potential of differential privacy mechanisms for census data. In *UNECE Conference of European Statisticans*, 2013.

[127] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):571–588, 2002.

[128] Latanya Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10:557–570, October 2002.

[129] Matthias Templ. Statistical disclosure control for microdata using the r-package sdcMicro. *Transactions on Data Privacy*, pages 67–85, 2008.

[130] Texas Department of State Health Services. Texas hospital inpatient discharge public use data file. `http://www.dshs.state.tx.us/thcic/hospitals/Inpatientpudf.shtm`, 2006.

[131] Texas Department of State Health Services. Texas Inpatient Public Use Data File. `https://www.dshs.state.tx.us/thcic/hospitals/Inpatientpudf.shtm`, 2014.

[132] Constantino Tsallis. Possible generalization of boltzmann-gibbs statistics. *Journal of Statistical Physics*, 1988.

[133] Stef van Buuren and Karin Groothuis-Oudshoorn. MICE: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3), 2011.

[134] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

[135] L. Willenborg and T. de Waal. *Elements of Statistical Disclosure Control*, volume 155. Springer, 2001.

[136] William E. Winkler. A contingency-table model for imputing data satisfying analytic constraints. *U.S. Census Bureau Statistical Research Division Research Report Series*, 2003.

[137] William E. Winkler. General discrete-data modeling methods for producing synthetic data with reduced re-identification risk that preserve analytic properties. *U.S. Census Bureau Statistical Research Division Research Report Series*, 2010.

[138] Xiaokui Xiao and Yufei Tao. Anatomy: simple and effective privacy preservation. In *Proceedings of the 32nd international conference on Very large data bases*, pages 139–150, 2006.

[139] Xiaokui Xiao and Yufei Tao. m-invariance: towards privacy preserving re-publication of dynamic datasets. In *Proceedings of SIGMOD*, 2007.

[140] Xialin Yang, Stephen E. Fienberg, and Alessandro Rinaldo. Differential privacy for protecting multi-dimensional contingency table data: Extensions and applications. *Journal of Privacy and Confidentiality*, 4(1):101–125, 2012.

[141] Huaiyu Zhu and Richard Rohwer. Information geometric measurements of generalization. Technical Report 4350, Aston University, 1995.