DISCLOSURE RISK MEASUREMENT OF ANONYMIZED DATASETS AFTER PROBABILISTIC ATTACKS

A Thesis by

Nafia Malik

Bachelor of Science, Bangladesh University of Professionals, 2011

Submitted to the Department of Electrical Engineering and Computer Science
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

July 2014

DISCLOSURE RISK MEASUREMENT OF ANONYMIZED DATASETS AFTER PROBABILISTIC ATTACKS

The following faculty members have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Computer Networking.

Rajiv Bagai, Committee Chair

Murtuza Jadliwala, Committee Member

Khawaja Saeed, Committee Member

DEDICATION

To my loving parents and my beloved husband.
All I have and will accomplish are only possible due to their unconditional support and countless sacrifices.

ACKNOWLEDGEMENTS

I wish thank my advisor Dr. Rajiv Bagai for his constant guidance and continuous motivation to complete my thesis. His positive encouragements helped me to start my masters at the first place and his infinite support kept me focused all the way through to finishing. I would like to thank Dr. Murtuza Jadliwala for encouraging my technical writings and inspiring me with LaTeX, and my fellow research group member Huabo Lu for helping me to start with the template. Lastly, I convey my heartfelt thank to my family and friends for their endless support, unconditional love and limitless tolerance throughout the entire period of writing the thesis.

ABSTRACT

We present a unified metric for analyzing the risk of disclosing anonymized datasets. Datasets containing privacy sensitive information are often required to be shared with unauthorized users for utilization of valuable statistical properties of the data. Anonymizing the actual data provides a great opportunity to share the data while preserving its statistical properties and privacy. The risk of disclosure remains, as hackers may perform a de-anonymization attack to breach the privacy from released datasets. Existing metrics for analyzing this risk were established in the context of infeasibility attacks where each consistent matching (i.e., feasible mapping between actual data and anonymized data) appears equally likely to the hacker. In practice, the hacker may possess some background knowledge for assigning unequal probabilities to all the matchings. We consider these unequal probabilities assigned to matchings to compute the expected closeness of the matchings to the actual mapping adopted for anonymization. We find that our metric delivers a more practical risk assessment for decision makers but has a high computational complexity. Hence, we propose an efficient heuristic for our metric and analyze its accuracy. We also show that our heuristic results in a very close estimation to the actual metric.

TABLE OF CONTENTS

Chapter **Page**

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Risk of unwanted disclosure of data has always been a major concern in data sharing over the Internet. Preserving data privacy while data mining was initially introduced for hiding actual attributes of data from unauthorized recipients [1]. In 1997 Moor [2] reveals his worries about privacy breach of data through the Internet and states,

> *"Our challenge is to take advantage of computing without allowing computing to take advantage of us. When information is computerized, it is greased to slide easily and quickly to many ports of call. This makes information retrieval quick and convenient, but legitimate concerns about privacy arise when this speed and convenience lead to the improper exposure of information*."

Today in the age of *Big data* and *Cloud computing*, privacy preservation while mining the data is more of a compulsion than an option. This allows computational access on sensitive information while preventing unexpected privacy breach [3].

Owners of datasets containing privacy sensitive data (such as, health-care related information) have to release some form of data for important purposes, e.g., research on diseases requires data from huge population for investigation [4] and software testing for health insurance company or at hospital requires actual data [1] for analyzing the real time performance. Sharing these sets of actual data is a violation of privacy law according to HIPAA [5]. Possible alternatives include releasing a transformed (secured) version of data instead of real data. One of these options is encryption of data before release, but this hides the statistical attribute of data and makes the substitution cipher [6] futile for research or software testing. Another alternative is to use fake data for testing, but constructing fake data [1] with properties of real data requires inspection on real data and also requires expensive man-hours to produce. The widely acceptable form of data release considering the privacy is to release a *sanitized* version of data [7, 8]. In this thesis we preferred anonymity over all other method of data sanitization. In literature, *anonymity* is defined as the

lack of capability to identify a particular item of interest among the set of other items (anonymity set) [9]. Prior to release of the anonymized data set, the dataset owner requires to analyze the risk of unwanted disclosure considering the possible de-anonymization attacks. These attacks are performed by unauthorized recipients (e.g., hackers) of data with an intention to re-identify sensitive information by revealing the actual mapping between real and anonymized datasets. Edman et al. [10] and Laskmanan et al. [7, 8] have given acceptable metrics for measuring the remaining anonymity after such de-anonymization attacks. Existing metrics for measuring the remaining anonymity, are established on the important assumption, that each consistent matching (i.e. feasible mappings) being equally likely to the hacker. These metrics are based on infeasibility attacks. They consider infeasible matchings to have no chance and other matchings to have equal chances of being the actual mapping. In a real life scenario, the hacker may be equipped with a set of background knowledge for assigning unequal chances to all the matchings. Unlike in an infeasibility attack, now the hacker may favor a matching over other matchings. A set of normalized values of these unequal chances gives a real valued probability distribution over all possible matchings. A metric computing on this probability distribution on all possible matchings will capture the practical scenario and give a better risk assessment for the dataset owners.

In this thesis, we give a new metric considering the probability assigned to each matching as the weight for calculating expected closeness of the matching to the actual mapping. We start with analyzing the limitations of existing metrics and give an effective unified metric based on the uneven probability distribution on all possible matchings, for measuring the remaining anonymity of an anonymized dataset after an attack. We examine the computational complexity of our metric and realizing the severe requirement of an efficient heuristic, we propose an efficient heuristic for computation. We also perform a detailed analysis of the accuracy of our heuristic and propose three interesting directions for future work.

The rest of the thesis is organized as follows. First, we present short a survey and a detailed analysis of the existing permanent based metrics by Edman et al. [10] and Laskmanan et al. [7] in Chapter 2. In Section 2.2 of this chapter we also describe the infeasibility attack model in context.

Then, we introduce the probability attack model in Section 3.1 of Chapter 3. In Sections 3.2 and 3.3 of this chapter, we explain the induced probability distribution all possible matching, and introduce uneven and flat probability matrices. The concepts of truly uneven and flat probability matrices are used for constructing our unified metric. In Chapter 4 we present our metric and the heuristic for our metric. Finally, we conclude and highlight some directions for future work in Chapter 5.

CHAPTER 2

BACKGROUND: EXISTING METRICS FOR MEASURING ANONYMITY

In this chapter, we give a literature survey of the existing methods adopted for data privacy, the infeasibility attack model and a detailed analysis of existing metrics for measuring remaining anonymity.

2.1 Quick Survey on Existing Methods

We start with a quick survey on techniques for data sanitization. Later, in this section we discus the methods for anonymity measurement and features observed for anonymity attack models.

First, we give a short survey on widely used methods for data privacy and their limitations in comparison to anonymity. A common concept of data privacy is limiting the disclosure of information by data sanitization before sharing the data. The majority of work done in data sanitization involves applying statistical disclosure limitation, e.g., data swapping, cell suppression, rounding, sampling and generation synthetic data as described by Moore [11], Fienberg et al. [12] and Domingo-Ferrer et al. [13]. These methods involved perturbation of actual data characters for the cost of privacy. Another method proposed by Evmievski et al. [14] is association rule mining, which involves randomizing data items in a transaction for preserving the privacy of individual transactions. In this method, privacy breaches of transactions are proposed to be prevented by randomization of operator, thus hiding the association rules. But, as pointed by Verykios et al. [15], association rule hiding changes data frequency as randomizing of transactions involves insertions and removals of data items in each transaction. Agrawal and Srikant [16] proposed generation of synthetic data with a perturbed distribution which is close to actual distribution of data. Interesting analysis of relative effectiveness of such generalized data to original data is by Agrawal and Aggarwal [17], presents the trade-off between privacy and information loss. Statistical databases prevent privacy breach by answering only statistical queries but various methods presented in the survey

by Adam and Wortmann [18] shows that an analysis of sequential queries can help to deduce information of an individual. The k-anonymity model [19–21] for domain generalization hierarchies to replace each recorded value with a generalized value. The problem remains as it becomes difficult to reconstruct data model with actual characteristics or statistical properties. These problems can be solved by releasing anonymized datasets. As with all other benefits of anonymity for satisfying the privacy constrains in a dataset, the most attractive property of anonymized datasets is that it preserves the actual characteristics and statistical properties of original data. Anonymity is a widely used method for privacy preservation. In a proposal for privacy terminologies, Pfitzmann and Hansen [9] described anonymity as the state being unidentifiable with a set of subjects called anonymity set. In an anonymized dataset, each entry of original item is replaced by a unique anonymized representative. This method is adopted in most cases to preserve information privacy, e.g., message communication, database transactions, location services, software testing and in many other applications.

Second, we give the methods of measuring the remaining anonymity in the aftermath of an attack. The methods include information theoretic entropy based and most importantly permanent of underlying attack matrix based measures. Serjantov and Danezis [22] gave their metric in context of anonymous communication as effective anonymity set size by separating equally likely senders of messages from all senders. Degree of anonymity proposed by Diaz et al. [23] is an effective anonymity set size normalized with maximum set anonymity size. Further analysis is performed by Edman et al. [10] for a collective measure of anonymity for all users of anonymity set. The metric by Lakshmanan et al. [8] introduced the metric for disclosure risk analysis of anonymized datasets. This metric is the collective correctness expressed in terms of expected cracks of matchings. Both these metrics [10] [8] are based on the calculation of permanent of underlying attack matrices. These metrics are established on the concept of the elements of anonymity set being equally likely, thus they are applicable to infeasibility attacks mostly. The practical scenario where these elements are unequally likely, demands a metric for probabilistic attacks. In the following sections a detailed analysis of metric by Edman et al. [10] and Lakshmanan

et al. [8] is given and based on the analysis, we develop our a new metric for probabilistic attacks in Chapter 4.

2.2 Infeasibility Attack Model and An Example

To release sanitized form of n sensitive entries of information, we assume R to be the set of recorded (actual) entries and T to be the set of transformed (anonymized) entries. Set R may contain privacy sensitive entries of health information, e.g., names of diseases, syndrome of illness, charges on medical billing, or personal identification records (e.g., Names, DOBs, or Addresses). Set T contains the anonymized form of entries in R, such that each entry from R is uniquely mapped to an entry from T and *vice versa*, i.e., $R \cap T = \emptyset$, and $|R| = |T| = n$. We assume R and T are public information but the exact mapping between the entries deployed by the dataset owner is private. We also assume that the hacker is equipped with some background knowledge and attempts to reveal the exact mapping between these two sets. We consider a *hacker* is an unauthorized recipient of data willing to de-anonymize and gain information from anonymized dataset. The maximum anonymity is observed when for any entry $r \in R$, every $t \in T$ seems a feasible anonymized entry to the hacker. In Figure 2.1(a) we represent this situation in form of a complete bipartite graph $K_{n,n}$ between R and T for $n = 5$. Here, an edge $\langle r_i, t_j \rangle$ shows the feasibility of t_j being the anonymized entry for r_i. As described by Edman et al. [10], the hacker can perform an infeasibility attack or a probabilistic attack, based on some prior knowledge about the dataset. In an *infesibility attack*, the hacker performs the attack based on his background knowledge and removes some edges as infeasible to arrive at a subgraph G of $K_{n,n}$. By producing the subgraph G, the hacker tries to find the matching representing the exact correspondence between every $r \in R$ and $t \in T$. A *matching* is a one to one correspondence between the entries of actual data and anonymized data. The exact matching M_0 is the matching that is considered as the key to anonymization and is a secret by the dataset owner. A completely successful attack will result in a subgraph or an exact matching with n edges connecting each $r \in R$ to a unique $t \in T$ and an instance of no anonymity. Figure 2.1(b) shows an arbitrary matching out of $n!$ possible matchings from $K_{n,n}$. Thus, more edges identified as infeasible by the hacker results in a stronger attack than

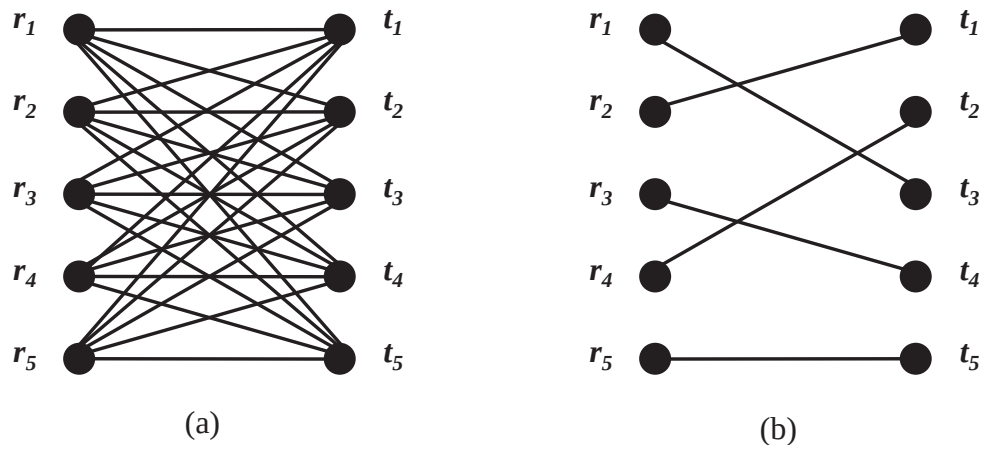


Figure 2.1: (a) Complete anonymity; (b) An instance of no anonymity.

a few edges. These attacks lie between two extreme ends of full anonymity (i.e., Figure 2.1(a)) and no anonymity(i.e., Figure 2.1(b)). We also assume the hacker makes correct analysis of infeasibility, so subgraph G always contain the exact mapping deployed by data owner. In a *probabilistic attack*, instead of removing edges, the the hacker assigns real valued probabilities to all the edges. Probabilistic attacks are discussed further in Chapter 3.

Now, we give an example of an infeasibility attack. Let us, suppose an anonymized medical history record of 10 transactions is released from respiratory care unit of a hospital as in Figure 2.2(a). Each of these transactions is the anonymized set of respiratory disease diagnosis in a patients medical records. Here, we assume, $T = \{u, v, x, y, z\}$ be the pool of transformed entries of $R = \{Flu, Viral\ Fever, Cold, Asthma, Tuberculosis\}$ recorded diagnosis. We consider the hacker is equipped with some background knowledge that the patients are recognized with *Flu* in case of 40% to 90% times for respiratory illness. The hacker also computes frequency of appearance of anonymized entries from the transactions and finds x, y, and z appeared between the range of *Flu*. He also finds u and v appearing outside the frequency range. These findings lead the hacker to determine the edges $\langle Flu, x\rangle$, $\langle Flu, y\rangle$, and $\langle Flu, z\rangle$ as feasible and edges $\langle Flu, u\rangle$, and $\langle Flu, v\rangle$ as infeasible. He establishes the knowledge that Flu is anonymized among x, y, and z and and not as u or v. The hacker determines and removes all the infeasible edges and arrives at a subgraph G of $K_{n,n}$ as in Figure 2.2(b). Graph G is also represented in a biadjacency matrix in

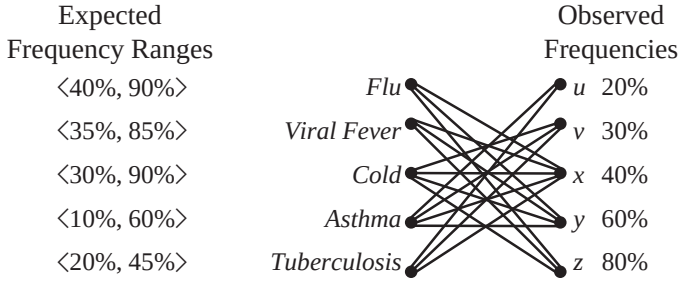
Released Anonymized Transactions

1	{x,z}	6	{v,z}
2	{v,y}	7	{y,z}
3	{u,y}	8	{v,z}
4	{x,z}	9	{u,y,z}
5	{x,y,z}	10	{x,y,z}

(a)

	u	v	x	y	z
Flu	0	0	1	1	1
Viral Fever	0	0	1	1	1
Cold	0	1	1	1	1
Asthma	1	1	1	1	0
Tuberculosis	1	1	1	0	0

(c)



(b)

Figure 2.2: (a) Released anonymized transactions; (b) Graph G arrived by hacker after removing infeasible edges; (c) A biadjacency matrix of Graph G.

Figure 2.2(c). Here the rows hold recorded entries and the columns hold the transformed entries. Presence and absence of edges are marked with 1s and 0s. The hacker tries to find the exact mapping used for anonymization by considering all the matchings that are *contained* in graph G. A matching M is contained in graph G if $E_M \in E_G$, i.e, all the edges of M are in graph G. Figure 2.3 shows the 18 possible matchings that are contained in G at Figure 2.2(b). Dataset owners required to assess the risk of information disclosure before releasing the transactions. Examples of such transactions appeared in Figure 2.2(a). Anonymity metrics help the dataset owners by giving measures of remaining anonymity on the dataset after an attack. Anonymity metrics are well studied fields of research. In the following sections we describe and compare two anonymity metrics from earlier research on infeasibility attacks.

2.3 Edman, Sivrikaya, and Yener's Metric d

Edman et al. [10] considered the size of anonymity set as the key to determine the remaining anonymity of a dataset after the hacker arrives at a particular graph G. The size of an anonymity

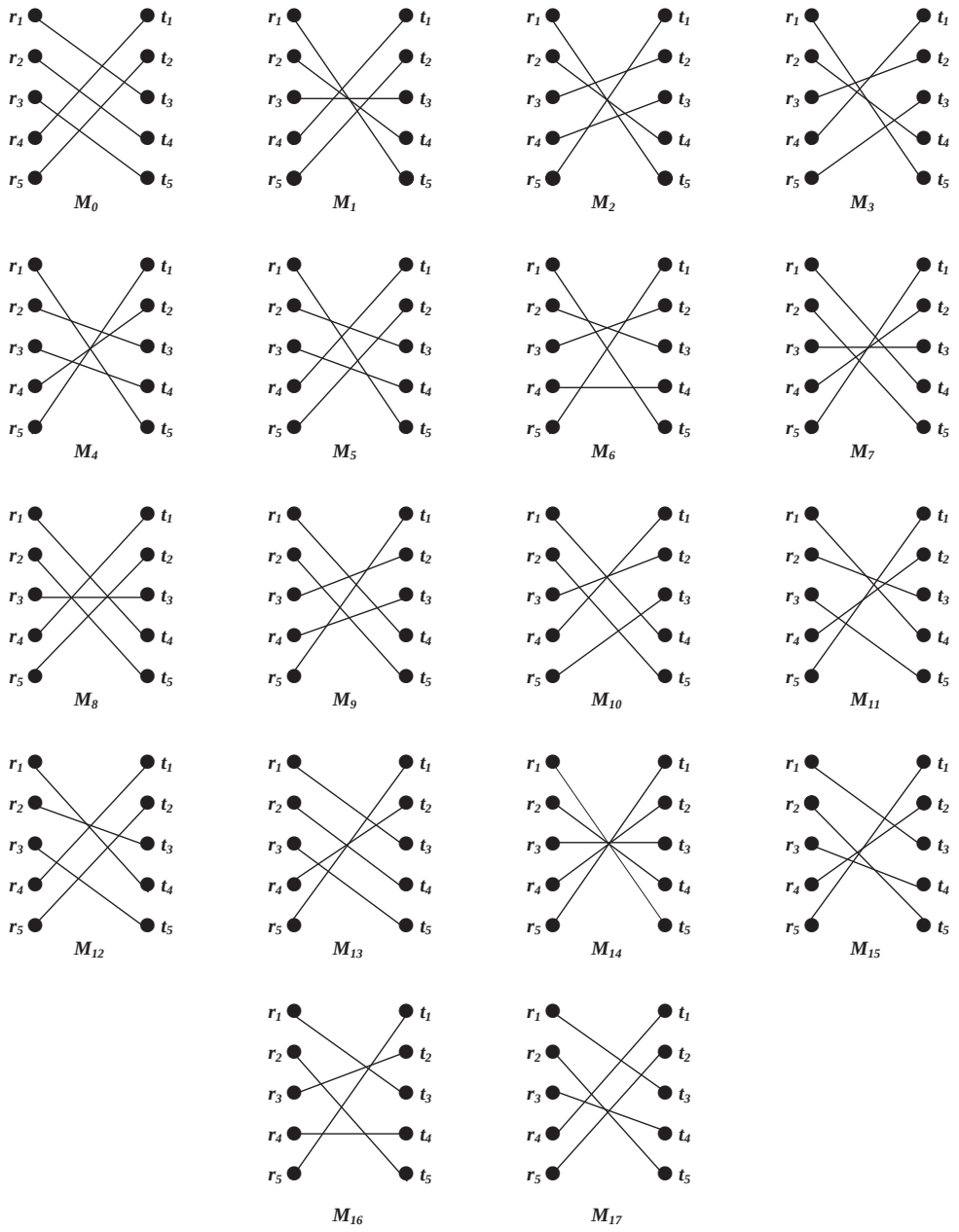


Figure 2.3: All possible matching produced from G of Figure 2.2(b).

set can be found by counting the number of matchings determined as a potential exact matching by the hacker. Potential exact matchings are the matchings that are contained in graph G. We let, $\mathcal{M}(G)$ denote the set of matchings contained in graph G. According to Asratian et al. [24]

the number of matchings contained in a graph G is equal to the *permanent* value of a biadjacency matrix A_G representing graph G. The permanent value of any $n \times n$ matrix A of real numbers is defined as:

$$permanent\,(A) = \sum_{\phi \in \Phi_n} A_{1\phi(1)} A_{2\phi(2)} ... A_{n\phi(n)} \tag{2.1}$$

where, Φ_n is the set of all bijections $\phi\colon\ \{1, 2, ..., n\} \rightarrow \{1, 2, ..., n\}$ i.e., permutations of the first n positive integers. It is also known that if A is a biadjacency matrix containing only 0s and 1s, then $permanent(A)$ is an integer value between 0 and $n!$. We assume M_0 from Figure 2.3 is the exact matching deployed by the dataset owner as the mapping between R and T for anonymity. Also, the hacker performs an infeasibility attack with correct background knowledge and arrives at G, hence $M_0 \in \mathcal{M}(G)$, and $|\mathcal{M}(G)| > 1$. Edman et al. [10] proposed the following metric as the degree of anonymity:

$$d(G) = \begin{cases} 0 & \text{if } n = 1 \\ \frac{log(|\mathcal{M}(G)|)}{log(n!)}, & \text{otherwise} \end{cases} \tag{2.2}$$

The value produced by $d(G)$ gives the measure of anonymity on a scale of $[0, 1]$. Here, the value $d(G) = 1$ indicates maximum anonymity. This is a scenario when the hacker finds all the matchings contained in of graph $K_{n,n}$ as potential candidates for being M_0 i.e., $G = K_{n,n}$ and $\mathcal{M}(G)$ has exactly $n!$ matchings. The value, $d(G) = 0$ indicates no anonymity at all. This is a scenario when the hacker has performed a completely successful attack and narrowed down his graph $G = M_0$ and $\mathcal{M}(G)$ has exactly 1 matching. Data set owner considers the anonymized transactions are safe to release if the value of $d(G)$ results in a greater than a minimum value chosen for safety level. Here, hacker background knowledge for arriving at graph G is also considered as public information. In our example attack in Section 2.2, the number of matchings contained in graph G in Figure 2.2(b) can be found by taking the permanent of A_G in Figure 2.2(c). The biadjacency matrix A_G gives the $permanent(A_G) = 18$, i.e., the exact matching M_0 is hidden among 18 potential matchings out of 120 maximum possible. The degree of anonymity given by $d(G)$ is $log(18)/log(120) = 0.6037$.

The degree of anonymity proposed by Edman et al. [10] is a rough metric for anonymity as $d(G)$ is determined only by the number of matchings in which exact matching M_0 is hiding.

Here, a higher value produced by the metric indicates a safer anonymized dataset to realease. In this context, any matching $M \in \mathcal{M}(G)$ chosen to be M_0 by dataset owner would result in the same degree of anonymity given by $d(G)$.

2.4 Lakshmanan, Ng, and Ramesh's Metric ε

A more accurate measure of anonymity after an attack is given by the collective closeness of matchings of the anonymity set to the exact matching deployed by the dataset owner. We define the closeness of matching as the number of cracks. A *crack* in any matching $M \in \mathcal{M}(G)$ is any edge of M that is also contained in M_0. The number of cracks $C(M)$ of a arbitrary matching $M \in \mathcal{M}(G)$ is the common number of edges between the matching M and the exact matching M_0 deployed by the dataset owner and also $0 \leq C(M) < n$. Lakshmanan et al. [8] measured the remaining anonymity of an anonymized data set in terms of expected number of cracks in a randomly choosen matching $M \in \mathcal{M}(G)$. The annonymized dataset is found to be safe to release for the mapping deployed by dataset owner, if the expected number of cracks are within some pre-established safety level. Here, a lower value produced by the metric gives a higher safety level to anonymized dataset.

We adopted the formulation given by Lakshmanan et al. [8] to calculate the number of matchings in $\mathcal{M}(G)$ with exactly c cracks. We make partitions with R^c vertices in $\mathcal{M}(G)$ where each partition contains only the matchings with exactly c cracks. There will be $n+1$ partitions for $c = 0, 1, 2, , n$ (because there can be at most n and at least 0 cracks found in any matching $M \in \mathcal{M}(G)$). Also there will be no matchings in the partition with $n-1$ cracks as the n^{th} non-cracked edge will have to incident to the vertices for n^{th} crack. Then, for every partition, we make sub partitions of S of matchings with common cracked edges. Hence, $R^c = \{S \subseteq R : |S| = c\}$. In the sub partitions of S, every matching has common sub graph $G(S')$ i.e., the vertices with c number of cracks the edges. For each sub partition, we identify that common sub graph $G(S')$(i.e., cracked edges and their incident vertices) from the matchings and remove it from the graph G. We also remove any other cracked edge(s) if found in G and arrive at $G(S) = \langle R', T', E' \rangle$. Here, $R' = R \setminus S$, R , $T' = T \setminus T''$,where $\{t : r \in S \text{ and } \langle r, t \rangle \in Mo)\}$, $E' = E \setminus E''$, and $E'' = \{$

$\langle r, t\rangle$: $r \in S$ or $t \in T''$ or ($r \in R'$ and $\langle r, t\rangle \in Mo$)}. Then, we calculate $permanent(A_{G(S)})$ to find the count of matchings in each sub partition of $\mathcal{M}(G)$ and then sum all the counts found for sub partitions of each partition with c cracks is $\sum_{S \in R^c} permanent(A_{G(S)})$ Lakshmanan et al. [8] formulated the expected number of cracks $\varepsilon(G)$ by taking the arithmetic average of total number of cracks. Hence,

$$\varepsilon(G) = \frac{1}{permanent(A_G)} \left[\sum_{c=0}^{n} \left\{ c \sum_{S \in R^c} permanent(A_{G(S)}) \right\} \right] \tag{2.3}$$

If the mapping deployed by dataset owner is

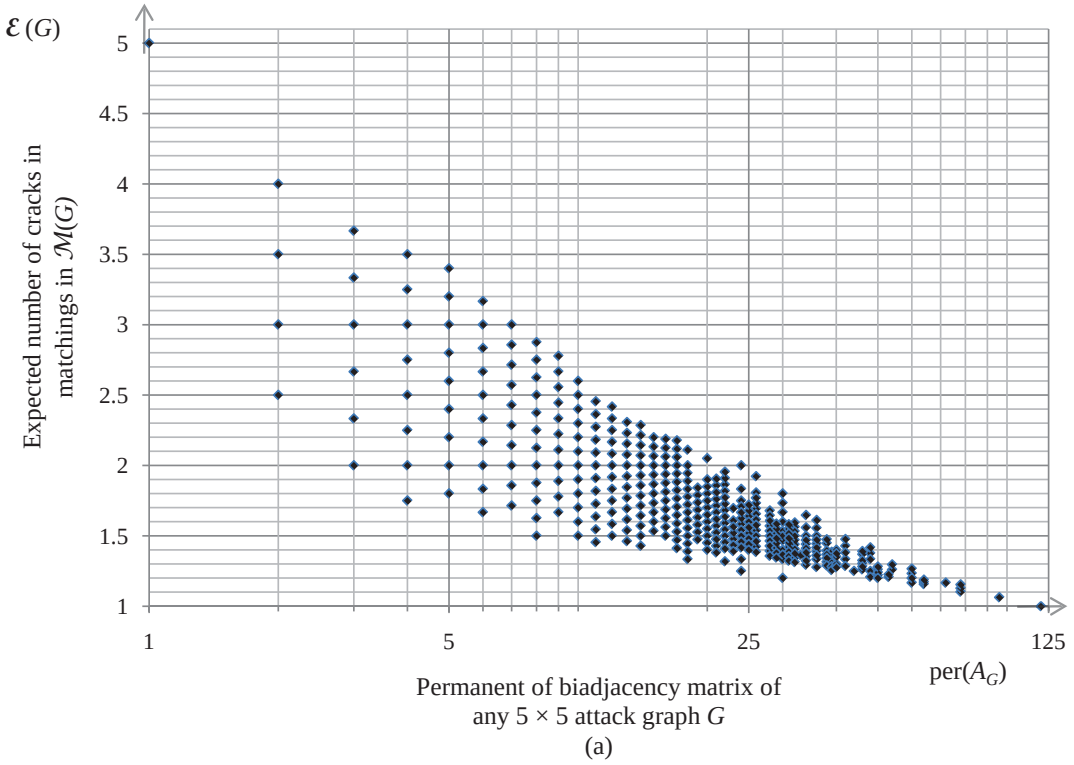
$$M_0 = \{\langle Flu, x\rangle, \langle Viral\ Fever, y\rangle, \langle Cold, z\rangle, \langle Asthma, u\rangle, \langle Tuberculosis, v\rangle\}$$

then, the hacker arrives at G and finds the 18 matchings on Figure 2.3 equally likely, then expected number of cracks in a randomly chosen matching is $(5+3+1+2+0+2+0+0+2+0+1+1+3+3+0+1+1+3)/18 = 1.56$. Lakshmanan et al. [8] formulated their metric based on the quality of the anonymity set deemed by the hacker. Here quality is considered as the closeness of the feasible matchings $M \in \mathcal{M}(G)$ to M_0.

2.5 Comparative Analysis of Existing Metrics d and ε

There are significant differences between the evaluation of remaining anonymity given by existing metric $d(G)$ and $\varepsilon(G)$. The metric $d(G)$ proposed by Edman et al. [10] gives the measure of anonymity in terms of the permanent value of underlying biadjacency matrix of graph G, which is the number of matchings deemed feasible by the hacker. The metric $\varepsilon(G)$ proposed by Lakshmanan et al. [8] measures the same based on the collective closeness of feasible matchings to the actual mapping for anonymity in terms of expected number of cracks. Here, metric $d(G)$ considers only the quantity of matchings in the anonymity set whereas metric $\varepsilon(G)$ also considers the quality of the matchings. In this section, we give a performance analysis of both the metrics to show that metric $\varepsilon(G)$ offers more accurate measure of remaining anonymity than metric $d(G)$, by considering both quality and quantity of matchings deemed feasible by the hacker from the anonymity set. To compare metric $d(G)$ and $\varepsilon(G)$, again we let,

$$M_0 = \{\langle Flu, x\rangle, \langle Viral\ Fever, y\rangle, \langle Cold, z\rangle, \langle Asthma, u\rangle, \langle Tuberculosis, v\rangle\}$$



	u	v	x	y	z
Flu	0	0	1	1	1
Viral Fever	0	1	0	1	1
Cold	0	1	0	0	1
Asthma	1	0	1	0	0
Tuberculosis	1	1	0	0	0

(b)

	u	v	x	y	z
Flu	1	1	1	1	0
Viral Fever	1	0	0	1	0
Cold	1	0	0	0	1
Asthma	1	1	1	1	1
Tuberculosis	1	1	0	0	0

(c)

Figure 2.4: (a) Permanents and expected crack values for all possible 5×5 attack graphs; (b) An example matrix with permanent 4 and expected crack value 1.75; (c) An example matrix with permanent 7 and expected crack value 3.

be the actual mapping deployed by the dataset owner which produced the anonymized transactions on Figure 2.2(a). We compute permanent values of 5×5 biadjacency matrices representing all possible attack graphs that contain M_0 and plot them on horizontal axis. Then we compute the expected crack values of same set of underlying attack graphs on the vertical axis. There are $2^{20} = 1,048,576$ such attack graphs with many of them producing overlapped values and the data

points form an inversely proportionate pattern. Also we plot the data points on horizontal axis in Figure 2.4(a) in logarithmic scale to improve visibility.

We observe in Figure 2.4(a) that, both the metrics agree with each other at most data points as the expected number of cracks fall with the increase of permanent values. Recall that safety levels of a dataset are higher, when permanent values are higher for metric d and when expected crack values are lower for metric ε. But in some of the data points, these metrics disagree and give different measures of safety level.

- Firstly, we discover expected crack values ranging between 1.8 to 3.5 for a stationary permanent value 5. Here, unlike metric $d(G)$, metric $\varepsilon(G)$ indicates ranged values of safety level, after different attacks. Here, different attacks with same anonymity set size, produce same quantities of different qualities of matchings.

- Secondly, we compare the data points $\langle 4, 1.75\rangle$ and $\langle 7, 3\rangle$, and their corresponding biadjacency matrices on Figure 2.4(b) and Figure 2.4(c). Here the tuple is defined as $\langle permanent\ value, expected\ crack\ value\rangle$. Metric $d(G)$ finds higher level of safety in the attack captured data point $\langle 7, 3\rangle$ than in data point $\langle 4, 1.75\rangle$ as 7 is a greater value to 4, but metric $\varepsilon(G)$ completely disagrees as expected crack value 1.75 compared to 3 gives higher safety.

We find metric $\varepsilon(G)$ proposes a better metric of anonymity with considering more information for measurement than in metric $d(G)$. We focus at the useful components and uncovered context by metric $\varepsilon(G)$ and propose a unified metric for more practical scenario of both infeasibility and probabilistic attacks.

CHAPTER 3

PROBABILITY DISTRIBUTION ON MATCHINGS BY PROBABILISTIC ATTACKS

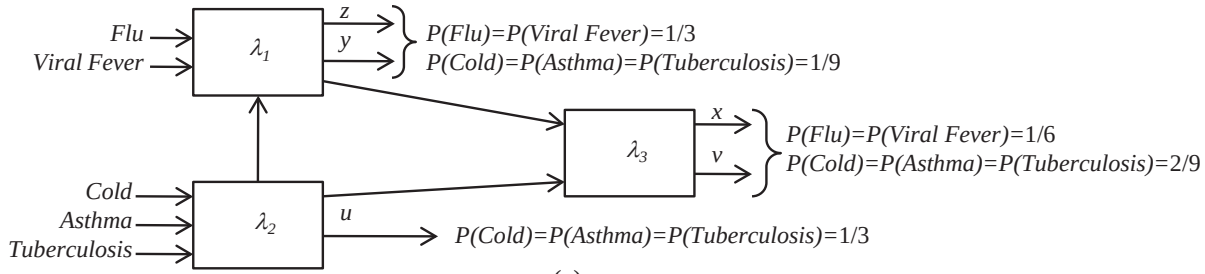
Earlier in Section 2.2, we mentioned two type of attacks performed by the attacker, namely infeasibility attacks and probabilistic attacks. We observed that in an infeasibility attack the graph G results in less than n^2 equally important edges as the edges determined infeasible by the hacker are removed and rest of the feasible edges are considered equally likely. In a more practical scenario, the hacker performs a probabilistic attack by assigning uneven real values between 0 and 1 to all n^2 edges of the complete bipartite graph $K_{n,n}$. Probabilities assigned to the edges of a complete bipartite graph induce a probability distribution on the set of all possible matchings and gives the probability of any randomly chosen matching $M \in \mathcal{M}(K_{n,n})$ being M_0.

In this chapter, we introduce the probabilistic attack and present the probabilities assigned on graph edges from these attacks. We also discuss the method for inducing the probability distribution on all possible matchings from probabilities on the edges.

3.1 Probabilistic Attack Model and Example

In a *probabilistic attack*, all the anonymized entries are considered with chances of being the anonymized representative of actual entries. Here, the hacker considers every entry $t_i \in T$ with a real valued probability to be the anonymized entry of each recorded entry $r_j \in R$. Recall that set R is the set of recorded entries and set T is the set of transformed entries representing the anonymized version of $r_i \in R$. The real valued probability assigned to an edge $\langle r_i, t_j \rangle$ gives the chance of $t_j \in T$ being the anonymized representative of $r_i \in R$. Also with the natural property of probability, these probabilities adds up to 1 $\sum_{j=1}^{n} P(r_i, t_j) = 1$ and $\sum_{i=1}^{n} P(r_i, t_j) = 1$ resulting in a doubly-stochastic probability matrix. dataset owner may use a dataflow network of randomizing nodes to construct the exact mapping M_0 before releasing the anonymized dataset. The Figure 3.1(a) gives an example of a data-flow network consisting of randomizing nodes λ_1, λ_2 and λ_3 that takes the recorded entries $Flu, Viral\ Fever, Cold, Asthma,\ and\ Tuberculosis$ as

input and gives the transformed entries $u, v, x, y, \ and \ z$ as the output based on some shuffling done by internal functionality of the each node. We assume the behavior of these internal functionalities



(a)

P	*u*	*v*	*x*	*y*	*z*
Flu	0	1/6	1/6	1/3	1/3
Viral Fever	0	1/6	1/6	1/3	1/3
Cold	1/3	2/9	2/9	1/9	1/9
Asthma	1/3	2/9	2/9	1/9	1/9
Tuberculosis	1/3	2/9	2/9	1/9	1/9

(b)

Figure 3.1: (a) The dataflow network adopted by dataset owner to construct M_0; (b) The probability matrix produces by the network at (a).

are known to the dataset owner and unknown to the hacker. We also assume the hacker knows the block structure of the data-flow network, output labels, and input values. But, he is unable to see the values coming out of the network, hence he assigns probabilities to each output label of being the inputs. He assigns these probabilities to edges of graph $K_{n,n}$ for every input and output pair if there is a channel in the data-flow network for the pair based on his background knowledge. For example, the outputs of node λ_1 gives probability $1/3$ to y and z be anonymized representative Flu and $Viral\ Fever$. Therefore the edges $\langle Flu, y\rangle$, $\langle Flu, y\rangle$, $\langle Viral\ Fever, y\rangle$, and $\langle Viral\ Fever, z\rangle$ are assigned with probability $1/3$. Also, the edges $\langle Flu, u\rangle$, $\langle Viral\ Fever, u\rangle$ are assigned with probability 0 as there are no channels for them in the data-flow network. The biadjacency matrix P representing the graph $K_{n,n}$ after the said probabilistic attack is given in Figure 3.1(b).

3.2 Induced Probability on Matchings by Probability of Graph Edge

Biadjacency probability matrix resulted from the probabilistic attacks have an interesting property, i.e., these matrices are doubly stochastic in nature. In this section we describe the method that induces the probability distribution on the set $\mathcal{M}(K_{n,n})$ of all possible matchings from the doubly stochastic probability matrix (i.e., biadjacency matrix resulted from the probabilistic attack).

In a doubly stochastic probability matrix P, the sum of the elements of each row or each column adds up to exactly 1. This is given by the fact that M_0 is essentially a bijection between $R = \{r_1, r_2, ..., r_n\}$ and $T = \{t_1, t_2, ..., t_n\}$. First, we define a few important terminologies:

A slice s is any subset of n cells of a $n \times n$ matrix P where no two cells are from the same row.

The weight of a slice $w(s)$ is the product of the values in all the cells in s.

A diagonal d is any slice $\mathcal{S}(P)$ where no two cells are from the same column of matrix P.

The normalized weight of a diagonal $\mathcal{W}(d)$ is the weight of any diagonal $d \in \mathcal{D}(P)$ normalized with $permanent(P)$. Here $\mathcal{W}(d) = w(d)/permanent(P)$.

Here, a slice s is a representation of a sub graph of complete bipartite graph $K_{n,n}$ between R and T with one edge connecting each $r \in R$ (i.e. a function from R to T). And a diagonal d is a representation of a sub graph between R and T with one edge connecting each $r \in R$ and each $t \in T$. We define such sub graphs as matchings, which implies, a diagonal is a representation of a set of probabilities corresponding to all the edges of a matching. We let $\mathcal{S}(P)$ define the set of all possible slices and $\mathcal{D}(P)$ define the set of all possible diagonals. For a $n \times n$ matrix P, $|\mathcal{S}(P)| = n^n$ and $|\mathcal{D}(P)| = n!$. From the definition of the *weight of a slice*, we propose the following.

Proposition 3.2.1 *For a given probability matrix P,*

(a) $\sum_{s \in S(P)} w(s) = 1$

(b) $\sum_{d \in D(P)} w(d) = permanent(P)$

Proof (a) Probability matrix P is of $n \times n$ size. By definition and performing algebraic rearrangement we get,

$$\sum_{s \in S(P)} w(s) = \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} ... \sum_{j_n=1}^{n} P_{1j_1} P_{2j_2} ... P_{nj_n} = \prod_{i=1}^{n} (P_{i1} + P_{i2} + ... + P_{in}) = 1$$

The last equality is followed from the definition of doubly stochastic matrix, that the elements of each row of P adds to 1.

(b) Is followed from the definition of $permanent(P)$. ∎

Therefore, $permanent(P)$ is the sum of weights of all diagonals of P. As $\mathcal{D}(P) \subseteq \mathcal{S}(P)$, we arrive at a a corollary of the Proposition 3.2.1 is that $permanent(P) \leq 1$. The equality is found when P contains exactly one 1 in each of its columns and rows. The minimum possible value of $permanent(P)$ is well known to be $n!/n^n$, when all cells in P are $1/n$ (see, for example, Egorychev [25]). We propose the following from the definition of normalized weight of a diagonal $d \in \mathcal{D}(P)$ and from Proposition 3.2.1(b).

Proposition 3.2.2 *For a given probability matrix P,*

$\sum_{d \in D(P)} \mathcal{W}(d) = 1$

Proof Since, $\sum_{d \in D(P)} \mathcal{W}(d) = \sum_{d \in D(P)} w(d)/permanent(P) = 1$. ∎

In matrix P, each value is a probability and each row of matrix P gives a probability distribution on the set T. The probabilities contained in any specific row i are the probabilities for each $t_j \in T$ of being associated with r_i in the matching M_0 employed by the dataset owner.

Now we consider the set T^R, shown in Figure 3.2, of all n^n functions $f : R \to T$, and let some fixed function $g \in T^R$ be given. Suppose a function f from the set T^R is constructed randomly as follows:

- We pick some $t_j \in T$, with a probability $P_{1,j}$ according to the distribution that is contained in the row 1 of P, and set that chosen t_j to be $f(r_1)$.

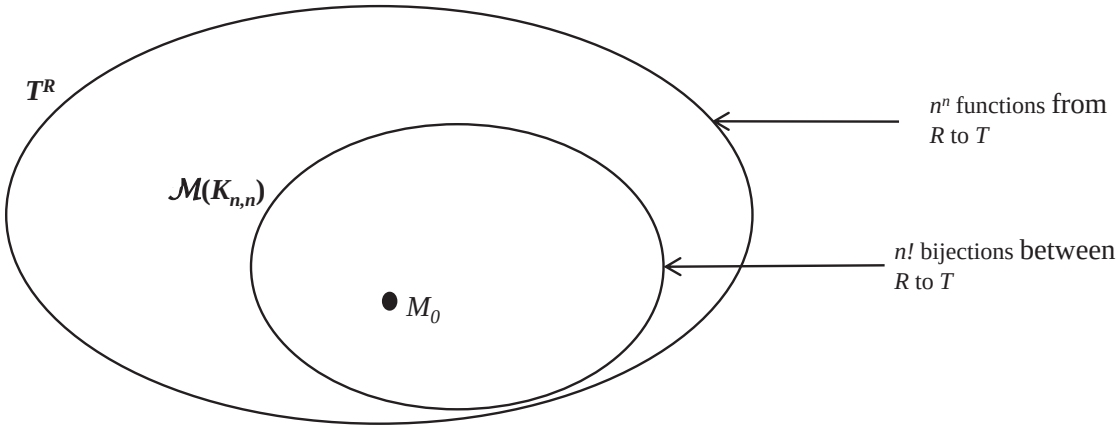


Figure 3.2: Sets T^R of all functions from R to T, and $\mathcal{M}(K_{n,n})$ of all $n!$ bijections between R and T.

- Then, we similarly set $f(r_2), f(r_3), ..., f(r_n)$ according to the distributions that is contained in rows $2, 3, ..., n$, respectively.

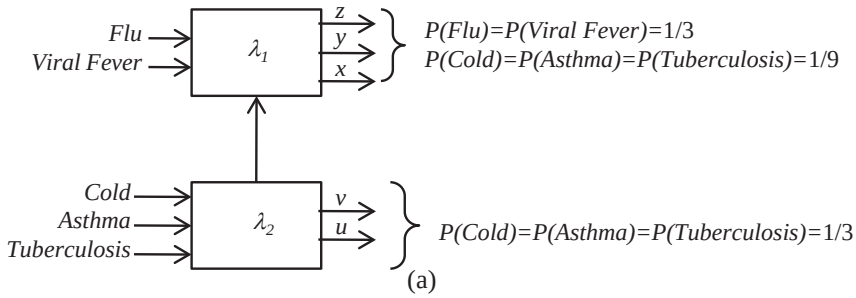
The probability that the function f constructed in this manner is identical to the given function $g \in T^R$ is $\prod \{P_{ij} | g(r_i) = t_j \}$, i.e, the weight of the slice of P that corresponds to function g. By Proposition 3.2.2(a), the said weights add up to 1, i.e., this gives a probability distribution on the entire set T^R. Also, by Proposition 3.2.1(b), $permanent(P)$ is the probability that our randomly constructed function f is a bijection, i.e., it represents a matching between R and T.

Now if the given function g is a bijection, i.e., $g \in \mathcal{M}(K_{n,n})$. Then, given the case when the function f constructed randomly as said is also a bijection, the *normalized weight* of the diagonal of P corresponding to g is the probability of the case: $f = g$. This induces a probability distribution on the set $\mathcal{M}(K_{n,n})$ since, by Proposition 3.2.2, these normalized weights add up to 1.

As the cells in P are the probabilities of edges being in M_0, the normalized weights of the individual diagonals of P are thus the probabilities induced by P to their corresponding matchings, of being M_0.

3.3 Truly Uneven and Flat Probability Matrices

In this section, we introduce the concept of truly uneven and flat probability matrices. We observe the dataflow network in Figure 3.3(a) and the probability matrix P produced by this network in Figure 3.3(b). The permanent of the matrix P of Figure 3.3(b) is $4/81$. The following are



(a)

P	u	v	x	y	z
Flu	0	0	1/3	1/3	1/3
Viral Fever	0	0	1/3	1/3	1/3
Cold	1/3	1/3	1/9	1/9	1/9
Asthma	1/3	1/3	1/9	1/9	1/9
Tuberculosis	1/3	1/3	1/9	1/9	1/9

(b)

Figure 3.3: (a) Another example of a dataflow network adopted by dataset owner to construct M_0; (b) The flat probability matrix produces by the network at (a).

two example matchings, of all the $5! = 120$ matchings contained in the graph $K_{5,5}$ corresponding to this matrix, along with their probabilities of being M_0:

$$M_1 = \{\langle Flu, u\rangle, \langle Viral\ Fever, z\rangle, \langle Cold, u\rangle, \langle Asthma, x\rangle, \langle Tuberculosis, v\rangle\},$$

$$\mathcal{W}(M_1) = \{\frac{1}{4/81}(0.\frac{1}{3}.\frac{1}{3}.\frac{1}{9}.\frac{1}{3})\} = 0$$

$$M_2 = \{\langle Flu, x\rangle, \langle Viral\ Fever, z\rangle, \langle Cold, u\rangle, \langle Asthma, y\rangle, \langle Tuberculosis, v\rangle\},$$

$$\mathcal{W}(M_2) = \{\frac{1}{4/81}(\frac{1}{3}.\frac{1}{3}.\frac{1}{3}.\frac{1}{9}.\frac{1}{3})\} = \frac{1}{36} \approx 0.027778$$

An interesting characteristic of this probability matrix P is that the normalized weight of 84 of its 120 diagonals is 0, and that of each of the other 36 is $1/36$. Matchings M_1 and M_2 above correspond to one in each of those classes, respectively. In other words, matchings that have a non-zero probability of being M_0 are all equally likely to be M_0.

Probability matrices with this property are called flat matrices, and they provide no additional probabilistic information to the hacker than their corresponding $0-1$ biadjacency matrices that possess an identical zero-pattern. The biadjacency matrix A of Figure 3.3(a) has the same zero-pattern as that of the above probability matrix P, i.e. both matrices contain 0 values in exactly the same cells, thereby declaring the same 36 matchings to be feasible, and all those matchings have, according to P, an equal probability of being M_0. In contrast, the probability matrix Q of Figure 3.4(b) contains the same zero-pattern, but is not flat. Here, the normalized weight of 84 of its

A	*u*	*v*	*x*	*y*	*z*
Flu	0	0	1	1	1
Viral Fever	0	0	1	1	1
Cold	1	1	1	1	1
Asthma	1	1	1	1	1
Tuberculosis	1	1	1	1	1

(a)

Q	*u*	*v*	*x*	*y*	*z*
Flu	0	0	57/100	2/5	3/100
Viral Fever	0	0	3/10	23/100	47/100
Cold	17/50	29/100	3/100	3/10	1/25
Asthma	29/100	13/25	2/25	1/20	3/50
Tuberculosis	37/100	19/100	1/50	1/20	2/5

(b)

Figure 3.4: (a) The biadjacency matrix A containing same information as the flat matrix P of Figure 3.3(b); (b) An example probability matrix Q that assigns truly uneven probabilities to matchings declared feasible by A and P.

120 diagonals is 0, and the other 36 uneven weight ranging from 0.0002 to 0.2543. The permanent of Q is $326/5361$. According to Q, the probabilities of those 72 feasible matchings of being M_0 vary from $79/421189$ to $252/991$, as shown by the following three example matchings:

$$M_3 = \{\langle Flu, u\rangle, \langle Viral\ Fever, x\rangle, \langle Cold, v\rangle, \langle Asthma, z\rangle, \langle Tuberculosis, y\rangle\}\,,$$

$$\mathcal{W}(M_3) = \{\frac{1}{326/5361}(\frac{57}{100}.\frac{47}{100}.\frac{3}{10}.\frac{13}{25}.\frac{37}{100})\} = \frac{252}{991} \approx 0.254289$$

$$M_4 = \{\langle Flu, v\rangle, \langle Viral\ Fever, u\rangle, \langle Cold, z\rangle, \langle Asthma, y\rangle, \langle Tuberculosis, x\rangle\}\,,$$

$$\mathcal{W}(M_4) = \{\frac{1}{326/5361}(\frac{2}{5}.\frac{3}{10}.\frac{29}{100}.\frac{29}{100}.\frac{2}{5})\} = \frac{815}{12277} \approx 0.066384$$

$$M_5 = \{\langle Flu, x\rangle, \langle Viral\ Fever, v\rangle, \langle Cold, u\rangle, \langle Asthma, y\rangle, \langle Tuberculosis, z\rangle\},$$

$$\mathcal{W}(M_5) = \{\frac{1}{326/5361}(\frac{3}{100}.\frac{23}{100}.\frac{3}{100}.\frac{29}{100}.\frac{19}{100})\} = \frac{79}{421189} \approx 0.000188$$

The matrix Q can assign truly uneven probabilities to the feasible matchings contained in A_G and P. Another example of flat matrix F is given in Figure 3.5(b). This matrix F is constructed by the

A_G	*u*	*v*	*x*	*y*	*z*
Flu	1	0	1	1	0
Viral Fever	0	1	0	1	0
Cold	0	0	1	0	1
Asthma	0	1	0	1	0
Tuberculosis	1	0	1	0	1

(a)

F	*u*	*v*	*x*	*y*	*z*
Flu	(√5-1)/2	0	(3-√5)/2	0	0
Viral Fever	0	1/2	0	1/2	0
Cold	0	0	(3-√5)/2	0	(√5-1)/2
Asthma	0	1/2	0	1/2	0
Tuberculosis	(3-√5)/2	0	√5-2	0	(3-√5)/2

(b)

Figure 3.5: (a) The biadjacency matrix A_G of an example graph G; (b) The flat matrix $F \in \mathcal{R}(G)$.

method outlined by Bagai et al. [27]. Figure 3.5(a) and 3.5(b) show the biadjacency matrix A_G and the flat matrix $F \in \mathcal{R}(G)$, respectively, for an example graph G

For a given graph G, let $\mathcal{R}(G)$ be the set of all doubly-stochastic probability matrices, such that matrices of $\mathcal{R}(G)$ assign non-zero probabilities to the matchings that are in $\mathcal{M}(G)$. We observe that, if $\mathcal{M}(G)$ contains at least two matchings, there can be uncountably infinite number of matrices in $\mathcal{R}(G)$. Still, exactly one of these matrices in $\mathcal{R}(G)$ assigns even probabilities, i.e., flat and all other matrices in $\mathcal{R}(G)$ assign truly uneven probabilities, i.e., non-flat to matchings in $\mathcal{M}(G)$. This follows from Corollary 2.6.6 in Bapat and Raghavan [26].

CHAPTER 4

UNIFIED METRIC Ψ FOR INFEASIBILITY AND PROBABILISTIC ATTACKS

In this chapter we develop a more accurate and unified metric for measuring the remaining anonymity in the aftermath of an attack. We observe the comparisons between the metric ε by Lakshmanan et al. [8] and the metric d by Edman et al. [10] presented in Section 2.5. In these comparisons, metric ε gives a better measure of remaining anonymity than metric d by considering the quality of anonymization. We develop our unified metric, which is a generalization of metric ε by Lakshmanan et al. [8] for infeasibility attack models and probabilistic attack models presented in Chapters 2 and 3 respectively.

4.1 Metric Ψ: A Generalization of Lakshmanan, Ng, and Ramesh's Metric ε

We observed, metric ε by Lakshmanan et al. [8] gives a more accurate measure considering all the matching being equally likely(i.e. for infeasibility attacks). Here, we give a more precise metric for both infieasibility and probabilistic attacks by improving the structure of metric ε.

We note that, for a graph G produced after an attack, metric $\varepsilon(G)$ gives the measure of expected cracks by calculating the average number of cracks among the matchings contained in $\mathcal{M}(G)$, i.e.,

$$\varepsilon(G) = \frac{\sum_{M \in \mathcal{M}(G)} C(M)}{|\mathcal{M}(G)|} \tag{4.1}$$

In a probabilistic attack, the crack count $C(M)$ of any matching $M \in \mathcal{M}(K_{n,n})$ is associated with a probability of M being M_0, given by a probability distribution. This probability distribution is induced by the normalized weights $\mathcal{W}(d)$ of the diagonals of the biadjacency or probability matrix resulted by an attack. We develop our metric considering this normalized weight $\mathcal{W}(d)$ of a diagonal $d \in D(P)$ be the *weight* for computing the weighted expected crack value of matchings contained in $\mathcal{M}(K_{n,n})$.

Let P be any $n \times n$ biadjacency or probability matrix resulting from an attack. The expected

crack value among all matchings in $\mathcal{M}(K_{n,n})$ after this attack is:

$$\Psi(P) = \frac{\sum_{d \in D(P)} \mathcal{W}(d).C(m(d))}{\sum_{d \in D(P)} \mathcal{W}(d)} = \sum_{d \in D(P)} \mathcal{W}(d).C(m(d)) \tag{4.2}$$

where $m(d)$ is a matching in $\mathcal{M}(K_{n,n})$ corresponding to the diagonal $d \in \mathcal{D}(P)$.

Metric Ψ is a unified metric. We observed in Section 2.4, metric ε by Lakshmanan et al. [8] is only applicable to the biadjacency matrices produced after infeasibility attacks. Out metric Ψ is applicable to both biadjacency and probability matrices produced after infeasibility and probabilistic attacks, respectively. Theorem 4.1.1 shows Ψ is a unified metric and a generalization of ε.

Theorem 4.1.1 *For any graph G with biadjacency matix A_G, and $F \in \mathcal{R}(G)$ be flat. Then, $\varepsilon(G) = \Psi(A_G) = \Psi(F)$.*

Proof The first equality is from the fact that for any $d \in D(A_G)$,
$\mathcal{W}(d) = 1/permanent(A_G) = 1/|\mathcal{M}(G)|$ if $m(d) \in \mathcal{M}(G)$ and 0 otherwise. The second equality is from the already noted fact that the normalized weight of any diagonal of A_G is equal to that of the corresponding diagonal of F. ∎

Now we give an example of the above theorem. Let G be the graph corresponding to the biadjacency matrix A at Figure 3.4(b) and we also consider the flat probability matrix P at Figure 3.3(b). Recall that, since P is a flat probability matrix, it supplies no more information to normalized weights of the diagonals, than its equivalent biadjacency matrix A. So here, $A = A_G$ and $P = F$. We compute expected crack value $\varepsilon(G)$, weighted expected cracks $\Psi(A)$ and $\Psi(P)$. We find, $\varepsilon(G) = \Psi(A) = \Psi(P) = 13/9 \approx 1.4$. With this example we observe that our metric Ψ is a unified metric as this gives the same measures as to infeasibility attack and probability attack, when the matchings contained in $\mathcal{M}(K_{n,n})$ are equally likely. Now we give an example for the case when matchings contained in $\mathcal{M}(K_{n,n})$ are not equally likely i.e. a case when probabilistic attack assigns truly uneven probabilities to the matchings. We consider the probability matrix Q at Figure 3.4(b) which has the same zero pattern to matrix A and we find $\Psi(Q) = 1369/4355 \approx$

0.3144. We also observe an interesting behavior of metric Ψ as does not always indicate a greater risk as $\Psi(Q) < \Psi(A)$. This situation is demonstrated by an attack captured by matrix Q that assigns lower probabilities to the matchings with higher crack values, i.e., matchings that are close to the actual mapping are determined less probable by the hacker. Thus, the "closeness" of the this probabilistic attack resulted correctly than infeasibility attack, therefore, metric Ψ gives a more precise anonymity measure.

4.2 Requirements of Heuristic for Ψ

Computation of metric Ψ is extremely difficult as it is a permanent based metric. The normalized weight $\mathcal{W}(d)$ of the diagonals $d \in \mathcal{D}$ are normalized by the permanent of the probability matrix P associated by the attack considered. We study the literature and find the following information about computational complexity regarding permanent values. Valiant [28] showed that, if all values of the matrix are just 0 or 1, solution to this is #P-complete. Thus, a polynomial-time solution for computing the permanent is improbable as that would imply P $=$ NP. In literature, evidence of more work is found for arriving at approximations to permanents more efficiently by Jerrum and Vazirani [29] and Chien et al. [30]. We find, the state-of-the-art polynomial-time approximation method of Jerrum et al. [31] that runs in $\mathcal{O}(t^{22})$. This provides an approximation to our metric, but infeasible to adopt in real life computations. Hence, we require to provide an efficient heuristic for a closer approximation to our metric.

4.3 Heuristic for Ψ

The metric Ψ, measures the remaining anonymity depending on the actual mapping M_0 deployed by the dataset owner. The construction of matching M_0 determines the distribution of crack values $C(M)$ of matchings contained in $\mathcal{M}(K_{n,n})$. The weight assigned to a matching $M \in \mathcal{M}(K_{n,n})$ depends on the probability assigned on its associated edges in $K_{n,n}$ and on the distribution of the cracks over $\mathcal{M}(K_{n,n})$. We let, m_i be the index of unique transformed entry in T for the recorded entry $r_i \in R$, indexed with i. We rewrite the construction of M_0 using this new

scheme of notations as,

$$M_0 = \{\langle r_1, t_{m1}\rangle, \langle r_2, t_{m2}\rangle, ..., \langle r_n, t_{mn}\rangle\}.$$

Since, P is a doubly-stochastic probability matrix, each row of P gives a probability distribution on the set T. The values $\langle P_{i,1}, P_{i,2}, ..., P_{i,n}\rangle$ of i^{th} row of P gives the probabilities assigned to each $t_j \in T$ to be anonymized representative of r_i. The value P_{i,m_i} assigned to an edge $\langle r_i, t_{m_i}\rangle \in M_0$ gives the probability of closeness between r_i and t_{m_i} determined by the hacker. This probability P_{i,m_i}, is an estimate of the contribution of entry r_i to the overall weighted expected crack value $\Psi(P)$. We take the total of the estimations for all entry indexes with $i = \{1, 2, ..., n\}$ and give our heuristic as,

$$\mathcal{H}(P) = \sum_{i=1}^{n} P_{i,m_i} \tag{4.3}$$

The heuristic at equation 4.3 is a huge improvement over $\Psi(P)$, as $\mathcal{H}(P)$ can be computed only in $\mathcal{O}(n)$ steps.

Later, we show that $\mathcal{H}(P)$ is a close approximation of $\Psi(P)$, but now, we emphasize that they are not identical. $\mathcal{H}(P)$ is, in fact, the expected crack value among all functions in R^T, which as shown in Figure 3.2, is usually a proper superset of $\mathcal{M}(K_{t,t})$. To establish this, we first give the following stronger lemma.

Lemma 1: Let for each $k \in \{1, 2, ..., n\}, \mathcal{F}_k$ denote the set of all n^k partial functions f :$\{r_1, r_2, ..., r_n\} \to T$. Then, for any M_0 and a given probability matrix P, we have that for all k, the expected weighted crack value among all partial functions in $\mathcal{F}_k$ is $\sum_{i=1}^{k} P_{i,m_i}$.

Proof: (By weak induction on k) In base case, let k = 1. $\mathcal{F}_1$ contains n partial functions, exactly 1 of which has 1 crack, as it maps r_1 to t_{m_1}. The probability of this function being chosen is P_{1,m_1}. None of the other $(n-1)$ partial functions in $\mathcal{F}_1$ has any cracks, and the total probability of choosing one of those is $(1 - P_{1,m_1})$. The expected weighted crack value among all partial functions in $\mathcal{F}_1$ is thus $1 \cdot P_{1,m_1} + 0 \cdot (1 - P_{1,m_1})$ = $\sum_{i=1}^{k} P_{i,m_i}$.

For the inductive case, we assume $2 \leq k \leq n$ and, as the inductive hypothesis, suppose the expected weighted crack value among all of the n^{k-1} partial functions in $\mathcal{F}_{k-1}$ is $\sum_{i=1}^{k-1} P_{i,m_i}$. $\mathcal{F}_k$

has n^k partial functions, exactly n^{k-1} of which, with a total probability of P_{k,m_k} , map r_k to t_{m_k}, thus adding 1 crack to each partial function in $\mathcal{F}_{k-1}$. The expected weighted crack value among these elements of $\mathcal{F}_k$ is thus $1 + \sum_{i=1}^{k-1} P_{i,m_i}$. The remaining $(n^k - n^{k-1})$ partial functions in $\mathcal{F}_k$, with a total probability of $(1 - P_{k,m_k})$, map r_k to other element of T. As r_k contributes no crack in these partial functions, their expected weighted crack value is still $\sum_{i=1}^{k-1} P_{i,m_i}$. The expected weighted crack value among all partial functions in $\mathcal{F}_k$ is thus

$$\left(1 + \sum_{i=1}^{k-1} P_{i,m_i}\right) \cdot P_{k,m_k} + \left(\sum_{i=1}^{k-1} P_{i,m_i}\right) \cdot (1 - P_{k,m_k})$$

which then simplifies to $\sum_{i=1}^{k} P_{i,m_i}$. ∎

The following theorem follows from the case k =n of the above lemma. The function C in it is extended from matchings to functions in T^R in a straightforward way.

Theorem 4.3.1 *For any M_0 and a given probability matrix P, $\mathcal{H}(P)$ is the expected weighted crack value over all functions in T^R, i.e., $\mathcal{H}(P) = \sum_{s \in \mathcal{S}(P)} w(s) \cdot C(f(s))$, where $f(s)$ is the function in T^R that corresponds to the slice s of P.*

We found it interesting to observe that while the expected weighted crack value over T^R can be computed in just linear-time, that over $\mathcal{M}(K_{t,t})$ can not be computed in even polynomial-time. Now we show that, these values are not too far apart. Thus, $\mathcal{H}(P)$ can used as a reasonable heuristic for $\Psi(P)$.

4.4 Accuracy of Heuristic $\mathcal{H}$

Number of cracks $C(f)$, contained in any function $f \in T^R$ is based on the choice of actual mapping M_0 made by dataset owner from $\mathcal{M}(K_{n,n})$. There are $n!$ possible options to choose the actual mapping M_0 from. Let, an extension of notation μ be the representation of M_0 and $C_\mu(f)$ be the number of cracks for in function f where $M_0 = \mu$, i.e., $C_\mu(f)$ gives the number of common edges contained in μ and in f. The following proposition gives the total count of cracks for any function $f \in T^R$, considering all $f \in \mathcal{M}(K_{n,n})$ being μ .

Proposition 4.4.1 *For any function $f \in T^R$, $\sum_{\mu \in \mathcal{M}(K_{n,n})} C_\mu(f) = n!$.*

Proof For a given $r \in R$, the edge $\langle r, f(r)\rangle$ is contained in exactly $1/|T| = 1/n$ of all the $n!$ matchings in $\mathcal{M}(K_{n,n})$. Thus, $\sum_{\mu\in\mathcal{M}(K_{n,n})} C_\mu(f) \sum_{r\in R} (n/n!) = |R|(n!/n) = n!$.

To observe, heuristic $\mathcal{H}(P)$ is a reasonable estimation of the metric $\Psi(P)$, a plotting for both metric Ψ and estimation $\mathcal{H}$ is given in Figure 4.1, for all $\mu \in M \in \mathcal{M}(K_{n,n})$. Here, the underlying 5×5 matrix P is the probability matrix Q of Figure 3.4(b). The metric and the estimation pair produced for each matching $M \in \mathcal{M}(K_{n,n})$ are sorted in ascending order of its metric values. Here we

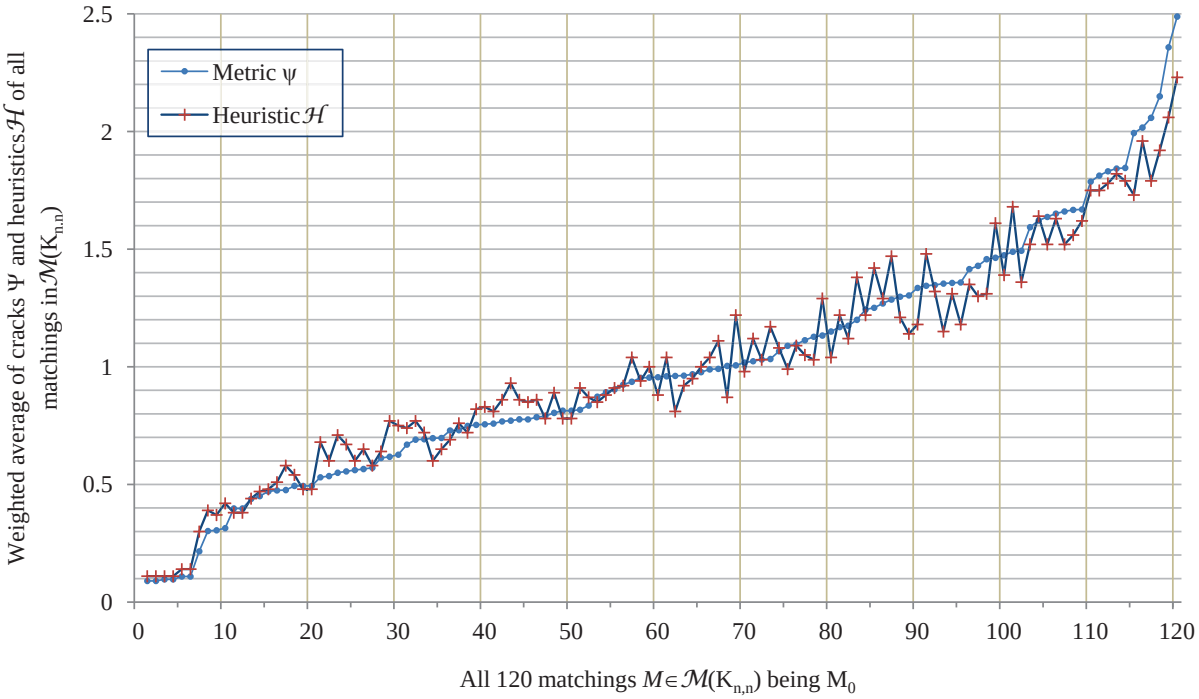


Figure 4.1: (a) Metric Ψ and heuristic $\mathcal{H}$ considering all possible $M \in \mathcal{M}(K_{n,n})$ as M_0 for a given matrix Q resulted from a probabilistic attack ; (b) The matrix Q from Figure 3.4 (b).

observe, heuristic $\mathcal{H}(P)$ produces not the exact but a very close estimation to metric $\Psi(P)$ values. This phenomenon is typical over all values of $n!$ options of μ and all $n \times n$ probability matrices. Given the scenario, a theorem proving the equality of the areas under both the curves established the faith for accuracy of heuristic $\mathcal{H}$.

Theorem 4.4.2 *For a given $n \times n$ probability matrix P,*

$$\sum_{\mu \in \mathcal{M}(K_{n,n})} \mathcal{H}_\mu(P) = \sum_{\mu \in \mathcal{M}(K_{n,n})} \Psi_\mu(P). \tag{4.4}$$

Proof The left side of equality, by Theorem 4.3.1, is $\sum_{\mu \in \mathcal{M}(K_{n,n})}(\sum_{s \in \mathcal{S}(P)} w(s) \cdot C_\mu(f(s)))$. After rearrangement, this becomes $(\sum_{s \in \mathcal{S}(P)} w(s) \cdot \sum_{\mu \in \mathcal{M}(K_{n,n})} C_\mu(f(s)))$. Proposition 4.4.1 simplifies this to $n! \sum_{s \in \mathcal{S}(P)} w(s)$, which then further reduced to, $n!$ by Proposition 3.2.1(a). The reduction of the right side is similar. By definition, it is $\sum_{\mu \in \mathcal{M}(K_{n,n})}(\sum_{d \in \mathcal{D}(P)} \mathcal{W}(d) \cdot C_\mu(m(d)))$, which is rearranged to $(\sum_{d \in \mathcal{D}(P)} \mathcal{W}(d) \cdot \sum_{\mu \in \mathcal{M}(K_{n,n})} C_\mu(m(d)))$. Again Proposition 4.4.1(a) simplifies it to $n! \sum_{d \in \mathcal{D}(p)}$, which is by definition of $\mathcal{W}$ is
$\frac{n!}{permanent(P)} \sum_{d \in \mathcal{D}(p)} w(d)$. Finally, Proposition 4.4.1(b) reduces this to $n!$. ∎

This theorem is established on the fact that the sum total of the overestimates made by the heuristic, always coincides with the sum total of its underestimates, across all possible chosen μ. While actually, the accuracy of the heuristic depends on the amount of deviation of $\mathcal{H}$ from Ψ for each μ. In literature, much work is being done for capturing this deviation in terms of error. Armstrong [32] has given a good collection of basic principles in time-series forecasting, such as the heuristic $\mathcal{H}$. Several other methods are there for similar tasks, like the Root Mean Square Error, Mean Absolute Percentage Error, Geometric Mean Absolute Relative Error, to name a few, each with its own strengths and limitations. The choice of a method, for any particular application, usually depends upon the nature of the underlying data values. Hyndman and Koehler [33], and Shcherbakov et al. [34] are some critical surveys on existing methods.

The most relevant method observed is *Normalized Mean Absolute Percentage Error* (NMAPE), which is already adopted in many other domains, e.g., the wind power forecasting application of Chen et al. [35]. For a given $n \times n$ probability matrix P, this value is given by

$$\text{NMAPE}(P) = \frac{1}{n!} \left(\sum_{\mu \in \mathcal{M}(K_{n,n})} \frac{|\mathcal{H}_\mu(P) - \Psi_\mu(P)|}{n} \right) \times 100 \tag{4.5}$$

Equation 4.5 takes absolute values of errors into account in order to prevent positive and negative error values from canceling each other out. As heuristic $\mathcal{H}$ and metric Ψ values always lie between

0 and n, the largest absolute error is n, which is employed as the normalizing factor to standardize to the scale of 0 to 1. The summation over all μ and division by $n!$ result in averaging these values and, finally, multiplication by 100 expresses this average absolute error on an intuitive percentage scale. NMAPE(P) thus gives the average-case absolute error for P as a percentage of the worst-case. Its values close to 0% indicate accurate computation by the heuristic, while those close to 100% indicate inaccurate computation. Another noteworthy observation from Figure 4.2

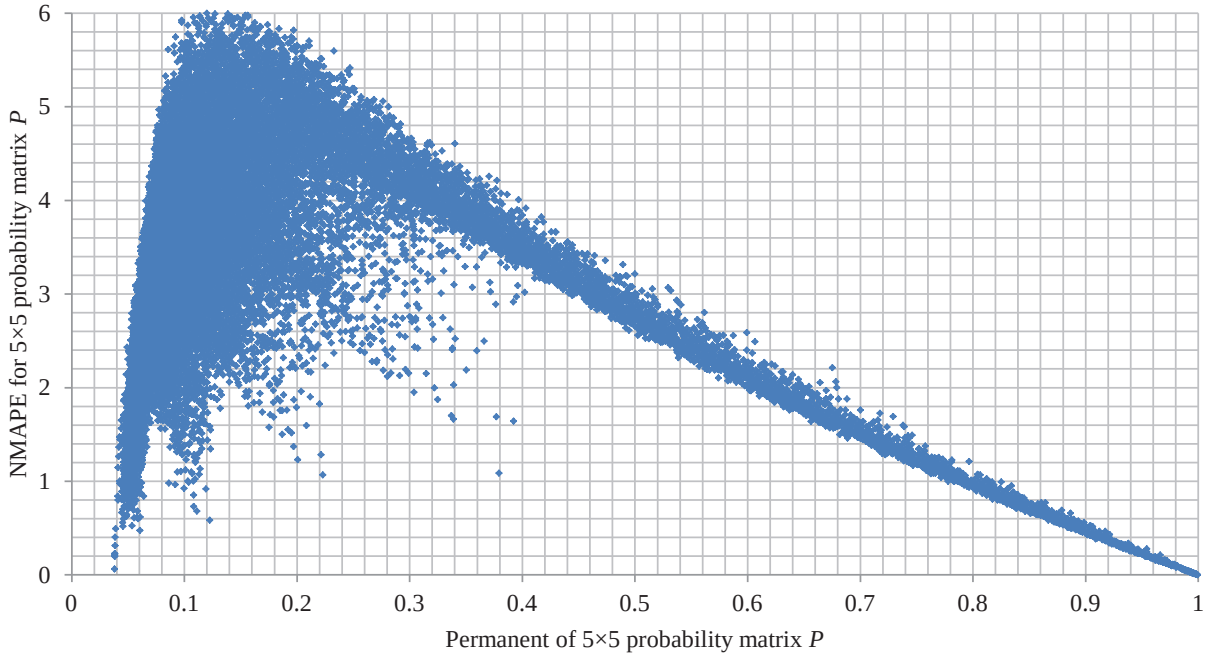


Figure 4.2: NMAPE(P) for 30,000 randomly generated 5×5 doubly stochastic probability matrix P, plotted against $permanent(P)$.

is that the higher values of NMAPE(P) occurs with matrices with lower $permanent(P)$, i.e., estimation of weaker attacks gives a higher range of errors compared to stronger attacks with higher $permanent(P)$. This characteristics makes the heuristic more acceptable to the dataset owner for decision making.

The permanent of a $n \times n$ probability matrix P is a real value in the range $[n!/n^n, 1]$ and, it can be easily shown that in the extreme cases, i.e., when $permanent(P)$ is either $n!/n^n$ or

1, NMAPE(P) is 0%. To get a better picture of the distribution of NMAPE(P) values over the uncountably innite space of all probability matrices, we randomly generated $30,000$ such matrices of size 5×5. Figure 4.2 plots the NMAPE(P) versus $permanent(P)$ values of these random matrices, and shows that NMAPE(P) is just within 6% which indicates a fairly high degree of heuristic accuracy.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

We present a new metric for measuring the risk of disclosing anonymized datasets, after any probabilistic attack. The previously proposed metrics are limited in their applicability to just the infeasibility attacks, which are a subclass of probabilistic attacks. Our metric is a unified metric as this measures for both even and uneven probability distributions over all possible matchings, produced by infeasibility and probability attacks. This metric considers the real valued probabilities of all matchings of being the actual mapping adopted for anonymization and delivers a more practical risk assessment for decision makers. The dataset owners can assess the risk of disclosure for a given attack, if the measure produced by the metric is lower than a pre-defined level of safely. Similar to existing metrics, we observe that the hardness of computational complexity remains with our permanent based metric. This helps us realize the severe requirement of an efficient heuristic for estimating our metric. We propose our heuristic as a close estimation of our metric and confer a detailed analysis of its accuracy.

We propose three interesting directions for future work following the contributions of this thesis.

1. We intend to extend our metric for attacks on frequent set mining. Currently, our metric focuses on attacks for re-identifying individual anonymized entries of the recorded dataset. In case of re-identifying a set of entries, our metric will result in indicating minimal anonymity remaining on the subsets. Here, a subset of recorded entries mapped to a subset of transformed entries represents a successful de-anonymization, while the individual entries remain anonymous inside the anonymity subset. An extension of our metric capturing the remaining anonymity of anonymity subsets after an attacks may produce an effective metric for datasets with categorical subsets of entries.

2. We observed that the maximum amount of deviation between the metric Ψ and its heuristic $\mathcal{H}$ decreases with the increase in dataset size. In this thesis, we have measured the amount of deviation between the measures by the metric Ψ and its estimation $\mathcal{H}$ in terms of Normalized Mean Absolute Percentage Error (NMAPE). An experiment on producing the deviations for larger datasets resulted in smaller range of NMAPE for a set of randomly chosen probability matrices P. We conjecture a decreasing trend of maximum of NMAPE(P) values, given an increase in the anonymity set size, and plan to investigate that this is in fact always the case.

3. We plan to generalize the construction of our heuristic $\mathcal{H}$ which is independent of the underlying attack matrix P. Currently, the construction of metric Ψ is independent of matrix P, as metric Ψ can compute using any probability distribution over the matchings in $\mathcal{M}(K_{n,n})$. But its heuristic $\mathcal{H}$, is dependent on the probability values of underlying attack matrix P. Recall that $\mathcal{R}(K_{n,n})$ is the set of all (doubly-stochastic) probability matrices that assign real-valued probabilities to all the matchings that are in $\mathcal{M}(K_{n,n})$ (i.e., the set of all possible matching for a dataset with n entries). We let, $\eta(K_{n,n})$ be the set of all possible probability distributions over $\mathcal{M}(K_{n,n})$ and $\zeta(\mathcal{R}(K_{n,n}))$ be the set of all possible distributions induced by the probability matrices in $\mathcal{R}(K_{n,n})$. Though, the elements of both $\eta(K_{n,n})$ and $\zeta(\mathcal{R}(K_{n,n}))$ are uncountably infinite, $\zeta(\mathcal{R}(K_{n,n})) \subset \eta(K_{n,n})$, i.e., the set of probability distributions induced by all possible probability matrices is a proper subset of all possible probability distributions. As heuristic $\mathcal{H}$ is dependent on the values of matrix P, the computation of this estimation is limited to the distributions only contained in $\zeta(\mathcal{R}(K_{n,n}))$. We plan to extend our heuristic to all the distributions contained in $\eta(K_{n,n})$, that would be an estimation for Ψ, given any probability distribution over the matchings in $\mathcal{M}(K_{n,n})$. A generalized heuristic, independent of P, will be more general, as this will be for all possible probability distributions.

REFERENCES

LIST OF REFERENCES

[1] Mark Grechanik, Christoph Csallner, Chen Fu, and Qing Xie. Is data privacy always good for software testing? *Software Reliability Engineering (ISSRE), 2010 IEEE 21st International Symposium on*, pages 368–377. IEEE, 2010.

[2] James H Moor. Towards a theory of privacy, in the information age. *Computers and Society*, 27(3):27–32, 1997.

[3] Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar. Random-data perturbation techniques and privacy-preserving data mining. *Knowledge and Information Systems*, 7(4):387–414, 2005.

[4] Ramakrishnan Srikant. Privacy preserving data mining: challenges and opportunities. *Advances in Knowledge Discovery and Data Mining*, pages 13–13. Springer, 2002.

[5] George J Annas. HIPAA regulations-a new era of medical-record privacy? *New England Journal of Medicine*, 348(15):1486–1490, 2003.

[6] Alan G Konheim. *Cryptography, a primer*. John Wiley & Sons, Inc., 1981.

[7] Laks VS Lakshmanan, Raymond T Ng, and Ganesh Ramesh. To do or not to do: the dilemma of disclosing anonymized data. *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 61–72. ACM, 2005.

[8] Laks VS Lakshmanan, Raymond T Ng, and Ganesh Ramesh. On disclosure risk analysis of anonymized itemsets in the presence of prior knowledge. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(3):13, 2008.

[9] Andreas Pfitzmann and Marit Hansen. Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management-a consolidated proposal for terminology. *Version v0*, 31:15, 2008.

[10] Matthew Edman, Fikret Sivrikaya, and Bülent Yener. A combinatorial approach to measuring anonymity. *Intelligence and Security Informatics, 2007 IEEE*, pages 356–363. IEEE, 2007.

[11] Richard A Moore Jr. Controlled data-swapping techniques for masking public use microdata sets. *Statistical Research Division Report Series*, pages 96–04, 1996.

[12] Stephen E Fienberg, Udi E Makov, and RJ Steel. Disclosure limitation using perturbation and related methods for categorical data. *JOURNAL OF OFFICIAL STATISTICS-STOCKHOLM-*, 14:485–502, 1998.

LIST OF REFERENCES (continued)

[13] J Domingo-Feffer, Anna Oganian, and Vicenç Torra. Information-theoretic disclosure risk measures in statistical disclosure control of tabular data. *Proceedings of the 14th International Conference on Scientific and Statistical Database Management, 2002.*, pages 227–231. IEEE, 2002.

[14] Alexandre Evfimievski, Ramakrishnan Srikant, Rakesh Agrawal, and Johannes Gehrke. Privacy preserving mining of association rules. *Information Systems*, 29(4):343–364, 2004.

[15] Vassilios S Verykios, Ahmed K Elmagarmid, Elisa Bertino, Yücel Saygin, and Elena Dasseni. Association rule hiding. *Knowledge and Data Engineering, IEEE Transactions on*, 16(4):434–447, 2004.

[16] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. *ACM Sigmod Record*, 29(2):439–450, 2000.

[17] Dakshi Agrawal and Charu C Aggarwal. On the design and quantification of privacy preserving data mining algorithms. *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 247–255. ACM, 2001.

[18] Nabil R Adam and John C Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys (CSUR)*, 21(4):515–556, 1989.

[19] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, Technical report, SRI International, 1998.

[20] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[21] Gagan Aggarwal, Tomás Feder, Krishnaram Kenthapadi, Rajeev Motwani, Rina Panigrahy, Dilys Thomas, and An Zhu. Anonymizing tables. *Database Theory-ICDT 2005*, pages 246–258. Springer, 2005.

[22] Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. *Privacy Enhancing Technologies*, pages 41–53. Springer, 2003.

[23] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. *Privacy Enhancing Technologies*, pages 54–68. Springer, 2003.

[24] Armen S Asratian. *Bipartite graphs and their applications*. Number 131. Cambridge University Press, 1998.

LIST OF REFERENCES (continued)

[25] G P Egorychev. The solution of van der waerden's problem for permanents. *Advances in Mathematics*, 42(3):299–305, 1981.

[26] Ravindra B Bapat and Tirukkannamangai Echambadi Srinivasa Raghavan. *Nonnegative matrices and applications*, volume 64. Cambridge University Press, 1997.

[27] Rajiv Bagai, Huabo Lu, Rong Li, and Bin Tang. An accurate system-wide anonymity metric for probabilistic attacks. *Privacy Enhancing Technologies*, pages 117–133. Springer, 2011.

[28] Leslie G Valiant. The complexity of computing the permanent. *Theoretical computer science*, 8(2):189–201, 1979.

[29] Mark Jerrum and Umesh Vazirani. A mildly exponential approximation algorithm for the permanent. *Algorithmica*, 16(4-5):392–401, 1996.

[30] Steve Chien, Lars Rasmussen, and Alistair Sinclair. Clifford algebras and approximating the permanent. *Journal of Computer and System Sciences*, 67(2):263–290, 2003.

[31] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM (JACM)*, 51(4):671–697, 2004.

[32] Jon Scott Armstrong. *Principles of forecasting: a handbook for researchers and practitioners*, volume 30. Springer, 2001.

[33] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

[34] Maxim Vladimirovich Shcherbakov, Adriaan Brebels, Nataliya Lvovna Shcherbakova, Anton Pavlovich Tyukov, Timur Alexandrovich Janovsky, and Valeriy Anatolevich Kamaev. A survey of forecast error measures. 2013.

[35] Niya Chen, Zheng Qian, Xiaofeng Meng, and Ian T Nabney. Short-term wind power forecasting using gaussian processes. *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2790–2796. AAAI Press, 2013.