High-Performance Algorithms for Privacy-Preserving Mining

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

Master of Engineering

IN

COMPUTER SCIENCE AND ENGINEERING

by

Shipra Agrawal

Computer Science and Automation

Indian Institute of Science

BANGALORE – 560 012

July 2004

TO

my parents

for their unconditional love and support

Acknowledgements

I am deeply indebted to my supervisor Prof. Dr. J. R. Haritsa whose help, suggestions and encouragement helped me in all the time of research for and writing of this thesis. I learned many things from him.

I thank Prof. Dr. R. Vittal Rao for his help, interest and valuable suggestions on this work. My colleagues in Database Systems lab were very amiable and supportive. I thank them for providing a positive work environment.

My colleague and friend Emtiyaz deserves special thanks and gratitude from me. His support and valuable suggestions kept me motivated throughout the work. Stimulating discussions with him honed my interest and understanding of various topics related to the work. I thank my friends Niranjan, Sujit, Kalyan, Anoop for their moral support and encouragement throughout my stay at IISc. I also thank Piyush, Shilpi, Shweta, who were not there with me physically at IISc, but always made me feel their supportive presence.

I would like to give special thanks to my parents whose patient love and support enabled me to do this work.

Publications based on this Thesis

1. S. Agrawal, V. Krishnan, J. R. Haritsa, “On Addressing Efficiency concerns in Privacy Preserving Data Mining”, 9th International Conference on Database Systems For Advanced Applications (DASFAA) 2004, Jeju Island, Korea

2. S. Agrawal and J. Haritsa, “A Framework for High-Accuracy Privacy-Preserving Mining”, Tech. Rep. TR-2004-02, DSL/SERC, Indian Institute of Science, 2004. http://dsl.serc.iisc.ernet.in/pub/TR/TR-2004-02.pdf

Abstract

Data mining services require accurate input data for their results to be meaningful, but privacy concerns may influence users to provide spurious information. To preserve client privacy in the data mining process, a variety of techniques based on random perturbation of data records have been proposed recently.

However, mining the perturbed database can be orders of magnitude more time-consuming as compared to mining the original database. In this work, we address this issue and demonstrate that by (a) generalizing the distortion process to perform symbol-specific distortion, (b) appropriately choosing the distortion parameters, and (c) applying a variety of optimizations in the reconstruction process, runtime efficiencies that are well within an order of magnitude of undistorted mining can be achieved.

Also, we present a generalized matrix-theoretic model of random perturbation, which facilitates a systematic approach to the design of perturbation mechanisms for privacy-preserving mining. Specifically, we demonstrate that (a) the prior techniques differ only in their settings for the model parameters, and (b) through appropriate choice of parameter settings, we can derive new perturbation techniques that provide highly accurate mining results even under strict privacy guarantees. We also propose a novel perturbation mechanism wherein the model parameters are themselves characterized as random variables, and demonstrate that this feature provides significant improvements in privacy at a very marginal cost in accuracy.

While our model is valid for random-perturbation-based privacy-preserving mining in general, we specifically evaluate its utility here with regard to frequent-itemset mining on a variety of real datasets. The experimental results indicate that our mechanisms incur substantially lower identity and support errors as compared to the prior techniques.

Contents

List of Tables

List of Figures

Notation and Abbreviations

ρ	Support Error
σ^+	Identity Error : false positives
σ^-	Identity Error : false negatives
$\tilde{A}$	transition probability matrix as a matrix of random variables
A	transition probability matrix
A_i	value of i^{th} row of matrix A
A_{ij}	value of i^{th} row and j^{th} column of matrix A
C_i	i^{th} client
$E(a)$	Expectation of a
I_U	Index set of set S_U
I_V	Index set of set S_V
M	number of attributes in mining dataset
M_b	number of boolean attributes corresponding to M categorical attributes
N	number of records in mining dataset
s_j	support of j^{th} item

S_U	domain of records in dataset U
S_V	domain of records in dataset V
S_U^j	domain of j^{th} attribute in dataset U
S_V^j	domain of j^{th} attribute in dataset V
U	original dataset of user records
U_i	i^{th} record of dataset U
U_{ij}	j^{th} attribute of i^{th} record of U
V	original dataset of user records
V_i	i^{th} record of dataset V
V_{ij}	j^{th} attribute of i^{th} record of V
$Var(a)$	Variance of a
con_{min}	minimum confidence threshold for association rule mining
sup_{min}	minimum support threshold for association rule mining
BP	Basic Privacy
RP	Reinterrogation privacy
SP	Strict Privacy

Chapter 1

Introduction

1.1 Privacy Concerns in Data Mining

The knowledge models produced through data mining techniques are only as good as the accuracy of their input data. One source of data inaccuracy is when users, due to privacy concerns, deliberately provide wrong information.

The compulsion for doing so may be the (perhaps well-founded) worry that the requested information may be misused by the service provider to harass the customer. As a case in point, consider a pharmaceutical company that asks clients to disclose the diseases they have suffered from in order to investigate the correlations in their occurrences – for example, "Adult females with malarial infections are also prone to contract tuberculosis". While the company may be acquiring the data solely for genuine data mining purposes that would eventually reflect itself in better service to the client, at the same time the client might worry that if her medical records are either inadvertently or deliberately disclosed, it may adversely affect her employment opportunities.

With the dramatic increase in digital data, concerns about informational privacy have emerged globally [17] [18] [27] [34]. Privacy issues are further exacerbated now that the World Wide Web makes it easy for the new data to be automatically collected and added to databases [13] [14] [23] [39] [40] [41].

1.2 Existing Techniques for Privacy-Preserving Mining

To encourage users to submit correct inputs, a major research in recent years in data mining has been for the development of techniques that incorporate privacy concerns. Specifically, such techniques address the following question. Since the primary task in data mining is the development of models about aggregated data, can we develop accurate models without access to precise information in individual data records? [3]

In a nutshell, the privacy preserving mining methods modify the original data in some way, so that the privacy of the user data is preserved and at the same time the mining models can be reconstructed from the modified data with reasonably accuracy. Various approaches have been proposed in the existing literature for privacy-preserving data mining which differ with respect to their assumptions of data collection model and user privacy requirements.

The perturbation approach used in *random perturbation model* works under the strong privacy requirement that even the dataset forming server is not allowed to learn or recover precise records. Users trust nobody and perturb their record at their end before providing it to any other party. In the pioneering work of [3], privacy-preserving data classifiers based on adding random noise to the record values were proposed. This work was extended in [7] and [25] to address a variety of subtle privacy loopholes. New randomization operators for maintaining data privacy were presented and analyzed in [19, 28]. These methods are for categorical/boolean data and are based on probabilistic mapping from domain space to the range space rather than by incorporating additive noise to continuous valued data. A theoretical formulation of privacy breaches for such methods and a methodology for limiting them were given in the foundational work of [20].

Another model of privacy preserving data mining is *k-anonymity model* [33]. k-anonymity model[33] does not satisfy the strong privacy requirements of random perturbation model. The *condensation approach* discussed in [10] also requires the relaxation of the assumption that even the data forming server is not allowed to learn or recover records, as in k-anonymity model.

[4, 5, 6, 26] deal with *Hippocratic databases* which are the database systems that take responsibility of the privacy of data they manage. It involves specification of how the data is

to be used in a privacy policy and enforcing limited disclosure rules.

Maintaining input data privacy is considered in [24, 35, 36, 37] in the context of databases that are *distributed* across a number of sites with each site only willing to share data mining results, but not the source data.

Finally, [11, 16, 29, 30] addresses the problem of how to prevent *sensitive rules* from being inferred by the data miner – it addresses concerns about *output* privacy, rather than the privacy of the *input* data.

1.3 Our Contributions

Our work is in context of *random perturbation model* for privacy preserving mining. In this context we make the following contributions :

Efficient privacy-preserving mining While various schemes for privacy-preserving mining have been proposed for simultaneously achieving privacy of user data and reasonable accuracy of mining results, a problem left unaddressed was characterizing the *runtime efficiency* of mining the distorted data as compared to directly mining the original data. Our subsequent analysis has shown that this issue is indeed a *major concern*: Specifically, taking MASK algorithm [8] as our base for analysis, we have found that on typical market-basket databases, privacy-preserving mining can take as much as *two to three orders of magnitude* more time as compared to direct mining. In this work, we address this runtime efficiency issue in privacy-preserving association rule mining, which to the best of our knowledge has never been previously considered in the literature. We demonstrate that it is possible to bring the efficiency to *well within an order of magnitude* with respect to direct mining, while retaining satisfactory privacy and accuracy levels. This improvement is achieved through changes in both the distortion process and the mining process of MASK, resulting in a new algorithm that we refer to as EMASK. Our new design is validated against a variety of synthetic and real datasets.

Generalized framework for strict privacy-preserving mining The trend in the prior literature has been to propose *specific* perturbation techniques, which are then analyzed for their

privacy and accuracy properties. We move on, in this work, to presenting FRAPP (FRamework for Accuracy in Privacy-Preserving mining), a generalized matrix-theoretic *framework* that facilitates a systematic approach to the design of random perturbation schemes for privacy-preserving mining. FRAPP quantitatively characterizes the *sources of error* in random data perturbation and model reconstruction processes. While various privacy metrics have been discussed in the literature, FRAPP supports a particularly strong notion of privacy, originally proposed in [20]. Specifically, it supports a measure called "amplification", which guarantees strict limits on privacy breaches of individual user information, *independent of the distribution of the original data*. We demonstrate that various prior techniques for random perturbation fall into this framework differing only in their settings of the FRAPP parameters.

New perturbation mechanisms We demonstrate that through appropriate choice of FRAPP parameter settings, new perturbation techniques can be constructed that provide highly accurate mining results even under strict privacy guarantees. In particular, we show that our choice of parameters is optimal for minimizing estimation error under given conditions. Efficient implementations for these new perturbation techniques are also presented.

Novel random parameter approach We investigate here, for the first time, the possibility of *randomizing the perturbation parameters themselves*. The motivation is that it could lead to an increase in privacy levels since the exact parameter values used by a specific client will not be known to the data miner. This scheme has the obvious downside of perhaps reducing the model reconstruction accuracy. However, our investigation shows that the trade-off is very attractive in that the privacy increase is substantial whereas the accuracy reduction is only marginal. This opens up the possibility of using FRAPP in a *two-step* process: First, given a user-desired level of privacy, identifying the deterministic values of the FRAPP parameters that both guarantee this privacy and also maximize the accuracy; and then, (optionally) randomizing these parameters to obtain even better privacy guarantees at a minimal cost in accuracy.

Empirical evaluation of new mechanisms We specifically evaluate the performance of our new perturbation mechanisms on the popular mining task of finding frequent itemsets, the cornerstone of association rule mining. Here, we focus on its applications to *categorical databases*, where the domain of each attribute is finite. Note that boolean data is a special case of this class, and further, that continuous-valued attributes can be converted into categorical attributes by partitioning the domain of the attribute into fixed length intervals. Our evaluation on a variety of real datasets shows that both identity and support errors are substantially lower than those incurred by the prior privacy-preserving techniques.

1.4 Organization of the report

The report is organized as follows :

In Chapter 2 we give the basic notions of privacy and accuracy quantifications used in this report.

Chapter 3 addresses the efficiency issues in MASK algorithm for privacy preserving mining. Section 3.1 provides background information about the original MASK algorithm. Section 3.2 explains the efficiency concerns in MASK. Then, in Section 3.3, we present the details of our new EMASK(Efficient-MASK) privacy-preserving scheme. The performance model and the experimental results are highlighted in Sections 3.4 and 3.5, respectively.

Chapter 4 deals with the framework for random perturbation schemes under strict privacy guarantees. The FRAPP framework for data perturbation and model reconstruction is presented in Section 4.1. Appropriate choices of FRAPP parameters for simultaneously guaranteeing strict data privacy and providing high model accuracy are discussed in Section 4.2. The impact of randomizing the FRAPP parameters is investigated in Section 4.3. Efficient schemes for implementing the new perturbation mechanisms are described in Section 4.4. In Section 4.5, we discuss the application of our mechanisms to association rule mining. Then, in Section 4.6, the utility of FRAPP in the context of association rule mining is empirically investigated.

Finally, in Chapter 5, we summarize the conclusions of our study and outline future research avenues.

Chapter 2

Basic Notions

2.1 Data Collection Model

Suppose there are N clients $C_1, \ldots, C_N$ connected to the server. Each client C_i contributes a tuple U_i of M attributes to the mining dataset U . We denote the j^{th} attribute of this tuple as U_{ij}. Each client perturbs her tuple before submitting it to the server in order to preserve its privacy. The modified dataset formed as a result of perturbation of original records is denoted by V. MASK [28] and subsequently the EMASK algorithm proposed by us assume the dataset U and V to be boolean, i.e. each attribute is a boolean attribute with only two possible values '0' and '1' resulting in a tuple of 0's and 1's from each client. FRAPP provides a more generalized framework with each attribute j being a categorical attribute with domain S_U^j and S_V^j respectively for original and perturbed datasets.

2.2 Privacy Quantification

To measure the extent to which a given privacy-preserving mining is able to keep the private information of the user private, various quantifications of privacy have been proposed.

Basic Privacy (BP)

This notion of privacy had been proposed in [28] in context of boolean database. The *basic privacy* measures the probability that given a random customer C_i who bought an item j, her original entry for j, i.e. '1', can be accurately reconstructed from the distorted entry, *prior* to the mining process. We can calculate this privacy in the following manner, Let U_{ij} be the original entry corresponding to item j in a tuple U_i and V_{ij} be its distorted entry. The reconstruction probability of a '1' is given by

$$\begin{aligned} \mathcal{R}_1 &= \Pr(V_{ij}=1|U_{ij}=1)\Pr(U_{ij}=1|V_{ij}=1) + \Pr(V_{ij}=0|U_{ij}=1)\Pr(U_{ij}=1|V_{ij}=0) \\ &= \frac{\Pr(V_{ij}=1|U_{ij}=1)^2\Pr(U_{ij}=1)}{\Pr(V_{ij}=1)} + \frac{\Pr(V_{ij}=0|U_{ij}=1)^2\Pr(U_{ij}=1)}{\Pr(V_{ij}=0)} \end{aligned} \tag{2.1}$$

The Basic privacy offered to 1's is simply $100(1-\mathcal{R}_1)$.

Reinterrogation privacy (RP)

Reinterrogation privacy takes into account the reduction in privacy due to the knowledge of the *output* of the mining process – namely, the association rules, or equivalently, the support of frequent itemsets [2]. Privacy breach due to reinterrogation stems from the fact that an individual entry in the distorted database may not reveal enough information to reconstruct it, but on seeing a long frequent itemset in the distorted database and knowing the distorted and original supports of the itemset one may be able to predict the presence of an item of the itemset with more certainty. As an example situation, suppose the reconstructed support of a 3-itemset present in a transaction distorted with flipping probability $p = 0.1$ [28], is 0.01. Then the probability of this 3-itemset coming from a '000' in the original transaction is as low as $0.1 * 0.1 * 0.1 * 0.99 = 0.00099$. Thus, with the knowledge of support of higher length itemsets the miner can predict the presence of an item of the itemset in the original transaction with higher probability.

Our method of estimating reinterrogation privacy breach is based on that described in [19] for computing privacy breaches. We make the *extremely conservative* assumption here that the data miner is able to accurately estimate the *exact* supports of the original database during the

mining process. The method of calculating reinterrogation privacy is as follows:

- First, for each item that is part of a frequent itemset, we compare the support of the frequent itemset in the distorted data with the support of the singleton item in the original data. For example, let an itemset $\{$a,b,c$\}$ occur a hundred times in the randomized data and let the support of the item b in the corresponding hundred transactions of the original database be twenty. We then say that the itemset $\{$a,b,c$\}$ caused a 20% privacy breach for item b due to reinterrogation, since for these hundred distorted transactions, we estimate with 20% confidence that the item b was present in the original transaction.

- Then, we estimate the privacy of each '1' appearing in a frequent item column in the original database. There are two cases: 1's that have remained 1's in the distorted database, and 1's that have been flipped to 0's. For the former, the privacy is estimated with regard to the worst of the privacy breaches (computed as discussed above) among all the frequent itemsets of which it is a part and which appear in the distorted version of this transaction. For example, for an item b in the original transaction $\{$a,b,c,d,e$\}$, the privacy breach of b is the worst of the privacy breaches due to all frequent itemsets which are subsets of $\{$a,b,c,d,e$\}$ and contain b.

  In the latter (flip) case, the privacy is set equal to the average Basic Privacy value – this is because for the flipping probability values for '0' that we consider in this work, which are close to 0.0 as discussed later, most of the large number of 0's in the original sparse matrix remain '0', so a '1' flipping to a '0' is resistant to privacy breaches due to reinterrogation.

- Next, for the 1's present in in-frequent columns, the privacy is estimated to be the Basic Privacy since these privacies are not affected by the discovery of frequent itemsets.

- Finally, the above privacy estimates are averaged over all 1's in the original database to obtain the overall average reinterrogation privacy.

Strict Privacy (SP)

Strict privacy refers to a worst case notion of privacy given by [20]. It gives a criterion that should be satisfied by any privacy preserving algorithm in order to ensure privacy guarantees independent of the distribution of data.

A *privacy breach* is defined as a situation when, for some client, the disclosure of its randomized private information to the server reveals that a certain property of client's private information holds with high probability [19]. *Prior probability* is the likelihood of the property in the absence of any knowledge about private information; *posterior probability* is the likelihood of the property given the randomized value. A $\psi_1 - to - \psi_2$ privacy breach is said to occur if for some client, a property with prior probability less than ψ_1 has posterior probability equal to or more than ψ_2. Thus a privacy preserving technique is said to have *strict privacy* (ψ_1, ψ_2) if it guarantees that no $\psi_1 - to - \psi_2$ breaches could occur, regardless of the prior distribution.

[20] also gives conditions under which a privacy preserving technique can provide such strict (ψ_1, ψ_2) privacy guarantees.

The algorithms discussed in Chapter 3 are designed to satisfy Basic Privacy and Reinterrogation privacy requirements. Chapter 4 discusses algorithms which perform with reasonable accuracy even with Strict Privacy guarantees.

2.3 Mining Objectives

As most of the study carried out in this work has been evaluated in context of the popular mining task of association rule mining, we present in this section a brief background on association rule mining objectives.

In association rule mining task, the goal of the miner is to compute *association rules* on the above database. Denoting the set of transactions in the database by $\mathcal{T}$ and the set of items in the database by $\mathcal{I}$, an association rule is a (statistical) implication of the form $\mathcal{X} \Longrightarrow \mathcal{Y}$, where $\mathcal{X}, \mathcal{Y} \subset \mathcal{I}$ and $\mathcal{X} \cap Y = \phi$. A rule $\mathcal{X} \Longrightarrow \mathcal{Y}$ is said to have a *support* (or frequency) factor *s* iff at least $s\%$ of the transactions in $\mathcal{T}$ satisfy $\mathcal{X} \cup \mathcal{Y}$. A rule $\mathcal{X} \Longrightarrow \mathcal{Y}$ is satisfied in the set of

transactions $\mathcal{T}$ with a *confidence* factor *c* iff at least $c\%$ of the transactions in $\mathcal{T}$ that satisfy $\mathcal{X}$ also satisfy $\mathcal{Y}$. Both support and confidence are fractions in the interval [0,1]. The support is a measure of statistical significance, whereas confidence is a measure of the strength of the rule.

A rule is said to be "interesting" if its support and confidence are greater than user-defined thresholds sup_{min} and con_{min}, respectively, and the objective of the mining process is to find all such interesting rules. It has been shown in [1] that achieving this goal is effectively equivalent to generating all subsets $\mathcal{X}$ of $\mathcal{I}$ that have support greater than sup_{min} – these subsets are called *frequent* itemsets. Therefore, the mining objective is, in essence, to efficiently discover all frequent itemsets that are present in the database.

2.3.1 Mining Accuracy Quantification

Privacy-preserving association rule mining aims at preserving privacy of data while allowing the miner to discover the frequent itemsets with reasonable accuracy through reconstruction. But, Since random-perturbation-based privacy preserving techniques take a *probabilistic* approach, fundamentally we cannot expect the reconstructed support values to coincide exactly with the actual supports. This means that we may have errors in the estimated supports of frequent itemsets with the reported values being either larger or smaller than the actual supports.

Errors in support estimation can have an even more pernicious effect than just wrongly reporting the support of a frequent itemset. They can result in errors in the *identities* of the frequent itemsets. That is, we can encounter both *false positives* and *false negatives*.

Thus association rule mining errors can be quantified in terms for two metrics : Support Error and Identity Error [28]

Support Error (ρ) :

This metric reflects the (percentage) average relative error in the reconstructed support values for those itemsets that are correctly identified to be frequent. Denoting the number of frequent itemsets by $|f|$, the reconstructed support by rec_sup and the actual support

by act_sup, the support error is computed over all frequent itemsets as

$$\rho = \frac{1}{\mid f \mid} \Sigma_f \frac{\mid rec\_sup_f - act\_sup_f \mid}{act\_sup_f} * 100$$

Identity Error (σ) :

This metric reflects the percentage error in identifying frequent itemsets and has two components: σ^+, indicating the percentage of false positives, and σ^- indicating the percentage of false negatives. Denoting the reconstructed set of frequent itemsets with R and the correct set of frequent itemsets with F, these metrics are computed as:

$$\sigma^+ = \frac{|R-F|}{|F|} * 100 \qquad \sigma^- = \frac{|F-R|}{|F|} * 100$$

Chapter 3

Efficiency Concerns in Privacy-Preserving Mining

3.1 Background on MASK algorithm

We present here background information on the MASK algorithm, which we recently proposed in [28] for providing acceptable levels of both privacy and accuracy.

Given a customer tuple with 1's and 0's, the MASK algorithm has a simple distortion process: Each item value (i.e. 1 or 0) is either kept the same with probability p or is flipped with probability $1 - p$. All the customer tuples are distorted in this fashion and make up the database supplied to the miner – in effect, the miner receives a *probabilistic function* of the true customer database. For this distortion process, the Basic Privacy for 1's was computed to be

$$BP = 100(1 - \mathcal{R}_1(p)) = 100(1 - \frac{s_0 \times p^2}{s_0 \times p + (1-s_0) \times (1-p)} - \frac{s_0 \times (1-p)^2}{s_0 \times (1-p) + (1-s_0) \times p}) \qquad (3.1)$$

where s_0 is the average support of individual items in the database and p is the distortion parameter mentioned above.

Since the privacy graph as a function of p has a large range where it is almost constant, it means that we have considerable flexibility in choosing the p value – in particular, we can

choose it in a manner that will *minimize the error* in the subsequent mining process. Specifically, the experiments in [28] showed that choosing $p = 0.9$ (or, equivalently, $p = 0.1$, since the graph is symmetric about $p = 0.5$) was most conducive to accuracy.

The concept of Re-interrogated Privacy was not considered in [28]; we include it in this paper and compute both Basic Privacy and Re-interrogated Privacy for EMASK.

3.2 Efficiency Concerns

While MASK is successful in achieving the dual objectives of privacy and accuracy, its runtime efficiency proves to be rather poor. For example, Figure 3.1 shows the running time (on log scale) of MASK, implemented as an extension to Apriori[2], as compared to Apriori[2] itself, for various settings of the minimum support parameter.The graph in Figure 3.1 shows that there are huge differences in running times of the two algorithms – specifically, mining the distorted database can take as much as *two to three orders of magnitude* more time than mining the original database. Obviously, such enormous overheads make it difficult for MASK to be viable in practice.

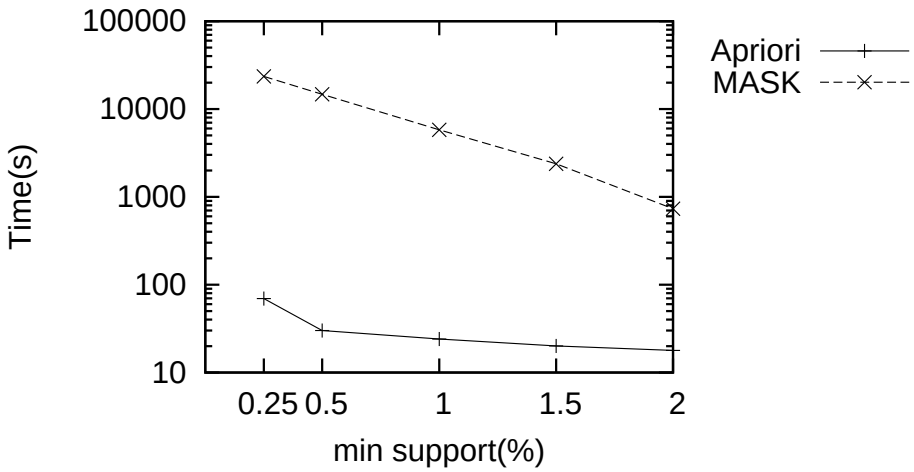


Figure 3.1: Comparison of run time of Apriori and MASK (log scale)

The reason for the huge overheads are the following:

Increased database density: This overhead is inherent to the methods employing random

distortion method to achieve privacy. The random perturbation methods flip 0's to 1's to hide the original 1's. Due to the generation of false 1's, the *density* of the database matrix is increased considerably. For example, given a supermarket database with an average transaction length of 10 over a 1000 item inventory, a flipping probability as low as 0.1 increases the average transaction length by an order of magnitude, i.e. from 10 to 108. The reason that flipping of true 1's to false 0's cannot compensate for this increase is that the datasets we are considering here are sparse databases, with the number of 0's orders of magnitude larger than the number of 1's. Hence the effect of flipping 0's highly dominates the effect of flipping 1's.

As a result of increased transaction lengths, counting the itemsets in the distorted database requires much more processing as compared to the original database, substantially increasing the mining time. In [20], a technique for compressing large transactions is proposed – however, its utility is primarily in reducing storage and communication cost, and not in reducing the mining runtime since the compressed transactions have to be decompressed during the mining process.

Counting Overhead: On distortion, a k-itemset may produce any of 2^k combinations. For example, a '11' may distort to '00','01','10' or '11'. In order to accurately reconstruct the support of the k-itemset, we need to, in principle, keep track of the counts of all 2^k combinations. To reduce these costs, MASK took the approach of maintaining *equal* flipping probabilities for both 1's and 0's – with this assumption, the number of counters required is only k [28]. Further, only $k-1$ of the counts need to be explicitly maintained since the sum of the counts is equal to the database cardinality, N – the counter chosen to be implicitly counted was the one with the expected largest count.

While these counting optimizations did appreciably reduce runtime costs, yet the overhead in absolute terms remains very significant – as mentioned earlier, it could be as much as two to three orders of magnitude compared to the time taken for direct mining.

3.3 EMASK algorithm

In this section, we describe how EMASK, the algorithm that we propose in this paper, is engineered to address the above-mentioned efficiency concerns with the basic MASK technique.

3.3.1 The Distortion Process

The new feature of EMASK's distortion process is that it applies *symbol-specific* distortion – that is, 1's are flipped with probability $(1-p)$, while 0's are flipped with probability $(1-q)$. Note that in this framework the original MASK can be viewed as a special case of EMASK wherein $p = q$.

The idea here is that MASK generates false items to hide the true items that are retained after distortion, resulting in an increase in database density. But, if a fewer number of true items are retained, a fewer number of false items need to be generated, and we can minimize this density increase. Thus the goal of reduced density could be achieved by reducing the value of p and increasing the value of q (specifically increasing it to beyond 0.9). However, note that a decrease in p value increases the distortion significantly which can reduce accuracy of reconstruction. Also, q or the non-flipping probability of 0s cannot be increased to very high values as it can decrease the privacy significantly. Thus the choices of p and q have to be made *carefully* to obtain a combination of p and q values, such that q is high enough to result in decreased transaction lengths but privacy and accuracy are still achievable.

We defer the discussion on how to select appropriate values of p and q to Section 3.3.5.

3.3.2 Privacy Quantification for EMASK

As discussed in Section 2.2, privacy can be computed at various levels. EMASK is designed to satisfy Basic Privacy(BP) and Reinterrogated Privacy (RP) requirements. We derive here the expression for Basic Privacy(BP) for distortion probabilities p and q in EMASK. Reinterrogation privacy(RP) is calculated as explained in Section 2.2.

Continuing from expression in Equation 2.1 for Basic Privacy(BP); for the distortion process

adopted by EMASK,

$$\Pr(V_{ij} = 1|U_{ij} = 1) = p$$
$$\Pr(V_{ij} = 0|U_{ij} = 1) = q$$

Let s_j be the support of the j^{th} item, then, from Equation 2.1,

$$\mathcal{R}_1(p, q, s_j) = \frac{p^2 s_j}{\Pr(V_i j = 1)} + \frac{(1-p)^2 s_j}{\Pr(V_i j = 0)}$$

But,

$$\begin{aligned}\Pr(V_{ij} = 1) &= \Pr(V_{ij} = 1|U_{ij} = 0)\Pr(U_{ij} = 0) + \Pr(V_{ij} = 1|U_{ij} = 1)\Pr(U_{ij} = 1) \\ &= s_j p + (1 - s_j)(1 - q)\end{aligned}$$

And,

$$\Pr(V_{ij} = 0) = 1 - \Pr(V_{ij} = 1) = s_j(1 - p) + (1 - s_j)q$$

Therefore,

$$\mathcal{R}_1(p, q, s_j) = \frac{p^2 s_j}{s_j p + (1 - s_j)(1 - q)} + \frac{(1-p)^2 s_j}{s_j(1 - p) + (1 - s_j)q}$$

The overall reconstruction probability of 1's is now given by

$$\mathcal{R}_1(p, q) = \frac{\sum_i s_j \mathcal{R}_1(p, q, s_j)}{\sum_i s_j}$$

and the Basic privacy offered to 1's is simply $100(1 - \mathcal{R}_1(p, q))$.

The above expression is *minimized* when all the items in the database have the same support, and increases with the variance in the supports across items. As a first-level approximation, we replace the item-specific supports in the above equation by s_0, the average support of an item in the database. With this, the reconstruction probability simplifies to

$$\mathcal{R}_1(p,q) = \frac{p^2 s_0}{s_0 p + (1-s_0)(1-q)} + \frac{(1-p)^2 s_0}{s_0(1-p) + (1-s_0)q} \tag{3.2}$$

In Figure 3.2 we plot $100(1-\mathcal{R}_1(p,q))$ i.e. privacy for different values of p and q at s_0=0.01 – the shadings indicate the privacy ranges. Note that the color of each square represents the privacy value of the lower left corner of the square. We observe here that there is a *band* of values for p and q in which the privacy is greater than the 80% mark. Specifically, for q between 0.1 and 0.9, high privacy values are attainable for all p values, and the whole of this region appears black. Beyond $q = 0.9$, privacy above 80% is attainable but only for low p values. Similarly for q below 0.1, high privacy can be obtained only at high p values.

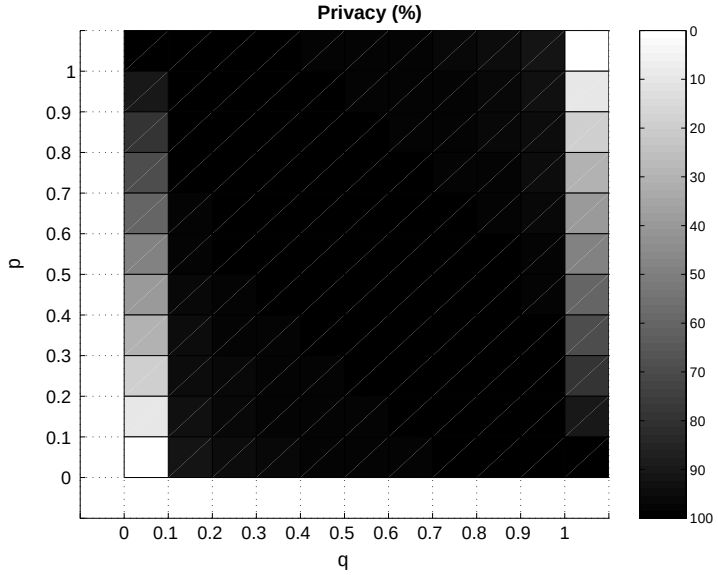


Figure 3.2: Basic Privacy of EMASK

3.3.3 The EMASK Mining Process

Having established the privacy obtained by EMASK's distortion process, we now move on to presenting EMASK's technique for estimating the true supports of itemsets from the distorted database. In the following discussion, we first show how to estimate the supports of 1-itemsets (i.e. singletons) and then present the general n-itemset support estimation procedure. In this

derivation, it is important to keep in mind that the miner is provided with both the distorted matrix[1] as well as the distortion procedure, that is, he *knows* the values of p and q that were used in distorting the true data matrix.

Estimating Singleton supports

We denote the original true matrix by U and the distorted matrix, obtained with distortion parameters p and q, as V. Now consider a random individual boolean attribute **j**. Let c_1^U and c_0^U represent the number of 1's and 0's, respectively, in the **j** column of U, while c_1^V and c_0^V represent the number of 1's and 0's, respectively, in the **j** column of V. With this notation, we estimate the support of **j** in U using the following equation:

$$\mathbf{C^U = A^{-1} C^V} \tag{3.3}$$

where

$$A = \begin{bmatrix} p & 1-q \\ 1-p & q \end{bmatrix} \quad C^V = \begin{bmatrix} c_1^V \\ c_0^V \end{bmatrix} \quad C^U = \begin{bmatrix} c_1^U \\ c_0^U \end{bmatrix}$$

Estimating n-itemset Supports

It is easy to extend Equation 3.3, which is applicable to individual items, to compute the support for an arbitrary n-itemset. For this general case, we define the matrices as:

$$C^V = \begin{bmatrix} c_{2^n-1}^V \\ . \\ . \\ . \\ c_1^V \\ c_0^V \end{bmatrix} \quad C^U = \begin{bmatrix} c_{2^n-1}^U \\ . \\ . \\ . \\ c_1^U \\ c_0^U \end{bmatrix}$$

Here c_k^U should be interpreted as the count of the tuples in U that have the binary form of k (in n digits) for the given itemset (that is, for a 2-itemset, c_2^U refers to the count of 10's in the

[1] The boolean dataset can be viewed as a large boolean matrix

columns of U corresponding to that itemset, c_3^U to the count of 11's, and so on). Similarly, c_k^V is defined for the distorted matrix V.

The column matrices can be simplified by observing that the distortion of an entry in the above distortion procedure depends only on whether the entry is 0 or 1, and *not* on the column to which the entry belongs, rendering distortion of all combinations with same number of 1s and 0s equivalent. Hence the above matrices can be represented as:

$$C^V = \begin{bmatrix} c_n^V \\ . \\ . \\ . \\ c_1^V \\ c_0^V \end{bmatrix} \quad C^U = \begin{bmatrix} c_n^U \\ . \\ . \\ . \\ c_1^U \\ c_0^U \end{bmatrix}$$

where c_k^U should be interpreted as the count of the tuples in U that have the binary form with k 1's (in n digits) for the given itemset. For example , for a 2-itemset, c_2^U refers to the count of 11's in the columns of U corresponding to that itemset, c_1^U to the count of 10's and 01's, and c_0^U to the count of 00's. Similarly, c_k^V is defined for the distorted matrix V.

Each entry $A_{i,j}$ in the matrix $\mathbf{A}$ is the probability that a tuple of the form corresponding to c_j^U in U goes to a tuple of the form corresponding to c_i^V in V. For example, $A_{2,1}$ for a 2-itemset is the probability that a 10 or 01 tuple distorts to a 11 tuple. Accordingly, $A_{2,1}$ = $p(1-q)$. The basis for this formulation lies in the fact that in our distortion procedure, the component columns of an n-itemset are distorted *independently*. Therefore, we can use the product of the probability terms. In general,

$$A_{i,j} = \sum_{k=max(0,i+j-n)}^{min(i,j)} \binom{j}{k} p^k (1-p)^{(j-k)} \binom{n-j}{i-k} q^{(n-j-i+k)} (1-q)^{(i-k)} \tag{3.4}$$

3.3.4 Eliminating Counting Overhead

We now present a simple but powerful optimization by which the entire extra overhead of counting all the combinations generated by the distortion can be *eliminated completely*. This

optimization is based on the following basic formula from set theory: Given a universe $\mathcal{U}$, and subsets $\mathcal{A}$ and $\mathcal{B}$,

$$N(\mathcal{A}' \cap \mathcal{B}) = N(\mathcal{B}) - N(\mathcal{A} \cap \mathcal{B})$$

where $N(\cdot)$ is the number of elements, i.e. cardinality, of the set denoted by the bracketed expression. This formula can be generalized[2] to

$$N(\mathcal{A}'_1\mathcal{A}'_2 \ldots \mathcal{A}'_m\mathcal{B}_1\mathcal{B}_2 \ldots \mathcal{B}_n)$$
$$= N(\mathcal{B}_1\mathcal{B}_2 \ldots \mathcal{B}_n) + \sum_{k=1}^{m} \sum_{\{x_1,\ldots,x_k\} \subset \{1,\ldots,m\}} (-1)^k N(\mathcal{A}_{x_1}\mathcal{A}_{x_2} \ldots \mathcal{A}_{x_k}\mathcal{B}_1\mathcal{B}_2 \ldots \mathcal{B}_n)$$

Using above formula the counts of all the combinations generated from an n-itemset can be calculated from the counts of itemsets and the counts of their subsets which are available from previous passes over the distorted database. *Hence, we only need to explicitly count only the '111...1's, just as in the direct mining case.*

For example, during the second pass we need to explicitly count only '11's which makes $N(\mathcal{A} \cap \mathcal{B})$ available at the end of the second pass. The counts of the remaining combinations, '10', '01' and '00' can then be calculated using the following set of formulae:

$$10 : N(\mathcal{A} \cap \mathcal{B}') = N(\mathcal{A}) - N(\mathcal{A} \cap \mathcal{B})$$
$$01 : N(\mathcal{A}' \cap \mathcal{B}) = N(\mathcal{B}) - N(\mathcal{A} \cap \mathcal{B})$$
$$00 : N(\mathcal{A}' \cap \mathcal{B}') = N(\mathcal{U}) - N(\mathcal{A}) - N(\mathcal{B}) + N(\mathcal{A} \cap \mathcal{B})$$

The above implies that the extra calculations for reconstruction are performed only at *the end of each pass*, the rest of the pass being exactly the same as that of the original mining algorithm. Further, the only additional requirement of the above approach as compared to traditional data mining algorithms such as Apriori is that we need to have available, at the end of each pass, the counts of all frequent itemsets generated during the previous passes. However, this requirement is easily met by keeping a hash table in memory of these previously identified frequent itemsets.

[2] $N(\mathcal{B}_1\mathcal{B}_2..\mathcal{B}_n)$ is replaced by $N(\mathcal{U})$ if $n = 0$

3.3.5 Choosing values of p and q

As promised earlier, we now discuss how the distortion parameters, p and q, should be chosen. Our aim here is to identify those parameter settings that achieve the three goals of privacy, accuracy and run time efficiency simultaneously or, in other words, minimize the privacy breach, error and run time.

The problem can be viewed as a multi-objective optimization problem,

$$\text{Find } X = \{p, q\}$$

which minimizes

$$f_1(X), f_2(X), f_3(X)$$

$$X \in \Re^2$$

$$X_l \leq X \leq X_u$$

Here, $X_l = \{0, 0\}$, $X_u = \{1, 1\}$ and $f_1(X)$,$f_2(X)$ and $f_3(X)$ are objective functions denoting the estimations for privacy breach, error and relative run time of EMASK to Apriori[2]. Exact expressions for these functions are discussed in section 3.3.5. As the three objectives are competing, there can be no solution that minimizes all the objectives simultaneously. Instead, the concept of non-inferiority [42](also called Pareto optimality [12], must be used to characterize the objectives. A feasible solution X is called Pareto optimal if there exists no other feasible solution Y such that $f_i(Y) \leq f_i(X)$ for i=1,2,..k with $f_i(Y) < f_i(X)$ for at least one j, here k is the number of objectives. In other words, a feasible vector X is called Pareto optimal if there is no other feasible solution Y that would reduce some objective function without causing a simultaneous increase in at least one other objective function.

Most of the multi-objective optimization methods basically generate a set of Pareto optimal solutions and use some additional criterion or rule to select one particular Pareto optimal solution as the solution of the multi-objective problem. For our purpose we need a method to chose the Pareto optimal solution which achieves the three objectives equally or gives designer specified weights to each of the three objectives. We use goal attainment method of Gembicki

[22] to solve this problem.

Goal attainment method involves expressing a set of design goals, $F^* = \{F_1^*, F_2^*, ...F_k^*\}$, associated with the objectives $F(X) = \{f_1(X), f_2(X), ..., f_k(X)\}$. The problem formulation allows the objectives to be under- or over-achieved enabling the designer to be relatively imprecise about initial design goals. The relative degree of under- or over-achievement of the goals is controlled by a vector of weighting coefficients, $w = \{w_1, w_2, ..., w_k\}$, and is expressed as a standard optimization problem using the following formulation.

$$minimize\, \gamma$$

$$\gamma \in \Re, X \in \Omega$$

such that

$$F_i(X) - w_i\gamma \leq F_i^* \;\; i = 1, ...k$$

The term $w_i\gamma$ introduces an element of slackness into the problem, which otherwise imposes that the goals be rigidly met. The weighting vector, w, enables the designer to express a measure of the relative trade-offs between the objectives. For instance, setting the weighting vector w equal to the initial goals indicates that the same percentage under- or over-attainment of the goals,F^* , is achieved. Hard constraints can be incorporated into the design by setting a particular weighting factor to zero (i.e.,$w_i = 0$). In our experiments, we will take w equal to initial goal, to ensure equal weightage to over-attainment or under-attainment of all the three objectives. For a specific problem the user can vary the weights.

The advantage of using the goal attainment method is that the user needs to specify only loose goals for the three objectives. The algorithm gives values of p and q for the best that be attained over or under it.

Objective functions

We use estimations of privacy breach, error and relative runtime of EMASK as the objective functions to be minimized. The average support value of the dataset and the dataset size(no. of tuples) are required for these estimations. We assume here that some initial sample data is

available based on which the determination of average support value can be made.

Privacy Breach Estimation Earlier in section 3.3.2, we had shown how the basic privacy(BP) for EMASK can be calculated. We use the basic privacy values as the estimation for privacy at a point (p,q) (only the basic privacy can be calculated before distortion.) Thus from equation 3.2,

$f_1(X) = 1 - BP = R_1(p, q)$

$$= \frac{s_0 \times p^2}{s_0 \times p + (1 - s_0) \times (1 - p)} + \frac{s_0 \times (1 - p)^2}{s_0 \times (1 - p) + (1 - s_0) \times p} \tag{3.5}$$

Error Estimation We now describe how to estimate the error of reconstruction beforehand. Our focus is specifically on 1-itemsets since their error percolates through the entire mining process.

Let the true database consists of *tuples*. Consider a single column in the true database which has n 1's and $N - n$ 0's. We expect that the n 1's will distort to np 1's and $n(1 - p)$ 0's when distorted with parameter p. Similarly, we expect the 0's to go to $(N - n)q$ 0's and $(N - n)(1 - q)$ 1's. However, note that this is assuming that "When we generate a random number, which is distributed as Bernoulli(p), then the number of 1's, denoted by P, in n trials is actually np". But, in reality, this will not be so. Actually, P is distributed as ***binomial(n,p)***, with mean np and variance $np(1 - p)$.

The total number of 1's in a column of the distorted matrix can be expressed as the sum of two random variables, X and Y:

- X: The 1's in the distorted database that come from 1's in the original database.
- Y: The 1's in the distorted database that come from 0's in the original database.

The variance in the total number of 1's generated in the distorted matrix is $var(X + Y)$, which is simply $var(X) + var(Y)$ since X and Y are ***independent*** random variables. So, we have $var(X + Y) = var(X) + var(Y) = np(1 - p) + (N - n)(1 - q)q$. Therefore, the deviation

of the number of 1's in the distorted database from the expected value is

$$\triangle n' = (np(1-p) + (N-n)(1-q)q)^{1/2} \tag{3.6}$$

Let the distorted column have n' 1's. Then, our estimate of the original count is given by

$$\overline{n} = \frac{n'}{p+q-1} - \frac{(1-q)N}{p+q-1}$$

and the possible error in this estimation is computed as

$$\triangle \overline{n} = abs(\frac{\triangle n'}{p+q-1})$$

where $\triangle n'$ is as per Equation 3.6. If we normalize the error to the true value (n), we obtain

$$Error = abs(\frac{(np(1-p) + (N-n)(1-q)q)^{1/2}}{n(p+q-1)})$$

Substituting $n = N * s$,

$$Error = abs(\frac{1}{(p+q-1)}\sqrt{\frac{(sp(1-p) + (1-s)(1-q)q)}{N*s}}) = f_2(X) \tag{3.7}$$

which is completely expressed in terms of known parameters. Equation 3.7 gives the second objective function $f_2(X)$ to be minimized.

The above equation estimates the error in reconstruction of counts of 1-itemsets. But as their error percolates through the entire mining process, we expect the p and q values giving higher Error by this measure to give higher overall error for other levels as well. Thus we can estimate the desirability of a particular p and q value in terms of reconstruction accuracy using the above estimation.

Runtime Estimation To estimate the run time, we use the observations in the section 3.3. After optimizations of section 3.3.4 the increased density of the database remains the only reason for inefficiency of EMASK. Thus the ratio of runtime of EMASK to Apriori can be

estimated to ratio of density of the randomized database to the non-randomized database. This ratio gives the third objective function to be minimized to achieve optimum efficiency.

$$f_3(X) = p + \frac{(1-s)}{s}(1-q) \tag{3.8}$$

The above estimations for privacy, accuracy and efficiency are not accurate and are likely to underestimate the respective measures. But as they uniformly underestimate, they are appropriate for selecting p and q by minimization. We shall illustrate this in experimental evaluation section.

3.4 Performance Framework

EMASK aims at simultaneously achieving satisfactory performance on three objectives: privacy, accuracy and efficiency. The specific metrics used to evaluate EMASK's performance w.r.t. privacy, accuracy and efficiency are given below.

3.4.1 Privacy Metrics

As discussed in previous chapter, privacy can be computed at various levels, namely : Basic Privacy (BP), Reinterrogated Privacy (RP) and Strict Privacy(SP). As outlined in Section 3.3.2, we quantify the privacy provided by EMASK with regard to Basic Privacy and Reinterrogated Privacy. Algorithms for achieving Strict Privacy will be discussed in the next chapter.

3.4.2 Accuracy Metrics

The association rule mining errors can be quantified in terms of *Support Error* ρ and *Identity Error* (false positives σ^+, false negatives σ^-), as explained in Section 2.3.1. We use these as accuracy metrics in our experiments.

3.4.3 Efficiency Metric

This metric determines the runtime overheads resulting from mining the distorted database as compared to the time taken to mine the original database. This is simply measured as the inverse ratio of the running times between Apriori [2] on the original database and executing the same code augmented with EMASK (i.e. EMASK-Apriori) on the distorted database. Denoting this slowdown ratio as Δ, we have

$$\Delta = \frac{Runtime\ of\ EMASK Apriori}{Runtime\ of\ Apriori}$$

For ease of presentation, we hereafter refer to this augmented algorithm simply as EMASK.

3.5 Experimental Results

3.5.1 Data Sets

We carried out experiments on a variety of large sparse synthetic and real datasets. Due to space limitations, we report the results for only three representative databases here:

1. A synthetic database generated from the IBM Almaden generator [2]. The synthetic database was created with parameters T10.I4.D1M.N1K (as per the naming convention of [2]), resulting in a million customer tuples with each customer purchasing about ten items on average.

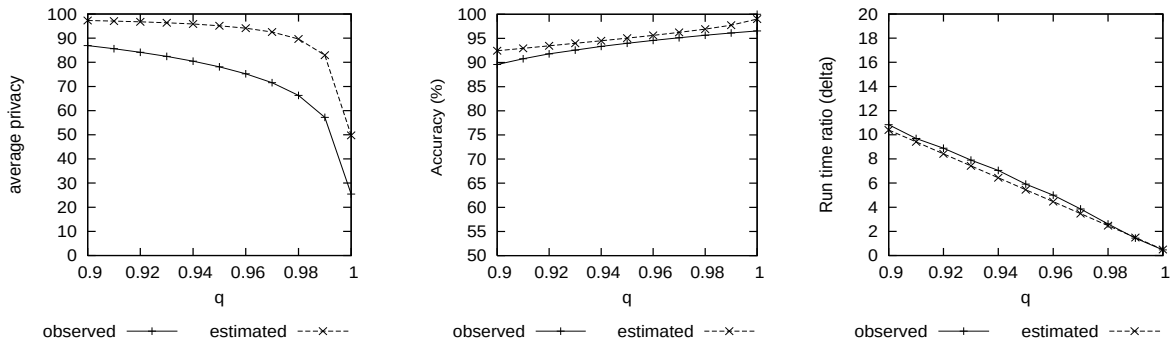
2. A real dataset, BMS-WebView-1 [43], placed in the public domain by Blue Martini Software. This database contains click-stream data from the web site of a (now defunct) legwear and legcare retailer. There are about 60,000 tuples with close to 500 items in the schema. In order to ensure that our results were applicable to large disk-resident databases, we scaled this database by a factor of ten, resulting in approximately 0.6 million tuples. The average transaction length was 2.5 for this database.

3. Another real dataset, BMS-WebView-2 [43], placed in public domain by Blue Martini Software. This dataset also contains click-stream data from e-commerce websites. There

are about 78,000 tuples in the database with 3340 items in the schema. This database was scaled by 50 times resulting in approximately 4 million tuples. The average transaction length was 5 for this database.

3.5.2 Support and Distortion settings

The theoretical basis for determining the settings of the distortion parameters, p and q, was presented in Section 3.3.5. Figure 3.3 illustrates how estimations are appropriate for choosing the values of p and q.



(a) (b) (c)

Figure 3.3: Comparison of estimates to observed performance behavior

The graphs are plotted for p=0.5 and varying q from 0.9 to 1 for the Synthetic database. The figure shows that estimated values are very close to the observed values for accuracy and run time ratio. Here the accuracy is *100-support error* averaged over all the levels. The difference between the values of estimated and observed privacy values in figure 3.3(a) are due to the fact that the observed privacy is average *reinterrogated privacy* which depends on the characteristics of database and cannot be estimated before hand. The estimation takes into account only the *basic privacy*. But the figure shows that the trends in the curve for estimated privacy does follow the curve for observed privacy. Thus the p and q values determined through optimization of estimations can be expected to give best actual performance too.

DataSet	*Specified Goals* {*privacy breach, error,* Δ}	*Attained by goal programming*	*p*	*q*
Synthetic	{0.06,0.03,5}	{0.075, 0.0375, 3.5192 }	0.5051	0.9696
BMS-WebView1	{0.06,0.03,5}	{0.1075, 0.0537, 3.0212}	0.5673	0.9877
BMS-WebView2	{0.06,0.03,5}	{0.0861, 0.0431, 3.3354}	0.5350	0.9958

Table 3.1: p, q combinations for EMASK obtained by goal programming

To use the goal programming method for multi-objective optimization as explained in section 3.3.5, we take an approximate goal of {0.06,0.03,5} (which is equivalent to 6% privacy breach, 3% error and $\Delta = 5$) and set the weight w as $abs(goal)$. Note that these are not rigid goals, the method shall obtain the best point $\{p, q\}$ for the database which may under or over-attain the specified goal.

The average support for the three databases are 0.01, 0.005 and 0.0015 respectively. The values of parameter {p,q} returned, and the goals expected to be attained for those values are shown in Table 3.1 for the three databases.

We evaluated the actual privacy, accuracy and efficiency of the EMASK privacy-preserving mining process for the obtained distortion parameters on the three datasets for a variety of minimum support values. Due to space limitations, we present here the results only for 0.3% and 0.5% sup_{min} value, which represents supports low enough for a large number of frequent itemsets to be produced, thereby stressing the performance of the EMASK algorithm. The results for the synthetic database are presented first, followed by those for the real databases.

3.5.3 Experiment Set 1 : Synthetic Dataset

Mining the synthetic dataset with a sup_{min} of 0.3 and 0.5% resulted in frequent itemsets of length upto 8. Table 3.2 and 3.3 present the EMASK accuracy results for this dataset. Here *Level* denotes the pass of Apriori or the length of frequent itemsets, $|F|$ denotes the number of frequent itemsets in the non-randomized database, σ^+, σ^- and ρ are error metrics as explained in the previous section.

The basic privacy BP for this setting was 92.5% where as reinterrogated privacy RP was

Level	$\|F\|$	σ^+	σ^-	ρ
1	664	1.05	1.35	3.34
2	1847	6	6.38	3.92
3	1310	3.58	5.11	3.97
4	864	7.98	6.82	5.7
5	419	7.15	13.12	8.46
6	115	6.08	9.56	12.12
7	21	0	4.76	16.61
8	2	0	0	16.53

Table 3.2: Synthetic : p=0.5051,q=0.9696,$\sup_{\min}$=0.3%

Level	$\|F\|$	σ^+	σ^-	ρ
1	560	0.71	0.71	2.55
2	470	8.51	4.89	2.79
3	326	4.29	8.58	3.38
4	208	8.65	6.73	5.68
5	125	0.8	10.4	8.64
6	43	0	11.62	13.57
7	10	0	10	19.08
8	1	0	0	20.19

Table 3.3: Synthetic : p=0.5051,q=0.9696,$\sup_{\min}$=0.5%

observed to be 71.3%. The time taken by EMASK was 3.94 times Apriori for sup_{min}=0.3% ,i.e. $\Delta_{0.3} = 3.94$ and 2.91 times for sup_{min}=0.5% i.e. , $\Delta_{0.5} = 2.91$. Thus EMASK takes only about 4 times the time taken by the undistorted mining.

3.5.4 Experiment Set 2: BMS-WebView-1 Dataset

We conducted a similar set of experiments on the real dataset (BMS-WebView-1 [43]), which had frequent itemsets of length upto 4. The accuracy results for this set of experiments are shown in Table 3.4 and 3.5.

The basic privacy BP for this setting was 89.23% where as reinterrogated privacy RP was observed to be 74.14%. For this database, EMASK took only 2.03 times Apriori for sup_{min}=0.3% and 1.58 times for sup_{min}=0.5% i.e. $\Delta_{0.3} = 2.03$, $\Delta_{0.5} = 1.58$. This run

Level	$\|F\|$	σ^+	σ^-	ρ
1	225	1.33	2.22	3.27
2	169	5.91	1.18	2.68
3	39	5.12	10.25	3.34
4	2	0	0	6.02

Table 3.4: BMS-WebView-1 : p=0.5673, q=0.9877, $\sup_{\min}$=0.3%

Level	$\|F\|$	σ^+	σ^-	ρ
1	150	2	0.66	2.28
2	45	0	2.22	2.22
3	6	33.33	0	2.55

Table 3.5: BMS-WebView-1 : p=0.5673, q=0.9877, $\sup_{\min}$=0.5%

time result is especially encouraging as it shows that the Privacy-preservimg mining is taking comparable time to undistorted mining.

3.5.5 Experiment Set 3: BMS-WebView-2 Dataset

The accuracy results for this dataset are shown in Table 3.6 and 3.7.

The basic privacy BP for this setting was 91.4% where as reinterrogated privacy RP was observed to be 75.7%. For the purpose of calculating reinterrogated privacy, we chose to find the frequent itemsets for a much lower sup_{min} of 0.1% for this database, as it is very sparse (average support of only 0.0015). The time taken by EMASK was 2.21 times Apriori for sup_{min}=0.3% i.e. $\Delta_{0.3} = 2.21$. and 2.17 times for sup_{min}=0.5% , $\Delta_{0.5} = 2.17$.

3.6 Summary

Overall, our experiments indicate that by a careful choice of distortion parameter settings, it is possible to simultaneously achieve satisfactory privacy, accuracy, *and efficiency*. Accuracy above 90%, privacy above 70% and run times of less than 4 times Apriori could be obtained for all databases. The results were especially encouraging for large and sparse databases like

Level	$\|F\|$	σ^+	σ^-	ρ
1	340	1.17	0.88	1.11
2	270	1.85	0.74	0.99
3	338	2.36	1.18	1.5
4	278	6.83	1.07	2.54
5	101	20.79	3.96	3.91
6	13	7.69	0	3.82

Table 3.6: BMS-WebView-2 : p=0.5350,q=0.9958,$\sup_{\min}$=0.3%

Level	$\|F\|$	σ^+	σ^-	ρ
1	170	1.17	0.58	0.79
2	120	1.66	0	0.84
3	88	3.4	0	1.39
4	28	25	0	2.37
5	2	0	0	3.27

Table 3.7: BMS-WebView-2 : p=0.5350, q=0.9958, $\sup_{\min}$=0.5%

BMS-WebView-3.

In particular, the experiments show that there is a “window of opportunity” where these triple goals can be all met. The size and position of the window is primarily a function of the database density and could be quite accurately characterized with our estimation optimization method.

Chapter 4

A Framework for High-Accuracy Strict-Privacy-Preserving Mining

We present here a generalized matrix theoretic framework FRAPP for random perturbation methods for privacy preserving mining under *strict privacy guarantees*. The framework is meant for categorical datasets. Note that boolean attributes are special case of categorical attributes and continuous attributes can be converted into categorical by partitioning the domain of the attribute into fixed size intervals.

4.1 The FRAPP Framework

In this section, we describe the construction of the FRAPP framework, and its quantification of privacy and accuracy measures.

Data Model. We assume that the original database U consists of N records, with each record having M categorical attributes. The domain of attribute j is denoted by S_U^j, resulting in the domain S_U of a record in U being given by $S_U = \prod_{j=1}^{M} S_U^j$. We map the domain S_U to index set $I_U = \{1, \ldots, |S_U|\}$, so that we can model the database as set of N values from I_U. Thus, if we denote i^{th} record of U as U_i, we have

$$U = \{U_i\}_{i=1}^{N}, \qquad U_i \in I_U$$

4.1.1 Perturbation Model

We consider the privacy situation wherein the customers trust *no one except themselves*, that is, they wish to perturb their records at their client site before the information is sent to the the miner, or any intermediate party. This means that perturbation is done at the level of *individual* customer records U_i, without being influenced by the contents of the other records in the database.

For this situation, there are two possibilities: a simple *independent column perturbation*, wherein the value of each attribute in the record is perturbed independently of the rest, or a more generalized *dependent column perturbation*, where the perturbation of each column may be affected by the perturbations of the other columns in the record. Most of the prior perturbation techniques, including [19, 20, 28], fall into the independent column perturbation category. The FRAPP framework, however, includes both kinds of perturbation in its analysis.

Let the perturbed database be $V = \{V_1, \ldots, V_N\}$, with domain S_V, and corresponding index set I_V.

For each original customer record $U_i = u, u \in I_U$, a new perturbed record $V_i = v, v \in I_V$ is randomly generated with probability $p(u \to v)$. Let A denote the matrix of these transition probabilities, with $A_{vu} = p(u \to v)$. This random process maps to a Markov process, and the perturbation matrix A should therefore satisfy the following properties [32]:

$$\begin{aligned} \sum_{v \in I_V} A_{vu} &= 1 \qquad \forall u \in I_U \\ A_{vu} &\geq 0 \qquad \forall u \in I_U, v \in I_V \end{aligned} \tag{4.1}$$

Due to the constraints imposed by Equation 4.1, the domain of A is not $\mathbf{R}^{|\mathbf{S_U}| \times |\mathbf{S_V}|}$ but a subset of it. This domain is further restricted by the choice of perturbation method. For example, for the MASK technique [28] mentioned in the Introduction, all the entries of matrix A are

decided by the choice of the single parameter p.

In this paper, we propose to explore the *preferred choices* of A to simultaneously achieve privacy guarantees and high accuracy, without restricting ourselves ab initio to a particular perturbation method.

4.1.2 Privacy Guarantees

The miner receives the perturbed database V and attempts to reconstruct the original probability distribution of database U using this perturbed data and the knowledge of the perturbation matrix A.

The *prior probability* of a property of a customer's private information is the likelihood of the property in the absence of any knowledge about the customer's private information. On the other hand, the *posterior probability* is the likelihood of the property given the perturbed information from the customer and the knowledge of the prior probabilities through reconstruction from the perturbed database. As discussed in [20], in order to preserve the privacy of some property of a customer's private information, we desire that the posterior probability of that property should not be much higher than the prior probability of the property for the customer. This is quantified by saying that a perturbation method has privacy guarantees (ψ_1, ψ_2) if, for any property $Q(U_i)$ with prior probability less than ψ_1, the posterior probability of the property is guaranteed to be less than ψ_2.

For our formulation, we derive (using Definition 3 and Statement 1 from [20]) the following condition on the perturbation matrix A in order to support (ψ_1, ψ_2) privacy.

$$\frac{A_{vu_1}}{A_{vu_2}} \leq \gamma \leq \frac{\psi_2(1-\psi_1)}{\psi_1(1-\psi_2)} \qquad u_1, u_2 \in I_U, \forall v \in I_V \tag{4.2}$$

That is, the choice of perturbation matrix A should follow the restriction that the ratio of any two entries should not be more than γ.

4.1.3 Reconstruction Model

We now analyze how the distribution of the original database can be reconstructed from the perturbed database. As per the perturbation model, a client C_i with data record $U_i = u, u \in I_U$ generates record $V_i = v, v \in I_V$ with probability $p[u \rightarrow v]$. This event of generation of v can be viewed as a Bernoulli trial with success probability $p[u \rightarrow v]$. If we denote outcome of i^{th} Bernoulli trial by random variable Y_v^i, then the total number of successes Y_v in N trials is given by sum of the N Bernoulli random variables. i.e.

$$Y_v = \sum_{i=1}^{N} Y_v^i \tag{4.3}$$

That is, the total number of records with value v in the perturbed database will be given by the total number of successes Y_v.

Note that Y_v is the sum of N independent but non-identical Bernoulli trials. The trials are non-identical because the probability of success in a trial i varies from another trial j and actually depends on the values of U_i and U_j, respectively. The distribution of such a random variable Y_v is known as the Poisson-Binomial distribution [38].

Now, from Equation 4.3, the expectation of Y_v is given by

$$E(Y_v) = \sum_{i=1}^{N} E(Y_v^i) = \sum_{i=1}^{N} P(Y_v^i = 1) \tag{4.4}$$

Let X_u denote the number of records with value u in the original database.
Since $P(Y_v^i = 1) = p[u \rightarrow v] = A_{vu}$, for $U_i = u$, we get

$$E(Y_v) = \sum_{u \in I_U} A_{vu} X_u \tag{4.5}$$

Let $X = [X_1 X_2 \cdots X_{|S_U|}]^T$, $Y = [Y_1 Y_2 \cdots Y_{|S_V|}]^T$, then from Equation 4.5 we get

$$E(Y) = AX \tag{4.6}$$

We estimate X as $\widehat{X}$ given by the solution of following equation

$$Y = A\widehat{X} \tag{4.7}$$

which is an approximation to Equation 4.6. This is a system of $|S_V|$ equations in $|S_U|$ unknowns. For the system to be uniquely solvable, a necessary condition is that the space of the perturbed database is larger than or equal to the original database (i.e. $|S_V| \geq |S_U|$). Further, if the inverse of matrix A exists, then we can find the solution of above system of equations by

$$\widehat{X} = A^{-1}Y \tag{4.8}$$

That is, Equation 4.8 gives the estimate of the distribution of records in the original database, which is the objective of the reconstruction exercise.

4.1.4 Estimation Error

To analyze the error in the above estimation process, we use the following well-known theorem from linear algebra [32]:

Theorem 1: For an equation of form $Ax = b$, the relative error in solution $x = A^{-1}b$ satisfies

$$\frac{\| \delta x \|}{\| x \|} \leq c\frac{\| \delta b \|}{\| b \|}$$

where c is the *condition number* of matrix A. For a positive definite matrix, $c = \lambda_{max}/\lambda_{min}$, where λ_{max} and λ_{min} are the maximum and minimum eigen values of $n \times n$ matrix A. Informally, the condition number is a measure of stability or sensitivity of a matrix to numerical operations. Matrices with condition numbers near one are said to be *well-conditioned*, whereas those with condition numbers much greater than one (e.g. 10^5 for a $5 * 5$ Hilbert matrix [32]) are said to be *ill-conditioned*.

From Equations 4.6, 4.8 and the above theorem, we have

$$\frac{\| \widehat{X} - X \|}{\| X \|} \leq c \frac{\| Y - E(Y) \|}{\| E(Y) \|} \tag{4.9}$$

This inequality means that the error in estimation arises from two sources: First, the sensitivity of the problem which is measured by the condition number of matrix A; and, second, the deviation of Y from its mean as measured by the variance of Y.

As discussed above, Y_v is a Poisson-Binomial distributed random variable. Hence, using the expression for variance of a Poisson-Binomial random variable [38], we can compute the variance of Y_v to be

$$Var(Y_v) \;=\; A_v X(1 - \frac{1}{N} A_v X) - \sum_{u \in I_U} (A_{vu} - \frac{1}{N} A_v X)^2 X_u \tag{4.10}$$

which depends on the perturbation matrix A and the distribution X of records in the original database. Thus the effectiveness of the privacy preserving method is *critically dependent on the choice of matrix* A.

4.2 Choice of Perturbation Matrix

The various perturbation techniques proposed in the literature primarily differ in their choice for perturbation matrix A. For example,

- MASK [28] uses the matrix A with

$$A_{vu} = p^k (1-p)^{M_b - k} \tag{4.11}$$

where M_b is the number of boolean attributes when each categorical attribute j is converted into $\mid S_U^j \mid$ boolean attributes, k is the number of attributes with matching values in perturbed record v and original record u, and $1 - p$ is the value flipping probability.

- The *cut-and-paste* randomization operator [19] employs a matrix A with

$$A_{vu} = \sum_{z=0}^{M} p_M[z] \sum_{q=max\{0,z+l_u-M,l_u+l_v-M_b\}}^{min\{z,l_u,l_v\}} \frac{\binom{l_u}{q}\binom{M-l_u}{z-q}}{\binom{M}{z}} \cdot \binom{M_b-l_u}{l_v-q} p^{(l_v-q)}(1-p)^{(M_b-l_u-l_v+q)}$$

where (4.12)

$$p_M[z] = \sum_{w=0}^{min\{K,z\}} \binom{M-w}{z-w} p^{(z-w)}(1-p)^{(M-z)} \cdot \begin{cases} 1-M/(K+1) & \text{if } w=M \text{ and } w<K \\ 1/(K+1) & \text{o.w.} \end{cases}$$

Here l_u and l_v are the number of 1^s in the original record u and its corresponding perturbed record v, respectively, while K and p are operator parameters.

For enforcing strict privacy guarantees, the parameters for the above methods are decided by the constraints on the values of perturbation matrix A given in Equation 4.2. It turns out that for practical values of privacy requirements, the resulting matrix A for these schemes is extremely *ill-conditioned* – in fact, we found the condition numbers in our experiments to be of the order of 10^5 and 10^7 for MASK and the Cut-and-Paste operator, respectively.

Such ill-conditioned matrices make the reconstruction very sensitive to the variance in the distribution of the perturbed database. Thus, it is important to carefully choose the matrix A such that it is well-conditioned (i.e has a low condition number). If we decide on a distortion method apriori, as in the prior techniques, then there is little room for making specific choices of perturbation matrix A. Therefore, we take the opposite approach of first designing matrices of the required type, and then devising perturbation methods that are compatible with the chosen matrices.

To choose the appropriate matrix, we start from the intuition that for $\gamma = \infty$, the matrix choice would be the unity matrix, which satisfies the constraints on matrix A imposed by Equations 4.1 and 4.2, and has condition number 1. Hence, for a given γ, we can choose the

following matrix:

$$A_{ij} = \begin{cases} \gamma x, & \text{if } i = j \\ x, & \text{o.w.} \end{cases}$$

$$where$$

$$x = \frac{1}{\gamma + (|S_U| - 1)} \tag{4.13}$$

This matrix will be of the form

$$x \begin{bmatrix} \gamma & 1 & 1 & \dots \\ 1 & \gamma & 1 & \dots \\ 1 & 1 & \gamma & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

It is easy to see that the above matrix, which incidentally is a symmetric Toeplitz matrix [32], satisfies the conditions given by Equations 4.1 and 4.2. Further, its condition number can be computed to be $1+\frac{|S_U|}{\gamma-1}$, For ease of exposition, we will hereafter refer to this matrix informally as the "gamma-diagonal matrix".

At this point, an obvious question is whether it is possible to design matrices that have even lower condition number than the gamma-diagonal matrix. In the remainder of this section, we prove that within the constraints of our problem, the gamma-diagonal matrix has the *lowest* possible condition number, that is, it is an *optimal choice* (albeit non-unique).

Proof. To prove this, we will first derive the expression for minimum condition number for such matrices and the conditions under which that condition number is achieved. Then we show that our gamma-diagonal matrix satisfies these conditions, and has minimum condition number.

For a symmetric positive definite matrix, the condition number is given by

$$c = \frac{\lambda_{max}}{\lambda_{min}} \tag{4.14}$$

where λ_{max} and λ_{min} are the maximum and minimum eigenvalues of the matrix. As the matrix A is a Markov matrix (refer Equation 4.1), the following theorem for eigenvalues of a matrix can be used

Theorem 2 [32] *For an $n \times n$ Markov matrix,*

- 1 *is an eigenvalue*
- *the other eigenvalues satisfy* $\mid \lambda_i \mid \leq 1$

Theorem 3 [32] *The sum of n eigenvalues equals the sum of n diagonal entries:*

$$\lambda_1 + \cdots + \lambda_n = A_{11} + \cdots + A_{nn}$$

Using Theorem 2 we get,

$$\lambda_{max} \quad = \quad 1$$

As the least eigenvalue λ_{min} will always be less than or equal to average of the eigenvalues other than λ_{max}, we get,

$$\lambda_{min} \quad \leq \quad \frac{1}{n-1} \sum_{i=2}^{n} \lambda_i$$

where $\lambda_1 = \lambda_{max}$ Using Theorem 3,

$$\lambda_{min} \quad \leq \quad \frac{1}{n-1} \left(\sum_{i=1}^{n} A_{ii} - 1 \right), \tag{4.15}$$

Hence, condition number,

$$c = \frac{1}{\lambda_{min}} \geq \frac{n-1}{\sum_{i=1}^{n} A_{ii} - 1} \tag{4.16}$$

Now, due to privacy constraints on A given by Equation 4.2,

$$A_{ii} \leq \gamma A_{ij} \text{ for any } j \neq i,$$

i.e.,

$$\begin{aligned} A_{ii} &\leq \gamma A_{i1} \\ A_{ii} &\leq \gamma A_{i2} \\ &\vdots \end{aligned}$$

Summing above,

$$\begin{aligned} (n-1)A_{ii} &\leq \gamma \sum_{j \neq i} A_{ij} \\ &= \gamma(1 - A_{ii}) \end{aligned}$$

where the last step is due to the condition on A given by Equation 4.1. Solving for A_{ii}, we get,

$$A_{ii} \leq \frac{\gamma}{\gamma + n - 1} \tag{4.17}$$

Using above inequality in Equation 4.16, we get

$$c \geq \frac{n-1}{\frac{n\gamma}{\gamma+n-1} - 1} = \frac{\gamma + n - 1}{\gamma - 1} \tag{4.18}$$

Hence minimum condition number for the symmetric perturbation matrices under privacy constraints represented by γ is $\frac{\gamma+n-1}{\gamma-1}$. This condition number is achieved when $A_{ii} = \frac{\gamma}{\gamma+n-1}$.

The diagonal values of gamma-diagonal matrix given by Equation 4.1 is $\frac{\gamma}{\gamma+n-1}$. Thus it is *minimum condition number* symmetric perturbation matrix, with condition number $\frac{\gamma+|S_U|-1}{\gamma-1}$.

4.3 Randomizing the Perturbation Matrix

The estimation model in the previous section implicitly assumed the perturbation matrix A to be *deterministic*. However, it appears intuitive that if the perturbation matrix parameters are

themselves *randomized*, so that each client uses a perturbation matrix that is not specifically known to the miner, the privacy of the client will be further increased. Of course, it may also happen that the reconstruction accuracy may suffer in this process.

In this section, we explore this trade-off. Instead of deterministic matrix A, the perturbation matrix here is matrix $\tilde{A}$ of random variables, where each entry $\tilde{A}_{vu}$ is a random variable with $E(\tilde{A}_{vu}) = A_{vu}$. The values taken by the random variables for a client C_i provide the specific values for his/her perturbation matrix.

4.3.1 Privacy Guarantees

Let $Q(U_i)$ be a property of client C_i's private information, and let record $U_i = u$ be perturbed to $V_i = v$. Denote the prior probability of $Q(U_i)$ by $P(Q(U_i))$. On seeing the perturbed data, the posterior probability of the property is calculated to be:

$$\begin{aligned} P(Q(U_i)|V_i = v) &= \sum_{Q(u)} P_{U_i|V_i}(u|v) \\ &= \sum_{Q(u)} \frac{P_{U_i}(u)P_{V_i|U_i}(v|u)}{P_{V_i}(v)} \end{aligned}$$

When we use a fixed perturbation matrix A for all clients i, then $P_{V_i/U_i}(v/u) = A_{vu}, \forall i$. Hence

$$P(Q(U_i)|V_i = v) = \frac{\sum_{Q(u)} P_{U_i}(u)A_{vu}}{\sum_{Q(u)} P_{U_i}(u)A_{vu} + \sum_{\neg Q(u)} P_{U_i}(u)A_{vu}}$$

Let $\max_{Q(u')} A_{vu'} = maxp$ and $\min_{\neg Q(u')} A_{vu'} = minp$. As discussed in [20], the data distribution P_{U_i} in the worst case can be such that $P(U_i = u) > 0$ only if $\{u \in I_U | Q(u) \text{ and } A_{vu} = maxp\}$ or

$\{u \in I_U | \neg Q(u) \text{ and } A_{vu} = minp\}$ so that

$$P(Q(U_i)/V_i = v) = \frac{P(Q(u)) \cdot maxp}{P(Q(u)) \cdot maxp + P(\neg Q(u))minp}$$

Since the distribution P_U is known through reconstruction to the miner, and matrix A is fixed,

the above posterior probability can be determined by the miner. For example, if $P(Q(u)) = 5\%, \gamma = 19$, the posterior probability can be computed to be 50% for perturbation with the gamma-diagonal matrix.

But, in the randomized matrix case where $P_{V_i/U_i}(v/u)$ is a realization of random variable $\tilde{A}$, only its distribution and not the exact value for a given i is known to the miner. Thus determinations like the above cannot be made by the miner for a given record U_i. For example, suppose we choose matrix A such that

$$A_{uv} = \begin{cases} \gamma x + r, & \text{if } u = v \\ x - \frac{r}{|S_U|-1}, & \text{o.w.} \end{cases}$$

where $x = \frac{1}{\gamma+(|S_U|-1)}$ and r is a random variable uniformly distributed between $[-\alpha, \alpha]$. Thus, the worst case posterior probability for a record U_i is now a function of the value of r, and is given by

$$\psi_2(r) = \frac{P(Q(u)) \cdot \gamma x + r}{P(Q(u)) \cdot (\gamma x + r) + P(\neg Q(u))(x - \frac{r}{|S_U|-1})}$$

Therefore, only the posterior probability *range*, i.e. $[\psi_2^-, \psi_2^+] = [\psi_2(-\alpha), \psi_2(+\alpha)]$, and the distribution over the range, can be determined by the miner. For example, for the situation $P(Q(u)) = 5\%, \gamma = 19, \alpha = \gamma x/2$, he can only say that the posterior probability lies in the range $[33\%, 60\%]$ with its probability of being greater than 50% (ψ_2 corresponding to $r = 0$) equal to its probability of being less than 50%.

4.3.2 Reconstruction Model

The reconstruction model for the deterministic perturbation matrix A was discussed in Section 4.1.3. We now describe the changes to this analysis for the randomized perturbation matrix $\tilde{A}$.

The probability of success for Bernoulli variable Y_v^i is now modified to

$$P(Y_v^i = 1) = \tilde{A}_{vu}^i, \text{ for } U_i = u$$

where $\tilde{A}^i_{vu}$ denotes the i^{th} realization of random variable $\tilde{A}_{vu}$.

Thus, from Equation 4.4,

$$\begin{aligned} E(Y_v) &= \sum_{i=1}^{N} P(Y_v^i = 1) \\ &= \sum_{u \in I_U} \sum_{\{i|U_i=u\}} \tilde{A}^i_{vu} \\ &= \sum_{u \in I_U} \overline{A}_{vu} X_u \qquad (4.19) \\ \Rightarrow E(Y) &= \overline{A}_{vu} X \qquad (4.20) \end{aligned}$$

where $\overline{A}_{vu} = \frac{1}{X_u} \sum_{\{i|U_i=u\}} \tilde{A}^i_{vu}$ is the average of the values taken by $\tilde{A}_{vu}$ for the clients whose original data record had value u.

$\tilde{A}_{vu}$ is a random variable with expectation $E(\tilde{A}_{vu}) = A_{vu}$, it can be easily seen that,

$$E(\overline{A}_{vu}) = A_{vu} \qquad (4.21)$$

Hence, from Equation 4.19, we get

$$E(E(Y)) = AX \qquad (4.22)$$

We estimate X as $\widehat{X}$ given by the solution of following equation

$$Y = A\widehat{X} \qquad (4.23)$$

which is an approximation to Equation 4.22. From *Theorem 1* in Section 4.1.3, the error in estimation is bounded by:

$$\frac{\| \widehat{X} - X \|}{\| X \|} \leq c \frac{\| Y - E(E(Y)) \|}{\| E(E(Y)) \|} \qquad (4.24)$$

where c is the condition number of perturbation matrix A.

We now compare these bounds with the corresponding bounds of the deterministic case. Firstly, note that, due to the use of the randomized matrix, there is a *double expectation* for Y on the RHS of the inequality, as opposed to the single expectation in the deterministic case. Secondly, only the numerator is different between the two cases since $E(E(Y)) = AX$. Now,

we have

$$\begin{aligned} \| Y - E(E(Y)) \| &= \| (Y - E(Y)) + (E(Y) - E(E(Y))) \| \\ &\leq \| Y - E(Y) \| + \| E(Y) - E(E(Y)) \| \end{aligned}$$

Here $\| Y - E(Y) \|$ is given by the variance of random variable Y. Since Y_v, as discussed before, is Poisson-binomial distributed, its variance is given by [38]

$$Var(Y_v) = N\overline{p}_v - \sum_i (p_v^i)^2 \tag{4.25}$$

where $\overline{p}_v = \frac{1}{N}\sum_i p_v^i$ and $p_v^i = P(Y_v^i = 1)$.

It is easily seen (by elementary calculus or induction) that among all combinations $\{p_v^i\}$ such that $\sum_i p_v^i = n\overline{p}_v$, the sum $\sum_i (p_v^i)^2$ assumes its minimum value when all p_v^i are equal. It follows that, if the average probability of success $\overline{p}_v$ is kept constant, $Var(Y_v)$ assumes its maximum value when $p_v^1 = \cdots = p_v^N$. In other words, the variability of p_v^i, or *its lack of uniformity, decreases the magnitude of chance fluctuations*, as measured by its variance [21]. On using random matrix $\tilde{A}$ instead of deterministic A we increase the variability of p_v^i (now p_v^i assumes variable values for all i), hence decreasing the fluctuation of Y_v from its expectation, as measured by its variance.

Hence, $\| Y - E(Y) \|$ is likely to be decreased as compared to the deterministic case, thereby reducing the error bound.

On the other hand, the positive value $\| E(Y) - E(E(Y)) \| = \| (\overline{A} - A)X \|$, which depends upon the variance of the random variables in $\tilde{A}$, was 0 in the deterministic case. Thus, the error bound is increased by this term.

So, we have a classic trade-off situation here, and as shown later in our experiments of Section 4.6, the trade-off turns out very much in our favor with the two opposing terms almost canceling each other out, making the error only marginally worse than the deterministic case.

4.4 Implementation of Perturbation Algorithm

To implement the perturbation process discussed in the previous sections, we effectively need to generate for each $U_i = u$, a discrete distribution with PMF $P(v) = A_{vu}$ and CDF $F(v) = \sum_{i \leq v} A_{iu}$, defined over $v = 1, \ldots, \mid S_V \mid$.

A straightforward algorithm for generating the perturbed record v from the original record u is the following

1. Generate $r \sim \mathcal{U}(0, 1)$
2. Repeat for $v = 1, \ldots, \mid S_V \mid$

   if $F(v-1) < r \leq F(v)$

   return $V_i = v$

where $\mathcal{U}(0, 1)$ denotes uniform distribution over range $[0, 1]$.

This algorithm, whose complexity is proportional to the *product* of the cardinalities of the attribute domains, will require $\mid S_V \mid /2$ iterations on average which can turn out to be very large. For example, with 31 attributes, each with two categories, this amounts to 2^{30} iterations for each customer! We therefore present below an alternative algorithm whose complexity is proportional to the *sum* of the cardinality of the attribute domains.

Given that we want to perturb the record $U_i = u$, we can write

$$
\begin{aligned}
& P(V_i; U_i = u) \\
& = P(V_{i1}, \ldots, V_{iM}; u) \\
& = P(V_{i1}; u) \cdot P(V_{i2} | V_{i1}; u) \cdots P(V_{iM} | V_{i1}, \ldots, V_{i(M-1)}; u)
\end{aligned}
$$

For the perturbation matrix A, we get the following expressions for the above probabilities:

$$\begin{aligned} P(V_{i1} = a; u) &= \sum_{\{v|v(1)=a\}} A_{vu} \\ P(V_{i2} = b|V_{i1} = a; u) &= \frac{P(V_{i2} = b, V_{i1} = a; u)}{P(V_{i1} = a); u} \\ &= \frac{\sum_{\{v|v(1)=a \& v(2)=b\}} A_{vu}}{P(V_{i1} = a; u)} \\ & \ldots \text{and so on} \end{aligned}$$

where $v(i)$ denotes value of i^{th} column for record value =v.

For the gamma-diagonal matrix A, and using n_j to represent $\prod_{k=1}^{j} \mid S_U^k \mid$, we get the following expressions for these probabilities after some simple algebraic calculations:

$$\begin{aligned} P(V_{i1} = b; U_{i1} = b) &= (\gamma + \frac{n_M}{n_1} - 1)x \\ P(V_{i1} = b; U_{i1} \neq b) &= \frac{n_M}{n_1}x \end{aligned}$$

Then, for the j^{th} attribute

$$P(V_{ij}/V_{i1}, \ldots, V_{i(j-1)}; U_i) = \begin{cases} \frac{(\gamma + \frac{n_M}{n_j} - 1)x}{\prod_{k=1}^{j-1} p_k}, & \text{if } \forall k \leq j, V_{ik} = U_{ik} \\ \\ \frac{(\frac{n_M}{n_j})x}{\prod_{k=1}^{j-1} p_k}, & \text{o.w.} \end{cases} \tag{4.26}$$

where p_k is the probability that V_{ik} takes value a, given that a is the outcome of the random process performed for k^{th} attribute. i.e.

$$p_k = P(V_{ik} = a/V_{i1}, \ldots, V_{i(k-1)}; U_i)$$

Therefore, to achieve the desired random perturbation for a value in column j, we use as input both its original value and the perturbed value of the previous column $j - 1$, and generate the perturbed value as per the discrete distribution given in Equation 4.26. Note

that is an example of *dependent column perturbation*, in contrast to the independent column perturbation used in most of the prior techniques.

To assess the complexity, it is easy to see that the average number of iterations for the j^{th} discrete distribution will be $|S_U^j|/2$, and hence the average number of iterations for generating a perturbed record will be $\sum_j |S_U^j|/2$ (this value turns out to be exactly M for a boolean database).

4.5 Application to Association Rule Mining

To illustrate the utility of the FRAPP framework, we demonstrate in this section how it can be used for enhancing privacy-preserving mining of *association rules*, a popular mining model that identifies interesting correlations between database attributes [1, 31].

The core of the association rule mining is to identify "frequent itemsets", that is, all those itemsets whose support (i.e. frequency) in the database is in excess of a user-specified threshold. Equation 4.8 can be directly used to estimate the support of itemsets containing all M categorical attributes. However, in order to incorporate the reconstruction procedure into bottom-up association rule mining algorithms such as *Apriori* [2], we need to also be able to estimate the supports of itemsets consisting of only a *subset* of attributes.

Let $\mathcal{C}$ denotes the set of all attributes in the database, and $\mathcal{C}_s$ be a subset of attributes. Each of the attributes $j \in \mathcal{C}_s$ can assume one of the $|S_U^j|$ values. Thus, the number of itemsets over attributes in $\mathcal{C}_s$ is given by $n_{\mathcal{C}_s} = \prod_{j \in \mathcal{C}_s} |S_U^j|$. Let $\mathcal{L}, \mathcal{H}$ denote itemsets over this subset of attributes.

We say that record supports an itemset $\mathcal{L}$ over $\mathcal{C}_s$ if the entries in the record for the attributes $j \in \mathcal{C}_s$ are same as in $\mathcal{L}$.

Let *support* of an itemset $\mathcal{L}$ in original and distorted database be denoted by $sup_{\mathcal{L}}^U$ and $sup_{\mathcal{L}}^V$, respectively. Then,

$$sup_{\mathcal{L}}^V = \frac{1}{N} \sum_{v \text{ supports } \mathcal{L}} Y_v$$

where Y_v denotes the number of records in V with value v (refer Section 4.1.3). From Equation 4.7, we know

$$Y_v = \sum_{u \in I_U} A_{vu} \widehat{X}_u \tag{4.27}$$

Hence,

$$\begin{aligned} sup_{\mathcal{L}}^{V} &= \frac{1}{N} \sum_{v \text{ supports } \mathcal{L}} \sum_{u} A_{vu} \widehat{X}_u \\ &= \frac{1}{N} \sum_{u} \widehat{X}_u \sum_{v \text{ supports } \mathcal{L}} A_{vu} \\ &= \frac{1}{N} \sum_{\mathcal{H}} \sum_{u \text{ supports } \mathcal{H}} \widehat{X}_u \sum_{v \text{ supports } \mathcal{L}} A_{vu} \end{aligned}$$

If for all u which support a given itemset $\mathcal{H}$, $\sum_{v \text{ supports } \mathcal{L}} A_{vu} = \mathcal{A}_{\mathcal{H}\mathcal{L}}$, i.e. it is equal for all u which support a given itemset, then the above equation can be written as:

$$\begin{aligned} sup_{\mathcal{L}}^{V} &= \frac{1}{N} \sum_{\mathcal{H}} \mathcal{A}_{\mathcal{H}\mathcal{L}} \sum_{u \text{ supports } \mathcal{H}} \widehat{X}_u \\ &= \sum_{\mathcal{H}} \mathcal{A}_{\mathcal{H}\mathcal{L}} \quad \widehat{sup^{U}}_{\mathcal{H}} \end{aligned}$$

Now we find the matrix $\mathcal{A}$ for our gamma-diagonal matrix. Through some simple algebra, we get following matrix $\mathcal{A}$ corresponding to itemsets over subset $\mathcal{C}_s$, Hence,

$$\mathcal{A}_{\mathcal{H}\mathcal{L}} = \begin{cases} \gamma x + (\frac{n_{\mathcal{C}}}{n_{\mathcal{C}_s}} - 1)x, & \text{if } \mathcal{H} = \mathcal{L} \\ \frac{n_{\mathcal{C}}}{n_{\mathcal{C}_s}} x, & \text{o.w.} \end{cases} \tag{4.28}$$

Using the above $n_{\mathcal{C}_s} \times n_{\mathcal{C}_s}$ matrix we can estimate support of itemsets over any subset $\mathcal{C}_s$ of attributes. Thus our scheme can be implemented on popular bottom-up association rule mining algorithms.

4.6 Performance Analysis

We move on, in this section, to quantify the utility of the FRAPP framework with respect to the privacy and accuracy levels that it can provide for mining frequent itemsets.

Table 4.1: CENSUS Dataset description

Attribute	*Categories*
age	$(15-35], (35-55], (55-75], >75$
fnlwgt	$(0-1e5], (1e5-2e5], (1e5-3e5], (3e5-4e5], >4e5$
hours-per-week	$(0-20], (20-40], (40-60], (60-80], >80$
race	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
sex	Female, Male
native-country	United-States, Other

Table 4.2: HEALTH Dataset description

Attribute	*Categories*
AGE (Age)	$[0-20), [20-40), [40-60), [60-80), \geq 80)$
BDDAY12 (Bed days in past 12 months)	$[0-7), [7-15), [15-30), [30-60), \geq 60$
DV12 (Doctor visits in past 12 months)	$[0-7), [7-15), [15-30), [30-60), \geq 60$
PHONE (Has Telephone)	Yes,phone number given; Yes, no phone number given; No
SEX (Sex)	Male ; Female
INCFAM20 (Family Income)	Less than $20,000; $20,000 or more
HEALTH (Health status)	Excellent; Very Good; Good; Fair; Poor

Datasets. We use the following real world datasets in our experiments:

CENSUS : This dataset contains census information for approximately 50,000 adult American citizens. It is available from the UCI repository [44], and is a popular benchmark in data mining studies. It is also representative of a database where there are fields that users may prefer to keep private – for example, the "race" and "sex" attributes. We use three continuous (`age, fnlwgt, hours-per-week`) and three nominal

attributes (`native-country, sex, race`) from the census database in our experiments. The continuous attributes are partitioned into (five) equiwidth intervals to convert them into categorical attributes. The categories used for each attribute are listed in Table 4.1.

HEALTH : This dataset captures health information for over 100,000 patients collected by the US government [45]. We selected 3 continuous and 4 nominal attributes from the dataset for our experiments. The continuous attributes were partitioned into equi-width intervals to convert them into categorical attributes. The attributes and their categories are listed in Table 4.2.

We evaluated the association rule mining accuracy of our schemes on the above datasets for $sup_{min} = 2\%$. Table 4.3 gives the number of frequent itemsets in the datasets for $sup_{min} = 2\%$.

Table 4.3: CENSUS & HEALTH Dataset : Frequent Itemsets for $sup_{min} = 0.02$

	Itemset Length						
	1	2	3	4	5	6	7
CENSUS	19	102	203	165	64	10	–
HEALTH	23	123	292	361	250	86	12

Privacy Metric. The (ψ_1, ψ_2) strict privacy measure from [20] is used as the privacy metric. While we experimented with a variety of privacy settings, due to space limitations, we present results here for a sample $(\psi_1, \psi_2) = (5\%, 50\%)$, which was also used in [20]. This privacy value results in $\gamma = 19$.

Accuracy Metrics. The association rule mining errors can be quantified in terms of *Support Error* ρ and *Identity Error* (false positives σ^+, false negatives σ^-), as explained in Section 2.3. We use these as accuracy metrics in our experiments.

Perturbation Mechanisms. We show frequent-itemset-mining accuracy results for our proposed perturbation mechanisms as well as representative prior techniques. For all the perturbation mechanisms, the mining from the distorted database was done using *Apriori* [2] algorithm, with an additional support reconstruction phase at the end of each pass to recover the original supports from the perturbed database supports computed during the pass [28, 8].

The perturbation mechanisms evaluated in our study are the following:

DET-GD: This schemes uses the deterministic gamma-diagonal perturbation matrix A (Section 4.2) for perturbation and reconstruction. The implementation described in Section 4.4 was used to carry out the perturbation, and the results of Section 4.5 were used to compute the perturbation matrix used in each pass of Apriori for reconstruction.

RAN-GD: This scheme uses the randomized gamma-diagonal perturbation matrix $\tilde{A}$ (Section 4.3) for perturbation and reconstruction. Though in general, any distribution can be used for $\tilde{A}$, here we evaluate the performance of uniformly distributed $\tilde{A}$ given by Equation 4.19 over the entire range of the randomization parameter α.

MASK: This is the perturbation scheme proposed in [28], which is intended for boolean databases and is characterized by a single parameter $1-p$, which determines the probability of an attribute value being flipped. In our scenario, the categorical attributes are mapped to boolean attributes by making each value of the category an attribute. Thus, the M categorical attributes map to $M_b = \sum_j \mid S_U^j \mid$ boolean attributes.

The flipping probability $1-p$ was chosen as the lowest value which could satisfy the constraints given by Equation 4.2

The constraint $\forall v : \forall u_1, u_2 : \frac{A_{vu_1}}{A_{vu_2}} \leq \gamma$ is satisfied for MASK [28], if $\frac{p^{M_b}}{(1-p)^{M_b}} \leq \gamma$. But, for each categorical attribute, one and only one of its associated boolean attributes takes value 1 in a particular record. Therefore, all the records contain exactly M 1^s, and the following condition is sufficient for the privacy constraints to be satisfied:

$$\frac{p^{2M}}{(1-p)^{2M}} \leq \gamma$$

. This equation was used to determine the appropriate value of p. Value of p turns out be 0.5610 and 0.5524 respectively for CENSUS and HEALTH datasets for $\gamma = 19$.

C&P: This is the Cut-and-Paste perturbation scheme proposed in [19], with algorithmic parameters K and p. To choose K, we varied K from 0 to M, and for each K, p was chosen such that the matrix (Equation 4.12) satisfies the privacy constraints (Equation 4.2). The results reported here are for the (K, p) combination giving the best mining accuracy. For $\gamma = 19$ $K = 3, p = 0.494$ turn out to be appropriate values.

4.6.1 Experimental Results

For the CENSUS dataset, the support (ρ) and identity (σ^-, σ^+) errors of the four perturbation mechanisms (DET-GD, RAN-GD, MASK, C&P) is shown in Figure 4.1, as a function of the length of the frequent itemsets. The corresponding graphs for the HEALTH dataset are shown in Figure 4.2. In this graph for comparison, the performance of RAN-GD is shown for randomization parameter $\alpha = \gamma x/2$. Note that the support error (ρ) is plotted on a *log-scale*.

In these figures, we first note that the performance of the DET-GD method is visibly better than that of MASK and C&P. In fact, as the length of the frequent itemset increases, the performance of both MASK and C&P degrades drastically. MASK is not able to find any itemsets of length above 4 for the CENSUS dataset, and above 5 for the HEALTH dataset, while C&P does not works after 3-length itemsets.

The second point to note is that the accuracy of RAN-GD, although dealing with a randomized matrix, is only marginally lower than that of DET-GD. In return, it provides a substantial increase in the privacy – its worst case (determinable) privacy breach is only 33% as compared to 50% with DET-GD. Figure 4.3 and 4.4 shows performance of RAN-GD over entire range of α, and the posterior probability range $[\psi_2^-, \psi_2^+]$. We can observe that the performance of RAN-GD does not deviate much from the deterministic case over the entire range, where as very low *determinable* posterior probability can be obtained for higher values of α.

The primary reason for DET-GD and RAN-GD's good performance is the low *condition*

number of their perturbation matrices. This is quantitatively shown in Figure 4.1(d) and Figure 4.2(d), which compares the condition numbers (on a *log-scale*) of the reconstruction matrices. Note that as the expected value of random matrix $\tilde{A}$ is used for estimation in RAN-GD, and the random matrix used in experiments has expected value A (refer Equation 4.19) used in DET-GD, the condition numbers for two methods are equal. Here we see that the condition number for DET-GD and RAN-GD is not only low but also constant over all lengths of frequent itemsets (as mentioned before, the condition number is equal to $1 + \frac{|S_U|)}{(\gamma-1)}$). In marked contrast, the condition number for MASK and C&P increase *exponentially* with increasing itemset length, resulting in drastic degradation in accuracy. Thus our choice of a gamma-diagonal matrix shows highly promising results for discovery of long patterns.

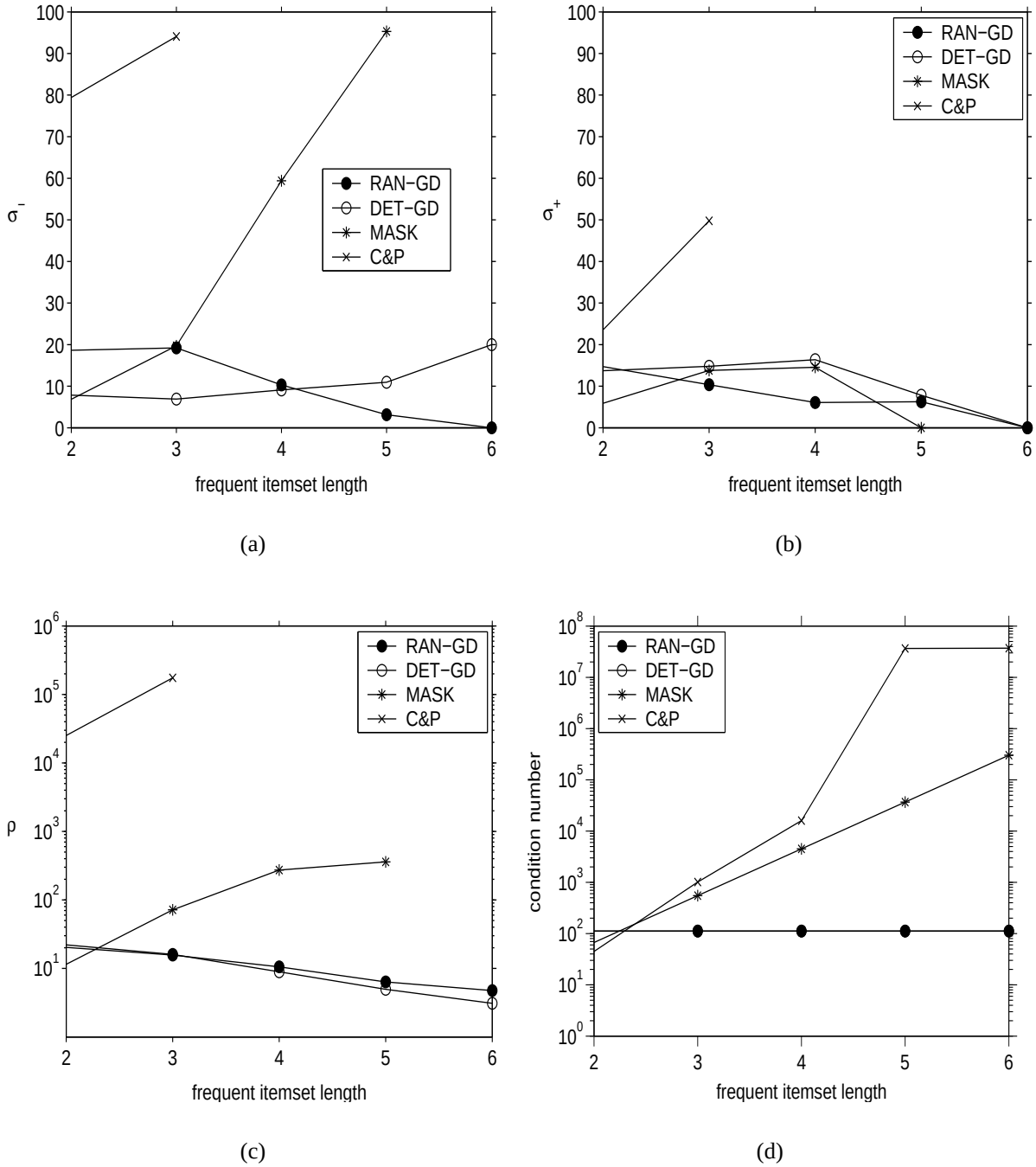
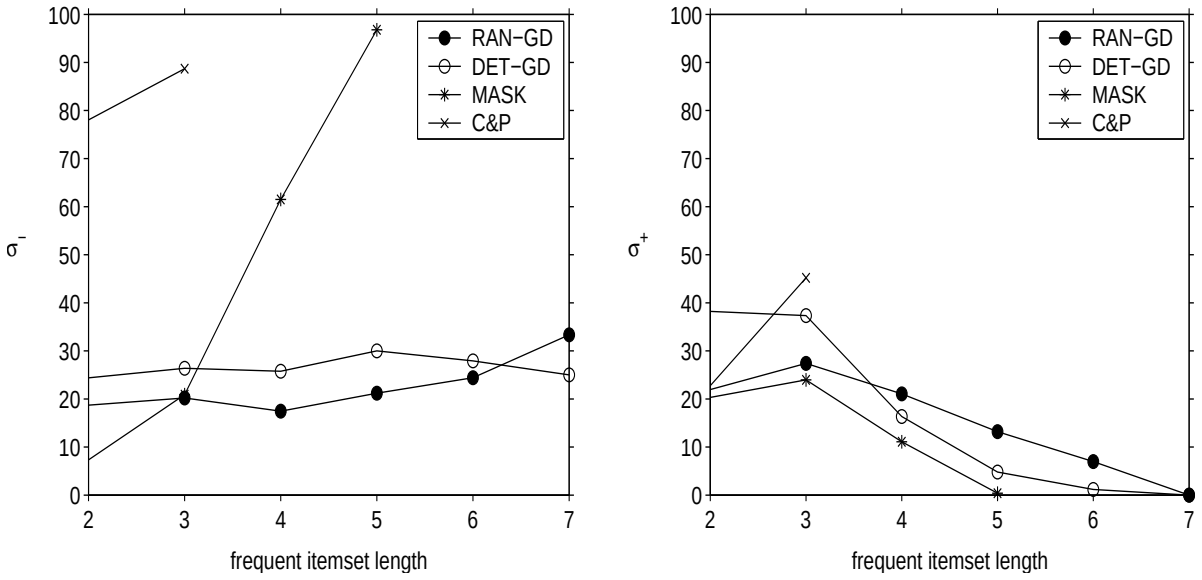
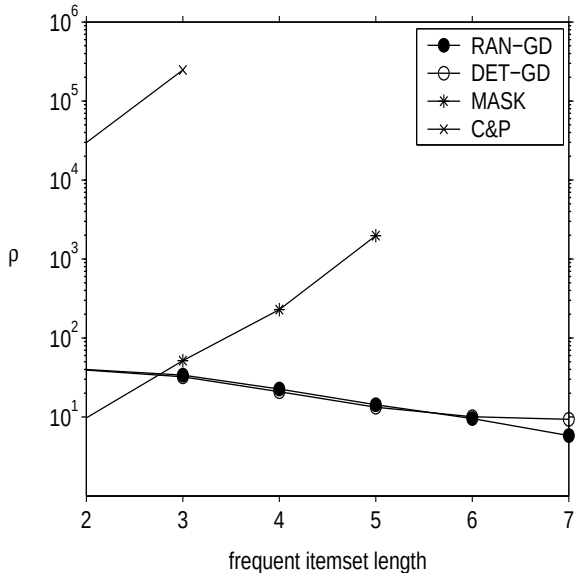


Figure 4.1: CENSUS Dataset mining accuracy results at $sup_{min} = 2\%$ (a) False negatives σ^- (b) False positives σ^+ (c) Support error ρ (d) Condition numbers

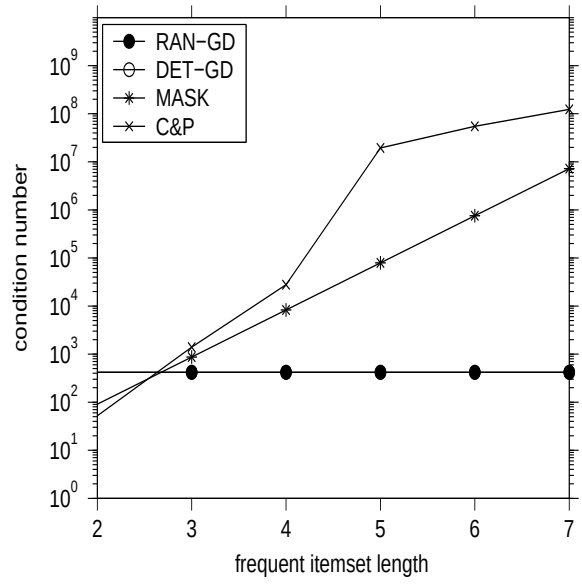


(a)

(b)

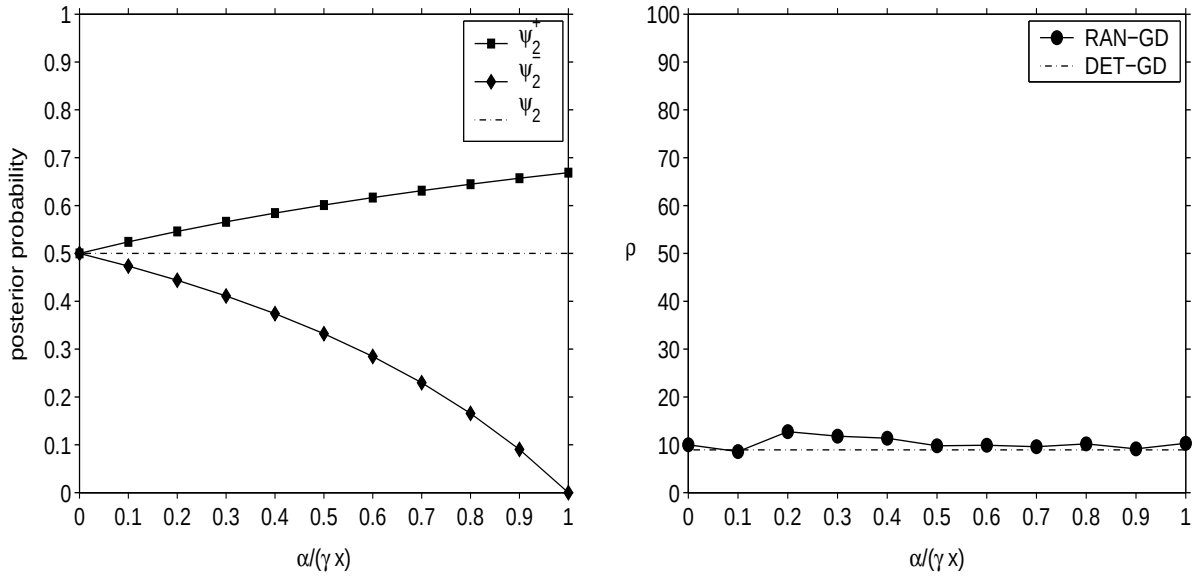


(c)

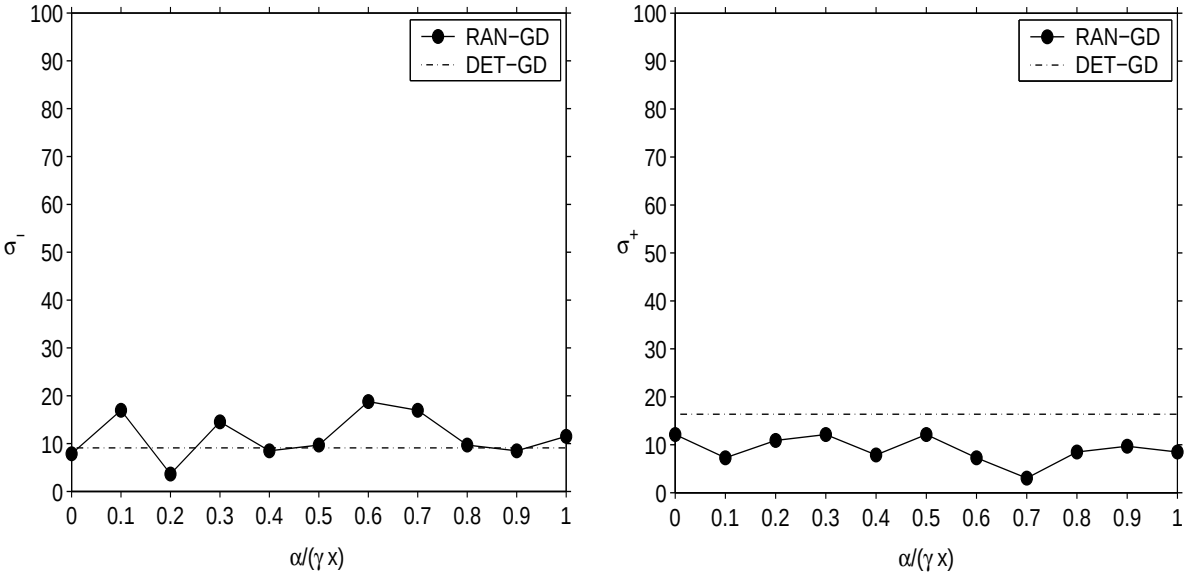


(d)

Figure 4.2: HEALTH Dataset mining accuracy results at $sup_{min} = 2\%$ (a) False negatives σ^- (b) False positives σ^+ (c) Support error ρ (d) Condition numbers



(a) (b)



(c) (d)

Figure 4.3: CENSUS Dataset (a) Posterior probability ranges (b) Support error ρ (c) False negatives σ^- (d) False positives σ^+ for itemset length 4 by RAN-GD with varying degree of randomization

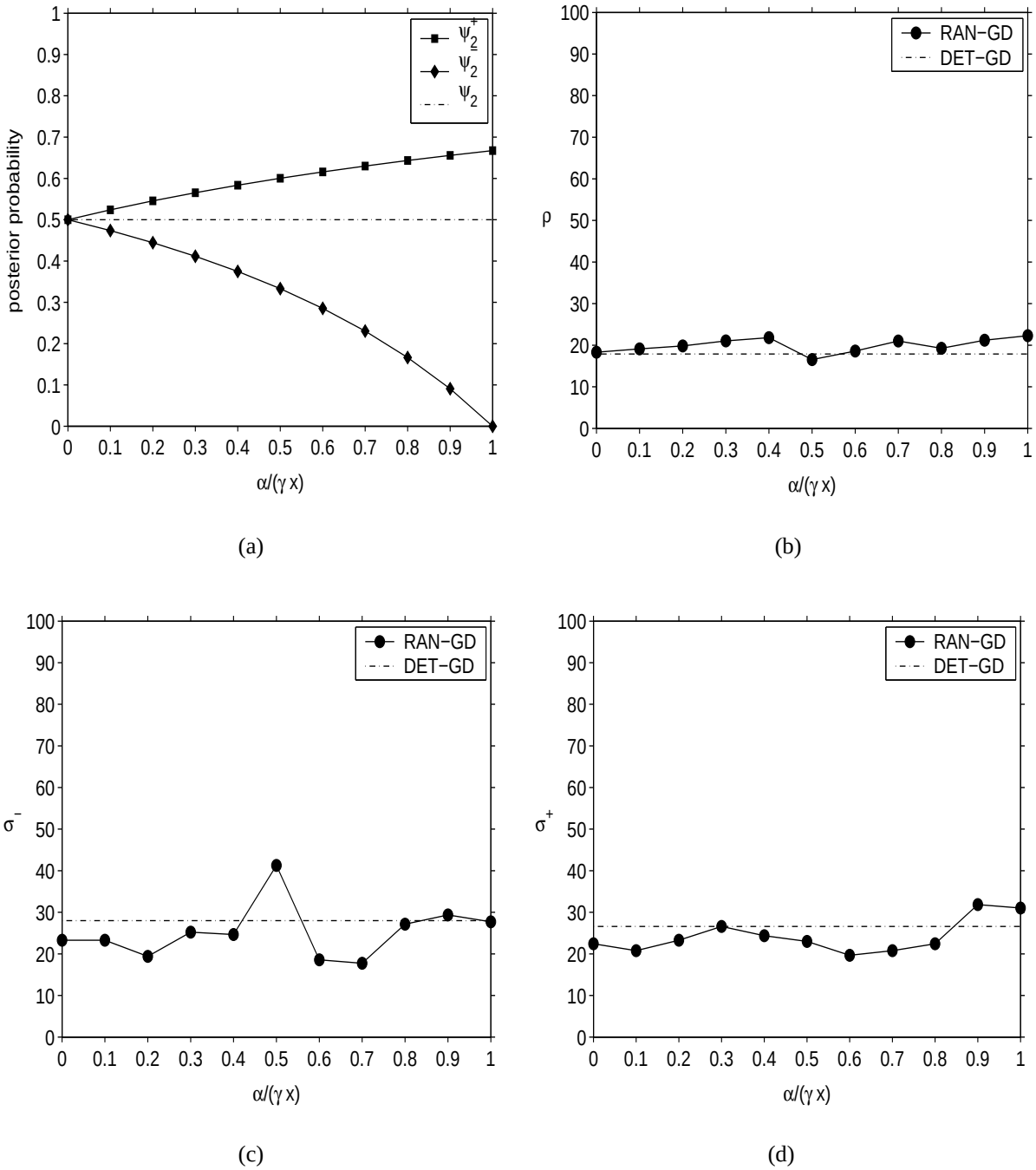


Figure 4.4: HEALTH Dataset (a) Posterior probability ranges (b) Support error ρ (c) False negatives σ^- (d) False positives σ^+ for itemset length 4 by RAN-GD with varying degree of randomization

Chapter 5

Conclusions

In this work, we have considered, for the first time, the issue of providing efficiency in privacy-preserving mining. Our goal was to investigate the possibility of simultaneously achieving high privacy, accuracy and efficiency in the mining process. We first showed how the distortion process required for ensuring privacy can have a marked negative side-effect of hugely increasing mining runtime. Then, we presented our new EMASK algorithm that is specifically designed to minimize this side-effect through the application of symbol-specific distortion. We derived simple but effective formulas for estimating the performance beforehand and used optimization method to find the settings of the distortion parameters to get best privacy, accuracy and efficiency possible. We also presented a simple but powerful optimization by which all additional counting incurred by privacy preserving mining is moved to the end of each pass over the database.

Our experiments show that EMASK could simultaneously provide good privacy, accuracy and efficiency. Specifically, less than 4 times slowdown with respect to Apriori in conjunction with 70-plus privacies and 90-plus accuracies, were achieved. In summary, EMASK takes a significant step towards making privacy-preserving mining of association rules a viable enterprise.

In the second part of the work, we develop FRAPP : a generalized model for random-perturbation-based methods operating on categorical data under strict privacy constraints. We showed that by making careful choices of the model parameters and building perturbation

methods for these choices, order-of-magnitude improvements in accuracy could be achieved as compared to the conventional approach of first deciding on a method and thereby implicitly fixing the associated model parameters. In particular, we proved that a "gamma-diagonal" perturbation matrix is capable of delivering the best accuracy, and is in fact, optimal with respect to its condition number, which decides the estimation error. We presented an implementation technique for gamma-diagonal-based perturbation, whose complexity is proportional to the *sum* of the domain cardinalities of the attributes in the database. Empirical evaluation of our new gamma-diagonal-based techniques showed substantial reductions in frequent itemset identity and support reconstruction errors.

We also investigated the novel strategy of having the perturbation matrix composed of not fixed values, but random variables instead. Our analysis of this approach for association rule mining showed that, at a marginal cost in accuracy, significant improvements in privacy levels could be achieved.

In our future work, we plan to extend our modeling approach to other flavors of mining tasks.

References

[1] R. Agrawal, T. Imielinski and A. Swami, “Mining association rules between sets of items in large databases”, *Proc. of ACM SIGMOD Intl. Conference on Management of Data (SIGMOD)*, May 1993.

[2] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules”, *Proc. of 20th Intl. Conf. on Very Large Data Bases (VLDB)*, September 1994.

[3] R. Agrawal and R. Srikant, “Privacy-Preserving Data Mining”, *Proc. of ACM SIGMOD Intl. Conf. on Management of Data*, May 2000.

[4] R. Agrawal, J. Kiernan, R. Srikant and Y. Xu, “Hippocratic Databases”, *Proc. of 28th Intl. Conf. on Very Large Data Bases (VLDB)*, 2002.

[5] R. Agrawal, A. Kini, K. LeFevre, A. Wang, Y. Xu and D. Zhou, “Managing Healthcare Data Hippocratically”, *Proc. of ACM SIGMOD Intl. Conf. on Management of Data*, 2004.

[6] R. Agrawal, R. Bayardo, C. Faloutsos, J. Kiernan, R. Rantzau and R. Srikant, “Auditing Compliance with a Hippocratic Database”, *Proc. of 30th Intl. Conf. on Very Large Data Bases (VLDB)*, 2004.

[7] D. Agrawal and C. Aggarwal, “On the Design and Quantification of Privacy Preserving Data Mining Algorithms”, *Proc. of Symposium on Principles of Database Systems (PODS)*, 2001.

[8] S. Agrawal, V. Krishnan and J. Haritsa, “On Addressing Efficiency Concerns in Privacy-Preserving Mining”, *Proc. of 9th Intl. Conf. on Database Systems for Advanced Applications (DASFAA)*, March 2004.

[9] S. Agrawal and J. Haritsa, “A Framework for High-Accuracy Privacy-Preserving Mining”, *Tech. Rep. TR-2004-02, DSL/SERC, Indian Institute of Science*, 2004. http://dsl.serc.iisc.ernet.in/pub/TR/TR-2004-02.pdf

[10] C. Aggarwal and P. Yu, “A Condensation Approach to Privacy Preserving Data Mining”, *Proc. of 9th Intl. Conf. on Extending DataBase Technology (EDBT)*, March 2004

[11] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim and V. Verykios, “Disclosure Limitation of Sensitive Rules”, *Proc. of IEEE Knowledge and Data Engineering Exchange Workshop (KDEX)*, November 1999.

[12] Y. Censor, “Pareto Optimality in Multiobjective Problems”, *Appl. Math. Optimiz.*, vol. 4, 1977.

[13] L.F. Cranor, J. Reagle, and M.S. Ackerman. “Beyond concern: Understanding net users’ attitudes about online privacy”. *Technical Report TR 99.4.3, AT&T Labs-Research*, April 1999.

[14] Lorrie Faith Cranor, editor. *Special Issue on Internet Privacy*. Comm. A12M, 42(2), Feb. 1999.

[15] Da Cunha, N.O. and E. Polak, “Constrained Minimization Under Vector-valued Criteria in Finite Dimensional Spaces,”*J. Math. Anal. Appl., Vol. 19, pp. 103-124,* 1967.

[16] E. Dasseni, V. Verykios, A. Elmagarmid and E. Bertino, “Hiding Association Rules by Using Confidence and Support”, *Proc. of 4th Intl. Information Hiding Workshop (IHW)*, April 2001.

[17] The Economist, “The End of Privacy”, May 1999.

[18] The European Union's Directive on Privacy Protection, October 1998. Available from http :// ww. echo. lu/legal/en/dataprot/ direct iv/direct iv. htlal.

[19] A. Evfimievski, R. Srikant, R. Agrawal and J. Gehrke, "Privacy Preserving Mining of Association Rules", *Proc. of 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, July 2002.

[20] A. Evfimievski, J. Gehrke and R. Srikant, "Limiting Privacy Breaches in Privacy Preserving Data Mining", *Proc. of ACM Symp. on Principles of Database Systems (PODS)*, June 2003.

[21] W. Feller, "An Introduction to Probability Theory and its Applications (Vol. I)", Wiley, 1988.

[22] F. Gembicki, "Vector Optimization for Control with Performance and Parameter Sensitivity Indices," PhD Dissertation, Case Western Reserve Univ., Cleveland, Ohio, 1974.

[23] C. Hine and J. Eve. "Privacy in the marketplace", The Information Society, 42(2):56-59, 1998.

[24] M. Kantarcioglu and C. Clifton, "Privacy-preserving Distributed Mining of Association Rules on Horizontally Partitioned Data", *Proc. of ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD)*, June 2002.

[25] H. Kargupta, S. Datta, Q. Wang and K. Sivakumar, "On the Privacy Preserving Properties of Random Data Perturbation Techniques", *Proc. of the Intl. Conf. on Data Mining (ICDM)*, 2003.

[26] K. LeFevre, R. Agrawal, V. Ercegovac, R. Ramakrishnan, Y. Xu and D. DeWitt, "Limiting Disclosure in Hippocratic Databases", *Proc. of 30th Intl. Conf. on Very Large Data Bases (VLDB)*, 2004.

[27] Office of the Information and Privacy Commissioner, Ontario. "Data Mining: Staking a Claim on Your Privacy, January 1998". Available from http ://www.ipc.on.ca/website.eng/matters/sum.pap/papers/datamine.htm

[28] S. Rizvi and J. Haritsa, ”Maintaining Data Privacy in Association Rule Mining”, *Proc. of 28th Intl. Conf. on Very Large Databases (VLDB)*, August 2002.

[29] Y. Saygin, V. Verykios and C. Clifton, “Using Unknowns to Prevent Discovery of Association Rules”, *ACM SIGMOD Record*, vol. 30, no. 4, 2001.

[30] Y. Saygin, V. Verykios and A. Elmagarmid, “Privacy Preserving Association Rule Mining”, *Proc. of 12th Intl. Workshop on Research Issues in Data Engineering (RIDE)*, February 2002.

[31] R. Srikant and R. Agrawal, “Mining Quantitative Association Rules in Large Relational tables”, *Proc. of ACM SIGMOD Intl. Conf. on Management of data*, 1996.

[32] G. Strang, “Linear Algebra and its Applications”, Thomson Learning Inc., 1988.

[33] P. Samarati and L. Sweeney, “Protecting Privacy when Disclosing Information: k-Anonymity and its enforcement through generalization and suppression”, *Proc. of the IEEE Symposium on Research in Security and Privacy*, 1998.

[34] Time, “The Death of Privacy”, August 1997.

[35] J. Vaidya and C. Clifton, “Privacy-preserving k-means clustering over vertically partitioned data”, *Proc. of 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, 2003.

[36] J. Vaidya and C. Clifton, “Privacy Preserving Naive Bayes Classifier for Vertically Partitioned Data”, *Proc. of SDM*, 2004.

[37] J. Vaidya and C. Clifton, “Privacy Preserving Association Rule Mining in Vertically Partitioned Data”, *Proc. of 8th ACM SIKGDD Intl. Conference on Knowledge Discovery and Data Mining (KDD)*, July 2002.

[38] Y. Wang, “On the number of successes in independent trials”, *Statistica Silica 3* (1993).

[39] A.F. Westin. “E-commerce and privacy: What net users want”. *Technical report, Louis Har- ris & Associates*, June 1998. Available from http ://www .privaeyexehange. org/iss/ surveys/eeommsum, html.

[40] A.F. Westin. “Privacy concerns & consumer choice”. *Technical report, Louis Har- ris & Associates*, Dec. 1998. Available from ht tp://.ww, privacyexchange, org/iss/ surveys/1298toc, html.

[41] A.F. Westin. “Freebies and privacy: What net users think”. *Technical report, Opinion Research Corporation*, July 1999. Available from http://www, privaeyexchange, org/iss/ surveys/sr990714, html.

[42] L. Zadeh, “Optimality and Nonscalar-valued Performance Criteria,” *IEEE Trans. Automat. Contr.*, AC-8, 1963.

[43] Z. Zheng, R. Kohavi and L. Mason, “Real World Performance of Association Rule Algorithms”, *Proc. of 7th ACM SIGKDD Intl. Conference on Knowledge Discovery and Data Mining (KDD)*, August 2001.

[44] *http://www.ics.uci.edu/ mlearn/MLRepository.html*

[45] *http://dataferrett.census.gov*