Mixture of Gaussian Models and Bayes Error under Differential Privacy

Bowei Xi
Department of Statistics
Purdue University
xbw@purdue.edu

Murat Kantarcıoğlu
Dept. of Computer Science
University of Texas at Dallas
muratk@utdallas.edu

Ali Inan
Dept. of Computer Eng.
Isik University
Istanbul, Turkey
ali.inan@isikun.edu.tr

ABSTRACT

Gaussian mixture models are an important tool in Bayesian decision theory. In this study, we focus on building such models over statistical database protected under differential privacy. Our approach involves querying necessary statistics from a database and building a Bayesian classifier over the noise added responses generated according to differential privacy. We formally analyze the sensitivity of our query set. Since there are multiple methods to query a statistic, either directly or indirectly, we analyze the sensitivities for different querying methods. Furthermore we establish theoretical bounds for the Bayes error for the univariate (one dimensional) case. We study the Bayes error for the multivariate (high dimensional) case in experiments with both simulated data and real life data. We discover that adding Laplace noise to a statistic under certain constraint is problematic. For example variance-covariance matrix is no longer positive definite after noise addition. We propose a heuristic method to fix the noise added variance-covariance matrix.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Multivariate statistics; H.2.7 [**Database Management**]: Database Administration—*Security, integrity, and protection*; H.2.8 [**Database Management**]: Database Applications—*Statistical databases*

General Terms

Algorithms, Experimentation, Security

Keywords

Differential Privacy, Statistical Databases, Mixture Models, Classification

1. INTRODUCTION

Mixture models are widely used, theoretically mature tools in statistical pattern recognition and pattern classification [2, 7]. The basic assumption behind mixture models is that the data are obtained by sampling a population consisting of several distinct sub-populations with their own distributions. Gaussian mixture models refer to the case where each model follows multivariate normal (Gaussian) distribution.

Mixture models are suitable for both unsupervised learning (e.g., clustering using the Expectation Maximization algorithm) and supervised learning (e.g., classification using the Bayes' decision rule). In this study, we assume that the records of an input data set belongs to different categories and focus on the classification task. Various studies have established tight bounds on Bayes classification error such as the Chernoff and Bhattacharyya bounds. We investigate the problem of building Gaussian mixture models in a privacy-preserving environment and try to establish similar bounds under differential privacy as the privacy protection mechanism.

Building Gaussian mixture models over a specific data set requires obtaining the mean vector and the covariance matrix for each class/category. This is often a straightforward task. However, when the data set in question contains sensitive information, special care has to be taken. Consider the following motivating scenario. A medical researcher believes that a certain disease (e.g., diabetes mellitus) can be diagnosed based on a series of attributes (e.g., blood pressure, weight, height, blood sugar, etc.) that is assumed to follow multivariate normal distribution and is recorded for every patient admitted to a hospital. The researcher would like to build a Gaussian mixture model and empirically test this belief using the resulting classifier. Yet, the hospital database contains highly sensitive information (e.g., disease history of the patient) and should prevent direct access to the data, even for research purposes.

Instead of granting direct access, the data users (i.e., the researcher in our example) are provided with a sanitized view of the database containing private information[1]. Various alternative privacy protection mechanisms have been suggested for producing a sanitized view. Among the first were anonymization methods such as k-anonymity [14], ℓ-diversity [12], and t-closeness [11]. Anonymization methods try to break the association between data records and individuals by grouping together similar records. Once the groups are formed, through generalization, suppression or partitioning [15] a sanitized version of the data set is released to the data user. Most definitions of anonymity (e.g.,

[1] Unless the data are distributed across multiple parties, methods based on Secure Multi-party Computation (SMC) do not apply here.

CODASPY'11, February 21–23, 2011, San Antonio, Texas, USA.

k-anonymity, ℓ-diversity, etc.) differ in the way the groups are formed.

Anonymization methods protect privacy only against adversaries with certain background information. Dwork proves in [3] that every privacy protection mechanism is vulnerable to some kind of background knowledge and "bad disclosures" might occur regardless of participation into the attacked database. Therefore, Dwork suggests that instead of tailoring privacy definitions against different types of background knowledge, one should minimize the risk of disclosure that arises from participation into a database. This notion is captured by the *differential privacy* protection mechanism [3]. Differential privacy restricts the access to a statistical interface, where users can only issue aggregate statistical queries to the database and the responses are perturbed with random noise. The magnitude of the noise depends on the privacy parameter (e.g., ϵ in ϵ-differential privacy) and sensitivity of the set of queries. Sensitivity is a function of the query set and not the database. As shown in [16], computing the sensitivity is NP-hard.

In this paper, we develop a privacy preserving method of building a Bayesian classifier for the mixture of Gaussian models. This is achieved by modeling the underlying database as a statistical database protected with differential privacy against disclosures, and querying necessary statistics from the database to build the classifier over the noisy responses. Main contributions of this work are as follows:

1. Sensitivity of statistical queries are formally analyzed. More accurate or exact bounds for sensitivity are established.

2. For the univariate (one dimensional) Gaussian case, we establish theoretical bounds on Bayes error under differential privacy based on the Bhattacharyya bound [2].

3. We show the applicability of our methods and examine the Bayes error for the multivariate (high dimensional) Gaussian case through experiments, using both simulated data and real-world data.

4. We propose a heuristic method to fix the noise added variance-covariance matrix, which is no longer positive definite and cannot be directly used in building a Bayesian classifier.

The rest of the paper is organized as follows. We formally define the problem in Section 1.1 and provide a brief overview of differential privacy as a protection mechanism in Section 1.2. Related work in the area is discussed in Section 2. In Section 3, we calculate the sensitivity of various query sets that retrieve necessary statistics from the database. Since the exact value of sensitivity depends on the number of records, our calculation is in terms of the database size. Then, in Section 4, we establish theoretical bounds on the Bayes error under differential privacy as the privacy protection mechanism. Section 5 gives experimental results and finally Section 6 concludes our discussion and presents future directions of research.

1.1 Problem Definition

Let $D = \{A_1, \cdots, A_d\}$ be a d-dimensional database such that the domain $Dom(A_i)$ of each attribute A_i, $i = 1, ..., d$, is continuous and bounded. For the analysis of sensitivity in Section 3, we assume that each domain is normalized to the range $[0, 1]$ to simplify the expression of sensitivity. Assume the database D is comprised of n records. Without loss of generality, we assume that D is represented as a relation. Then the value of attribute A_i of record x_k, $k = 1, ..., n$, is denoted by $x_k[A_i]$.

We are interested in building mixture of Gaussian models over databases D that fit the above description. When privacy is not a concern, this is a straightforward task. Without delving into too much details of Gaussian mixture models, let us restrict the discussion to the following: one only needs to compute the expected values of each attribute A_i and the variance-covariance matrix Σ:

$$\Sigma_{ij} = cov(A_i, A_j) = E[(A_i - \mu_i)(A_j - \mu_j)],$$

where $\mu_i = E(A_i)$. More details follow in Section 4.

In our definition of the problem, we consider a database D that contains privacy-sensitive information that is protected through differential privacy. This provides us with a statistical database interface. The interface answers aggregate queries only (e.g. count, sum etc.) and to each response adds random noise [3, 5]. In what follows, we briefly review differential privacy and analyze the sensitivities of certain queries.

1.2 Differential Privacy

Given a set of queries $Q = \{Q_1, ..., Q_q\}$, differential privacy adds Laplace noise with λ magnitude to the true response. Magnitude λ is determined by two parameters: privacy parameter ϵ and query set sensitivity $S(Q)$. Here, ϵ is assumed to be set by the data curator (i.e. the party that holds the database D). Sensitivity $S(Q)$, on the other hand, is a function of the query set Q.

Sensitivity of a query set is defined over all possible pairs of databases that differ in only one record, referred to as sibling databases.

$$S(Q) = \max_{\forall \text{ sibling databases } D_1, D_2} \sum_{i=1}^{q} |Q_i^{D_1} - Q_i^{D_2}| \tag{1}$$

That is, sensitivity of Q is the maximum difference in the total L_1 norm that a single record update can possibly cause in the query responses. Notice that the definition is independent of the original database D.

Once ϵ and $S(Q)$ are known, λ can be set such that $\lambda \geq S(Q)/\epsilon$ to facilitate uninterrupted querying[2]. The rest is straightforward. In response to each query Q_i, the database first computes the result Q_i^D over all records in D and then adds Laplace noise to obtain the noisy response R_i^D:

$$R_i^D = Q_i^D + r, \tag{2}$$

where $r \sim \text{Laplace}(\lambda)$. Obviously, the key to designing accurate differential privacy mechanism is to minimize the sensitivity $S(Q)$. In our problem definition, the query set Q is already fixed. However, there are multiple methods to query a statistic. Therefore we examine the sensitivities for different query approaches separately.

[2] If Q is not available ahead of the time and therefore $S(Q)$ cannot be computed, λ will be fixed heuristically. In such scenarios, the database must keep track of the sensitivity of the queries answered so far. If the pre-specified sensitivity threshold λ is exceeded, the database simply stops responding.

2. RELATED WORK

Gaussian mixture models are classical models that are widely used in practice [2, 7]. Despite their popularity in practice, so far, privacy issues related to building mixture models have received little attention. Merugu et al. propose in [13] that instead of perturbing original data to protect privacy, in distributed settings, statistical information describing mixture models can be released. The basic idea is to generate data samples based on mixture models and run data mining tasks over the samples. However, as discussed by Kantarcioglu et al. in [9], releasing (non-perturbed) two-class mixture models might violate individual privacy. Our approach is motivated by the results of [9].

Privacy preserving data mining has been studied extensively in recent years. Initial works in the area consisted mostly of two approaches: 1) perturbation methods (e.g., random noise addition method by Agrawal et al.[1]); 2) anonymization methods (e.g., k-anonymity method proposed by Sweeney [14]) that yield a *sanitized* version of the original data set. However, successful attack strategies against proposed solutions in both directions necessitated new definitions of privacy and anonymity. For example, Kargupta et al. shows in [10] that the random noise added according to [1] could be problematic since "in many cases the original data can be accurately estimated from the perturbed data". Similarly, ℓ-diversity [12] presents an attack scenario against k-anonymity definition of [14] based on lack of diversity over sensitive attributes. Such vulnerabilities have lead to the definition of differential privacy [3]. Dwork proves in [3] that for every privacy definition, there exists some background knowledge that results in disclosure of sensitive information and therefore violation of individual privacy. Consequently, a new and much stronger privacy definition that minimizes the risk of disclosure irrespective of attendance to a database is proposed, namely, differential privacy.

Differential privacy [3] models the database as a statistical database that only responds to statistical queries and adds to the responses random noise, whose magnitude is proportional to the privacy parameter ϵ and the sensitivity of the query set. Here, sensitivity is a function of the query set and not the database in question.

Various different formulations of differential privacy have been suggested. Initial definitions of sensitivity operate over sibling data sets that have the same size but differ in only record (i.e., one data set can be mapped to another by updating only one record) [3, 5]. Some later studies consider insertion of a new record when defining sibling data sets [4]. The distinction between the two approaches might appear minor. However, for most query sets, the prior definition asks for sensitivity computations twice that of the later. We follow [3] in our sensitivity computations.

Sensitivity calculations of many important functions are analyzed in [5], including some statistics used in this paper as well. However, the bounds achieved by [5] are admittedly crude. Dwork et al. calculate the sensitivity of querying the mean vector as $2\gamma/n$, where n is the number of records in the database and $\gamma = \max_x ||v(x)||_1$ (i.e., the maximum L_1 norm of any record). We establish the exact sensitivity on the same query, which equals to one half of the previously established bound: d/n, where d represents the dimensionality (i.e., the number of attributes)[3]. Similarly, [5] crudely calculates the sensitivity of the variance-covariance matrix Σ. Here, we provide a complete, more formal analysis of the sensitivity of the query retrieving Σ, and establish much tighter bounds.

Privacy preserving classification with differential privacy as the underlying privacy protection mechanism has received little attention so far. In [6], Friedman et al. presented a method of ID3 classification that builds a decision tree through recursive queries retrieving the information gain across an attribute and the partitioning mechanism. A different solution to ID3 classification by Jagannathan et al. [8] builds multiple random decision trees using sum queries. In this study, we present a Bayes classifier based on Gaussian mixture models by querying the mean vector and the covariance matrix for each class category. To the best of our knowledge, we are the first to explore Bayes error for Gaussian mixture models in detail under differential privacy as the protection mechanism.

3. SENSITIVITY AS FUNCTIONS OF SAMPLE SIZE AND DIMENSIONALITY

Assume two sibling databases D_1 and D_2 have n records each, and they differ by one record. Next we establish the sensitivity of queries given sample size n and d attributes. [5] provided upper bounds for the sensitivity of querying mean and variance-covariance matrix. [5] defined $\gamma = max||x'||_1$. Since all the attributes are normalized to $[0, 1]$, $\gamma = d$ in our setting. [5] showed that the sensitivity of directly querying the mean is smaller than or equal to $2d/n$, and the sensitivity of querying the variance-covariance matrix is smaller than or equal to $8d^2/n$. In this section we obtain the exact sensitivity of directly querying the mean, and indirectly through querying sum and sample size, or indirectly querying the median, which is the mean for symmetric distributions. We also obtain a much tighter upper bound for querying the variance-covariance matrix.

We notice there are multiple ways to query a statistic. For example, the value of sample mean can be obtained indirectly through the sample median for any symmetric distribution. The sample mean can also be obtained through the sum divided by the sample size. Users can attempt various methods to query a statistic and to reduce sensitivity. We discuss the different sensitivities associated with the different methods to query a statistic in this section. The following summarize the findings in this section:

1. The sensitivity of directly querying mean is d/n, which decreases with increasing sample size n.

2. The sensitivity of directly querying sum is d, not affected by the sample size n, so is the sensitivity of directly querying median.

3. Notice mean can be obtained indirectly through querying median for symmetric distributions, or through querying sum and sample size. These two indirect query methods for mean have sensitivity not affected by sample size.

[3] We assume that all domains are normalized to the range [0,1], therefore having the value of γ to be fixed, $\gamma = d$. This is a trivial task if the domains are bounded, which has to be the case since differential privacy requires a bounded domain.

4. Directly querying variance has sensitivity between $\frac{1}{n} - \frac{1}{n^2}$ and $\frac{3}{n} - \frac{3}{n^2}$, so does directly querying covariance. Directly querying variance-covariance matrix (upper triangle only) has sensitivity between $(\frac{1}{n} - \frac{1}{n^2})d(d+1)/2$ and $(\frac{3}{n} - \frac{3}{n^2})d(d+1)/2$.

3.1 Directly Querying Mean and Sum

We examine the sensitivity of directly querying the mean and the sum. These two statistics are closely related. One can be solved from another. Yet the sensitivity for querying these two statistics are quite different.

THEOREM 3.1. *Assume we have two sibling databases and each has n records, i.e.* $|D_1| = |D_2| = n$, *where sample size* $n \geq 1$. *Let* $Q = \{Mean_1, ..., Mean_d\}$, *where* $d \geq 1$. *Hence*

$$S(Q) = d/n.$$

Proof: Let $Mean_i^{(n-1)}$ be the mean of A_i over the common $n-1$ records shared by D_1 and D_2. Let the unique record in D_1 be x_1 and the unique record in D_2 be x_2. Then the mean values of A_i in D_1 and D_2 are

$$Mean_i^{(n),1} = \frac{(n-1) \times Mean_i^{(n-1)} + x_1[A_i]}{n},$$

$$Mean_i^{(n),2} = \frac{(n-1) \times Mean_i^{(n-1)} + x_2[A_i]}{n}.$$

We have

$$|Mean_i^{(n),1} - Mean_i^{(n),2}| = \frac{|x_1[A_i] - x_2[A_i]|}{n}.$$

Then we have

$$\begin{array}{rccc} & max_{\{D_1,D_2\}} \sum_1^d |Mean_i^{(n),1} - Mean_i^{(n),2}| & & \\ = & \left(max_{\{D_1,D_2\}} \sum_{i=1}^d |x_1[A_i] - x_2[A_i]|\right)/n & & \\ = & d/n & = & S(Q). \end{array}$$

When all the d attributes in the x_1 and x_2 differ by 1, we reach the maximum, which determines the sensitivity. ■

THEOREM 3.2. *Assume we have two sibling databases and each has n records, i.e.* $|D_1| = |D_2| = n$, *where sample size* $n \geq 1$. *Let* $Q = \{Sum_1, ..., Sum_d\}$, *where* $d \geq 1$. *Hence*

$$S(Q) = d.$$

Proof: Let $Sum_i^{(n-1)}$ be the sum of attribute A_i over the common $n-1$ records shared by D_1 and D_2. Again let the unique record in D_1 be x_1 and the unique record in D_2 be x_2. Then the sum of A_i in D_1 and D_2 are

$$Sum_i^{(n),1} = Sum_i^{(n-1)} + x_1[A_i],$$

$$Sum_i^{(n),2} = Sum_i^{(n-1)} + x_2[A_i].$$

When all the d attributes in the x_1 and x_2 differ by 1, we have

$$\begin{array}{rccc} & max_{\{D_1,D_2\}} \sum_1^d |Sum_i^{(n),1} - Sum_i^{(n),2}| & & \\ = & max_{\{D_1,D_2\}} \sum_{i=1}^d |x_1[A_i] - x_2[A_i]| & & \\ = & d & = & S(Q). \end{array}$$

■

The two theorems do not rely on the distribution of A_i over the interval $[0, 1]$. The sensitivity of $Q = \{Mean_1, ..., Mean_d\}$ improves linearly as the sample size n increases given a fixed d. It requires the sample size to be much larger than the dimensionality, $n >> d$, to have a small sensitivity. On the other hand increasing the sample size n will not improve the sensitivity of $Q = \{Sum_1, ..., Sum_d\}$, which is determined solely by dimensionality.

Since sensitivity is defined over all possible sibling databases with all possible sample sizes, the following corollary establishes the overall sensitivity of directly querying the mean.

COROLLARY 3.1. *Let* $Q = \{Mean_1, ..., Mean_d\}$, *where* $d \geq 1$. $S(Q) = d$, *for all possible pairs of sibling databases.*

Proof: Following Theorem 3.1, when we set n=1, we obtain the maximum change of L_1 norm over all possible sibling databases. The problem can be solved in a more straightforward fashion. Note $Mean_i$ has minimum value 0 and maximum value 1. Let D_1 and D_2 each contains 1 record. $x_1 = \vec{0}$ and $x_2 = \vec{1}$. Then D_1 has the minimum $Mean_i$ $\forall i = 1, ..., d$ and D_2 has the maximum $Mean_i$ $\forall i = 1, ..., d$. The maximum L_1 difference is $d = S(Q)$. ■

3.2 Directly Querying Median

For Gaussian distribution, or in general any symmetric distribution, median equals to mean. However the sensitivity of directly querying the median is quite different than that of directly querying the mean. The sensitivity of directly querying the median of d attributes is a constant d, same as directly querying the sum, regardless of sample size n.

THEOREM 3.3. *Let* $Q = \{\ Median_1, ..., Median_d\ \}$, *such that* $Median_i$ *retrieves the median of attribute* A_i. *Hence the overall sensitivity for for all possible pairs of sibling databases is:*

$$S(Q) = d.$$

Proof: First consider one attribute A_i. Since attribute A_i is normalized to interval $[0, 1]$, the minimum value of the median is 0 and the maximum is 1. Therefore, it is sufficient to show that there is a pair of sibling databases (D_1, D_2) such that the response to $Median_i$ shifts by 1.

Let database D_1 have $2m + 1$ records, $m \geq 0$, where

$$x_j[A_i] = \begin{cases} 0, & if\ 1 \leq j \leq m+1 \\ 1, & otherwise. \end{cases}$$

Construct database D_2 by changing the value of $x_m[A_i]$ from 0 to 1. Notice the response to $Median_i$ over D_1 is 0, while it is 1 over D_2, which achieves the maximum L_1 difference.

For $Q = \{Median_1, ..., Median_d\}$, similarly we let D_1 have

$$x_j = \begin{cases} \vec{0}, & if\ 1 \leq j \leq m+1 \\ \vec{1}, & otherwise. \end{cases}$$

Construct database D_2 by changing the value of x_{m+1} from $\vec{0}$ to $\vec{1}$. Hence the responses to the query over D_1 and D_2 achieve the maximum difference in L_1 norm. We conclude $S(Q) = d$, $\forall n \geq 1$. ■

3.3 Indirectly Querying Mean

There are multiple ways of estimating a statistic. For example, querying the median is equivalent to querying the mean for any symmetric distribution. Another choice is to issue two queries, one for sum and the other for sample size.

THEOREM 3.4. *Assume we have two sibling databases and each has n records, i.e.* $|D_1| = |D_2| = n$*, where sample size* $n \geq 1$*. Let* $Q = \{Sum_1, ..., Sum_d, SampleSize\}$*, where* $d \geq 1$*. Hence* $S(Q) = d$*.*

Proof: The query for sample size has sensitivity 0, since both D_1 and D_2 have the same sample size. Then we only need to consider the sensitivity of Sum_i. Similar to the proof of Theorem 3.2, we obtain $S(Q) = d$. ∎

3.4 Directly Querying Variance and Covariance

Next we examine the sensitivity of directly querying variance, covariance, and the whole variance-covariance matrix. We establish much tighter bounds for the sensitivity in this section.

THEOREM 3.5. *Assume we have two sibling databases and each has n records, i.e.* $|D_1| = |D_2| = n$*, where sample size* $n \geq 2$*. Without loss of generality let* $Q = \{Var_1\}$ *for attribute* A_1*. Then*

$$\frac{1}{n} - \frac{1}{n^2} \leq S(Q) \leq \frac{3}{n} - \frac{3}{n^2}.$$

Proof: Assume x_3, ..., x_{n+1} are the $n-1$ common records shared by the two databases D_1 and D_2. Let x_1 be the unique record in D_1 and x_2 be the unique record in D_2. Here we estimate the sample variance as the following:

$$Var_1 = \frac{1}{n}\sum_{i=1}^{n}(x_i[A_1] - \bar{x}[A_1])^2 = \frac{\sum_{i=1}^{n} x_i^2[A_1]}{n} - \bar{x}^2[A_1].$$

Let Var_1^i be the sample variance of database D_i, $i = 1, 2$. Then we have

$$\begin{aligned} & Var_1^1 - Var_1^2 \\ = & \left[\frac{\sum_{i=3}^{n+1} x_i^2[A_1] + x_1^2[A_1]}{n} - (\frac{\sum_{i=3}^{n+1} x_i[A_1] + x_1[A_1]}{n})^2\right] \\ - & \left[\frac{\sum_{i=3}^{n+1} x_i^2[A_1] + x_2^2[A_1]}{n} - (\frac{\sum_{i=3}^{n+1} x_i[A_1] + x_2[A_1]}{n})^2\right] \\ = & (x_1^2[A_1] - x_2^2[A_1])(\frac{1}{n} - \frac{1}{n^2}) \\ + & \frac{2(x_2[A_1] - x_1[A_1])(\sum_{i=3}^{n+1} x_i[A_1])}{n^2} \end{aligned}$$

When $x_i[A_1] = 0, i = 3, ..., n+1$, $x_1[A_1] = 1$, and $x_2[A_1] = 0$, we have

$$Var_1^1 - Var_1^2 = \frac{1}{n} - \frac{1}{n^2}.$$

This is a lower bound for $S(Q)$.

On the other hand we have

$$\begin{aligned} |Var_1^1 - Var_1^2| \leq & \left|(x_1^2[A_1] - x_2^2[A_1])(\frac{1}{n} - \frac{1}{n^2})\right| \\ + & \left|\frac{2(x_2[A_1] - x_1[A_1])(\sum_{i=3}^{n+1} x_i[A_1])}{n^2}\right| \end{aligned}$$

We obtain an upper bound by letting every component on the right hand side of the above inequality reach their maximum individually.

$$\begin{aligned} max|Var_1^1 - Var_1^2| \leq & 1 \times (\frac{1}{n} - \frac{1}{n^2}) + \frac{2 \times 1 \times (n-1)}{n^2} \\ = & \frac{3}{n} - \frac{3}{n^2} \end{aligned}$$

Therefore we have

$$\frac{1}{n} - \frac{1}{n^2} \leq S(Q) \leq \frac{3}{n} - \frac{3}{n^2}.$$

∎

THEOREM 3.6. *Assume we have two sibling databases and each has n records, i.e.* $|D_1| = |D_2| = n$*, where sample size* $n \geq 2$*. Without loss of generality let* $Q = \{Cov_{1,2}\}$ *for attributes* A_1 *and* A_2*. Then*

$$\frac{1}{n} - \frac{1}{n^2} \leq S(Q) \leq \frac{3}{n} - \frac{3}{n^2}.$$

Proof: Again assume x_3, ..., x_{n+1} are the $n-1$ common records shared by the two databases D_1 and D_2. Let x_1 be the unique record in D_1 and x_2 be the unique record in D_2. The sample covariance is the following:

$$\begin{aligned} Cov_{1,2} = & \frac{1}{n}\sum_{i=1}^{n}(x_i[A_1] - \bar{x}[A_1])(x_i[A_2] - \bar{x}[A_2]) \\ = & \frac{\sum_{i=1}^{n} x_i[A_1]x_i[A_2]}{n} - \bar{x}[A_1]\bar{x}[A_2]. \end{aligned}$$

We have the difference as

$$\begin{aligned} & Cov_{1,2}^1 - Cov_{1,2}^2 \\ = & \frac{\sum_{i=3}^{n+1} x_i[A_1]x_i[A_2] + x_1[A_1]x_1[A_2]}{n} \\ - & (\frac{\sum_{i=3}^{n+1} x_i[A_1] + x_1[A_1]}{n}) \times (\frac{\sum_{i=3}^{n+1} x_i[A_2] + x_1[A_2]}{n}) \\ - & \frac{\sum_{i=3}^{n+1} x_i[A_1]x_i[A_2] + x_2[A_1]x_2[A_2]}{n} \\ + & (\frac{\sum_{i=3}^{n+1} x_i[A_1] + x_2[A_1]}{n}) \times (\frac{\sum_{i=3}^{n+1} x_i[A_2] + x_2[A_2]}{n}) \end{aligned}$$

Cleaning up the above expression we have

$$\begin{aligned} & Cov_{1,2}^1 - Cov_{1,2}^2 \\ = & (x_1[A_1]x_1[A_2] - x_2[A_1]x_2[A_2])\,(\frac{1}{n} - \frac{1}{n^2}) \\ - & (x_1[A_1] - x_2[A_1])\left(\frac{\sum_{i=3}^{n+1} x_i[A_2]}{n^2}\right) \\ - & (x_1[A_2] - x_2[A_2])\left(\frac{\sum_{i=3}^{n+1} x_i[A_1]}{n^2}\right) \end{aligned}$$

Let $x_i[A_1] = x_i[A_2] = 0$ for $i = 3, ..., n+1$, $x_1[A_1] = x_1[A_2] = 1$, and $x_2[A_1] = x_2[A_2] = 0$. We have $Cov_{1,2}^1 - Cov_{1,2}^2 = 1/n - 1/n^2$. Hence this is a lower bound of $S(Q)$.

We also have

$$\begin{aligned}
& |Cov^1_{1,2} - Cov^2_{1,2}| \\
\leq\ & |x_1[A_1]x_1[A_2] - x_2[A_1]x_2[A_2]|\,(\frac{1}{n} - \frac{1}{n^2}) \\
+\ & |x_1[A_1] - x_2[A_1]| \left| \frac{\sum_{i=3}^{n+1} x_i[A_2]}{n^2} \right| \\
+\ & |x_1[A_2] - x_2[A_2]| \left| \frac{\sum_{i=3}^{n+1} x_i[A_1]}{n^2} \right|
\end{aligned}$$

Let every component reach their maximum values, we have

$$\begin{aligned}
max|Cov^1_{1,2} - Cov^2_{1,2}| &\leq (\frac{1}{n} - \frac{1}{n^2}) + \frac{n-1}{n^2} + \frac{n-1}{n^2} \\
&= \frac{3}{n} - \frac{3}{n^2}.
\end{aligned}$$

Therefore we have

$$\frac{1}{n} - \frac{1}{n^2} \leq S(Q) \leq \frac{3}{n} - \frac{3}{n^2}.$$

∎

For large sample size n, the above result shows the sensitivity of a single variance or a single covariance decreases as $O(1/n)$. Next we consider querying the whole variance-covariance matrix.

Theorem 3.7. *Assume we have two sibling databases and each has n records, i.e.* $|D_1| = |D_2| = n$, *where sample size* $n \geq 2$. *Without loss of generality let* $Q = \{\Sigma\}$ *for d attributes. We consider only the upper triangle. Then*

$$(\frac{1}{n} - \frac{1}{n^2})\frac{d(d+1)}{2} \leq S(Q) \leq (\frac{3}{n} - \frac{3}{n^2})\frac{d(d+1)}{2}.$$

Proof: Again assume x_3, ..., x_{n+1} are the $n-1$ common records shared by the two databases D_1 and D_2. Let x_1 be the unique record in D_1 and x_2 be the unique record in D_2. We follow the thread in the above two theorems. Then we have

$$\begin{aligned}
& |Q^1 - Q^2| \\
=\ & \sum_{k=1}^{d-1}\sum_{l=k+1}^{d} |(x_1[A_k]x_1[A_l] - x_2[A_k]x_2[A_l])(\frac{1}{n} - \frac{1}{n^2}) \\
-\ & (x_1[A_k] - x_2[A_k]) \left(\frac{\sum_{i=3}^{n+1} x_i[A_l]}{n^2} \right) \\
-\ & (x_1[A_l] - x_2[A_l]) \left(\frac{\sum_{i=3}^{n+1} x_i[A_k]}{n^2} \right) | \\
+\ & \sum_{k=1}^{d} |(x_1^2[A_k] - x_2^2[A_k])(\frac{1}{n} - \frac{1}{n^2}) \\
-\ & 2(x_1[A_k] - x_2[A_k]) \left(\frac{\sum_{i=3}^{n+1} x_i[A_k]}{n^2} \right) |
\end{aligned}$$

When $x_3 = ... = x_{n+1} = \vec{0}$, $x_2 = \vec{0}$, and $x_1 = \vec{1}$, we have the above sum equal to $(\frac{1}{n} - \frac{1}{n^2})\frac{d(d+1)}{2}$. This forms a lower bound of $S(Q)$. We also have

$$\begin{aligned}
& |Q^1 - Q^2| \\
\leq\ & \sum_{k=1}^{d-1}\sum_{l=k+1}^{d} \{\ |x_1[A_k]x_1[A_l] - x_2[A_k]x_2[A_l]| \times (\frac{1}{n} - \frac{1}{n^2}) \\
+\ & |x_1[A_k] - x_2[A_k]| \times \left| \frac{\sum_{i=3}^{n+1} x_i[A_l]}{n^2} \right| \\
+\ & |x_1[A_l] - x_2[A_l]| \times \left| \frac{\sum_{i=3}^{n+1} x_i[A_k]}{n^2} \right|\ \} \\
+\ & \sum_{k=1}^{d} \{\ |x_1^2[A_k] - x_2^2[A_k]| \times (\frac{1}{n} - \frac{1}{n^2}) \\
+\ & 2|x_1[A_k] - x_2[A_k]| \times |\frac{\sum_{i=3}^{n+1} x_i[A_k]}{n^2}|\ \}
\end{aligned}$$

Let each component reach their maximum values (i.e. $x_3 = ... = x_{n+1} = \vec{1}$), we have

$$max|Q^1 - Q^2| \leq (\frac{3}{n} - \frac{3}{n^2})\frac{d(d+1)}{2}.$$

Hence we establish an upper bound for $S(Q)$ too. Combining the lower and upper bounds we have:

$$(\frac{1}{n} - \frac{1}{n^2})\frac{d(d+1)}{2} \leq S(Q) \leq (\frac{3}{n} - \frac{3}{n^2})\frac{d(d+1)}{2}.$$

∎

We obtain a much tighter bound for querying the variance-covariance matrix. The above result indicates that in order to reduce sensitivity for querying the whole variance-covariance matrix, we need the sample size to be much larger than d^2, $n >> d^2$. Next as what we do for directly querying the mean, we can obtain an upper bound for the maximum change in L_1 norm for querying the variance-covariance matrix for all possible sibling databases with all possible sample sizes. The following establishes an upper bound for the overall sensitivity of directly querying the variance-covariance matrix.

Corollary 3.2. *Let* $Q = \{\Sigma\}$, *where* Σ *retrieves the variance-covariance matrix.* $S(Q) \leq 3d(d+1)/8$.

Proof: We let $n = 2$ in the upper bound specified by Theorem 3.7. We then obtain the overall upper bound for all possible sample size: $S(Q) \leq 3d(d+1)/8$. ∎

The primary reason behind high overall sensitivity in Corollaries 3.1 and 3.2 calculations is the small sample size of the databases. Even though any databases that will be used to build Gaussian mixture models would contain thousands if not millions of records, by definition sensitivity is calculated over all possible sibling databases.

3.5 Multiple Querying Methods for A Statistic and The Effect on Sensitivity

Different methods to issue the queries for the same statistic are associated with very different sensitivity values. To obtain the sample mean, we can query the median instead if the attribute is from a symmetric distribution, or we can query the sum and the sample size. Based on the above theorems, we discover that querying the median or the sum together with sample size has sensitivity d, which is not affected by sample size n. Directly querying the mean has sensitivity d/n, fast approaching 0 as sample size increases. Some indirect queries can result in high sensitivity.

There are also alternative methods to issues a set of queries to construct variance, covariance, and a variance-covariance matrix, instead of directly querying the statistics. For example, for attribute A_1, we can query the sums and the sample size, i.e. $\sum_{i=1}^n x_i[A_1]$, $\sum_{i=1}^n x_i^2[A_1]$, and n. Another method is to query the means, i.e. $(\sum_{i=1}^n x_i[A_1])/n$ and $(\sum_{i=1}^n x_i^2[A_1])/n$. We then construct the variance from the sums or the means. However querying the sums and querying the means have very different sensitivity values.

While working with differential privacy, we usually try to come up with query methods that will perturb the results as little as possible. However, most accurate results need not be computed with query sets of smaller sensitivities. Comparing the direct query for mean in Corollary 3.1 and the indirect query in Theorem 3.4, we observe the indirect query is more resilient to noise. Any positive or negative noise with magnitude larger than 1 completely disguise the mean value retrieved by direct querying (as in Corollary 3.1). Yet Laplace distribution has support over $(-\infty, \infty)$. The conclusion we would like to draw is that, directly querying a statistic may not always be the best idea, especially for databases with small sample size.

Later in simulation we assume databases have relatively large sample size and we apply sensitivity values of directly querying the mean and variance-covariance matrix, after adjusting for the range.

4. BAYES ERROR OF GAUSSIAN MIXTURE MODELS UNDER DIFFERENTIAL PRIVACY

Let $D = \{A_1, \ldots, A_d, W\}$ be a database of n records, where W represents a binary class attribute with the domain

$$Dom(W) = \{w_1, w_2\},$$

and each attribute A_i, $1 \leq i \leq d$ represents a continuous attribute with the domain $Dom(A_i) = \mathbf{R}$.

Our purpose is to build a classifier using D that, given a non-classified record in terms of a d-dimensional feature vector $\mathbf{x} \in \mathbf{R}^d$, assigns a class value to $\mathbf{x}$ such that the probability of mis-classification

$$P(error|\mathbf{x}) = \begin{cases} P(w_1|\mathbf{x}) & \text{if } \mathbf{x} \in w_2 \\ P(w_2|\mathbf{x}) & \text{if } \mathbf{x} \in w_1 \end{cases}$$

is minimized. The following Bayes' decision rule describes one such classifier:

$$\text{Assign } w_1 \text{ if } P(w_1|\mathbf{x}) > P(w_2|\mathbf{x}) \text{ ; otherwise assign } w_2. \tag{3}$$

Here, the probabilities $P(w_i|\mathbf{x})$ can easily be calculated based on Bayes' theorem:

$$P(w_i|\mathbf{x}) = \frac{p(\mathbf{x}|w_i)P(w_i)}{p(\mathbf{x})}.$$

The specific case where $p(\mathbf{x}|w_i)$ has multivariate normal (Gaussian) density is known as the "mixture of Gaussian models" problem and it has been studied extensively due to its tractability [2]. For each class value w_i, the mean μ_i and the covariance matrix Σ_i of the distribution of $p(\mathbf{x}|w_i) \sim N(\mu_i, \Sigma_i)$ are estimated from the data set D. Based on the parameters of these distributions, the feature space $\mathbf{R}^d$ can be partitioned into possibly disconnected decision regions $\mathcal{R}_i$ such that $\mathbf{x} \in \mathcal{R}_i$ implies $\mathbf{x}$ will be classified as w_i.

The Bayes error is calculated by integrating the probability of incorrect decision(s) over decision regions. For binary classification, this implies [2]:

$$\begin{aligned} \text{Bayes Error} &= P(\mathbf{x} \in R_1, w_2) + P(\mathbf{x} \in R_2, w_1) \\ &= P(\mathbf{x} \in R_1|w_2)P(w_2) + P(\mathbf{x} \in R_2|w_1)P(w_1) \\ &= \int_{\mathcal{R}_1} p(\mathbf{x}|w_2)P(w_2)d\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}|w_1)P(w_1)d\mathbf{x} \end{aligned}$$

In mixture of Gaussian models, such error can be bounded from above using the *Chernoff* bound or the *Bhattacharyya* bound as explained in [2]. Among these two approaches, the Chernoff bound is never looser than the Bhattacharyya but computationally more complex.

Our purpose is to calculate similar error bounds for privacy preserving Gaussian mixture models. Specifically, data set D will act as a statistical database that only responds to aggregate queries about the records. Using differential privacy as the underlying privacy protection mechanism, all responses to the queries will be perturbed with independent Laplace noise $\mathcal{L}(0, \lambda)$, where $\lambda \geq S(Q)/\epsilon$ is the magnitude of the added noise, $S(Q)$ is the sensitivity of the query set issued to the database (as defined in [3]) and ϵ is the privacy parameter.

In order to build a Gaussian mixture model, the query set Q including the following statistical information has to be issued to the database D:

- The number of records in D (sensitivity of this query is 0),
- The distribution of class values / categories (i.e., $P(w_1)$ and $P(w_2)$),
- For each category, parameters of the multivariate Gaussian distribution (i.e., $p(\mathbf{x}|w_i)$) in terms of μ_i and Σ_i.

4.1 Truncated Gaussian Distribution

Differential privacy works well for bounded variables. For unbounded variables one extremely large or small record has the ability to cause an extremely large change in any statistic queried and inflate the sensitivity. However Gaussian distribution has support over the entire real line. Assume we truncate a Gaussian variable to interval $[\mu - k\sigma, \mu + k\sigma]$ and the original Gaussian variable $X \sim N(\mu, \sigma^2)$ has density $f(x)$. The truncated Gaussian variable has density:

$$I_{\{\mu - k\sigma \leq x \leq \mu + k\sigma\}}(x) \frac{f(x)}{Z(k) - Z(-k)},$$

where $Z(\cdot)$ is the cumulative distribution function of the standard normal variable, and $I_{\mu - k\sigma \leq x \leq \mu + k\sigma}(x)$ is an indicator function. If we choose sufficiently large k, $Z(k) - Z(-k)$ is almost 1, and the truncated Gaussian variable and the genuine Gaussian variable have almost identical properties, such as density, mean, variance etc. We notice a Gaussian variable has probability 0.999999998 to fall into the bounded interval $[\mu - 6\sigma, \mu + 6\sigma]$. Therefore in the simulation study we choose $k = 6$.

4.2 One Dimensional Bayes Error Bound

We can obtain an upper bound for the one dimensional Bayes error with Gaussian mixture models under differential privacy for binary classes. Assume class $\omega_1 \sim N(\mu_1, \sigma_1^2)$ and class $\omega_2 \sim N(\mu_2, \sigma_2^2)$. Further assume class ω_1 has n_1 records and class ω_2 has n_2 records. First note the *Bhattacharyya* bound [2] states that

$$\text{Bayes Error} \le \sqrt{P(\omega_1)P(\omega_2)}e^{-K}, \tag{4}$$

where

$$K = \frac{1}{4}\frac{\mu_1^2 + \mu_2^2 - 2\mu_1\mu_2}{\sigma_1^2 + \sigma_2^2} + \frac{\log(\sigma_1^2 + \sigma_2^2)}{2} - \frac{\log(4\sigma_1^2\sigma_2^2)}{4}. \tag{5}$$

Considering the Laplace noises added to the queries of mean and variances in each class, we have the following theorem.

THEOREM 4.1. *The Gaussian mixture models are as specified above. Assume under differential privacy the query responses are the sample means and the sample variances plus independent Laplace noises:*

$$\hat{\mu}_1 = \bar{x}_1 + r_1,\ \hat{\mu}_2 = \bar{x}_2 + r_2,\ \hat{\sigma}_1^2 = S_1^2 + r_3,\ \hat{\sigma}_2^2 = S_2^2 + r_4.$$

Since there are multiple ways to query a statistic, we simply assume the independent Laplace noises $r_i \sim L(0, \lambda_i)$ for a general result. We have for $0 < p < 1$,

$$P(K^L(p) < K < K^U(p)) = p^8,$$

and

$$Pr(\text{Bayes Error} < \sqrt{P(\omega_1)P(\omega_2)}e^{-K^L(p)}) \ge p^8,$$

where

$$\begin{aligned} K^U(p) \ &= \\ &\frac{\sum_{i=1}^2\{\mu_i + \sqrt{\frac{\sigma_i^2}{n_i}}Z(1-\frac{p}{2}) - \lambda_i\log(1-2|\frac{1-p}{2}|)\}^2}{4\{\sum_{i=1}^2 \frac{\sigma_i^2}{n_i}\chi^2_{n_i-1}(\frac{p}{2}) + \sum_{i=3}^4 \lambda_i\log(1-2|\frac{1-p}{2}|)\}} \\ &- \frac{\prod_{i=1}^2\{\mu_i - \sqrt{\frac{\sigma_i^2}{n_i}}Z(1-\frac{p}{2}) + \lambda_i\log(1-2|\frac{1-p}{2}|)\}^2}{2\{\sum_{i=1}^2 \frac{\sigma_i^2}{n_i}\chi^2_{n_i-1}(1-\frac{p}{2}) - \sum_{i=3}^4 \lambda_i\log(1-2|\frac{1-p}{2}|)\}} \\ &+ \frac{\log\{\sum_{i=1}^2 \frac{\sigma_i^2}{n_i}\chi^2_{n_i-1}(1-\frac{p}{2}) - \sum_{i=3}^4 \lambda_i\log(1-2|\frac{1-p}{2}|)\}}{2} \\ &- \frac{\log\{4\prod_{i=1}^2[\frac{\sigma_i^2}{n_i}\chi^2_{n_i-1}(\frac{p}{2}) + \lambda_{i+2}\log(1-2|\frac{1-p}{2}|)]\}}{4}, \end{aligned}$$

and

$$\begin{aligned} K^L(p) \ &= \\ &\frac{\sum_{i=1}^2\{\mu_i - \sqrt{\frac{\sigma_i^2}{n_i}}Z(1-\frac{p}{2}) + \lambda_i\log(1-2|\frac{1-p}{2}|)\}^2}{4\{\sum_{i=1}^2 \frac{\sigma_i^2}{n_i}\chi^2_{n_i-1}(1-\frac{p}{2}) - \sum_{i=3}^4 \lambda_i\log(1-2|\frac{1-p}{2}|)\}} \\ &- \frac{\prod_{i=1}^2\{\mu_i + \sqrt{\frac{\sigma_i^2}{n_i}}Z(1-\frac{p}{2}) - \lambda_i\log(1-2|\frac{1-p}{2}|)\}^2}{2\{\sum_{i=1}^2 \frac{\sigma_i^2}{n_i}\chi^2_{n_i-1}(\frac{p}{2}) + \sum_{i=3}^4 \lambda_i\log(1-2|\frac{1-p}{2}|)\}} \\ &+ \frac{\log\{\sum_{i=1}^2 \frac{\sigma_i^2}{n_i}\chi^2_{n_i-1}(\frac{p}{2}) + \sum_{i=3}^4 \lambda_i\log(1-2|\frac{1-p}{2}|)\}}{2} \\ &- \frac{\log\{4\prod_{i=1}^2[\frac{\sigma_i^2}{n_i}\chi^2_{n_i-1}(1-\frac{p}{2}) - \lambda_{i+2}\log(1-2|\frac{1-p}{2}|)]\}}{4}. \end{aligned}$$

$Z(r)$ is the r quantile of the standard normal distribution. $\chi^2_{n-1}(r)$ is the r quantile of χ^2_{n-1}. $\lambda\log(1-2|\frac{1-p}{2}|)$ and $-\lambda\log(1-2|\frac{1-p}{2}|)$ are $p/2$ and $(1-p/2)$ quantile of Laplace distribution $L(0, \lambda)$.

Proof: Since both classes follow Gaussian distribution, we have the following distribution for the sample means and the sample variances:

$$\begin{aligned} \bar{x}_1 &\sim N(\mu_1, \frac{\sigma_1^2}{n_1}), \\ \bar{x}_2 &\sim N(\mu_2, \frac{\sigma_2^2}{n_2}), \\ \frac{n_1S_1^2}{\sigma_1^2} = \frac{\sum_{i=1}^{n_1}(x_{1,i} - \bar{x}_1)^2}{\sigma_1^2} &\sim \chi^2_{n_1-1}, \\ \frac{n_2S_2^2}{\sigma_2^2} = \frac{\sum_{i=1}^{n_2}(x_{2,i} - \bar{x}_2)^2}{\sigma_2^2} &\sim \chi^2_{n_2-1}. \end{aligned}$$

Note the sample means and the sample variances are independent. Also note we add independent Laplace noises $r_i \sim L(0, \lambda_i)$,

$$\hat{\mu}_1 = \bar{x}_1 + r_1,\ \hat{\mu}_2 = \bar{x}_2 + r_2,\ \hat{\sigma}_1^2 = S_1^2 + r_3,\ \hat{\sigma}_2^2 = S_2^2 + r_4.$$

With probability p (for example $p = 0.90,\ 0.95$, etc.), we have:

$$\mu_i - \sqrt{\frac{\sigma_i^2}{n_i}} \times Z(1-\frac{p}{2}) < \bar{x}_i < \mu_i + \sqrt{\frac{\sigma_i^2}{n_i}} \times Z(1-\frac{p}{2}),\ i = 1, 2,$$

$$\frac{\sigma_i^2}{n_i} \times \chi^2_{n_i-1}(\frac{p}{2}) < S_i^2 < \frac{\sigma_i^2}{n_i} \times \chi^2_{n_i-1}(1-\frac{p}{2}),\ i = 1, 2,$$

$$\lambda_i\log(1-2|\frac{1-p}{2}|) < r_i < -\lambda_i\log(1-2|\frac{1-p}{2}|),\ i = 1-4,$$

where $Z(1-p/2)$ is the $(1-p/2)$ quantile of the standard normal distribution, $\chi^2_{n_i-1}(r)$ is the r quantile of $\chi^2_{n_i-1}$ ($r = p/2$ *or* $1-p/2$), and $\lambda_i\log(1-2|\frac{1-p}{2}|)$ and $-\lambda_i\log(1-2|\frac{1-p}{2}|)$ are $p/2$ and $(1-p/2)$ quantile of Laplace distribution $L(0, \lambda_i)$.

In Equation 5, plugging in the bounds of the sample means, the sample variances, and the Laplace noises, we have:

$$Pr(K^L(p) < K < K^U(p)) = p^8,$$

where $K^L(p)$ and $K^U(p)$ are specified in the main theorem. Because $Pr(K^L(p) < K) \ge p^8$, we have

$$Pr(\text{Bayes Error} < \sqrt{P(\omega_1)P(\omega_2)}e^{-K^L(p)}) \ge p^8.$$

■

The proof is based on genuine Gaussian distributions. Bhattacharyya bound can be applied to truncated Gaussian distribution [2] and we can obtain useful information if k is sufficiently large. We are not able to develop theoretical results for multivariate Gaussian distribution. We obtain information for high dimensional Bayes error through experiments in the next section.

5. EXPERIMENTAL EVALUATION

In order to evaluate the performance of Gaussian mixture models learned from data under differential privacy, we have conducted extensive experiments. Since our goal is to understand how differential privacy affects the Bayes error of Gaussian mixture models, we try to avoid introducing other types of errors. Clearly, one of the issues with using Gaussian mixture models in practice is that such models may not represent the underlying data accurately. To sidestep this issue, and make sure that we do not have additional errors due to modeling real data distribution inaccurately, we generated data sets from known Gaussian mixture parameters. The parameters are estimated from real life data in one experiment, and synthetic in the rest. Later on, we use these data sets for our experiments. By using such generated data sets, we ensure that we do not introduce errors due to wrong model selection.

Each reported experiment is repeated five times using the following steps:

1. Given the parameters of the Gaussian mixture models, we generate training sets of increasing sample sizes.
2. Since differential privacy requires bounded attribute values, we truncate the generated training samples using $\mu \pm \ 6\sigma$ confidence intervals for every attribute.
3. Using the truncated training data set, pre-specified ϵ value, and the sensitivity values computed after adjusting for the actual range of every attribute, we add Laplace noise to the mean and variance-covariance matrix of each Gaussian component. One issue we have to deal with in our experiments is the fact that after noise addition, the variance-covariance matrix cease to be a positive definite matrix. In order to make sure that the privacy properties of the Laplace noise addition are protected, we employ the following heuristic processing on the *noise added variance-covariance matrix* M:
   (a) Copy the noise added upper triangle of the matrix M to lower triangle to make M symmetric.
   (b) Using eigenvalue decomposition, represent M as $V \times D \times V'$ where V is an orthogonal matrix of eigenvectors and D is a diagonal matrix of eigenvalues. If any of the values in D is negative, replace it with the minimum positive eigenvalue of the matrix. [4]
   (c) Repeat step (b) until we obtain a positive definite matrix. [5]
4. We generate a separate test data set of size 50,000 using the original parameters without the noises, and report the effectiveness of the Gaussian mixture models using the Laplace noise added parameters. Meanwhile, we report the effectiveness of the regular Gaussian mixture models based on the parameters learned from the truncated training data sets. Test data set of size 50,000 is chosen to make sure that the estimated Bayes errors are as accurate as possible.

5.1 Experimental Results

In our first set of experiments, the Gaussian mixture distributions have the following parameter values.

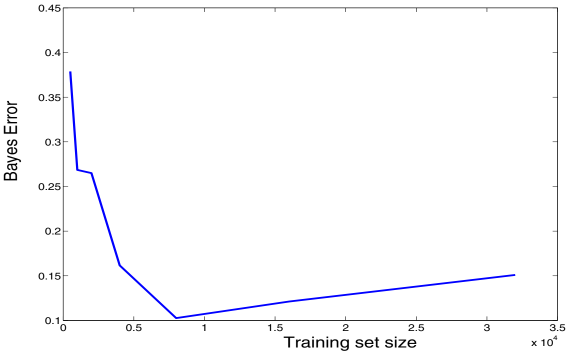

Figure 1: Bayes error versus training set size for $\epsilon = 0.1$.

$$\begin{aligned}
\mu_1 &= [1.8,\ 3.2,\ 3.8,\ 6,\ 5.5],\\
\Sigma_1 &= \text{diag}\{0.36,\ 1.21,\ 3.24,\ 5,\ 76,\ 0.64\},\\
\mu_2 &= [0.5,\ 1,\ 1.5,\ 2.5,\ 3.5],\\
\Sigma_2 &= \text{diag}\{2.56,\ 0.64,\ 4.00,\ 1.44,\ 0.16\}.
\end{aligned}$$

The mixing probability is 0.7. We tested different ϵ values ranging from 0.01 to 0.1. For the training data set size, we have conducted experiments with 500, 1000, 2000, 4000, 8000, 16000, and 32000. For this Gaussian mixture, *all the models built with ϵ less than 0.1 resulted in Bayes errors more than 0.3.* Please note that predicting everything as class one has a Bayes error 0.3. For these reasons, we do not report the experiments with ϵ values less than 0.1. In all of our experiments with different training data sets, Bayes error without noise addition was less than 0.01. In Figure 1, we show the Bayes error rates for different training set sizes for fixed $\epsilon = 0.1$. As the results indicate, even for large training set sizes the Bayes error is more than 0.1. If we compare this with 0.01, the Bayes error from estimated parameters without noise addition, this result indicates that directly adding noise to variance-covariance matrix may cause significantly larger Bayes error.

In another set of experiments, we want to understand the joint effect of correlated attributes, dimensionality, and the training set size. In each of these experiments, the first Gaussian component has the identity covariance matrix with mean $\vec{0}$. For the second Gaussian component, we fixed the mean vector to $\vec{1}$. For covariance matrix Σ, we set $\Sigma_{i,i} = \sigma$ and $\Sigma_{i,j} = \Sigma_{j,i} = 0.5 \times \sigma$ for various σ values. As a reference, we report the Bayes error rates without the noise addition for different σ values for training set size 500 in Table 1. We would like to stress that this is the worst case for the scenarios without noise addition, since Bayes error will be smaller as the training set size increases. The results for Gaussian Mixture models under differential privacy are reported in Figures 2, 3, 4, and 5. The results indicate that for training samples of sizes less than 16000, the Bayes error caused by differential privacy is prohibitively high. Again these results suggest that differential privacy needs to be applied to large data sets with large ϵ values to provide useful results.

Finally, we used the Parkinson data set from the UCI Machine learning repository. We computed the mean and

[4] Please note that a symmetric matrix is positive definite iff all the eigenvalues are positive.

[5] This problem could be represented as the following optimization problem: Given M, find M' such that M' is symmetric positive definite and $s(M,\ M')$ is minimum for some distance metric s. We leave the exploration of such optimization problem as a future work.

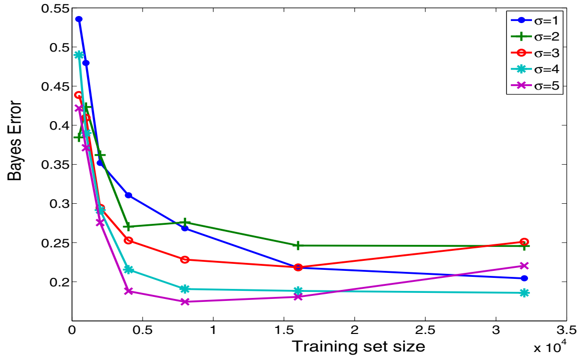

Figure 2: Bayes error versus training set size for $\epsilon = 0.1$, 5 dimensional Gaussian mixture.

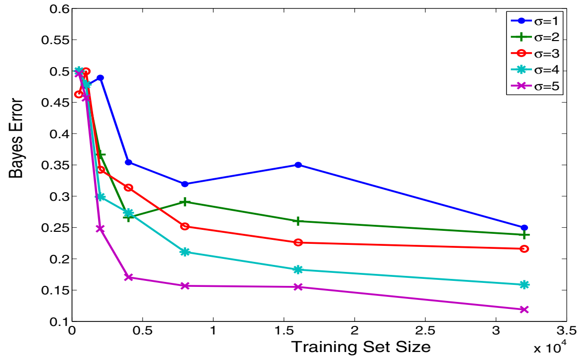

Figure 3: Bayes error versus training set size for $\epsilon = 0.1$, 10 dimensional Gaussian mixture.

covariance matrix of each class in the Parkinson data set and used these parameters in our experiments. In all of the experiments, we set $\epsilon = 0.1$. For the Parkinson data set, a classifier that put all the records into the majority class has Bayes error 0.2462. Unfortunately, all the Gaussian mixture models with different sample sizes under differential privacy have Bayes error around 0.24. On the other hand, in non-differentially private case, the Bayes error is less than 0.01. The above results suggest that direct noise addition to Gaussian mixture parameters could cause significant distortion.

6. SUMMARY

In this article we examine sensitivities of various statistics queried from a statistical database, and the performance of Bayesian classifier using the noise added mean and variance-covariance matrix. In the process we identify an interesting issue associated with random noise addition: The variance-covariance matrix without the added noise is positive definite. However simply adding noise can only return a symmetric matrix, which is no longer positive definite. Consequently the query result cannot be used to construct a Bayesian classifier. We implement a heuristic method to repair the noise added matrix to achieve positive definiteness in the experiments.

dimension v.s. σ	1	2	3	4	5
5	0.18	0.22	0.20	0.17	0.14
10	0.12	0.19	0.16	0.12	0.09
15	0.10	0.18	0.15	0.09	0.06
20	0.08	0.18	0.14	0.08	0.05

Table 1: Bayes Error for increasing dimensionality and σ values for fixed training set size 500.

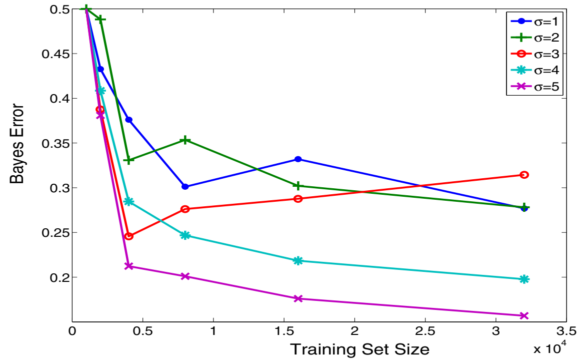

Figure 4: Bayes error versus training set size for $\epsilon = 0.1$, 15 dimensional Gaussian mixture.

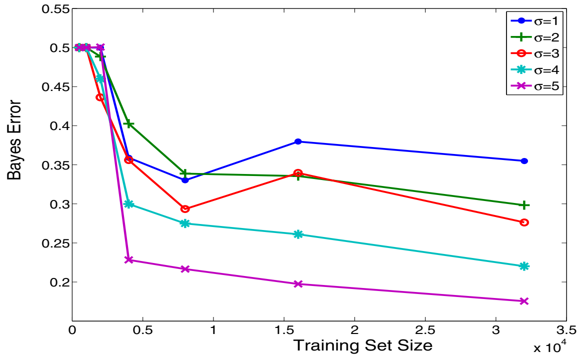

Figure 5: Bayes error versus training set size for $\epsilon = 0.1$, 20 dimensional Gaussian mixture.

This is a general issue for random noise addition. Adding noise to a statistic under certain constraint may return query results that no longer satisfy the constraint. The query results need to be further modified in order to be used in subsequent studies. An interesting question is how to provide query results that are helpful for subsequent studies while safely protecting database participant's privacy. Each constrained statistic may need an algorithm to achieve its original properties after noise addition.

7. ACKNOWLEDGMENTS

This work was partially supported by Air Force Office of Scientific Research MURI Grant FA9550-08-1-0265, Army Research Office MURI Grant W911NF-08-1-0238, National Institutes of Health Grant 1R01LM009989, National Science Foundation Grants Career-0845803, CNS- 0964350, CNS-1016343, and DMS-0904548.

8. REFERENCES

[1] R. Agrawal and R. Srikant. Privacy-preserving data mining. *SIGMOD Rec.*, 29(2):439–450, 2000.

[2] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, New York, 2. edition, 2001.

[3] C. Dwork. Differential privacy. In *ICALP (2)*, pages 1–12. Springer, 2006.

[4] C. Dwork. Differential privacy: A survey of results. In *Theory and Applications of Models of Computation*, pages 1–19. Springer Berlin / Heidelberg, 2008.

[5] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *In Proceedings of the 3rd Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

[6] A. Friedman and A. Schuster. Data mining with differential privacy. In *KDD '10: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 493–502, New York, NY, USA, 2010. ACM.

[7] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[8] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright. A practical differentially private random decision tree classifier. In *ICDM Workshops*, pages 114–121, 2009.

[9] M. Kantarcioğlu, J. Jin, and C. Clifton. When do data mining results violate privacy? In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 599–604, New York, NY, USA, 2004. ACM.

[10] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, page 99, Washington, DC, USA, 2003. IEEE Computer Society.

[11] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE '07*, pages 106–115, Istanbul, Turkey, 2007. IEEE.

[12] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *ICDE '06*, page 24, Atlanta, GA, USA, 2006. IEEE Computer Society.

[13] S. Merugu and J. Ghosh. Privacy-preserving distributed clustering using generative models. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, page 211, Washington, DC, USA, 2003. IEEE Computer Society.

[14] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[15] X. Xiao and Y. Tao. Anatomy: simple and effective privacy preservation. In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, pages 139–150, Seoul, Korea, 2006. VLDB Endowment.

[16] X. Xiao and Y. Tao. Output perturbation with query relaxation. *PVLDB*, 1(1):857–869, 2008.