Naive Bayes Classification under Local Differential Privacy

Emre Yilmaz
Department of Computer and Data Sciences
Case Western Reserve University
Cleveland, OH, USA
exy109@case.edu

Mohammad Al-Rubaie
Department of Electrical Engineering
University of South Florida
Tampa, FL, USA
mtilink@gmail.com

J. Morris Chang
Department of Electrical Engineering
University of South Florida
Tampa, FL, USA
chang5@usf.edu

***Abstract*—Supervised learning techniques such as classification algorithms learn from training data to predict the correct label for newly presented input data. In many real-world scenarios, training data required by such techniques can contain personal information and data collection can be a significant problem due to privacy concerns. Cryptographic techniques have been used before to do training on encrypted data. However, such techniques are computationally expensive and they are not scalable most of the time. If a dataset in another party will be used for training, differential privacy technology can be used to preserve the privacy of the individuals in the dataset. When there is no such dataset and data needs to be collected from individuals directly for training, local differential privacy can be used. Local differential privacy is a technology to preserve privacy during data sharing with an untrusted data collector. In this work, we propose to use local differential privacy techniques to train a Naive Bayes classifier. Using the proposed solution, an untrusted party collects perturbed data from individuals that keep the relationship between the feature values and class labels. By estimating probabilities needed by the Naive Bayes classifier using the perturbed data, the untrusted party can classify new instances with high accuracy. We develop solutions that work for both discrete and continuous data. We also propose utilizing dimensionality reduction techniques to decrease communication cost and improve accuracy. We show the accuracy of the proposed Naive Bayes classifier achieving local differential privacy via experiments on several datasets. We also show how dimensionality reduction enhances the accuracy.**

***Index Terms*—Local Differential Privacy, Naive Bayes, Classification, Dimensionality Reduction**

I. Introduction

Predictive analytics is the process of making prediction about future events by analyzing the current data using statistical techniques. It is used in many different areas such as marketing, insurance, financial services, mobility, and healthcare. For predictive analytics many techniques can be used from statistics, data mining, machine learning, and artificial intelligence. Classification methods in machine learning such as neural networks, support vector machines, regression techniques, and Naive Bayes are widely used for predictive analytics. These methods are supervised learning methods in which labeled training data is used to generate a function which can be used for classifying new instances. In these supervised learning methods, the accuracy of the classifier highly depends on the training data. Using a larger training set improves the accuracy most of the time. Hence, one needs to have a large training data in order to do classification accurately. However, collecting a large dataset brings privacy concerns. In many real life applications, the classification tasks require training sets containing sensitive information about individuals such as financial, medical or location information. For instance, insurance companies need financial information of individuals for risk classification. If there is a company that wants to build a model for risk classification, the data collection may be a critical problem because of privacy concerns. Therefore, we address the problem of doing classification while protecting the privacy of the individuals who provide the training data; thus enabling companies and organizations to achieve their utility targets, while helping individuals to protect their privacy.

Differential privacy is a commonly used standard for quantifying individual privacy. In the original definition of differential privacy [1], there is a trusted data curator which collects data from individuals and applies techniques to obtain differentially private statistics about the population. Then, the data curator publishes privacy-preserving statistics about the population. Satisfying differential privacy in the context of classification has been widely studied [2]–[4]. However, these techniques are not suitable when individuals do not trust the data curator completely. To eliminate the need of trusted data curator, techniques to satisfy differential privacy in the local setting have been proposed [5]–[8]. In local differential privacy (LDP), individuals send their data to the data aggregator after privatizing data by perturbation. Hence, these techniques provide plausible deniability for individuals. Data aggregator collects all perturbed values and makes an estimation of statistics such as the frequency of each value in the population.

In order to guarantee the privacy of the individuals who provide training data in a classification task, we propose using LDP techniques for data collection. We apply LDP techniques to Naive Bayes classifiers which are set of simple probabilistic

978-1-7281-8206-3/20/$31.00 ©2020 IEEE
DOI 10.1109/DSAA49011.2020.00081

classifiers based on Bayes' theorem. Naive Bayes classifiers use the assumption of independence between every pair of features. They are highly scalable and particularly suitable when the number of features is high or when the size of training data is small. Naive Bayes is a popular method for text classification (e.g. spam detection and sentiment classification), and it is also used in many other practical applications such as medical diagnosis, digit recognition, and weather prediction. Despite its simplicity, Naive Bayes can often perform better than or close to more sophisticated classification methods.

Given a new instance, Naive Bayes basically computes the conditional probability of each class label, and then assigns the class label with maximum probability to the given instance. Using Bayes' theorem and the assumption of independence of features, each conditional probability can be decomposed as the multiplication of several probabilities. One needs to compute each of these probabilities using training data in order to do Naive Bayes classification. Since the training data must be collected from individuals by preserving privacy, we utilize LDP frequency and statistics estimation methods for collecting perturbed data from individuals and estimating conditional probabilities in Naive Bayes classification. To be able to estimate the conditional probability that a feature would have a specific value given a class label, the relationship between class labels and each feature must be preserved during data collection. Therefore, a new instance can be classified based on the collected privatized training data with Naive Bayes classifier. We developed techniques to perform this privatized training for discrete and continuous data using Naive Bayes classifiers.

Our contributions can be summarized as follows:

First, for the discrete features, we developed LDP Naive Bayes classifier using LDP frequency estimation techniques; where each possible probability that can be used to classify an instance with Naive Bayes is estimated, by preserving the relationships between class labels and features. For perturbation, we utilized five different LDP mechanisms: Direct Encoding (DE), Symmetric and Optimal Unary Encoding (SUE and OUE), Summation with Histogram Encoding (SHE), and Thresholding with Histogram Encoding (THE).

Second, for the continuous features, we propose two approaches: (a) discretizing the data, and then applying LDP techniques (similar to the previous discussion), and (b) applying Gaussian Naive Bayes after adding Laplace noise to the data to satisfy LDP. For the second approach, we utilized and compared three types of continuous data perturbation methods. In both approaches, we also propose to utilize dimensionality reduction to improve accuracy and to decrease the communication cost and the amount of noise added.

Third, we conducted experiments with real datasets using various LDP techniques. The results demonstrate that the accuracy of the Naive Bayes classifier is maintained even when the LDP guarantees are satisfied. Our experiment results also show that dimensionality reduction improves classification accuracy without decreasing the privacy level.

TABLE I: An example dataset

Age	Income	Gender	Missed Payment
Young	Low	Male	Yes
Young	High	Female	Yes
Medium	High	Male	No
Old	Medium	Male	No
Old	High	Male	No
Old	Low	Female	Yes
Medium	Low	Female	No
Medium	Medium	Male	Yes
Young	Low	Male	No
Old	High	Female	No

The rest of the paper is organized as follows. We explain Naive Bayes classification, locally differentially private frequency and statistics estimation methods as background in Section II. In Section III, we present our methods to apply LDP techniques into Naive Bayes classification. We experimentally evaluate the accuracy of the classification under LDP in Section IV. Related work is reviewed in Section V. Finally, Section VI concludes the paper.

II. Preliminaries

A. Naive Bayes Classification

In probability theory, Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event. It is stated as follows:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Naive Bayes classification technique uses Bayes' theorem and the assumption of independence between every pair of features. Let the instance to be classified be n-dimensional vector $X = \{x_1, x_2, ..., x_n\}$, the names of the features be $F_1, F_2, ..., F_n$, and the possible classes that can be assigned to the instance be $C = \{C_1, C_2, ..., C_k\}$. Naive Bayes classifier assigns the instance X to the class C_s if and only if $P(C_s \mid X) > P(C_j \mid X)$ for $1 \leq j \leq k$ and $j \neq s$. Hence, the classifier needs to compute $P(C_j \mid X)$ for all classes and compare these probabilities. Using Bayes' theorem, the probability $P(C_j \mid X)$ can be calculated as

$$P(C_j \mid X) = \frac{P(X \mid C_j) \cdot P(C_j)}{P(X)}$$

Since $P(X)$ is same for all classes, it is sufficient to find the class with maximum $P(X \mid C_j) \cdot P(C_j)$. With the assumption of independence of features, it is equal to $P(C_j) \cdot \prod_{i=1}^{n} P(F_i = x_i \mid C_j)$. Hence, the probability of assigning C_j to given instance is proportional to $P(C_j) \cdot \prod_{i=1}^{n} P(F_i = x_i \mid C_j)$.

1) Discrete Naive Bayes: To demonstrate the concept of the Naive Bayes classifier for discrete (categorical) data, we use the dataset given in Table I. In this example, the classification task is predicting whether a customer will miss a mortgage

TABLE II: Conditional probabilities for F_1 (i.e. Age) of the example dataset.

$P(Age = Young \mid C_1) = 2/4$
$P(Age = Young \mid C_2) = 1/6$
$P(Age = Medium \mid C_1) = 1/4$
$P(Age = Medium \mid C_2) = 2/6$
$P(Age = Old \mid C_1) = 1/4$
$P(Age = Old \mid C_2) = 3/6$

payment or not. Hence, there are two classes such as C_1 and C_2 representing missing a previous payment or not, respectively. $P(C_1) = \frac{4}{10}$ and $P(C_2) = \frac{6}{10}$. In addition, conditional probabilities for the feature "Age" is given in Table II. Similarly, conditional probabilities for the other features can be calculated.

In order to predict whether a young female with medium income will miss a payment or not, we can set $X = (Age = Young,\ Income = Medium,\ Gender = Female)$. To use Naive Bayes classifier, we need to compare $P(C_1) \cdot \prod_{i=1}^{3} P(F_i = x_i \mid C_1)$ and $P(C_2) \cdot \prod_{i=1}^{3} P(F_i = x_i \mid C_2)$. Since the first one is equal to 0.025 and the second one is equal to 0.055, it can be concluded that C_2 is assigned for the instance X by Naive Bayes classifier. In other words, it can be predicted that a young female with medium income will not miss her payments.

2) Gaussian Naive Bayes: For continuous data, a common approach is assuming the values are distributed according to Gaussian distribution. Then, the conditional probabilities can be computed using the mean and the variance of the values. Let a feature F_i has a continuous domain. For each class $C_j \in C$ the mean $\mu_{i,j}$ and the variance $\sigma^2_{i,j}$ of the values of F_i in the training set are computed. For the given instance X, the conditional probability $P(F_i = x_i \mid C_j)$ is computed using Gaussian distribution as follows:

$$P(F_i = x_i \mid C_j) = \frac{1}{\sqrt{2\pi\sigma^2_{i,j}}} e^{-\frac{(x_i - \mu_{i,j})^2}{2\sigma^2_{i,j}}}$$

Gaussian Naive Bayes can also be used for features with large discrete domain. Otherwise, the accuracy may reduce because of the high number of values which are not seen in the training set.

B. Local Differential Privacy

Local differential privacy (LDP) is a way of measuring the individual privacy in the case where the data curator is not trusted. In LDP setting, individuals perturb their data before sending it to a data aggregator. Hence, the data aggregator only sees perturbed data. It aggregates all reported values and *estimates* privacy-preserving statistics. LDP states that for any reported value, the probability of distinguishing two input values by the data aggregator is at most $e^{-\epsilon}$. The formal definition of local differential privacy is as follows:

Definition 1: A protocol P satisfies ϵ-local differential privacy if for any two input values v_1 and v_2 and any output o in the output space of P,

$$\Pr\left[P(v_1) = o\right] \leq \Pr\left[P(v_2) = o\right] \cdot e^{\epsilon}$$

Randomized response mechanism is one method to satisfy LDP. In the binary randomized response mechanism, the input is a single bit. An individual sends the correct bit to the data aggregator with probability p and incorrect bit with probability $1 - p$. The aggregator can estimate the actual number of 0s and 1s by using the probability p and the reported numbers of 0s and 1s. To satisfy ϵ-LDP, p can be selected as $\frac{e^{\epsilon}}{1+e^{\epsilon}}$. This problem can be generalized into frequency estimation problem where the inputs can be selected from a larger set containing more than two values.

1) LDP Frequency Estimation: In the problem of frequency estimation, there are m individuals having a value from the set $\mathcal{D} = \{1, 2, ..., d\}$. The aim of data aggregator is to find the number of individuals having a value $i \in \mathcal{D}$ for all values in the set. Wang et al. [9] proposed a framework to generalize the LDP frequency estimation protocols in the literature, and they also proposed two new protocols. Here, we summarize the LDP protocols which are explained in [9] in detail. All of them can be used for frequency estimation in our solution. We empirically compare their effect on accuracy in our problem setting in Section IV.

Direct encoding (DE): In this method, there is no encoding of input values. For perturbation, an individual reports her value v correctly with probability $p = \frac{e^{\epsilon}}{e^{\epsilon}+d-1}$, or reports one of the remaining $d - 1$ values with probability $q = \frac{1}{e^{\epsilon}+d-1}$ per each. When the aggregator collects all perturbed values from m individuals, it estimates the frequency of each $i \in \{1, 2, ..., d\}$ as follows: Let c_i be the number of times i is reported. Estimated number of occurrence of value i in the population is computed as $E_i = \frac{c_i - m \cdot q}{p - q}$.

Histogram encoding: An individual encodes her value v as length-d vector $[0.0, ..., 1.0, ..., 0.0]$ where only v^{th} component is 1.0 and the remaining are 0.0. Then, she perturbs her value by adding Lap($\frac{2}{\epsilon}$) to each component in the encoded value, where Lap($\frac{2}{\epsilon}$) is a sample from Laplace distribution with mean 0 and scale parameter $\frac{2}{\epsilon}$. When the data aggregator collects all perturbed values, it can use two estimation methods. In summation with histogram encoding (SHE), it calculates the sum of all values reported by individuals. To estimate the number of occurrence of value i in the population, the data aggregator sums the i^{th} components of all reported values. In thresholding with histogram encoding (THE), the data aggregator sets all values greater than a threshold θ to 1, and the remaining to 0. Then it estimates the number of i's in the population as $E_i = \frac{c_i - m \cdot q}{p - q}$, where $p = 1 - \frac{1}{2}e^{\frac{\epsilon}{2}(1-\theta)}$, $q = \frac{1}{2}e^{-\frac{\epsilon}{2}\theta}$, and c_i is the number of 1's in the i^{th} components of all reported values after applying thresholding.

Unary encoding: In this method, an individual encodes her value v as length-d binary vector $[0, ..., 1, ..., 0]$ where only

v^{th} bit is 1 and the remaining are 0. Then, for each bit in the encoded vector, she reports correctly with probability p and incorrectly with probability $1-p$ if the input bit is 1. Otherwise, she reports correctly with probability $1-q$ and incorrectly with probability q. In symmetric unary encoding (SUE), p is selected as $\frac{e^{\epsilon/2}}{e^{\epsilon/2}+1}$ and q is selected as $1-p$. In optimal unary encoding (OUE), p is selected as $\frac{1}{2}$ and q is selected as $\frac{1}{e^{\epsilon}+1}$. The data aggregator estimates the number of 1's in the population as $E_i = \frac{c_i - m \cdot q}{p-q}$, where c_i denotes the number of 1's in the i^{th} bit of all reported values.

2) LDP Mean Estimation: As explained in Section II-A2, Gaussian Naive Bayes is suitable for large discrete domains and continuous domains. Conditional probabilities are computed using the mean and the variance. In order to compute the mean under LDP, Laplace mechanism can be used [10]. Let the domain be normalized, and an individual has a value $v \in [-1, 1]$. The individual adds Laplace noise Lap($\frac{2}{\epsilon}$) to her value and reports noisy value ($v' = v + \text{Lap}(\frac{2}{\epsilon})$) to the data aggregator. Since the mean of noises that are drawn from Laplace distribution is 0, the data aggregator calculates the sum of all noisy values reported by individuals, and divides the sum by the number of individuals to estimate the mean. As for estimating the variance, we explain our proposed method in Section III-B.

3) LDP with Multi-dimensional Data: The frequency and mean estimation methods described in Section II-B1 and II-B2 work for one-dimensional data. If the data owned by individuals is multi-dimensional, reporting each value with these methods may cause privacy leaks due to the dependence of features. Hence, the following approaches were proposed to deal with n-dimensional data.

Approach 1: For the Laplace mechanism described in Section II-B2, LDP can also be satisfied if the noise scaled with the number of dimensions n [10]. Hence, if an individuals' input is $V = (v_1, ..., v_n)$ such that $v_i \in [-1, 1]$ for all $i \in \{1, ..., n\}$, then she can report each v_i after adding Lap($\frac{2n}{\epsilon}$) (i.e. $v'_i = v_i + \text{Lap}(\frac{2n}{\epsilon})$). This approach is not suitable if the number of dimensions n is high because large amount of noise reduces the accuracy.

Approach 2: For mean estimation, Nguyên et al. [10] introduced an algorithm that requires reporting one bit by each individual to the data aggregator. An individual has an input value $V = (v_1, ..., v_n)$ such that $v_i \in [-1, 1]$ for all $i \in \{1, ..., n\}$. She can perturb and report her input as follows:

- She select $j \in \{1, ..., n\}$ uniformly at random.
- She samples Bernoulli variable u such that $\Pr[u = 1] = \frac{v_j(e^{\epsilon}-1)+e^{\epsilon}+1}{2e^{\epsilon}+2}$.
- She sets $v'_j = \frac{e^{\epsilon}+1}{e^{\epsilon}-1} \cdot n$ if $u = 1$, $v'_j = -\frac{e^{\epsilon}+1}{e^{\epsilon}-1} \cdot n$ otherwise.
- She reports $V' = \big(0, ..., 0, v'_j, 0, ..., 0\big)$ to the data aggregator.

Since the only non-zero value is v'_j and it has two possible values, it is sufficient to report one bit to indicate the sign of v'_j. Each feature is approximately reported by $\frac{m}{n}$ individuals. This approach is efficient in terms of communication cost.

Approach 3: The first two approaches are specific to continuous data. Hence, we outline a third approach that is more general. The data aggregator requests only one perturbed input from each individual to satisfy ϵ-LDP. Each individual can select the input to be reported uniformly at random or the data aggregator can divide the individuals into n groups and requests different input values from each group. As a result, each feature is approximately reported by $\frac{m}{n}$ individuals. This approach is suitable when the number of individuals m is high relative to the number of features n. Otherwise the accuracy decreases since the number of reported values is low for each feature.

C. Dimensionality Reduction

The approaches for dealing with multi-dimensional data suffer from the high number of dimensions which necessitates adding more noise that results in decreasing the accuracy. In the first approach, the amount of noise is directly proportional to the number of dimensions. In the second approach, the number of individuals who report each feature decreases for high number of dimensions because each feature is approximately reported by $\frac{m}{n}$ individuals. Therefore, we propose to utilize dimensionality reduction techniques to improve accuracy. Dimensionality reduction is a machine learning tool that is traditionally used to solve over-fitting issues, and to reduce the computational cost caused by high numbers of features. We utilize two commonly used methods for dimensionality reduction: Principal Component Analysis (PCA) and Discriminant Component Analysis (DCA) [11].

PCA reduces the dimensions while preserving most of the information by projecting the data on the principal components with the highest variance. By projecting the data in the direction of the highest variability, PCA also tends to decrease the reconstruction error; thus improving recoverability of the original data from its projection. On the other hand, **DCA** utilizes the class labels C_i's to project the data in the direction that can effectively discriminate between different classes. Such direction might not be necessarily the direction of the highest variance; thus DCA can be superior to PCA for labeled data.

III. Naive Bayes Classification under Local Differential Privacy

As explained in Section II-A, one needs to know the probability $P(C_j)$ for all classes, and $P(F_i = x_i \mid C_j)$ for all classes and all possible x_i values in order to use Naive Bayes classifier. These probabilities are calculated based on the training data. However, when individuals avoid sharing their data for training due to privacy reasons, it is impossible to calculate these probabilities. Since LDP provides plausible deniability for individuals, LDP methods can be used to train Naive Bayes classifier. In this section, we explain the

TABLE III: Notations used in the paper.

$X = (x_1, ..., x_n)$	instance to be classified
$C = \{C_1, C_2, ..., C_k\}$	the set of class labels
n	the number of features
k	the number of class labels
m	the number of individuals

estimation of such necessary probabilities using LDP methods. First we introduce a solution for classification for all discrete features (Section III-A), and then we explain the solutions to deal with continuous data (Section III-B). Table III shows the notations used in the paper.

A. LDP Naive Bayes with Discrete Features

We initially consider the case where all the features are numerical and discrete. There are m individuals who are reluctant to share their data to train a classifier. However, they can share perturbed data to preserve their privacy. By satisfying LDP during data collection, the privacy of individuals can be guaranteed. Here, we propose a solution that utilizes the LDP frequency estimation methods given in Section II-B in order to compute all necessary probabilities for a Naive Bayes classifier.

The data aggregator needs to estimate class probabilities $P(C_j)$ for all classes in $C = \{C_1, C_2, ..., C_k\}$ and conditional probabilities $P(F_i = x_i \mid C_j)$ for all classes and all possible x_i values. Let an individual's (e.g. Alice's) data be $(a_1, a_2, ..., a_n)$ and her class label be C_v. She needs to prepare her input and perturb it by satisfying LDP. We now explain the preparation and the perturbation of input values based on Alice's data and the estimation of the class probabilities and the conditional probabilities by data aggregator.

1) Computation of Class Probabilities: For the computation of class probabilities, Alice's input becomes $v \in \{1, 2, ..., k\}$ since her class label is C_v. Alice encodes and perturbs her value v, and reports to the data aggregator. Any LDP frequency estimation method which is explained in Section II-B1 can be used. Similarly, other individuals report their perturbed class labels to the data aggregator. The data aggregator collects all perturbed data and estimates the frequency of each value $j \in \{1, 2, ..., k\}$ as E_j. As a result, the probability $P(C_j)$ is estimated as $\frac{E_j}{\sum_{i=1}^{k} E_i}$. For the example dataset in Table I, Alice's input v becomes 1 if she has a missing payment or 2 if she does not have a missing payment.

2) Computation of Conditional Probabilities: To estimate the conditional probabilities $P(F_i = x_i \mid C_j)$, it is not sufficient to report feature values directly. To be able to compute these probabilities, the relationship between class labels and features must be preserved. To keep this relationship, individuals prepare their inputs using feature values and class labels. Let the total number of possible values for F_i be n_i. If Alice's value in i^{th} dimension is $a_i \in \{1, 2, ..., n_i\}$ and her class label value is $v \in \{1, 2, ..., k\}$, then Alice's input for feature F_i becomes $v_i = (a_i - 1) \cdot k + v$. Therefore, each individual calculates her input for the i^{th} feature in the range of $[1, k \cdot n_i]$. For instance, let "Age" values in the Table I be enumerated as (Young = 1), (Medium = 2), (Old = 3). For this feature, an individual's input can be a value between 1 and 6, where 1 represents the age is young and there is a missing payment, and 6 represents the age is old and there is no missing payment. Therefore, there is one input value that corresponds to each line of Table II. Similarly, the number of possible inputs for "Income" is 6 and the number of possible inputs for "Gender" is 4. After determining her input in i^{th} feature, Alice encodes and perturbs her value v_i, and reports the perturbed value to the data aggregator. To estimate the conditional probabilities for F_i, the data aggregator estimates the frequency of individuals having value $y \in \{1, 2, ..., n_i\}$ and class label $z \in \{1, 2, ..., k\}$ as $E_{y,z}$ by estimating the frequency of input $(y-1) \cdot k + z$. Hence, the conditional probability $P(F_i = x_i \mid C_j)$ is estimated as $\frac{E_{x_i,j}}{\sum_{h=1}^{n_i} E_{h,j}}$. For the example given above, to estimate the probability $P(Age = Medium \mid C_2)$, the data aggregator estimates the frequency of 2, 4, and 6 as $E_{1,2}$, $E_{2,2}$, and $E_{3,2}$, respectively. Then $P(Age = Medium \mid C_2)$ is estimated as $\frac{E_{2,2}}{E_{1,2}+E_{2,2}+E_{3,2}}$.

As a result, in order to contribute to the computation of class probabilities and conditional probabilities, each individual can prepare $n+1$ inputs (i.e. $\{v,\ v_1,\ v_2,\ ....,\ v_n\}$ for Alice) that can be reported after perturbation. As mentioned in Section II-B3, reporting multiple values which are dependent to each other decreases the privacy level. Reporting all $n+1$ perturbed values increases the probability of predicting the class labels of individuals by the data aggregator. This case is similar to requesting multiple queries in the centralized setting of differential privacy. Hence, each individual reports one input as described in Approach 3 in Section II-B3.

Finally, when the data aggregator estimates a value such as E_j or $E_{y,z}$, the estimation may give a negative result. In that case, we set all the negative estimations to 1 to obtain valid probability.

B. LDP Naive Bayes with Continuous Features

In order to satisfy LDP in Naive Bayes classification for continuous data, we propose two different solutions. First solution is discretizing the continuous data and applying the discrete Naive Bayes solution outlined in Section III-A. In this solution, continuous numerical data is divided into buckets to make it finite and discrete. Each individual perturbs her input after discretization. Second, the data aggregator can use Gaussian Naive Bayes to estimate the probabilities as given in Section II-A2. To estimate the mean and the variance, the data aggregator uses LDP methods given in Section II-B2. Figure 1 shows the steps of the proposed solutions. As explained in Section II-B3, the number of dimensions can be reduced to improve accuracy; hence, we utilize dimensionality reduction techniques. Now, we describe the solutions in detail.

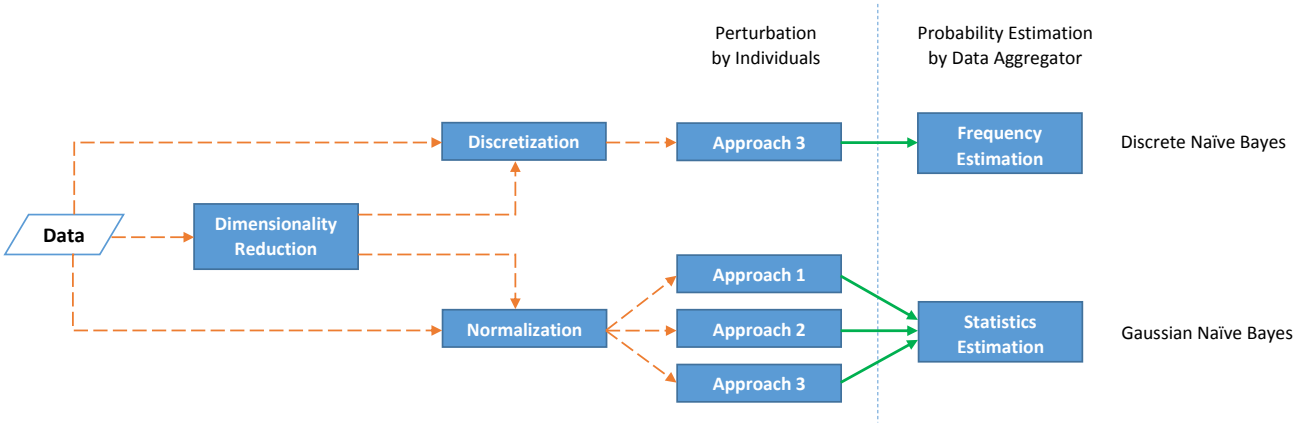


Fig. 1: Steps of LDP Naive Bayes for multi-dimensional continuous data.

Discrete Naive Bayes. We first propose to use the solution introduced for discrete data in Section III-A. Based on known feature ranges for features with continuous or large domain, the data aggregator determines the intervals for buckets in order to discretize the domain. Equal-Width Discretization (EWD) can be used for equally partitioning the domain. EWD computes the width of each bin as $\frac{max-min}{n_b}$ where max and min are the maximum and minimum feature values, and n_b is the number of desired bins. We utilized EWD in our experiments for discretization.

When the data aggregator shares the intervals with individuals, each individual firstly discretizes her continuous feature values, and then applies the procedure described in Section III-A for perturbation. The data aggregator also estimates the probabilities with the same procedure for LDP Naive Bayes for discrete data. As mentioned in Section III-A2, each individual should report just one perturbed value to guarantee ϵ-LDP.

Gaussian Naive Bayes. As explained in Section II-A2, a common approach for Naive Bayes classification for continuous data is assuming the data is normally distributed. For locally differentially private Gaussian Naive Bayes, computing the class probabilities is same with the computation for discrete features as given in Section III-A1. To compute conditional probabilities, the data aggregator needs to have the mean and the variance of training values for each feature given a class label. That is, to compute $P(F_i = x_i \mid C_j)$, the data aggregator needs to estimate the mean $\mu_{i,j}$ and the variance $\sigma^2_{i,j}$ using the F_i values of individuals with a class label C_j. Hence, the association between features and class labels has to be maintained (similar to the discrete Naive Bayes classifier).

The mean estimation was explained in Section II-B2. However to compute the mean $\mu_{i,j}$ and the variance $\sigma^2_{i,j}$ together, we propose the following method: the data aggregator divides the individuals into two groups. One group contributes to the estimation of the mean (i.e. $\mu_{i,j}$) by perturbing their inputs and sharing with the data aggregator, while the other group contributes to the estimation of the mean of squares (i.e. $\mu^s_{i,j}$) by perturbing the squares of their inputs and sharing with data aggregator.

Let Bob has class label C_j and his feature F_i value be b_i. Note that, the domain of each feature was assumed to be normalized to have a value in $[-1, 1]$. If Bob is in the first group, he adds Laplace noise to his value b_i and obtains perturbed feature value b'_i. When data aggregator collects all perturbed feature values from individuals in the first group having class label C_j, it computes the mean of the perturbed feature values which gives an estimation of the mean $\mu_{i,j}$ because the mean of noise added by individuals is 0. Similar operations could be followed by the second group. If Bob is in the second group, he adds noise to his squared value b_i^2 to obtain $b_i^{2'}$ and shares it with the data aggregator. Similarly, the data aggregator computes the estimation of the mean of squares ($\mu^s_{i,j}$). Finally, the variance $\sigma^2_{i,j}$ can be computed as $\mu^s_{i,j} - (\mu_{i,j})^2$. Once again, each individual reports only one of her value or square of her value after perturbation because they are dependent values.

In this explained method to compute the mean and the variance, the class label of individuals are not hidden from the data aggregator. To hide the class labels, we adopt the following approach: an individual (Bob) reporting a feature value $F_i = b_i$ associated with class C_j where $j \in \{1, 2, \cdots, k\}$, first constructs a vector of length k where k is the number of class labels. The vector is initialized to zeros except for the j^{th} element corresponding to the j^{th} class label which is set to the feature value b_i. After that, each element of the vector is perturbed as usual (i.e. by adding Laplace noise), and contributed to the data aggregator. Since noise is added even to the zero elements of the vector, the data aggregator will not be able to deduce the actual class label, or the actual values.

As for estimating the actual mean value (and mean of the squared values) for each class, the data aggregator only needs to compute the mean of the perturbed values as usual, and then dividing that value by the probability of that class. To understand why, assume that a specific class j has Probability

TABLE IV: Datasets used in the experiments.

Name	# Instances	# Features	# Class Labels
Car Evaluation	$1,728$	6	4
Chess	$3,196$	36	2
Mushroom	$8,124$	22	2
Connect-4	$67,557$	42	3

$P(C_j)$ (explained in Section III-A1). Hence, for a specific feature F_i, only $P(C_j)$ of the individuals have their actual values in j^{th} element of the input vector, while the remaining proportion $(1 - P(C_j))$ have zeros. Hence, after the noise clustered around the actual mean cancels each other, and the noise clustered around zero cancel each other, we would have $P(C_j) * \mu_{i,j} = observed\ (shifted)\ mean$. Hence, we can divide the observed mean by $P(C_j)$ to obtain the estimated mean. The same situation applies for the mean of the squared values, and hence for computing the variance.

IV. Experimental Evaluation

To evaluate the accuracy of Naive Bayes classification under local differential privacy, we have implemented the proposed methods in Python utilizing pandas and NumPy libraries. We have implemented 5 different LDP protocols for frequency estimation such as Direct Encoding (DE), Summation with Histogram Encoding (SHE), Thresholding with Histogram Encoding (THE), Symmetric Unary Encoding (SUE), and Optimal Unary Encoding (OUE) which are presented in Section II-B. We performed experiments with different θ values in THE and we achieved best accuracy when $\theta = 0.25$. Hence, we give the experiment results of SHE for $\theta = 0.25$. We repeated all experiments 100 times and present the average classification accuracy. We used datasets from UCI Machine Learning repository [12] and selected 80% of the datasets for training and the remaining 20% for testing. We firstly present the results for the datasets with categorical and discrete features in Section IV-A. The results for continuous data is given in Section IV-B.

A. LDP Naive Bayes with Discrete Features

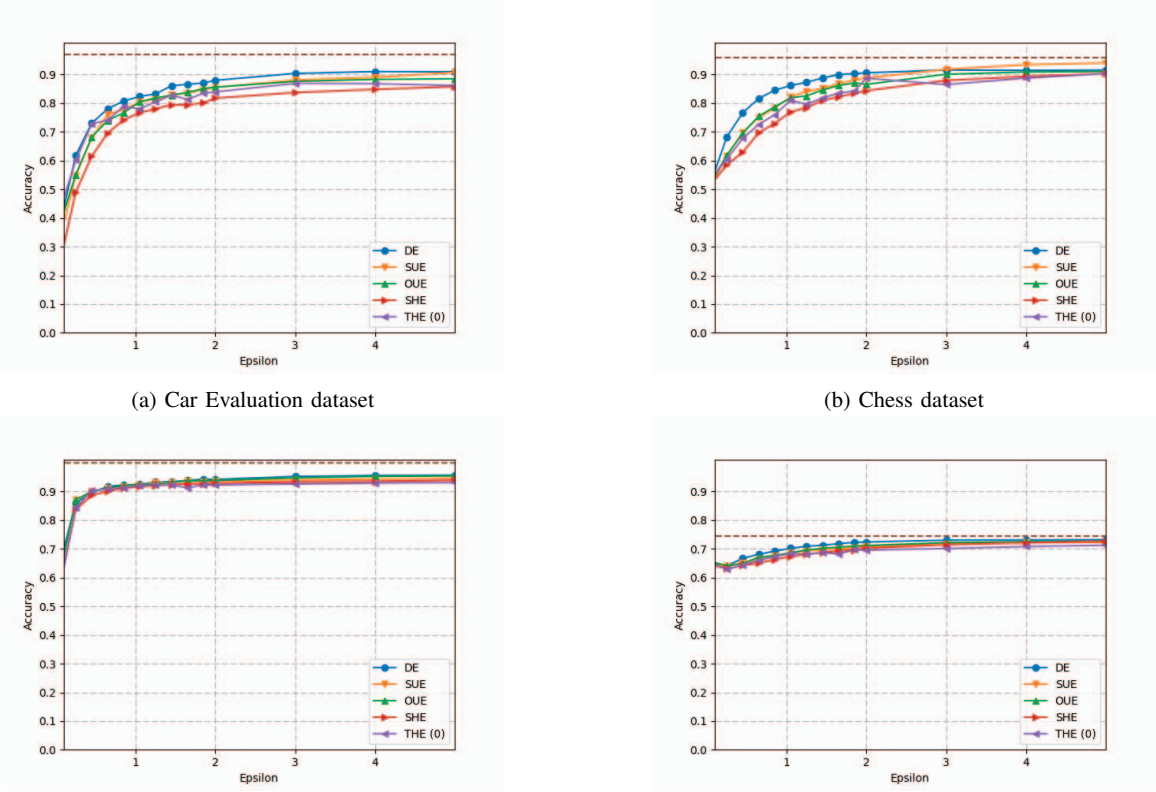
To evaluate the classification accuracy of the proposed method in Section III-A for classifying data with discrete features, we used Car Evaluation, Chess, Mushroom, and Connect-4 datasets from UCI ML repository. The number of instances, features, and class labels are given in Table IV. Initially, we performed Naive Bayes classification without local differential privacy to compare the accuracy under local differential privacy.

Experiment results for varying ϵ values up to 5 are shown in Figure 2. Dotted lines in the figures show the accuracy without privacy. As expected, when the number of instances in the training set increases, the accuracy is better for smaller ϵ values. For instance, in Connect-4 dataset, all protocols except SHE provide more than 65% accuracy even for very small ϵ values. Since the accuracy without privacy is approximately 75%, the accuracy of all of these protocols for ϵ values smaller than 1 is noticeable. The results are also similar for Mushroom dataset. For $\epsilon = 0.5$, all protocols except SHE provide nearly 90% classification accuracy. In all of the datasets, the protocol with worst accuracy is SHE. Since this protocol simply sums the all noisy values, its variance is higher than the other protocols. DE achieves the best accuracy for small ϵ values in Car Evaluation and Chess datasets because the input domains are small. The variance of DE is proportional to the size of the input domain. Therefore, its accuracy is better when the input domain is small. SUE and OUE provides similar accuracy in all of the experiments. They perform better than DE when the size of input domain is large. Although OUE is proposed by [9] to decrease variance, we did not observe considerable utility difference between SUE and OUE in our experiments.

B. LDP Naive Bayes with Continuous Features

In this section, we outline the results for the methods proposed in Section III-B for continuous data. We conducted the experiments on two different datasets: Australian and Diabetes. The Australian dataset has 14 original features, and the Diabetes dataset has 8 features. Initially, we applied the discretization method and implemented two dimensionality reduction techniques (i.e. PCA and DCA) to observe the effect of them in accuracy. The results for two datasets for different values of ϵ are given in Figure 3. We present the results for two LDP schemes (i.e. Direct Encoding and Optimized Unary Encoding) which provide the best accuracy for different domain sizes. The input domain is divided into $d = 2$ buckets for Australian dataset and $d = 4$ buckets for Diabetes dataset. For Australian dataset, we obtained the best results for PCA and DCA when the number of features is reduced to one. For Diabetes dataset, best accuracy is achieved when PCA reduces the number of features to 6 and when DCA reduces the number of features to one. As evident in Figure 3, DCA provides the best classification accuracy, which shows the advantage of using dimensionality reduction before discretization. As expected, DCA's accuracy is better than PCA since it is mainly designed for classification.

We also applied locally differentially private Gaussian Naive Bayes (LDP-GNB) on the same two datasets. We implemented all three perturbation approaches for multi-dimensional data explained in Section II-B3. Figure 4 shows the results of performing LDP-GNB on these two datasets. Among three approaches, the first one results in lowest utility since individuals report all features by adding more noise (i.e. propotional to the number of dimensions). In each figure, three curves are shown which correspond to using the original data (with 14 or 8 features for Australian and Diabetes datasets, respectively), or projecting the data using PCA or DCA before applying the LDP noise. The positive effect of reducing the dimensions can be clearly seen in all figures. In both datasets, and for PCA and DCA, the number of reduced dimensions were one. DCA



(a) Car Evaluation dataset

(b) Chess dataset

(c) Mushroom dataset

(d) Connect-4 dataset

Fig. 2: Classification accuracy for datasets with discrete features

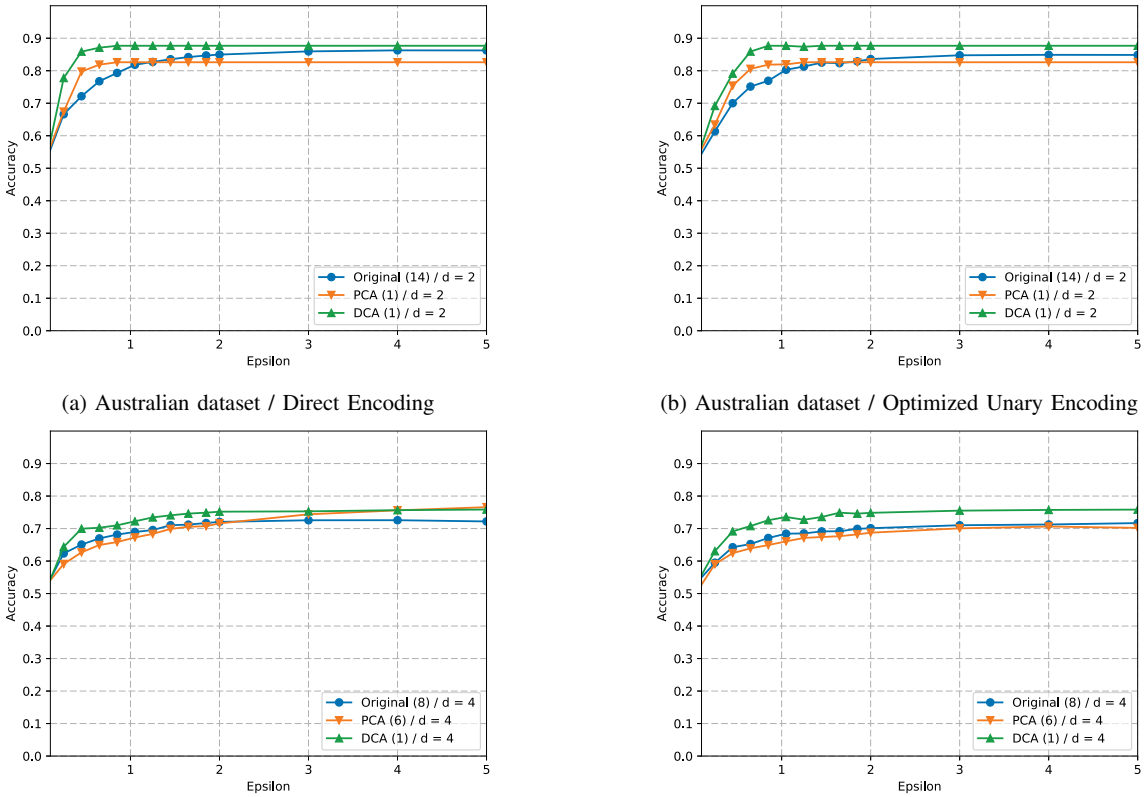
or PCA always performs better than the original data, and for all perturbation approaches.

Finally, when we compare discretization and Gaussian Naive Bayes for continuous data, it can be concluded that discretization provides better accuracy than Gaussian Naive Bayes. Especially for smaller ϵ values, the superiority of discretization is more apparent. Although it is not possible to compare the amount of noise for randomized response and Laplace mechanism, discretization possibly causes less noise due to smaller input domain.

V. Related Work

Privacy-preserving Naive Bayes classification has been studied before in different settings. Kantarcioglu et al. [13] proposed privacy-preserving Naive Bayes classifier for horizontally partitioned data. Their solution is secure in semi-honest threat model and utilizes computationally expensive cryptographic techniques such as oblivious transfer. Vaidya et al. [14] addressed the same problem for vertically partitioned data. They also used secure multi-party computation primitives which are computationally expensive operations. Naive Bayes classification under differential privacy has been studied in [15]. In [15], centralized setting for differential privacy is considered where the data owner has a training data and aims to release classifier by protecting privacy. They explain how to compute the sensitivity and add Laplace noise to satisfy differential privacy in Naive Bayes classifier. Li et al. [16] extended it to multiple data owners. Even though their problem setting is similar to our case, they guarantee the differential privacy at global level by calculating the global sensitivity and applying Laplace noise to the counts. Their solution does not satisfy the differential privacy in the local setting and preserves individual privacy with encryption techniques. Although privacy-preserving Naive Bayes classifier has been studied under different privacy settings such as horizontally or vertically partitioned data, and centralized differential privacy, none of them addresses the problem under LDP.

Most of the work in the literature about differential privacy consider the centralized setting. One of the earliest work on differential privacy in the local setting is Google's RAPPOR [6]. They proposed using randomized response mechanism to satisfy ϵ-LDP and using bloom filters to decrease communication cost. Bassily et al. [5] also proposed a method to satisfy LDP in frequency estimation utilizing random matrix projection. Wang et al. [9] introduced a framework of pure LDP protocols to generalize the frequency estimation protocols in the literature and they proposed two new protocols for frequency estimation. We utilize these protocols in our work



(a) Australian dataset / Direct Encoding

(b) Australian dataset / Optimized Unary Encoding

(c) Diabetes dataset / Direct Encoding

(d) Diabetes dataset / Optimized Unary Encoding

Fig. 3: Classification accuracy for datasets with continuous features using discretization

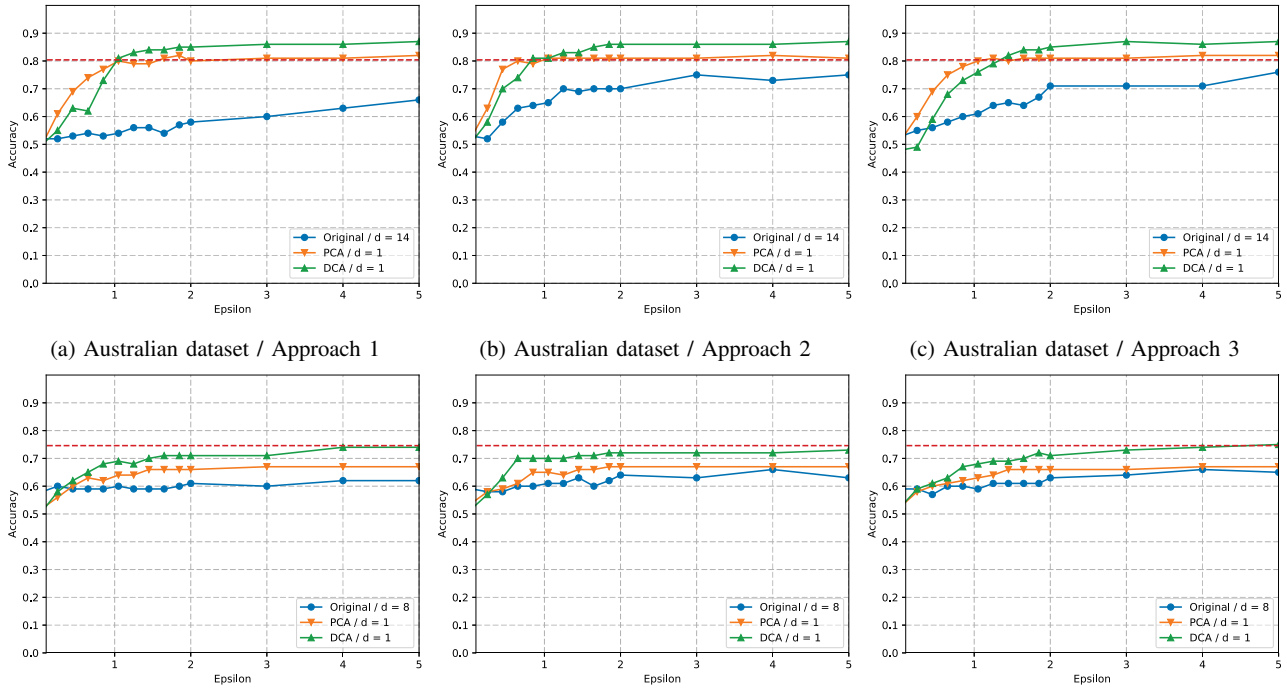
as mentioned in Section II-B. Other than frequency estimation, some other problems such as heavy hitters [17] and marginal release [18] have also been studied under LDP. The most similar work to our work is [19], which presents a system to do machine learning by satisfying LDP. To achieve better accuracy, they reduced the size of input domain to two and they also considered a binary classification model that has only two class labels. Using LDP frequency estimation the statistics about the features are estimated and using these statistics synthetic data is generated to train classification model. In our work, we do not especially address binary classification problem, and hence the number of class labels can be more than two. In addition, input domain for the features can have more than two values. By keeping the relationship between class labels and features, we allow estimation of probabilities for Naive Bayes classifier without a need for generating synthetic data.

VI. Conclusion

We proposed methods for applying locally differentially private frequency and statistics estimation protocols to collect training data in Naive Bayes classification. Using the proposed methods, one can estimate all necessary probabilities to be used in Naive Bayes classification for both discrete and continuous data. To be able to estimate the conditional probabilities, the proposed methods preserve the relationship between features and class labels during the selection of inputs. Our experiment results indicate that the classification accuracy of LDP Naive Bayes for $\epsilon > 2$ is very close to the accuracy without privacy. Even for smaller ϵ values, the accuracy is remarkable when Direct Encoding or Unary Encoding schemes are used for discrete data and when discretization is used for continuous data. In addition, experiment results show that using dimensionality reduction techniques such as DCA improves the accuracy of the proposed methods for continuous data. The proposed methods facilitate collecting large training data to use in Naive Bayes classifier without compromising the privacy of the individuals providing training data. Other than Naive Bayes, LDP techniques can be utilized in different machine learning methods which can be considered as potential future work.

Acknowledgment

This material is based on research sponsored by the DARPA Brandeis Program under agreement number N66001-15-C-4068. The views, opinions, and/or findings contained in this article are those of the author and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.



(a) Australian dataset / Approach 1 (b) Australian dataset / Approach 2 (c) Australian dataset / Approach 3

(d) Diabetes dataset / Approach 1 (e) Diabetes dataset / Approach 2 (f) Diabetes dataset / Approach 3

Fig. 4: Classification accuracy for datasets with continuous features using Gaussian Naive Bayes

REFERENCES

[1] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*. Springer, 2008, pp. 1–19.

[2] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *Journal of Machine Learning Research*, vol. 12, no. Mar, pp. 1069–1109, 2011.

[3] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright, "A practical differentially private random decision tree classifier," in *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*. IEEE, 2009, pp. 114–121.

[4] B. I. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft, "Learning in a large function space: Privacy-preserving mechanisms for svm learning," *Journal of Privacy and Confidentiality*, vol. 4, no. 1, pp. 65–100, 2012.

[5] R. Bassily and A. Smith, "Local, private, efficient protocols for succinct histograms," in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*. ACM, 2015, pp. 127–135.

[6] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. ACM, 2014, pp. 1054–1067.

[7] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," in *Advances in neural information processing systems*, 2014, pp. 2879–2887.

[8] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren, "Heavy hitter estimation over set-valued data with local differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 192–203.

[9] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *Proc. of the 26th USENIX Security Symposium*, 2017, pp. 729–745.

[10] T. T. Nguyên, X. Xiao, Y. Yang, S. C. Hui, H. Shin, and J. Shin, "Collecting and analyzing data from smart device users with local differential privacy," *arXiv preprint arXiv:1606.05053*, 2016.

[11] S. Y. Kung, *Kernel methods and machine learning*. Cambridge University Press, 2014.

[12] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[13] M. Kantarcıoglu, J. Vaidya, and C. Clifton, "Privacy preserving naive bayes classifier for horizontally partitioned data," in *IEEE ICDM workshop on privacy preserving data mining*, 2003, pp. 3–9.

[14] J. Vaidya and C. Clifton, "Privacy preserving naive bayes classifier for vertically partitioned data," in *Proceedings of the 2004 SIAM International Conference on Data Mining*. SIAM, 2004, pp. 522–526.

[15] J. Vaidya, B. Shafiq, A. Basu, and Y. Hong, "Differentially private naive bayes classification," in *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, vol. 1. IEEE, 2013, pp. 571–576.

[16] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, "Differentially private naive bayes learning over multiple data sources," *Information Sciences*, vol. 444, pp. 89–104, 2018.

[17] R. Bassily, K. Nissim, U. Stemmer, and A. G. Thakurta, "Practical locally private heavy hitters," in *Advances in Neural Information Processing Systems*, 2017, pp. 2288–2296.

[18] G. Cormode, T. Kulkarni, and D. Srivastava, "Marginal release under local differential privacy," in *Proceedings of the 2018 International Conference on Management of Data*. ACM, 2018, pp. 131–146.

[19] B. Cyphers and K. Veeramachaneni, "Anonml: Locally private machine learning over a network of peers," in *Data Science and Advanced Analytics (DSAA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 549–560.