Differentially Private Projected Histograms of Multi-Attribute Data for Classification

Dong Su *, Jianneng Cao *, Ninghui Li #

# Purdue University
{sul7, ninghui}@cs.purdue.edu

* Institute for Infocomm Research
caoj@i2r.a-star.edu.sg

ABSTRACT

In this paper, we tackle the problem of constructing a differentially private synopsis for the classification analyses. Several state-of-the-art methods follow the structure of existing classification algorithms and are all iterative, which is suboptimal due to the locally optimal choices and the over-divided privacy budget among many sequentially composed steps. Instead, we propose a new approach, PrivPfC, a new differentially private method for releasing data for classification. The key idea is to privately select an optimal partition of the underlying dataset using the given privacy budget in one step. Given one dataset and the privacy budget, PrivPfC constructs a pool of candidate grids where the number of cells of each grid is under a data-aware and privacy-budget-aware threshold. After that, PrivPfC selects an optimal grid via the exponential mechanism by using a novel quality function which minimizes the expected number of misclassified records on which a histogram classifier is constructed using the published grid. Finally, PrivPfC injects noise into each cell of the selected grid and releases the noisy grid as the private synopsis of the data. If the size of the candidate grid pool is larger than the processing capability threshold set by the data curator, we add a step in the beginning of PrivPfC to prune the set of attributes privately. We introduce a modified \( \chi^2 \) quality function with low sensitivity and use it to evaluate an attribute’s relevance to the classification label variable. Through extensive experiments on real datasets, we demonstrate PrivPfC’s superiority over the state-of-the-art methods.

1. INTRODUCTION

We study the problem of publishing histograms of datasets while satisfying differential privacy. A histogram is an important tool for summarizing data, and can serve as the basis for many data analysis tasks. Publishing noisy histograms for one-dimensional or two-dimensional datasets have been studied extensively in recent years [21, 41, 8, 26, 4, 40, 39, 33, 32]. However, as noticed in [39, 33], these approaches do not work well when the number of attributes/dimensions goes above a few. Many datasets that are of interest have multiple attributes. In this paper, we focus on multi-attribute datasets that have dozens of attributes, some of categorical and some are numerical.

For such a multi-attribute dataset, it is infeasible to publish a histogram with all the attributes, therefore it is necessary to select a subset of the attributes that are “interesting” for some intended data analysis tasks, and to determine how to discretize the attributes. These selection partition the domain into a number of cells. We call the result a “grid”. We consider a common optimization objective, where the dataset includes a label attribute and our goal is to ensure that classifiers that are accurate for the original dataset can be learnt from the published noisy histograms. Classification is an important tool for data analysis, and differentially private algorithms for learning classifiers have been considered an important problem, with many recent attempts [3, 6, 7, 30, 7, 24, 23, 39, 36, 45].

In this paper we propose the PrivPfC (Private Publication for Classification) approach for publishing projected histograms. The key novelty is to privately select a high-quality grid in a single step, while adapting to the privacy budget \( \epsilon \). We construct a set of candidate grids where the number of cells is under a certain threshold (determined by the dataset size and the privacy budget), and then use the exponential mechanism to select one grid using a novel quality function that minimizes expected number of misclassified records when a histogram classifier is constructed using the published histogram. By construction, our quality function considers the impact of injected Laplace noise to the histogram on the classification accuracy.

For high dimensional datasets, the size of the set of candidate grids might be larger than the processing capacity of the data curator. We add a feature selection step in the beginning to prune the set of attributes. This step first privately selects a small number of attributes which are most relevant to the class attribute by employing the exponential mechanism. We introduce a modified \( \chi^2 \) correlation function that has low sensitivity while evaluating an attribute’s relevance to the classification label variable. This feature selection step enables our PrivPfC framework to scale to higher dimensional datasets.

Our proposed PrivPfC outputs a histogram that can be used to generate synthetic data for multiple data analysis tasks, while being optimized for data classification. We show the effectiveness of PrivPfC by comparing it with several other approaches that output a classifier in a differentially private fashion.

For evaluation, we use two common classification algorithms, the decision tree and the SVM, because these have been used in the literature on learning classifiers while satisfying the differential privacy. Extensive experiments on real datasets show that PrivPfC consistently and significantly outperforms other state-of-the-art methods.

The contributions of this paper are summarized as follows:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright 20XX ACM X-XXXXXX-XX-X/XX/XX ...$15.00.
1. We propose PrivPfC, a novel framework for publishing data for classification under differential privacy. As part of PrivPfC, we introduce a new quality function that enables the selection of a good “grid” for publishing noisy histograms. We also introduce a way to enable private selection of most relevant features for classification, and to enable PrivPfC to scale to higher-dimension datasets.

2. Through extensive experiments on real datasets, we have compared PrivPfC against several other state-of-the-art methods for data publishing as well as private classification, demonstrating that PrivPfC improves the state-of-the-art.

The rest of the paper is organized as follows. In Section 2, we review the related work. Our PrivPfC approach is presented in Section 3. We report experimental results in Section 4. Section 5 concludes our work.

2. RELATED WORK

The notion of differential privacy was developed in a series of papers [11, 14, 3, 13, 12]. There are several primitives for satisfying $\epsilon$-differential privacy. In this paper we use two of them. The first primitive is the Laplacian mechanism [13]. It adds noise sampled from a Laplace distribution to a statistic $f$ to be released. The scale of the Laplace distribution is proportional to $SG_q$, the global sensitivity or the $L_1$ sensitivity of $f$. Another primitive is to sample the output of the data analysis mechanism according to an exponential distribution; this is generally referred to as the exponential mechanism [28]. The mechanism relies on a quality function $q : D \times \mathbb{R} \rightarrow \mathbb{R}$ that assigns a real valued score to one output $r \in \mathcal{R}$ when the input dataset is $D$, where higher scores indicate more desirable outputs. Given the quality function $q$, its global sensitivity $SG_q$ is defined as:

$$SG_q = \max_{r \in \mathcal{R}} \max_{D \in D} |q(D, r) - q(D', r)|.$$

The following method satisfies $\epsilon$-differential privacy:

$$\Pr \{ r \text{ is selected} \} \propto e\left(\frac{\epsilon}{2SG_q}q(D, r)\right).$$

There has been a large body of works on differentially private histogram construction for answering range queries or marginal queries [13, 40, 29, 19, 42, 33, 27, 34, 35].

Differentially Private Classification. Differentially private classification has received growing attention in the research community. Blum et al. [8] suggested a solution for constructing the private version of the ID3 decision tree classifier. When the ID3 algorithm needs to get the number of tuples with a specific feature value, it queries the SuLQ interface to get the corresponding noise count. Friedman and Schuster [17] improved this approach by redesigning the classic ID3 classifier construction algorithm to consider the feature quality function with low sensitivity and using exponential mechanism to evaluate all the attributes simultaneously. Chaudhuri et al. [9, 10] proposed a differentially private logistic regression algorithm and later generalized this idea to address the private empirical risk minimization which can be applied to a wider range of classification problems, such as SVM classification. Zhang et al. [36] proposed PrivGene, a general private model fitting framework based on genetic algorithms, that can be applied to the SVM classification and the logistic regression.

Besides the above interactive methods for constructing differentially private classifiers, several works proposed solutions to publish data for classification analysis tasks. Mohammed et al. [30] proposed the DiffGen algorithm which first partitions the data domain by iteratively selecting attributes and ways to discretize the attributes, and then injects Laplace noise into each cell of all the leaf partitions. Vinterbo [39] proposed another data publishing algorithm, called Private Projected Histogram (PPH). PPH first decides how many attributes are to be selected, then incrementally selects attributes via the exponential mechanism to maximize thediscriminability of the selected attributes. For each categorical attribute, the full domain is used. For numerical attribute, it uses the formula proposed in Lei [26] to decide how many bins to discretize them. In this method, the number of attributes and how attributes are partitioned are independent of the privacy budget. Furthermore, all selected attributes are treated equally. Zhang et al. [25] presented PrivBayes which constructs a private a Bayesian network through iteratively selecting sets of attributes that have maximum mutual information via the exponential mechanism. It then injects Laplace noise to perturb each conditional distribution of the network. We will further analyze the above approaches and compare our proposed method with them in the later sections.

3. PrivPfC FRAMEWORK

In this section we present the PrivPfC framework of privately publishing data for classification analysis.

3.1 Preliminaries

We consider a dataset with a set of predictor variables and one binary response variable. The predictor variables can be numerical or categorical. Following [2, 22, 13, 30], for each predictor variable $A_i$, we assume the existence of a taxonomy hierarchy (also called a generalized hierarchy in the literature) $T_i$. Figure 1 shows the taxonomy hierarchies of Relationship, a categorical variable, and Education-num, a numerical variable. In the hierarchy, the root node represents the whole domain of the variable, and a parent node is a generalization (or a cover) of its children. Child nodes under the same parent node are semantically related; they are closer to each other than to nodes under a different parent node.

Each level of a predictor variable’s taxonomy hierarchy forms a partition of its domain. On the basis of the taxonomy hierarchy and its levels, we introduce the notion of a grid.

**Definition 1 (Grid).** Let $A = \{A_1, \ldots, A_d\}$ be the set of predictor variables in a dataset and $\{T_1, \ldots, T_d\}$ be their taxonomy hierarchies respectively. Let $h_i$ be the height of $T_i$, $1 \leq i \leq d$. Then, a grid $g$ is given by $\{\ell_1, \ldots, \ell_d\}$, where $1 \leq \ell_i \leq h_i$ and $1 \leq i \leq d$. A grid defines a partition of the data domain into cells where each attribute $A_i$ is partitioned into the values at level $\ell_i$. The number of cells of a grid is $\Pi_{i=1}^d |T_i[\ell_i]|$, where $|T_i[\ell_i]|$ is the number of nodes in the level $\ell_i$ of the hierarchy $T_i$. The number of all possible grids is $\Pi_{i=1}^d h_i$.

**Definition 2 (Histogram).** Given a dataset $D$ and a grid $g$, a histogram $H(D, g)$ partitions $D$ into cells according to $g$, and outputs the numbers of positive instances and negative instances in each cell.

By injecting Laplace noise into the positive counts and negative counts of each cell in the histogram $H(D, g)$, we get the noisy version of it, $H(D, g)$.

3.2 Histogram Publishing for Classification

Given a dataset $D$, the taxonomy hierarchies of its predictor variables, a total privacy budget $\epsilon$, and the number of tuples in the dataset $N$ (a rough estimate suffices), we generate a candidate pool
of all grids whose number of cells are below a threshold, which is determined by \( \epsilon \) and \( N \). We compute the quality score for each grid, which measures the usefulness for classification of each grid in the pool. We then apply the exponential mechanism \( \text{EMMA} \) to privately select a grid, and finally publish a noisy histogram using \( g \).

A key technical challenge is to come up with a low-sensitivity quality function that can measure the desirability of choosing a particular grid \( g \). We publish \( H(D,g) \), a noisy histogram of \( D \) using \( g \) to partition the data domain, and desire that classifiers learned from \( H(D,g) \) are close to classifiers learned from \( D \). Furthermore, we desire this to hold regardless of which particular classification algorithm is used. We propose to define the quality function to minimize the misclassification error (when measured using the dataset \( D \)) for the classifier defined by the histogram \( H(D,g) \), i.e., each cell in the grid defined by \( g \), it predicts the majority class according to \( H(D,g) \). This classifier is in the same spirit as histogram classifiers [10][31], and we use \( HC_{H(D,g)} \) to denote it.

Suppose that a grid is able to separate positive and negative data points very well, then, even after adding the noises, this separation counts. Since when a grid \( g \) is fixed, the noisy histogram includes random noises, the misclassification error is a random variable, and we use the expected value of this error as the quality function.

**Definition 3 (Quality of Grid).** Given a dataset \( D \) and a grid \( g \), the grid quality is measured by the expected misclassification error of the histogram classifier \( HC_{H(D,g)} \):

\[
\text{qual}_D(g) = E[err(HC_{H(D,g)}, D)].
\]

The following Lemma shows how to compute \( \text{qual}_D(g) \).

**Lemma 1 (Quality of Grid).** Given a dataset \( D \) and a grid \( g \), \( \epsilon \) for the parameter of adding Laplacian noise to the counts, we have

\[
\text{qual}_D(g) = \sum_{c \in g} \left[ \min(n^+_c, n^-_c) \left( 1 - \frac{e^{-\epsilon x_c}}{2} \left( 1 + \frac{\epsilon x_c}{2} \right) \right) \right. \\
+ \left. \max(n^+_c, n^-_c) \left( \frac{e^{-\epsilon x_c}}{2} \left( 1 + \frac{\epsilon x_c}{2} \right) \right) \right]
\]

where \( c \) ranges over all cells in the grid, \( n^+_c \) is the number of positive data points in \( c \), \( n^-_c \) is the number of negative data points in \( c \), and \( x_c = |n^+_c - n^-_c| \).

To prove Lemma \( \text{II} \) we note that \( \text{qual}_D(g) \) can be further decomposed into the sum of expected misclassification error at each perturbed cell of the histogram after majority voting, and thus

\[
\text{qual}_D(g) = \sum_{c \in g} E[err(\tilde{c}, D)]
\]

Where \( \tilde{c} \) denotes the application of the histogram classifier \( HC_{H(D,g)} \) to the cell \( c \).

For cell \( c \), if the added Laplace noises do not change the majority class label, then the number of misclassified input tuples is \( \min(n^+_c, n^-_c) \); otherwise, it is \( \max(n^+_c, n^-_c) \). Thus,

\[
E[err(\tilde{c}, D)] = \min(n^+_c, n^-_c) \cdot p_c + \max(n^+_c, n^-_c) \cdot (1 - p_c).
\]

(2)

where \( p_c \) is the probability that the majority class label in \( c \) does not change after injecting Laplace noises.

Let \( Z_1 \) and \( Z_2 \) be the Laplace noises added to the majority class and the minority class of cell \( c \), respectively, then

\[
p_c = \Pr[|Z_2 - Z_1| \leq |n^+_c - n^-_c|].
\]

(3)

**Lemma 2 (Equation 3).** Let \( Z_1 \) and \( Z_2 \) be two i.i.d. random variables that follow the Laplace distribution with mean 0 and scale \( \frac{1}{\epsilon} \), then the density of their difference \( Y = Z_1 - Z_2 \) is

\[
f_Y(y) = \frac{e^{-\epsilon |y|}}{\epsilon} (1 + \epsilon |y|), \quad \infty < y < \infty,
\]

and the corresponding cumulative distribution function is

\[
F_Y(y) = \begin{cases} 
1 - \frac{e^{-\epsilon y}}{2} \left( 1 + \frac{\epsilon y}{2} \right), & \text{if } y \geq 0, \\
\frac{e^{\epsilon y}}{2} \left( 1 - \frac{\epsilon y}{2} \right), & \text{otherwise}.
\end{cases}
\]

(4)

From Equations (3) and (4), we have

\[
p_c = 1 - \frac{e^{-\epsilon |n^+_c - n^-_c|}}{2} \left( 1 + \frac{\epsilon |n^+_c - n^-_c|}{2} \right).
\]

(5)

Plugging Equation (5) into Equation (2) provides Lemma \( \text{II} \).

The lemma below bounds the sensitivity of our quality function.

**Lemma 3.** For any \( \epsilon > 0 \), the global sensitivity of the quality function \( \text{II} \) is \( B(\epsilon) \), where

\[
B(\epsilon) = x \left( e^{-\epsilon x} \left( 1 + \frac{\epsilon (x-1)}{2} \right) - e^{-\epsilon x} \left( 1 + \frac{\epsilon x}{2} \right) \right)
\]

\[
+ \frac{1 - e^{-\epsilon (x-1)}}{2} \left( 1 + \frac{\epsilon (x-1)}{2} \right),
\]

and

\[
x = \frac{\epsilon^2 \sqrt{2 - (4 - \epsilon^2) e^\epsilon + 2 \epsilon^2 e^\epsilon}}{-\epsilon + \epsilon e^\epsilon}.
\]

### 3.3 Correlation-based Feature Selection

Our basic solution privately selects a grid from the candidate pool to release synthetic data. As the number of predictor variables increases, the candidate pool size grows exponentially, giving rise to a scalability issue. Fortunately, usually in a real dataset some
Algorithm 1 PrivPIC: Privately Publishing Data for Classification

Input: dataset $D$, the set of predictor variables $F$ and their taxonomy hierarchies, total privacy budget $\epsilon$, maximum grid pool size $B$, median of the first branching factors of hierarchies $b$.

1: function main($D, F, N, \epsilon, B, b$)
2: $T \leftarrow \delta \cdot N \cdot \epsilon^2 / 2$
3: $H \leftarrow \text{Enumerate}(F, T)$
4: if $|H| \geq B$ then
5: $k \leftarrow \left\lfloor \frac{2 \log T}{\log b} \right\rfloor$
6: $\epsilon_{fs} \leftarrow 0.3e$, $\epsilon_{sh} \leftarrow 0.3e$, $\epsilon_{ph} \leftarrow 0.4e$
7: $X \leftarrow \text{selectFeature}(D, F, k, \epsilon_{fs})$
8: $H_X \leftarrow \text{Enumerate}(X, T)$
9: $\hat{I} \leftarrow \text{PrivateHistogramPublishing}(D, H_X, \epsilon_{sh}, \epsilon_{ph})$
10: else
11: $\epsilon_{sh} \leftarrow \frac{4}{5} e$, $\epsilon_{ph} \leftarrow \frac{4}{5} e$
12: $\hat{I} \leftarrow \text{PrivateHistogramPublishing}(D, H, \epsilon_{sh}, \epsilon_{ph})$
13: end if
14: return $\hat{I}$
15: end function

function PrivateHistogramPublishing($D, H, \epsilon_{sh}, \epsilon_{ph}$)
16: $h \leftarrow \text{selectHist}(D, H, \epsilon_{sh})$
17: $\hat{I} \leftarrow \text{perturbHist}(D, h, \epsilon_{ph})$
18: return $\hat{I}$
19: end function

function selectHist($D, H, \epsilon_{sh}$)
20: for $i = 1 \rightarrow |H|$ do
21: $q_i \leftarrow \text{qual}(H_i)$
22: $p_i \leftarrow e^{-(q_i \cdot \epsilon_{sh}) / 2}$
23: end for
24: $h \leftarrow \text{sample } i \in [1,|H|] \text{ according to } p_i$
25: return $h$
26: end function

function perturbHist($D, h, \epsilon_{ph}$)
27: Initialize $I$ to empty
28: for each cell $c \in h$ do
29: $n_c^a \leftarrow n_c^e + \text{Lap}(1/\epsilon_{ph})$
30: $n_c^b \leftarrow n_c^e + \text{Lap}(1/\epsilon_{ph})$
31: Add $(n_c^a, n_c^b)$ to $I$
32: end for
33: Round all counts of $I$ to their nearest non-negative integers.
34: return $I$
35: end function

function selectFeature($D, F, k, \epsilon_{fs}$)
36: Initialize $X$ to empty
37: Let $R$ be the response variable in $D$
38: for each $A_i \in F$ do
39: $cor_i \leftarrow \text{Cor}(A_i, R, D)$
40: $p_i \leftarrow e^{\frac{cor_i - \epsilon_{fs}}{4e}}$
41: end for
42: for $i = 1 \rightarrow k$ do
43: $f \leftarrow \text{sample } A_i \in F \text{ according to } p_i$
44: Add $f$ to $X$
45: Remove $f$ from $F$
46: end for
47: return $X$
48: end function

3.4 The Algorithm
We now present the full algorithm (Algorithm 1) for our framework of releasing private data for classification tasks.

Line 2 sets the threshold of the maximum number of cells in a grid, to prevent the average counts from being dominated by the injected noises. That is,

$$E \left[ \text{Lap} \left( \frac{1}{\tau} \right) \right] \leq \frac{1}{\tau} \cdot \frac{N}{T},$$

which means that the average noise magnitude is no more than the 20% of the average cell count.
When feature selection is deemed necessary, we allocate 30% of the privacy budget to privately select $k$ predictor variables that are strongly correlated with the response variable. These selected variables are then used to release private synthetic data.

The number of attributes to be selected, $k$, is based on $T$, the maximum grid size. We want to have enough attributes so that relevant attributes are included. At the same time, we do not want $k$ so large so that there are too many candidate grids with size below $T$. Let $b$ be the median of the first branching factors of hierarchies of all attributes. We set $k$ to be $\frac{2 \log(T)}{\log(b)}$.

**Theorem 1.** Algorithm $\mathcal{A}$ satisfies $\epsilon$-differential privacy.

Theorem 1 shows that Algorithm PrivPIC satisfies $\epsilon$-differential privacy. The proof of Theorem 1 is thus straightforward by considering the sequential composability of differential privacy as discussed in Section 2.

### 4. EXPERIMENT

#### 4.1 Experimental Settings

**Datasets.** We use 4 real datasets for our experiments. The first one is the Adult dataset from the UCI machine learning repository $\text{[1]}$. It contains 6 numerical attributes and 8 categorical attributes, and is widely used for evaluating the performance of classification algorithms. After removing missing values, the dataset contains 45,222 tuples. The second dataset is the Bank marketing dataset from the same repository. It contains 10 numerical attributes and 3 categorical attributes on 41,188 individuals. The third is the US dataset from the Integrated Public Use Microdata Series (IPUMS) $\text{[36]}$. It has 39,186 the United States census records in 2010, with 15 numerical attributes and 31 categorical ones. The last is the BR dataset (also from IPUMS), which contains 57,333 Brazil census records in 2010 and has 14 numerical attributes and 28 categorical ones. The classification tasks for the Adult, US and BR datasets are to predict whether an individual has an income above a certain threshold. The one for the Bank dataset is to predict whether a client will subscribe a term deposit. Table 1 summarizes the characteristics of the datasets.

**Taxonomy Hierarchies.** For the Adult dataset, we use the same taxonomy hierarchies as DiffGen $\text{[30]}$. For the remaining 3 datasets, we do the following. For numerical attributes, we partition each domain into equal size bins and build hierarchies over them. For categorical attributes, we build taxonomy hierarchies by considering the semantic meanings of the attribute values.

**Competing Methods.** We compare PrivPIC with 6 state-of-the-art methods in terms of misclassification rate. These include 3 non-interactive methods, DiffGen $\text{[39]}$, PrivBayes $\text{[25]}$, and Private Projected Histogram (PPH) $\text{[39]}$, which privately release synthetic datasets for classification analyses, and 3 interactive methods, PrivGene $\text{[46]}$, DiffPC-4.5 $\text{[17]}$, and PrivateERM $\text{[7]}$. We set one method for decision tree and two methods for SVM.

**DiffGen.** $\text{[30]}$ consists of two steps, partition and perturbation. The partition step first generalizes all attribute’s values into the topmost nodes in their taxonomy hierarchies and then iteratively selects one attribute at a time for specialization, using the exponential mechanism. The quality of each candidate specialization is based on the same heuristics as used by the decision tree algorithms, such as information gain and majority class. As suggested in $\text{[30]}$, we use the majority class to measure the candidate quality, and set the number of specialization steps to be 10 for the Adult dataset and the bank dataset. For the US and BR datasets, we set the number to be 6 and 8 respectively, as beyond these numbers, the DiffGen implementation runs into memory problems. The perturbation step injects Laplace noise into each cell of the partition and outputs all the cells with their noisy counts as the noisy synopsis of the data.

**PrivBayes.** $\text{[45]}$ determines the structure of a Bayesian network by first randomly select an attribute as the first node, and then iteratively select one attribute according to the set of attribute’s parent nodes, which have the maximum mutual information. After the structure is determined, PrivBayes perturbs the marginals needed for computing the conditional distributions. The performance of the PrivBayes algorithm depends on $k$. We set $k = 3$ for the Adult dataset and the Bank dataset, which is the same as the one used in $\text{[45]}$. For the US and BR datasets, which were not used in $\text{[45]}$, setting $k = 3$ runs out of memory in our experiments because of the larger dimensionality; we set $k = 2$ for them.

**PPH.** $\text{[39]}$ starts with a feature selection procedure to select a set of $k$ features that have the maximal discernibility. Then, it uses the selected features to build a histogram. For each categorical attribute, the full domain is used. For numerical attribute, it uses the formula proposed in Lei $\text{[26]}$ to decide how many bins to discretize them.

**PrivGene** $\text{[46]}$ is a general-purpose private model fitting framework based on genetic algorithms, which can be applied to SVM classification. DiffPC-4.5 $\text{[17]}$ is an interactive private algorithm for building a C4.5 decision tree classifier differentially-privately. PrivateERM $\text{[7]}$ is an interactive private algorithm for constructing SVM classifier by injecting noise into the risk function first and then optimizing the perturbed risk function.

The source codes of the DiffGen, PrivBayes, PPH, DiffPC-4.5, and PrivateERM were downloaded from $\text{[29]}$, $\text{[43]}$, $\text{[38]}$, $\text{[16]}$, and $\text{[44]}$, respectively. The source code of PrivateERM was shared by the authors of PrivBayes $\text{[45]}$.

**Evaluation Methodology.** We consider two baselines – Majority and NoiseFree. Majority is the misclassification rate by majority voting on the class attribute, which predicts each test case with the majority class label in the train dataset. NoiseFree is the misclassification rate of a decision tree or SVM classifier built on the true data. We expect that a good algorithm to perform better than Majority, and gets close to NoiseFree as $\epsilon$ increases.

The evaluation is based on two classification models: the CART decision tree classifier and the SVM classifier with radial basis kernel. Interactive approaches DiffPC-4.5 and PrivateERM build private classifiers directly. And we use parameters suggested by the corresponding papers. The non-interactive approaches PrivPIC, PPH, DiffGen, and PrivBayes generate private synthetic datasets. To evaluate their performance in terms of decision tree model, we use the rpart $\text{[37]}$ library to build decision trees on their generated synthetic datasets. For the evaluation in terms of SVM model, we use the LibSVM package $\text{[3]}$ to build SVM classifiers on the synthetic datasets. We use the same set of parameters of rpart and LibSVM respectively in evaluating the above non-interactive approaches.

For all the experiments, we vary $\epsilon$ from 0.05 to 1.0. Similar to the experiment settings of $\text{[17]}$, $\text{[30]}$, $\text{[39]}$, under each privacy budget, we execute 10-fold stratified cross-validation to evaluate the misclassification rate of the above methods. For each train-test pair, we run the target method 10 times. We report the average measurements over the 10 runs and the 10-fold crossvalidations. We set the maximum grid pool size to be 200,000. The implementation and experiments of PrivPIC were done in Python 2.7 and all experiments were conducted on an Intel Core i7-3770 3.40GHz PC with 16GB memory.
| Dataset | # Dim | # Numerical | # Categorical | # Records | Classification Task |
|---------|-------|-------------|---------------|-----------|---------------------|
| Adult   | 15    | 6           | 8             | 45,222    | Determine whether a person makes over 50K a year. |
| Bank    | 21    | 10          | 10            | 41,188    | Determine whether the client subscribed a term deposit. |
| US      | 47    | 15          | 31            | 39,186    | Determine whether a person makes over 50K a year. |
| BR      | 43    | 14          | 28            | 57,333    | Determine whether a person makes over 300 per month. |

Table 1: Dataset characteristics

| Methods              | Description                                      |
|----------------------|--------------------------------------------------|
| PrivPfC              | Our proposed method.                             |
| PrivPfC-SelNF        | Our proposed method with noise free feature selection. |
| PrivPfC-PSNF         | Our proposed method with noise free feature selection. |
| DiffGen [30]         | Private data release for classification via recursive partitioning. |
| DiffGen-NF           | Noise free DiffGen.                              |
| PrivBayes [45]       | Private Data Release via Bayes network.           |
| PrivBayes-NF         | Noise free PrivBayes.                            |
| PPH [39]             | Private data release for classification by projection and perturbation. |

Table 2: Summary of differentially private classification methods

Figure 2: Comparison of PrivPfC, DiffGen, PrivBayes, PPH and DiffPC-4.5 by decision tree classification. x-axis: privacy budget \( \epsilon \) in log-scale. y-axis: misclassification rate in log-scale.

4.2 Comparison against Competitors

Comparison on Decision Tree. Five approaches are involved: PrivPfC, DiffGen, PrivBayes, PPH and DiffPC-4.5. Figure 2 reports their average misclassification rates and the corresponding standard deviations. Clearly, PrivPfC has the best performance, followed by DiffGen, PPH, DiffPC-4.5. PrivBayes is the poorest in most cases. The performance of PrivPfC is also the most robust, as can be seen from the fact that the standard deviation of its misclassification rates is the lowest.

Comparison on SVM. We compare 6 approaches: PrivPfC, Diff-
Gen, PrivBayes, PPH, PrivGene, and PrivateERM. Figure 3 reports the experimental results. Once again, PrivPfC has the best performance, followed by DiffGen, PrivGene, PPH, PrivateERM and PrivBayes.

**Effectiveness of Private Feature Selection.** In Figure 4, we evaluate our private feature selection method on the US dataset under privacy budget 0.1. We create a variant of PrivPfC, called PrivPfC-FSNF, in which the feature selection step ofPrivPfC is noise-free and all the privacy budget is used in remaining steps. PPH is included in the comparison since it also has a private feature selection step. We create variants for each of the rest competitors, by adding our proposed feature selection method as preprocessing step which uses 30% of the total privacy budget.

From Figure 4, we can see that our PrivPfC algorithm has close performance to its counterpart (PrivPfC-FSNF). This justifies the fact that the set of attributes PrivPfC selects for grid partition is almost as good as those selected by PrivPfC-FSNF and the effectiveness of PrivPfC mainly comes from the private histogram selection. We can also see that although PrivBayes, DiffPC-4.5 and PrivateERM’s performances are improved significantly by doing our private feature selection step, they are still outperformed by PrivPfC.

### 4.3 Analyses of Sources of Errors

PrivPfC distributes the privacy budget among three steps, feature selection, grid selection and perturbation, in a 30%-30%-40% way. When feature selection is not needed, the privacy budget is divided between grid selection and perturbation in a ratio of 3:4. While these ratios are somewhat arbitrary, we have experimentally evaluated other ratios, allocating between 20% and 60% to each step. We have found that the differences among different budget allocations are minor, so long as the last step receives at least 30% of the privacy budget. Even with the worst allocation, which gives 20% to the last step, PrivPfC still clearly outperforms competing methods. We also consider a variant of PrivPfC, called PrivPfC-SelNF, in which the feature selection step and histogram selection step are noise free and all the privacy budget is used in the histogram perturbation step. PrivPfC-SelNF is not private; it shows the best one can hope to achieve by optimizing the division among steps.

We have seen that PrivPfC outperforms the other non-interactive methods such as DiffGen and PrivBayes. The key difference in PrivPfC is that we choose the grid $g$ holistically, instead of arriving at the final grid through a series of decisions. For example, DiffGen iteratively chooses the attributes and ways to partition them, and PrivBayes iteratively builds a Bayesian network. There are two reasons why such an iterative approach does not perform well. The first is that the decisions made in each iteration may be sub-optimal because of the perturbation necessary for satisfying differential privacy. The second is that even if the decision made in each iteration is locally optimal, the combination of them is not globally optimal. To see to what extent the latter factor affects accuracy, we consider noise free variants of them respectively, DiffGen-NF and PrivBayes-NF. In these variants the decisions in each iteration as well as the publishing of counts in the end are performed without any perturbation. They represent DiffGen and PrivBayes when the privacy budget $\epsilon$ goes to $\infty$. 

![Figure 3: Comparison of PrivPfC, DiffGen, PrivBayes, PPH, PrivGene and PrivateERM by SVM classification. x-axis: privacy budget $\epsilon$ in log-scale. y-axis: misclassification rate in log-scale.](image_url)
We also observe that the non-private noise-free version of PrivBayes still performs poorly; in fact, it performs significantly worse than the private PrivPfC. This suggests that the iterative Bayes network construction approach is not suitable for the purpose of building accurate classifiers. This is perhaps due in large part to the fact that it is not designed originally to optimize for classification.

The non-private DiffGen-NF performs similarly to PrivPfC and PrivPfC-SelNF on the Adult and US datasets. On the Bank dataset, it is outperformed by PrivPfC and PrivPfC-SelNF when $\epsilon \geq 0.15$. 

Figure 5 and Figure 6 report the experimental results of comparing these methods, using Decision Tree and SVM, respectively. We first observe that while PrivPfC-SelNF indeed outperforms PrivPfC, the difference is very small, especially for larger $\epsilon$ values in the range. In fact, on Adult, US, and BR datasets, the difference is barely noticeable when $\epsilon \geq 0.1$. This suggests that little improvement can be gained to further optimize the division of privacy budget or dataset among determining grid $g$ and publishing noisy histogram.

Figure 4: Effectiveness of Private Feature Selection on US dataset with $\epsilon = 0.1$. y-axis: misclassification rate.

(a) Decision Tree (b) SVM

(a) Adult (b) Bank (c) US (d) BR

Figure 5: Analyses of PrivPfC, DiffGen and PrivBayes by decision tree classification. x-axis: privacy budget $\epsilon$ in log-scale. y-axis: misclassification rate in log-scale.
Figure 6: Analyses of PrivPfC, DiffGen and PrivBayes by SVM classification. x-axis: privacy budget $\epsilon$ in log-scale. y-axis: misclassification rate in log-scale.

On the BR dataset, DiffGen-NF performs significantly worse than PrivPfC and PrivPfC-SelNF. This suggests that the inherent iterative structure of DiffGen is suboptimal, even without considering the effect of perturbation.

5. CONCLUSION

In this paper, we have introduced PrivPfC, a novel framework for publishing data for classification under differential privacy. As a core part of PrivPfC, we have introduced a novel quality function that enables the selection of a good “grid” for publishing noisy histograms. We have also introduced a new technique for privately selecting of most relevant features for classification, which enables PrivPfC to scale to higher-dimension datasets. We have conducted extensive experiments on four real datasets, and the results show that our approach greatly outperforms several other state-of-the-art methods for private data publishing as well as private classification.

6. REFERENCES

[1] A. Asuncion and D. Newman. UCI machine learning repository, 2010.
[2] R. J. Bayardo and R. Agrawal. Data privacy through optimal $k$-anonymization. In ICDE, pages 217–228, 2005.
[3] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the sulq framework. In PODS, pages 128–138, 2005.
[4] T.-H. H. Chan, E. Shi, and D. Song. Private and continual release of statistics. ACM Trans. Inf. Syst. Secur., 14(3):26:1–26:24, Nov. 2011.
[5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[6] K. Chaudhuri and C. Monteleoni. Privacy-preserving logistic regression. In NIPS, pages 289–296, 2008.
[7] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. The Journal of Machine Learning Research, 12:1069–1109, 2011.
[8] G. Cormode, M. Procopiuc, D. Srivastava, and V. S. P. Tran. Differentially private summaries for sparse data. ICDT ’12, pages 299–311, 2012.
[9] G. Cormode, M. Procopiuc, E. Shen, D. Srivastava, and T. Yu. Differentially private spatial decompositions. In ICDE, pages 258–269, 2012.
[10] L. Devroye, L. Györfi, and G. Lugosi. A probabilistic theory of pattern recognition. Applications of mathematics. Springer, New York, Berlin, Heidelberg, 1996.
[11] I. Dinur and K. Nissim. Revealing information while preserving privacy. In PODS, pages 202–210, 2003.
[12] C. Dwork. Differential privacy. In ICALP, pages 1–12, 2006.
[13] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In TCC, pages 265–284, 2006.
[14] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In CRYPTO, pages 528–544,
7. APPENDIX

Proof of Lemma\[4\] Without loss of generality, we assume the cell \(o_{11}\) is changed by 1. We use \(c_i\) to denote the number of tuples with \(R = r_i, i = 1, 2\).

\[
\Delta = |\text{corr}(f, D') - \text{corr}(f, D)|
\]

\[
= \left( o_{11} + 1 - \frac{(o_{11} + o_{12} + 1)(c_1 + 1)}{c_1 + c_2 + 1} \right) - \left( o_{11} - \frac{(o_{11} + o_{12})c_1}{c_1 + c_2} \right)
\]

\[
+ \sum_{i=2}^{m} \left( o_{11} - \frac{(o_{11} + o_{i2})(c_1 + 1)}{c_1 + c_2 + 1} \right) - \left( o_{11} - \frac{(o_{11} + o_{i2})c_1}{c_1 + c_2} \right)
\]

\[
\leq 1 - \frac{(o_{11} + o_{12} + 1)(c_1 + 1)}{c_1 + c_2 + 1} + \frac{(o_{11} + o_{12})c_1}{c_1 + c_2}
\]

\[
+ \sum_{i=2}^{m} \left( \frac{(o_{11} + o_{i2})(c_1 + 1)}{c_1 + c_2 + 1} + \frac{(o_{11} + o_{i2})c_1}{c_1 + c_2} \right)
\]

\[
= \frac{(c_1 + c_2)(c_1 + c_2 + 1) - (o_{11} + o_{12}c_2 - (c_1 + 1)(c_1 + c_2))}{(c_1 + c_2)(c_1 + c_2 + 1)}
\]

\[
+ 2c_2((c_1 - o_{11}) + (c_2 - o_{12}))
\]

\[
\frac{c_1 + c_2}{c_1 + c_2 + 1}
\]

\[
\leq 2.
\]

\[\square\]