공학석사학위논문

Differentially private multi-class classification using kernel supports and equilibrium points

커널 서포트와 평형점을 활용한 차분 프라이버시 다중 클래스 분류 기법

2022 년 2 월

서울대학교 대학원

산업공학과

박 진 성

Differentially private multi-class classification using kernel supports and equilibrium points

커널 서포트와 평형점을 활용한 차분 프라이버시 다중 클래스 분류 기법

지도교수 이 재 욱

이 논문을 공학석사 학위논문으로 제출함

2021 년 12 월

서울대학교 대학원

산업공학과

박 진 성

박진성의 공학석사 학위논문을 인준함

2021 년 12 월

위 원 장 조 성 준 (인)

부위원장 이 재 욱 (인)

위 원 장 우 진 (인)

Abstract

Differentially private multi-class classification using kernel supports and equilibrium points

Jinseong Park
Department of Industrial Engineering
The Graduate School
Seoul National University

In this paper, we propose a multi-class classification method using kernel supports and a dynamic system under differential privacy. We find support vector machine (SVM) algorithms have a fundamental weaknesses of implementing differential privacy because the decision function depends on some subset of the training data called the support vectors. Therefore, we develop a method using interior points called equilibrium points (EPs) without relying on the decision boundary. To construct EPs, we utilize a dynamic system with a new differentially private support vector data description (SVDD) by perturbing the sphere center in the kernel space. Empirical results show that the proposed method achieves better performance even on small-sized datasets where differential privacy performs poorly.

Keywords: Differential privacy, Machine learning, Support vector data description, Support vector machine, Dynamic System, Industrial engineering
Student Number: 2020-26472

Contents

List of Tables

List of Figures

Chapter 1

Introduction

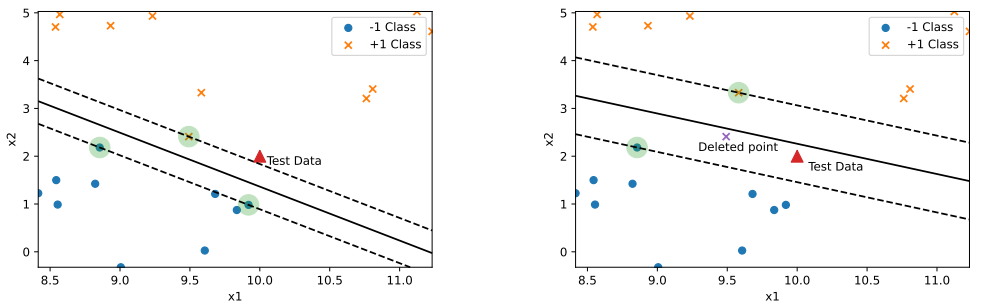
1.1 Problem Description: Data Privacy

Data privacy is an important issue for protecting data owner's sensitive information when collecting, storing, and sharing the data. As machine algorithms usually learn patterns from training data, the algorithms possess the risk of information leakages from the data and the training process. In other words, an attacker may learn the private data from models.

The EU implemented the General Data Protection Regulation (GDPR) in May 2018, proposing privacy standards such as the right to be informed, the right to be forgotten, and automated decision-making. Data management and control are increasingly being strengthened, including the enforcement of the California Consumer Privacy Act (CCPA) in the USA and the 'Three laws of data' in Korea. Furthermore, the vulnerability of anonymization, the most basic method for data protection, has emerged. In the Netflix Prize competition, the training data was anonymized and disclosed for the training of the recommendation algorithm, which was then attacked by linking with other rating data to personalize users [30].

To deal with the concerns, differential privacy (DP) [9] has become a formal mathematical framework for guaranteeing the privacy of data. Differential privacy

gives strong privacy for an individual's input to arbitrary functions while allowing useful computations on the private data. Therefore, guaranteeing differential privacy is essential to provide data analytic services. For instance, private medical data such as biomedical data and diseases cannot be revealed to the public. To achieve privacy protection, Laplace or Gaussian noise should be added to input, output, or parameters in the model according to their sensitivity.



(a) SVM of the whole dataset (b) SVM after deleting one support vector

Figure 1.1: Illustration of the vulnerability of SVM under differential privacy.

1.2 The Privacy of Support Vector Methods

Support vector based methods [3, 40, 42], one of the popular machine learning algorithms, have been successfully solved diverse pattern recognition problems. Support vector machine (SVM) by Vapnik [42] is the most popular method based on structured risk minimization and kernel functions. SVM primarily maximizes the margin of decision boundary between two-class and thus many studies have decomposed a multi-class problem into several two-class problems such as one-against-all or one-against-one strategies. In contrast to deep learning algorithms, which need massive data to train and thus relatively free from modification to one sample [1, 12], ma-

chine learning algorithms are usually trained with relatively small data since they perform well on generalization tasks through margin maximization and non-linear kernels. But at the same time, a small size of dataset is critical in terms of differential privacy since the sensitivity is usually inversely proportional to the number of data samples in ERM [6]. In summary, due to the high sensitivity on a small dataset, we should consider support vector methods that maintain comparable performance with a large dataset.

Several works have focused on privacy-preserving SVM [6, 17, 36]. However, we found that SVM framework may not be suitable for differential privacy with two drawbacks. First, the decision boundary of SVM depends on some subset of the training data called the support vectors. Therefore, publishing support vectors possesses high risk as the important information of the training set is concentrated on few points. Second, as shown in Figure 1.1, SVM is highly sensitive to a modification of a support vector considering that differential privacy focuses on the worst case of changing one point. In Figure 1.1, the predictions of a model could be radically changed when one support vector is modified, where differential privacy should guarantee the worst case. The test data (red triangle) was predicted as +1 class (O) in (a), but the decision boundary changed rapidly after deleting one support vector (purple x) and thus test data is classified to -1 class (X) in (b). The solid line represents the decision boundary and the points on dotted line with green circles are support vectors.

1.3 Research Motivation and Contribution

To deal with above issues, we propose a new multi-class classification method focusing on interior points called equilibrium points (EPs) which are local optimum of support level function. We choose support vector data description (SVDD) [40] as the support function, which finds a hyper-sphere with the minimum radius around data in the kernel space. This clustering-based approach solves the two problems of SVM mentioned above because the equilibrium points are not one of the train data and the new algorithm focuses not on the decision boundaries but on the interior points. Our research motivations and main contributions of the thesis are as follow:

- We prove differential privacy guarantees for support vector data description (SVDD) by perturbing the sphere center in the kernel space. To the best of our knowledge, this is the first approach for guaranteeing differential privacy of support vector data description.

- We propose a new multi-class classification with kernel supports focusing on interior points called equilibrium points (EPs).

- We achieve higher performance under differential privacy than previous differentially private SVM models on various datasets.

- The proposed method could publish its hyperparameter in two ways depending on the degree of protection: (1) publishing all private support functions and (2) publishing private equilibrium points and let a user classify the test data with k-nearest-neighbors (kNN).

1.4 Organization of the Thesis

The thesis is composed of 8 chapters. In Chapter 2, we review literatures related to the problem. In Chapter 3, we mention preliminaries of the proposed method in terms of privacy. In Chapter 4, we propose our clustering algorithm under guaranteeing privacy. In Chapter 5, we develop a private classification idea with the clustering method. Then, we present some inference scenarios of the proposed method. The experiments and results are in Chapter 7. Finally, in Chapter 8, we give concluding remarks and possible future research directions of this thesis.

Chapter 2

Literature Review

We summarize some works on guaranteeing differential privacy in support vector classifiers. As both SVM and SVDD can be solved by both primal and dual problems, we state two major approaches which are maximizing margin in the primal problem and utilizing dual variables and kernels in the dual problem.

2.1 Differentially private Empirical risk minimization

A support vector classifier like SVM can be formalized as a convex quadratic programming. Because the solution of SVM converges to a global optimum, earlier works primarily solved SVM from the point of view of empirical risk minimization (ERM). For ERM, after Dwork [7] proposed the perturbation techniques, Chaudhuri et al. [6] developed the basic algorithms of adding noise to the outputs or to the objective function. [19] proposed an improved version of differentially private convex ERM algorithm, especially for high-dimensional learning. These studies have focused on approximating the primal objective using the Huber loss [6] while solving SVM. Algorithms for other optimization problems also have been published such as the functional mechanism [46] which perturbed the coefficients of linear regression and logistic regression.

Another approach to achieve differential privacy in ERM is perturbing the gradient. Bassily et al. [2] investigated the stochastic gradient descent (SGD) method to solve the convex optimization, and achieved a tighter bound of convex ERM. Various works such as developing smooth objectives [47], reducing complexity, and tightening bound of the loss function [44] are under studied. Iyengar et al. [16] improved Huber SVM through gradient based differential private convex optimization algorithms. Note that the gradient approach is similar to deep learning [1] [2].

2.2 Differentially private Support vector machine

Support vector machine have only been studied under differential privacy among support vector classifiers. To be specific, Rubinstein et al. [36] proposed a privacy framework for SVM with kernel functions and corresponding reproducing kernel Hilbert spaces (RKHS). The algorithm obtains optimal dual variables by solving the dual problem and adds noise to the primal weight vector. Note that a dual variable matches to a training data, all dual variables could be affected by modifying just one input, which makes hard to obtain sensitivity in the dual formula. Jain and Thakurta [17] proposed three different ways to add noise to their corresponding three different situations on user's query: providing only prediction, publishing the differentially private weight vector fitted to the user's dataset, and publishing the weight vector privately learned with sampling training data. We will further discuss about [17, 36] in Section 6.3. Zhang et al. [49] suggested dual variable perturbation, but it may not fit the definition of differential privacy since they publish support vectors without any perturbations. Hall et al. [13] proposed the idea of adding an appropriate Gaussian process to the function, especially in the RKHS. Currently,

various application of [36] such as [29], [39], [26] and adding noise to kernel output [45] have been also published.

Chapter 3

Preliminaries

3.1 Differential privacy

Differential privacy [9], [10] provides a formal approach to privacy of individuals in datasets. It guarantees that the output of a mechanism is robust to any change (deletion, modification) of one sample, thus protecting the individual privacy from an adversary. Under differential privacy, each individual's data (record) is guaranteed to be private and cannot be revealed.

Definition 3.1. *(Differential privacy) A randomized mechanism* $\mathcal{M}$ *is* $(\boldsymbol{\epsilon}, \boldsymbol{\delta})$ ***-differentially private*** *if, for two neighboring datasets* D *and* D' *in* $\mathcal{D}$ *(i.e., differ by one data sample) and for all* $\mathcal{S} \subseteq Range(\mathcal{M})$*, it holds that*

$$Pr[\mathcal{M}(D) \in \mathcal{S}] \leq e^{\epsilon} Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta \tag{3.1}$$

where $\epsilon, \delta \geq 0$ *are parameters to control the amount of the difference between* D *and* D'*.*

We call ϵ the privacy budget, where smaller ϵ could guarantee better privacy of mechanism $\mathcal{M}$. The other parameter δ, introduced in [8], means that mechanism $\mathcal{M}$ can be broken with probability δ. Especially, **ϵ-differential privacy** represents

(ϵ, δ)-differential privacy with $\delta = 0$.

Definition 3.2. *(Sensitivity) For a query $g : \mathcal{D} \to \mathbb{R}^k$, and neighboring datasets D and D' the ℓ_1-sensitivity of g is defined as*

$$\Delta g = max_{D \sim D'} ||g(D) - g(D')||_1. \tag{3.2}$$

where $||g(D) - g(D')||_1$ represent the 1-norm distance between $g(D)$ and $g(D')$.

The ℓ_1 sensitivity also called **Sensitivity**. The sensitivity of g measures how much the output of g could be changed by modifying one data in D. The sensitivity gives an upper bound on how much we should perturb its output to satisfy differential privacy.

Properties 1. *For any function $g : \mathcal{D} \to \mathbb{R}^k$ with sensitivity Δg, randomized mechanism $\mathcal{M}$,*

$$\mathcal{M}(D) = g(D) + \boldsymbol{\mu} \tag{3.3}$$

*provides ϵ-differential privacy where each $\boldsymbol{\mu}_i$ is i.i.d. random variable drawn from $Lap(\frac{\Delta f}{\epsilon})$. Note that $Lap(\lambda) \sim \frac{1}{2\lambda} exp(-\frac{|\mathbf{x}|}{\lambda})$ where λ denotes the scale of noise. The randomized mechanism $\mathcal{M}$ called **Laplace Mechanism**. Similar to Laplace mechanism, Gaussian mechanism implies that adding Gaussian noise, $\mathcal{M}(D) = g(D) + \mathcal{N}(0, \frac{2ln(1.25/\delta) \cdot (\Delta f)^2}{\epsilon^2})$ for (ϵ, δ)-differential privacy [10].*

Properties 2. *(Post-processing [10]) Let a mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ is ϵ-differential private and let $h : \mathcal{R} \to \mathcal{R}'$ be a arbitrary randomized mapping. Then $h \circ \mathcal{M} : \mathcal{D} \to \mathcal{R}'$ is ϵ-differential private.*

Properties 3. *(Parallel composition [28]) For disjoint dataset chunks* $D_1 \cup D_2 \cup ... \cup D_n = D$*, suppose the combination of random algorithms* $\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_n$ *and the corresponding privacy budget* $\epsilon_1, \epsilon_2, ..., \epsilon_n$*. The combination mechanism* $\mathcal{M}(\mathcal{M}_1(D_1), \mathcal{M}_2(D_2), ..., \mathcal{M}_n(D_n)$ *provides* $max(\epsilon_i)$*-differential privacy where* $i = 1, ..., n$*.*

Chapter 4

Differential private support vector data description

For a robust privacy-friendly multi-class classification, we first propose a new method of injecting noise to guarantee differential privacy of SVDD.

4.1 Support vector data description

SVDD is a clustering and outlier detection method using kernel supports where Tax and Duin [40] proves that SVDD is identical to one class SVM [27] under RBF kernel. The basic idea of SVDD under a kernel function is to map the input data into a high dimensional feature space and to find the smallest enclosing sphere of radius R that contains most of the mapped data points in the feature space. Using the kernel function, the sphere can obtain non-linear boundary when mapped back to the data space. More specifically, let set of data $\{\mathbf{x}_i, y_i\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ be a given data set of $\mathbf{x}_i \in \mathbb{R}^d$ and its label y_i. With a nonlinear feature mapping ϕ from $\mathbb{R}^d$ to F-dimensional feature space $\mathbb{R}^F$, SVDD finds the smallest enclosing sphere of R with a sphere center $\boldsymbol{a}$ in RKHS described by the following model:

$$\min_{R,\boldsymbol{a},\xi} R^2 + \frac{C}{n}\sum_i \xi_i \quad \text{s.t.} \left\|\phi\left(\mathbf{x}_i\right) - \boldsymbol{a}\right\|_F^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0, \forall i. \tag{4.1}$$

Again, slack variables $\xi_i \geq 0$ allow a soft boundary and can be denoted as a loss function $\ell(\boldsymbol{a}, \mathbf{x}) := [\|\phi(\mathbf{x}) - \boldsymbol{a}\|^2 - R^2]_+$. Hyperparameter C controls the trade-off between penalties ξ_i. Points which fall outside the sphere, i.e., $\|\phi(\mathbf{x}_i) - \boldsymbol{c}\|^2 > R^2$, are deemed anomalous. Using the dual variables β_j, the solution of primal problem (4.1) can be obtained by solving the following Wolfe dual problem:

$$\begin{aligned} \max \quad & W = \sum_j \beta_j K(\mathbf{x}_j, \mathbf{x}_j) - \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \\ & s.t. \;\; 0 \leq \beta_j \leq \frac{C}{n}, \sum_j \beta_j = 1, j = 1, ..., n. \end{aligned} \tag{4.2}$$

By solving a convex optimization problem, the trained kernel support function, which measure the distance from the sphere center, is then given by

$$f(\mathbf{x}) = ||\phi(\mathbf{x}) - \boldsymbol{a}||^2 \quad \text{(in primal)}, \tag{4.3}$$

$$= K(\mathbf{x}, \mathbf{x}) - 2 \sum_j \beta_j K(\mathbf{x}_j, \mathbf{x}) + \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \quad \text{(in dual)}. \tag{4.4}$$

where $K(\mathbf{x}_i, \mathbf{x}_j)$ replaces the inner product of $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ and $\mathbf{x}_j$ with $0 < \beta_j < C/n$ are called support vectors and lie on the boundary of the sphere. Here, since the primal solution is equal to the dual solution, the center of the sphere $\boldsymbol{a} = \sum_{i=1}^{n} \beta_i \phi(\mathbf{x}_i)$.

4.2 Differentially private support vector data description

Now, we propose an algorithm for guaranteeing differential privacy in support vector data description and prove the algorithm. To prove differential privacy, we borrow proof techniques from differentially private SVM algorithms (Lemma 9 of [36] and

Appendix section A of [17]). From now on, we only consider Radial basis function (RBF) $K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma||\mathbf{x}_1 - \mathbf{x}_2||^2)$ as kernel, because RBF kernel is bounded by $K(\mathbf{x}, \mathbf{x}) \leq \kappa^2 = 1$, which helps reduce the sensitivity. Since the dual formula has a limitation that the kernel trick needs to utilize training data as mentioned above, we emphasize the reason of utilizing a primal formula again. To dealing with reproducing kernel Hilbert space induced by RBF kernel, we define F-dimensional feature mapping ϕ as

$$\phi(\cdot) = \sqrt{\frac{2}{F}} \left[\cos\left(\langle \boldsymbol{\rho}_1, \cdot \rangle\right), \sin\left(\langle \boldsymbol{\rho}_1, \cdot \rangle\right), \ldots, \cos\left(\left\langle \boldsymbol{\rho}_{\frac{F}{2}}, \cdot \right\rangle\right), \sin\left(\left\langle \boldsymbol{\rho}_{\frac{F}{2}}, \cdot \right\rangle\right) \right]^T \tag{4.5}$$

where F should be an even number. Algorithm 1 describes our Private SVDD, which perturbs the center of the sphere $\boldsymbol{a}$. First, we need to draw $\boldsymbol{\rho}$ to find feature mapping ϕ. Then, solve problem (4.2) and obtain optimal dual variables $\boldsymbol{\beta}^*$. With equation (4.5) for ϕ, we can form the according primal solution. At the end, we can obtain differentially private center $\hat{\boldsymbol{a}}$ of the sphere in RKHS by adding i.i.d. samples from Laplace noise $\boldsymbol{\mu}$ with corresponding λ which depends on L_1-sensitivity. In summary, we perturb the center of the sphere $\boldsymbol{a}^*$ compared to SVM that injects noise to $\mathbf{w}^*$ [36]. Then, with Laplace noise μ with sensitivity λ, differentially private support function is defined as

$$\hat{f}(\mathbf{x}) = ||\phi(\mathbf{x}) - \hat{\boldsymbol{a}}||^2 \quad \text{where } \hat{\boldsymbol{a}} = \boldsymbol{a} + \boldsymbol{\mu}. \tag{4.6}$$

From now on, to make the notation more convenient, we will denote a optimal solution with star(*) and corresponding differentially private notation with hat(ˆ).

Now, we calculate the sensitivity of $\boldsymbol{a}$ and privacy guarantee when a training

Algorithm 1 Private SVDD

Require: Training data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$; Translation-invariant kernel $K(\mathbf{x}, \mathbf{y}) = G(\mathbf{x} - \mathbf{y})$ with Fourier transform $p(\boldsymbol{\omega}) = 2^{-1} \int e^{-j\langle \boldsymbol{\omega}, \mathbf{x} \rangle} g(\mathbf{x}) d\mathbf{x}$; Convex loss function ℓ; Parameters $\lambda, C > 0$ and the desired dimension number of feature mapping F.

1: $\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_{\frac{F}{2}} \leftarrow$ Draw i.i.d. sample of $\frac{F}{2}$ vectors in $\mathbb{R}^d$ from p;
2: $\boldsymbol{\beta}^* \leftarrow$ Solve Equation (4.2) on D with parameter C, kernel K induced by map (4.5), and loss ℓ;
3: $\boldsymbol{a}^* \leftarrow \sum_{i=1}^n \beta_i \phi(\mathbf{x}_i)$ where ϕ is defined in Equation (4.5);
4: $\boldsymbol{\mu} \leftarrow$ Draw i.i.d. sample of F scalars from $Lap(\lambda)$; and
5: Return $\hat{\boldsymbol{a}} = \boldsymbol{a}^* + \boldsymbol{\mu}$ and $\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_{\frac{F}{2}}$.

example is changed. Before that, we can say a optimal solution of (4.2) be a solution of

$$\max_{\beta} \left\{ \sum_i T(\beta_i) - \frac{1}{2} \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \right\} \tag{4.7}$$

as determined at [11] where T is concave function.

Lemma 4.1. *(Zhang [48]) Let $\boldsymbol{a_D}$ be the solution of (4.7) under dataset D and $\boldsymbol{a_{D^i}}$ be a solution of the same problem where ith point is removed. Then,*

$$\|\boldsymbol{a_D} - \boldsymbol{a_{D^i}}\|_{\mathcal{F}} \leq |\beta_i| \sqrt{K(x_i, x_i)}.$$

Corollary 4.2. *(Sensitivity) For every pair of neighboring datasets D, D' of n entries, let $\boldsymbol{a}_D$ and $\boldsymbol{a}_{D'}$ are the optimal solutions to (4.1) when the underlying datasets are D and D'. Then, we have $\|\boldsymbol{a}_D - \boldsymbol{a}_{D'}\|_1 \leq \frac{2C\kappa\sqrt{F}}{n}$.*

Proof W.l.o.g. we can assume that the datasets D and D' differ in the nth data point, i.e. $\mathbf{x}_n \in D$ and $\mathbf{x}'_n \in D'$. Then, by Lemma 4.1:

$$\|\boldsymbol{a_D} - \boldsymbol{a_{D^n}}\|_{\mathcal{F}} \leq |\beta_n| \sqrt{K(\mathbf{x}_n, \mathbf{x}_n)} \quad \leq \frac{C}{n}\kappa$$

$$\|\boldsymbol{a_{D'}} - \boldsymbol{a_{D'^n}}\|_{\mathcal{F}} \leq |\beta_n| \sqrt{K(\mathbf{x}_n, \mathbf{x}_n)} \quad \leq \frac{C}{n}\kappa.$$

Note that $D^n = D'^n$. Then, by triangle inequality, we have

$$\|\boldsymbol{a_D} - \boldsymbol{a_{D'}}\|_{\mathcal{F}} \leq \|\boldsymbol{a_D} - \boldsymbol{a_{D^n}}\|_{\mathcal{F}} + \|\boldsymbol{a_{D'}} - \boldsymbol{a_{D^n}}\|_{\mathcal{F}} \quad \leq \frac{2C\kappa}{n}$$

and $\|\boldsymbol{a_D} - \boldsymbol{a_{D'}}\|_1 \leq \sqrt{F}\|\boldsymbol{a_D} - \boldsymbol{a_{D'}}\|_2 \leq \sqrt{F}\|\boldsymbol{a_D} - \boldsymbol{a_{D'}}\|_{\mathcal{F}} \leq \frac{2C\kappa\sqrt{F}}{n}$.

Theorem 4.3. *(Privacy Guarantee) Algorithm 1 is ϵ-differentially private.*

Proof Let D, D' be a pair of neighboring datasets with n entries, and $\boldsymbol{a_D}$, $\boldsymbol{a_{D'}}$ be the optimal solutions to (4.1) when the underlying datasets are D and D'. Let $\hat{\boldsymbol{a}} \in \mathbb{R}^F$ be the response of Algorithm 1 and let $\boldsymbol{\mu_D}$, $\boldsymbol{\mu_{D'}}$ denote i.i.d $Lap(0, \lambda)$. Then the ratio of probabilities $\Pr(M(D) = \hat{\boldsymbol{a}})$ and $\Pr(M(D') = \hat{\boldsymbol{a}})$ can be bounded by

$$\frac{\Pr(\boldsymbol{\mu_D} = \hat{\boldsymbol{a}} - \boldsymbol{a}_D)}{\Pr(\boldsymbol{\mu_{D'}} = \hat{\boldsymbol{a}} - \boldsymbol{a}_{D'})} = \frac{\exp(-\|\hat{\boldsymbol{a}} - \boldsymbol{a}_D\|_1 / \lambda)}{\exp(-\|\hat{\boldsymbol{a}} - \boldsymbol{a}_{D'}\|_1 / \lambda)} \leq \exp\left(\frac{\|\boldsymbol{a}_D - \boldsymbol{a}_{D'}\|_1}{\lambda}\right)$$

The equality holds by rule of product and law of exponent. The inequality follows by triangle inequality. By Lemma 4.1, for $\lambda \geq 2C\kappa\sqrt{F}/(\epsilon n)$, the algorithm guarantees ϵ-differential privacy. Note that for RBF kernel, we have $\kappa = 1$. The optimal solution of SVM is bounded by κ since $\boldsymbol{a}_D = \sum_{i=1}^n \beta_i \phi(\mathbf{x}_i) \leq \sum_{i=1}^n \beta_i \kappa = \kappa$. So, we

clip the output of Algorithm 1 by κ and because clipping is post-processing, we can get $\hat{\boldsymbol{a}} \leftarrow \mathit{clip}(\hat{\boldsymbol{a}})$ without without loss of privacy. Next, we show the utility of our method.

Theorem 4.4. *(Error bound) For any $\nu > 0$ and $\tau \in (0,1)$, Algorithm 1 run on D with loss ℓ, kernel K, noise parameter $0 < \lambda \leq \frac{\nu}{8\kappa\left(F+\ln\frac{1}{\tau}\right)}$, and regularization parameter C, is (ν,τ)-useful with respect to the SVM under the $\|\cdot\|_{\infty;\mathcal{X}\text{-norm.}}$ In other words, run with arbitrary noise parameter $\lambda > 0$, Algorithm 1 is (ν,τ)-useful for $\nu = \Omega\left(\lambda\kappa\left(F + \ln\frac{1}{\tau}\right)\right)$.*

Proof Consider the SVDD and Algorithm 1 on any arbitrary point $\mathbf{x} \in \mathcal{X}$ and i.i.d Lap$(0,\lambda)$ noise $\boldsymbol{\mu}$,

$$\begin{aligned}
|f_{\hat{\boldsymbol{a}}}(\mathbf{x}) - f_{\boldsymbol{a}_D}(\mathbf{x})| &= |\|\phi(\mathbf{x}) - \boldsymbol{a}_D\|^2 - \|\phi(\mathbf{x}) - \hat{\boldsymbol{a}}\|^2| \\
&= |\phi(\mathbf{x})^T(\hat{\boldsymbol{a}} - \boldsymbol{a}_D) + (\hat{\boldsymbol{a}} - \boldsymbol{a}_D)^T\phi(\mathbf{x}) + \boldsymbol{a}_D^T\boldsymbol{a}_D - \hat{\boldsymbol{a}}^T\hat{\boldsymbol{a}}| \\
&= |\phi(\mathbf{x})^T\boldsymbol{\mu} + \boldsymbol{\mu}^T\phi(\mathbf{x}) + \frac{1}{2}\{(\boldsymbol{a}_D + \hat{\boldsymbol{a}})^T\boldsymbol{\mu} + \boldsymbol{\mu}^T(\boldsymbol{a}_D + \hat{\boldsymbol{a}})\}| \\
&\leq 2\|\boldsymbol{\mu}\|_1\|\phi(\mathbf{x})\|_\infty + \|\boldsymbol{\mu}\|_1\|\boldsymbol{a}_D + \hat{\boldsymbol{a}}\|_\infty \\
&\leq 2\kappa\|\boldsymbol{\mu}\|_1 + 2\kappa\|\boldsymbol{\mu}\|_1 = 4\kappa\|\boldsymbol{\mu}\|_1.
\end{aligned}$$

The last inequality follows from $\boldsymbol{a}_D = \sum_{i=1}^n \beta_i\phi(\mathbf{x}_i) \leq \sum_{i=1}^n \beta_i\kappa = \kappa$, and without loss of privacy, we can assume $\hat{\boldsymbol{a}} \leq \kappa$.

The absolute value of a zero-mean Laplace random variable with scale λ is exponentially distributed with scale λ^{-1}. Moreover the sum of q i.i.d. exponential random variables has Erlang q-distribution with the same scale parameter. Thus we have,

for Erlang F-distributed random variable X and any $t > 0$

$$\forall \mathbf{x} \in \mathcal{X}, |f_{\hat{\boldsymbol{a}}}(\mathbf{x}) - f_{\boldsymbol{a}_D}(\mathbf{x})| \leq 4\kappa X$$

$$\Rightarrow \forall \nu > 0, \Pr\left(\|f_{\hat{\boldsymbol{a}}} - f_{\boldsymbol{a}_D}\|_{\infty;\mathcal{X}} > \nu\right) \leq \Pr(X > \nu/4\kappa) \leq \frac{\mathbb{E}\left[e^{tX}\right]}{e^{\nu t/4\kappa}}. \tag{4.8}$$

Here we have employed the Chernoff tail bound technique using Markov's inequality. The numerator of equation (4.8), the moment generating function of the Erlang F-distribution with parameter λ, is $(1 - \lambda t)^{-F}$ for $t < \lambda^{-1}$. With the choice of $t = (2\lambda)^{-1}$, this gives

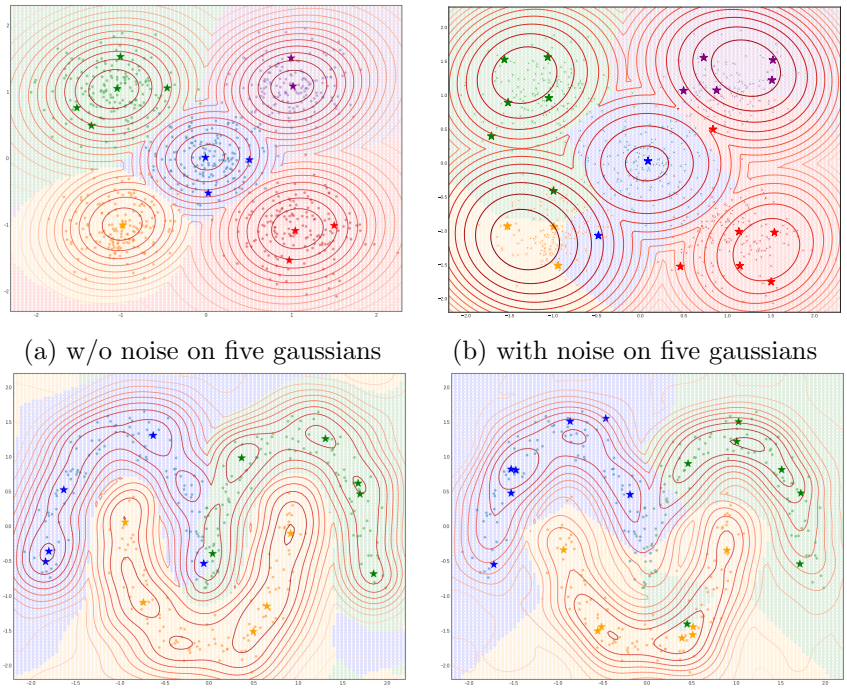
$$\begin{aligned}
\Pr\left(\|f_{\hat{\boldsymbol{a}}} - f_{\boldsymbol{a}_D}\|_{\infty;\mathcal{X}} > \nu\right) &\leq (1 - \lambda t)^{-F} e^{-\nu t/4\kappa} \\
&= 2^F e^{-\nu/(8\lambda\kappa)} \\
&= \exp\left(F \ln 2 - \frac{\nu}{8\lambda\kappa}\right) \\
&< \exp\left(F - \frac{\nu}{8\lambda\kappa}\right)
\end{aligned}$$

and provided that $\nu \geq \left(8\lambda\kappa\left(F + \ln\frac{1}{\tau}\right)\right)$ this probability is bounded by τ.

Chapter 5

Differentially private multi-class classification utilizing SVDD

We now present a new DP-friendly multi-class classification method utilizing differentially private SVDD. The basic idea of our approach is to classify data using interior points of the data space instead of the decision boundary of SVM. The proposed method consists of three phases: (I) constructing differentially private support level functions via SVDD, (II) classifying the data space privately via a dynamic system, and (III) two inference methods using equilibrium points. We illustrate the proposed method in Figure 5.1. Under differing colors based on samples' classes, train points are marked with circles. Equilibrium points (EPs), marked with stars, lie in the local minima of train data where the red lines indicate the contour of the support level function. The colored region means the class of each point in the grid of data space. Models with noise, even the EPs and decision boundaries become distorted, show similar decision regions to non-private models. To avoid confusion, we describe the details of the proposed method and check whether differential privacy is satisfied in the following subsections.



(a) w/o noise on five gaussians (b) with noise on five gaussians

(c) w/o noise on three moons (d) with noise on three moons

Figure 5.1: Illustration of classification regions of non-private models (left) and models with perturbation (right) of the proposed method on five gaussians (top) and three moons (bottom).

5.1 Phase I. Constructing a private support level function

A support level function [29] is defined as positive scalar function $f : \mathbb{X}^d \to \mathbb{X}^+$ where its level set is given by $L_f(r) = \{\mathbf{x} \in \mathbb{R}^d : f(\mathbf{x}) \leq r\}$ for some $r > 0$. As SVDD is designed to find a domain enclosing most of data points called as a support domain, $L_f(r)$ can estimates a kernel density function. Therefore, in this paper, we construct the level function with SVDD where its support level function f is defined by a square distance from the sphere center in RKHS. We note that other kernel density functions are also possible for the level function [18, 20, 22]. As we calculate a differentially private sphere center $\hat{\boldsymbol{a}}$ and corresponding support function

$\hat{f}$ in equation 4.6, we now build a differnetially private support level function $\hat{L}_f$. Using the level function, we can decompose the level function into several separate connected components C_i, where $i = 1, .., m$, i.e.,

$$\hat{L}_f(r) = \{\mathbf{x} \in \mathbb{R}^d : \hat{f}(\mathbf{x} \leq r)\} = C_1 \cup \cdots \cup C_m. \tag{5.1}$$

Here, constructing a support level function with a differentially private support function $\hat{f}$ satisfies post-processing property of Properties 2 in terms of differential privacy.

5.2 Phase II: Differentially private clustering on the data space via a dynamical system

The objective of phase II is to classify the data space via a dynamic system and find interior points called equilibrium points (EPs). To decompose a whole data space into separated cluster regions, we utilize the topological and dynamical properties of the dynamic system. The dynamic system helps us find the private interior points of the differentially private support level function $\hat{f}$ by solving the following dynamic system with $\hat{f}$, i.e.,

$$\frac{d\mathbf{x}}{dt} = -\nabla \hat{f}(\mathbf{x}). \tag{5.2}$$

Consequently, we can find a private state vector $\hat{\boldsymbol{s}}$, the local minimum point of the private support function $\hat{f}$, is called (asymptotically) *equilibrium points (EPs)*. For an equilibrium point, we define its *basin cell* $\overline{A(\hat{\boldsymbol{s}})}$ by as the closure of the set of all

the points converging to $\hat{\boldsymbol{s}}$ following the dynamic system (5.2), i.e.,

$$\overline{A(\hat{\boldsymbol{s}})} := cl\{\mathbf{x}(0) \in \mathbb{R}^d : \lim_{t\to\infty} \mathbf{x}(t) = \hat{\boldsymbol{s}}\} \tag{5.3}$$

from the initial point $\mathbf{x}(0)$ to $\mathbf{x}(t)$ in time t. Here, we apply gradient-descent methods to solve the system (5.2) [23]. Under the mild condition, it is well known that the whole data space can be partitioned into separate basin cells [24], [25] as

$$\mathbb{R}^d = \bigcup_{i=1}^{M} \overline{A(\hat{\boldsymbol{s}}_i)} \tag{5.4}$$

where the set of EPs is $\{\hat{\boldsymbol{s}}_i : i = 1, ..., M\}$. We could guarantee ϵ-differential privacy of Phase II with the parallel composition of Properties 3 because every data point converges to an EP and it cannot be assigned to multiple EPs.

5.3 Phase III: Classifying the decomposed regions under differential privacy

So far, we have not used any information of labels y while focusing on the clustering method into separate basin cells $\overline{A(\hat{\boldsymbol{s}})}$. In Phase III, we classify each decomposed region under differential privacy. Since all data points converge to their respective equilibrium points $\{\hat{\boldsymbol{s}}_i\}_{i=1}^{M}$ along the dynamics in equation 5.2, we can label each basin cell $\overline{A(\hat{\boldsymbol{s}}_i)}$ (hence, equilibrium point $\hat{\boldsymbol{s}}_i$). Specifically, as each basin cell $\overline{A(\hat{\boldsymbol{s}}_i)}$ contains at least one labeled data point, we can make a majority vote on the labeled data points and assign a class label to the most frequent class to the basin. The simple idea is to partition data and to aggregate similar to the majority voting on k-nearest neighbor (kNN) [35, 50] and the teacher ensemble in knowledge transfer

[14, 32]. We consider the label of each data point as a prediction and aggregate all predictions by choosing the label with the largest count in the basin cell. For each class $k = \{1, ..., c\}$ and the number of each class $\{n_1, ..., n_c\}$, we add random noise to the vote counts n_k of the basin cell to introduce ambiguity:

$$\hat{y}_{\boldsymbol{s}} = \underset{k=1,...,c}{\arg\max} \{n_k + Lap(\frac{1}{\epsilon})\} \tag{5.5}$$

where we add $Lap(\frac{1}{\epsilon})$ to each count as the sensitivity of the count is always 1 [9].

Almost real-world problems usually have high dimensional data, it's hard to gather a sufficient number of train samples to make a majority voting in a basin cell. To resolve this issue and increase the number of votes, we cluster the decomposed regions (i.e., basin cells) into hypercubes according to the coordinates of equilibrium points $\boldsymbol{s}$ in the high dimensional data space.

$$S_l := \{\boldsymbol{s}_j : ||\boldsymbol{s}_0 - \boldsymbol{s}_j||_\infty \leq r_{\max}\} \tag{5.6}$$

where $\boldsymbol{s}_0$ is the centroid of a hypercube S_l and $r_{\max}$ is the maximal threshold of distance between two points. To label S_l, we follow equation (5.5) where n_j measures the number of points converging to the equilibrium points in the hypercube for each possible label.

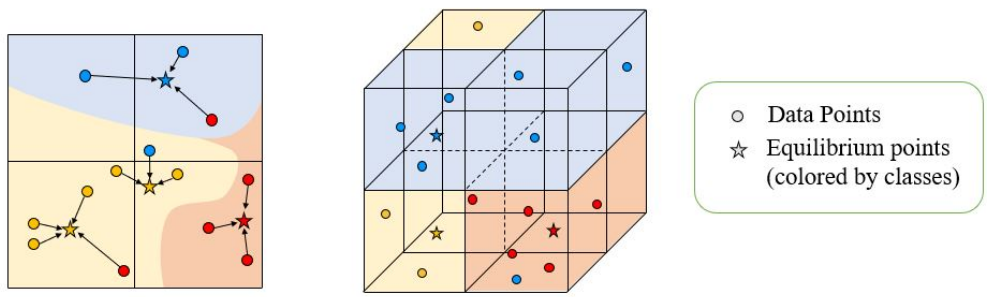


Figure 5.2: 2-dimensional classification using equilibrium points (left) and multi-dimensional classification using hypercubes (right) of the proposed method.

Chapter 6

Inference scenarios and releasing the differentially private model

For inference scenarios, we define three participants in the model: a data owner, a data publisher, and a data user. A data owner provides data for a training set and a data publisher builds differentially private (trusted) models using the training set. A user wants to make a prediction of his or her own test data using the model parameters of publishers. For example, consider a scenario as follows.

Scenario: A medical technology company has built a model to classify a person's disease according to a health medical state of data publishers. As the algorithm highly depends on each train data, it possesses a high risk to publish all the parameters for the algorithms. So, the company, which is the publisher of this situation, should keep the model parameters in a differentially private way.

Then, we present two ways for releasing the differentially private model: (1) a data publisher allows a data user to make use of the private support function and to classify new inputs with dynamic system and (2) a data publisher releases private EPs and their labels in the data space, which let a user classify new inputs by k-nearest neighbor (kNN).

6.1 Publishing support function

To enable a user to make his or her own predictions using the proposed method, a publisher needs to provide the private center of sphere $\hat{\boldsymbol{a}}$, feature mapping $\phi(\cdot)$, trained EPs (or hypercubes) and corresponding labels. Using the private center $\hat{\boldsymbol{a}}$ and feature mapping $\phi(\cdot)$, the user can calculate a support level function $\hat{f}(\mathbf{x})$ in equation (4.6). Then, the user estimates the equilibrium points of test data by equation (5.2) and classifies according to the label of the converged equilibrium point. If none of the EPs in training set exactly matches to the converged EP of test data, then the user may choose the nearest training equilibrium point and use its label.

6.2 Releasing equilibrium points

Even though we guarantee differential privacy of the published model, allowing an attacker to access the support function is still risky in terms of security and privacy. In other words, as the algorithm highly depends on each train data, publishing all the parameters for the algorithms may lead to a high risk of privacy leakage. Our solution is only publishing private EPs while preserving support functions. Because EPs exist in the input data space, classifying the test data by k-nearest neighbor ensures moderate accuracy for new data points. Note that EPs are not exactly matched with test data and keep the privacy of training set since EPs stand for private local minimum points of a private support level function.

6.3 Comparison to previous methods

We state details of previous differentially private SVM methods [17, 36] where $\boldsymbol{\mu}$ stands for Laplace noise of the corresponding sensitivity on each method. Rubinstein et al. [36] proposed the algorithm obtains optimal dual variables $\boldsymbol{\beta}^*$ by solving the dual problem and adds noise to the primal weight vector $\hat{\mathbf{w}} = \mathbf{w}^* + \boldsymbol{\mu} = \sum_{i=1}^{n} y_i \beta_i^* \phi(\mathbf{x}_i) + \boldsymbol{\mu}$ with feature map ϕ. Here, we denote the method perturbing primal weight vector with approximating RBF kernel as **SVM WEIGHT**.

Jain and Thakurta [17] proposed a method with adding noise on output of support function $\langle \mathbf{w}^*, \phi(\mathbf{z}) \rangle + \mu = \sum_{i=1}^{n} \beta_i^* K(\mathbf{z}, \mathbf{x}_i) + \mu$ where each $\mathbf{x}_i$ is in the training set. Here, the algorithm should use train data for inference, thus cannot publish parameters alone. We denote the method of perturbing output as **SVM OUTPUT**. In semi-interaction scenario when a user agrees to provide parts of his or her test set $\mathcal{Z}$ to a data publisher, the data publisher can publish a primal weight vector $\hat{\mathbf{w}}$ is $\mathbf{w}$ which minimizes the difference in perturbed outputs of test set $\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (\langle \mathbf{w} - \mathbf{w}^*, \phi(\mathbf{z}_i) \rangle - \mu)^2$. We denote the semi-supervised method [17] as **SVM SEMI**.

Chapter 7

Experiments

7.1 Models and Scenario setting

To evaluate the accuracy of models under differential privacy, it's important to set the scenario the same. For the experiments, we constraint the scenario to above mentioned one in Section 6, in which a publisher should release the parameters of a model and users can utilize them. **SVM WEIGHT** fits well to the scenario by publishing differentially private $\hat{\mathbf{w}}$ and **SVM SEMI**, which receives test sets from users, could provide appropriate differentially private $\hat{\mathbf{w}}$. To compare proposed method with the best accuracy of **SVM SEMI**, we gave the whole test data set and select δ as large as possible under differential privacy, which is $\frac{1}{n}$.

To set up a desired level of accuracy under differential privacy, we select **SVM OUTPUT**. Because of utilizing test data $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ in $\sum_{i=1}^{n} \beta_i{}^* K(\mathbf{x}, \mathbf{x}_i)$, **SVM**

Table 7.1: Benchmark data description and experimental settings

Data	data set description				SVM WEIGHT	SVM SEMI	Proposed	SVM OUTPUT
sets	dims	classes	train	test	($\frac{C}{n}$ / γ)	($\frac{C}{n}$ / γ)	($\frac{C}{n}$ / γ / lr / iters / r_{max})	($\frac{C}{n}$ / γ)
three moons	2	3	320	80	0.05 / 5	0.05 / 5	0.05 / 5 / 0.001 / 20 / -	0.05 / 5
five gaussians	2	5	800	200	0.1 / 1	0.1 / 1	0.1 / 5 / 0.001 / 20 / -	0.1 / 1
iris	3	4	80	20	0.05 / 1	0.05 / 1	0.05 / 1 / 0.001 / 10 / 0.5	0.05 / 1
wine	3	13	142	36	0.05 / 0.1	0.05 / 0.1	0.05 / 0.1 / 0.1 / 100 / 1	0.05 / 0.1
vowel	10	11	422	106	1 / 1	0.05 / 1	0.05 / 1 / 0.1 / 30 / 0.5	0.05 / 1
satimage	36	6	3548	887	0.05 / 0.1	0.05 / 1	0.05 / 0.1 / 0.001 / 20 / 1	0.05 / 1
segment	19	7	1232	308	0.5 / 1	0.5 / 1	0.5 / 1 / 0.001 / 20 / 0.5	0.5 / 1
shuttle	9	3	24360	6090	0.01 / 0.1	0.01 / 0.1	0.01 / 1 / 0.001 / 20 / 0.5	0.01 / 0.1

OUTPUT cannot publish the hyperparameters of the model, which needs a strong assumption that data users also trust the data publisher and all data should be calculated by the publisher. Under strong assumption, **SVM OUTPUT** shows higher accuracy than **SVM WEIGHT** and **SVM SEMI**. We exclude other models as follows: approximating primal solution with Huber loss in [6] showed lower performance than **SVM SEMI** mentioned in [17], [49] possesses the privacy issue of publishing support vectors without adding noise and [29], [39], [26] showed similar results with **SVM WEIGHT**.

7.2 Datasets

In order to evaluate the accuracy under differential privacy, we conducted experiments on 8 data sets and compared the performance on various privacy budgets ϵ. In this paper, we focused on small datasets where differential privacy performs poorly. Description of data set is given in Table 7.1. "three moons" and "five gaussians" are artificially generated two dimensional datasets for visualization as shown in Figure 5.1. "iris", "wine", "satimage", "segment" and "shuttle" are widely used multi-class classification data sets form University of California at Irvine (UCI) repository [4]. We divided "shuttle" data into three classes to alleviate the class data imbalance.

7.3 Experimental settings

In experiments, for the Fourier transformation in the proposed method and **SVM WEIGHT**, we use 400 dimensional feature map according to equation (4.5). Detailed hyperparameters such as C and γ are in Table 7.1. We try to match C, which affects the sensitivity, between methods. Additionally, for the proposed method, we

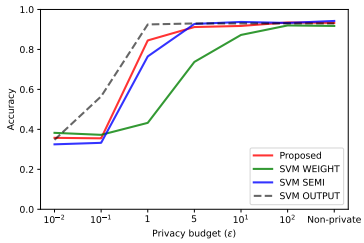
Table 7.2: Training and inference time (sec) for each dataset

Model — Dataset	three moons	five gaussians	iris	wine	vowel	satimage	segment	shuttle (3)
SVM WEIGHT	4.72	100.65	0.60	4.66	71.17	1376.24	298.70	49943.29
SVM SEMI	5.01	64.35	0.51	1.63	31.39	1158.10	140.61	52519.56
Proposed	**0.90**	**2.41**	**0.18**	**1.32**	**1.96**	**31.03**	**4.54**	**1334.80**

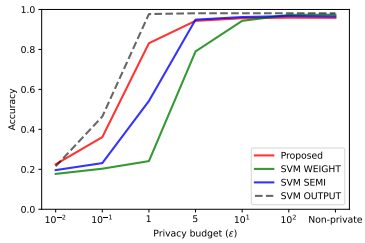
also make note of initial learning rates (lr), the number of iterations (iters) of a dynamic system and r_{max} of a hypercube in Table (4.5). We decayed the learning rate by a factor of 1/5 after 80% of iterations. We implement a random 80/20 train/test split and performance is evaluated over 5 runs. We use one-against-all approaches for the multi-class classification problem of SVM. All the experiments are run on an Intel(R) Xeon(R) Gold 5218R CPU @ 2.10GHz Processor using Python 3.7.4 along with Sklearn [33], Scipy [43], and Cvxopt [41] packages.

7.4 Empirical results on various datasets under publishing support function

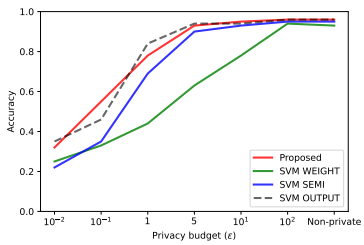
Figure 7.1 summarizes the evaluation results under different ϵ for each dataset. Each non-private model in the graph measures the baseline accuracy without adding noise. We compared the proposed method with publishing support function under privacy budgets $\epsilon = \{0.01, 0.1, 1, 5, 10, 100\}$ and Non-private model with **SVM WEIGHT**, **SVM SEMI** (and **SVM OUTPUT** as a desired level of accuracy). In general, the proposed method shows higher accuracy than **SVM WEIGHT** and **SVM SEMI** in all datasets and shows almost close performance to **SVM OUTPUT**. In Vowel (Figure 7.1e) and Segment (Figure 7.1g) datasets, the proposed method shows even better performance.



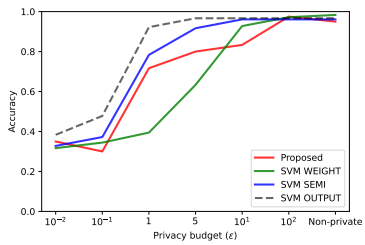
(a) Three moons



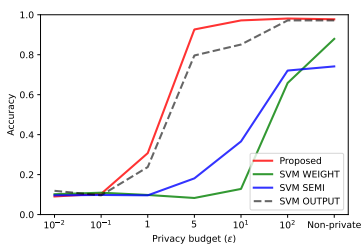
(b) Five Gaussians



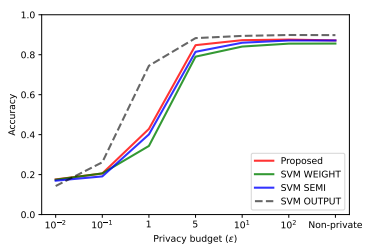
(c) Iris



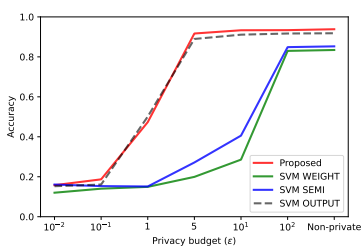
(d) Wine



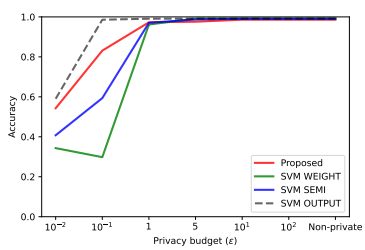
(e) Vowel



(f) Satimage



(g) Segment

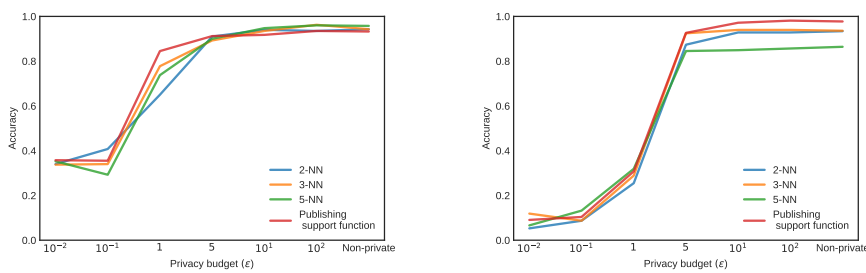


(h) Shuttle

Figure 7.1: Test accuracy of **SVM WEIGHT**, **SVM SEMI** and the proposed method (publishing support function) on various datasets.

Furthermore, the results show that preserving privacy becomes harder under the same privacy budget ϵ as the number of data decreases. For example, since shuttle dataset contains 24360 training samples, **SVM WEIGHT** and **SVM SEMI** also show similar accuracy until $\epsilon = 1$ with non-private models. However, we can see an apparent accuracy drop in the small datasets. It represents the hardness of protecting the privacy of one individual data. The proposed method shows consistent performance in a wide range of privacy budgets ϵ since the decisions with interior points enhance the robustness against noise and modifications than the decision boundary of SVM.

In terms of time complexity, the proposed method shows faster computation time than other methods. Both SVM and SVDD methods have a QP procedure and QP solvers take $O(n^3)$ [34]. One-against-all SVM algorithm should calculate all samples in each class, so the complexity is $O(c \cdot n^3)$. However, the proposed algorithm only takes $O(n^3)$ for solving QP. For the dynamic system, it only takes at $O(n^2 \cdot \text{iters})$ for gradient descent. Table 7.2 shows it takes lower time for classification.



(a) Three moons (b) Vowel

Figure 7.2: Test accuracy of the proposed method with releasing equilibrium points varying the number of neighbors in kNN.

Table 7.3: Test accuracy of each model on shuttle by sub-sampling 10% (left) and 30% (right) of whole dataset.

	sampling ratio = 0.1					sampling ratio = 0.3				
ϵ	0.01	0.1	1	5	Non-private	0.01	0.1	1	5	Non-private
SVM WEIGHT	0.366	0.310	0.892	0.917	0.912	0.135	0.456	0.957	0.955	0.948
SVM SEMI	0.332	0.484	0.899	0.932	0.930	0.330	0.514	0.935	0.950	0.947
Proposed	0.456	0.644	0.908	0.983	0.987	0.635	0.786	0.958	0.981	0.986
SVM OUTPUT	0.493	0.919	0.926	0.926	0.926	0.559	0.943	0.943	0.946	0.946

7.5 Evaluating robustness under diverse data size

In this subsection, we evaluate the empirical robustness between data set size by sub-sampling shuttle dataset. We sampled 10% and 30% of the dataset and implemented a random 80/20 train/test split on the subsets and settings are the same as Table 7.1. As shown in Table 7.3, even if the non-private accuracy of each model is similar, the proposed method shows high accuracy under a small privacy budget ϵ.

7.6 Inference through equilibrium points

Here, we compare releasing equilibrium points with publishing support vectors with three moons and vowel datasets. All the hyperparameters and experimental settings are the same as Table 7.1. We compared the performance on varying the number of neighbors in kNN, as shown in Figure 7.2. For vowel, an easy dataset, 1-NN and 3-NN show compatible results with the publishing support function. It shows similar results to the method of publishing support function for three moons and vowel dataset. The results show that publishing private EPs might be a good choice for some differentially private situations when we want to secure our support function.

Chapter 8

Conclusion

8.1 Conclusion

In this paper, we have investigated a multi-class classification method under differential privacy. By perturbing the sphere center in the kernel space, we can guarantee the differential privacy of SVDD. Accordingly, we propose a new three-phase classification method based on equilibrium points. Empirical results demonstrate the proposed method is more robust, fast, and useful compared to other existing differentially private SVM methods.

8.2 Future Work

The privacy issue becomes more important in real-world problems. In this paper, we run the experiments on the UCI dataset only, but the method can be applied in many datasets. Through unifying various differential privacy methodologies, the ultimate goal is to create a unified protocol and framework that can be applied to the data-driven machine learning industry. We hope this research could help improve practical solutions to real-world problems under guaranteeing the privacy of training data.

Bibliography

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

[3] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik. Support vector clustering. *Journal of machine learning research*, 2(Dec):125–137, 2001.

[4] C. Blake. Uci repository of machine learning databases. *http://www. ics. uci. edu/˜ mlearn/MLRepository. html*, 1998.

[5] C. J. Burges et al. Simplified support vector decision rules. In *ICML*, volume 96, pages 71–77. Citeseer, 1996.

[6] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

[7] C. Dwork. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*, pages 1–12. Springer, 2006.

[8] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[9] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[10] C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[11] A. Elisseeff, M. Pontil, et al. Leave-one-out error and stability of learning algorithms with applications. *NATO science series sub series iii computer and systems sciences*, 190:111–130, 2003.

[12] 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*, pages 1054–1067, 2014.

[13] R. Hall, A. Rinaldo, and L. Wasserman. Differential privacy for functions and functional data. *The Journal of Machine Learning Research*, 14(1):703–727, 2013.

[14] J. Hamm, Y. Cao, and M. Belkin. Learning privately from multiparty data. In *International Conference on Machine Learning*, pages 555–563. PMLR, 2016.

[15] S. Han, U. Topcu, and G. J. Pappas. Differentially private distributed con-

strained optimization. *IEEE Transactions on Automatic Control*, 62(1):50–64, 2016.

[16] R. Iyengar, J. P. Near, D. Song, O. Thakkar, A. Thakurta, and L. Wang. Towards practical differentially private convex optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 299–316. IEEE, 2019.

[17] P. Jain and A. Thakurta. Differentially private learning with kernels. In *International conference on machine learning*, pages 118–126. PMLR, 2013.

[18] K.-H. Jung, N. Kim, and J. Lee. Dynamic pattern denoising method using multi-basin system with kernels. *Pattern Recognition*, 44(8):1698–1707, 2011.

[19] D. Kifer, A. Smith, and A. Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pages 25–1. JMLR Workshop and Conference Proceedings, 2012.

[20] K. Kim, Y. Son, and J. Lee. Voronoi cell-based clustering using a kernel support. *IEEE Transactions on Knowledge and Data Engineering*, 27(4):1146–1156, 2014.

[21] D. Lee and J. Lee. Domain described support vector classifier for multi-classification problems. *Pattern Recognition*, 40(1):41–51, 2007.

[22] D. Lee and J. Lee. Equilibrium-based support vector machine for semisupervised classification. *IEEE Transactions on Neural Networks*, 18(2):578–583, 2007.

[23] D. Lee, K.-H. Jung, and J. Lee. Constructing sparse kernel machines using attractors. *IEEE transactions on neural networks*, 20(4):721–729, 2009.

[24] J. Lee and D. Lee. An improved cluster labeling method for support vector clustering. *IEEE Transactions on pattern analysis and machine intelligence*, 27 (3):461–464, 2005.

[25] J. Lee and D. Lee. Dynamic characterization of cluster structures for robust and inductive support vector clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1869–1874, 2006.

[26] H. Li, L. Xiong, L. Ohno-Machado, and X. Jiang. Privacy preserving rbf kernel support vector machine. *BioMed research international*, 2014, 2014.

[27] L. M. Manevitz and M. Yousef. One-class svms for document classification. *Journal of machine Learning research*, 2(Dec):139–154, 2001.

[28] F. D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30, 2009.

[29] R. Mochaourab, S. Sinha, S. Greenstein, and P. Papapetrou. Robust explanations for private support vector machines. *arXiv preprint arXiv:2102.03785*, 2021.

[30] A. Narayanan and V. Shmatikov. How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*, 2006.

[31] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125. IEEE, 2008.

[32] N. Papernot, M. Abadi, Úlfar Erlingsson, I. Goodfellow, and K. Talwar. Semi-supervised knowledge transfer for deep learning from private training data, 2017.

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[34] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

[35] J. Rauch, I. E. Olatunji, and M. Khosla. Achieving differential privacy for k-nearest neighbors based outlier detection by data partitioning, 2021.

[36] 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*, 4(1):65–100, 2012.

[37] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, J. C. Platt, et al. Support vector method for novelty detection. In *NIPS*, volume 12, pages 582–588. Citeseer, 1999.

[38] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[39] M. Senekane. Differentially private image classification using support vector machine and differential privacy. *Machine Learning and Knowledge Extraction*, 1(1):483–491, 2019.

[40] D. M. Tax and R. P. Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

[41] L. Vandenberghe. The cvxopt linear and quadratic cone program solvers. *Online: http://cvxopt. org/documentation/coneprog. pdf*, 2010.

[42] V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

[43] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[44] D. Wang, M. Ye, and J. Xu. Differentially private empirical risk minimization revisited: Faster and more general. *Advances in Neural Information Processing Systems*, 30, 2017.

[45] H. Wang and S. Li. Differential private multiple classification algorithm for

svm. In *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pages 604–609. IEEE, 2018.

[46] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett. Functional mechanism: regression analysis under differential privacy. *Proceedings of the VLDB Endowment*, 5(11):1364–1375, 2012.

[47] J. Zhang, K. Zheng, W. Mou, and L. Wang. Efficient private erm for smooth objectives. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3922–3928, 2017.

[48] T. Zhang. A leave-one-out cross validation bound for kernel methods with applications in learning. In *International Conference on Computational Learning Theory*, pages 427–443. Springer, 2001.

[49] Y. Zhang, Z. Hao, and S. Wang. A differential privacy support vector machine classifier based on dual variable perturbation. *IEEE Access*, 7:98238–98251, 2019.

[50] Y. Zhu, X. Yu, M. Chandraker, and Y.-X. Wang. Private-knn: Practical differential privacy for computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11854–11862, 2020.

국문초록

본 논문에서는 커널 서포트와 평형점을 활용한 차분 프라이버시 다중 클래스 분류 기법을 제시한다. 서포트 벡터 분류 기법은 데이터 분석과 머신 러닝에 활용성이 높아 사용자의 데이터를 보호하며 학습하는 것이 필수적이다. 그 중 가장 대중적인 서포트 벡터 머신(SVM)은 서포트 벡터라고 불리는 일부 데이터에만 분류에 의존하기 때문에 프라이버시 차분 기법을 활용하기 어렵다. 데이터 하나가 변경되었을 때 결과의 변화가 적어야 하는 차분 프라이버시 상황에서 서포트 벡터 하나가 없어진다면 분류기의 결정 경계는 그 변화에 매우 취약하다는 문제가 있다. 이 문제를 해결하기 위해 본 연구에서는 평형점이라고 불리는 군집 내부에 존재하는 점을 활용하는 차분 프라이버시 다중 클래스 분류 기법을 제시한다. 이를 위해, 먼저 커널 공간에서 구의 중심에 섭동을 더해 차분 프라이버시를 만족하는 서포트 벡터 데이터 디스크립션(SVDD)을 구하고 이를 레벨집합으로 활용해 동역학계로 극소점들을 구한다. 평형점을 활용하거나 고차원 데이터의 경우 초입방체를 만들어, 학습한 모델을 추론에 활용할 수 있는 (1) 서포트 함수를 공개 하는 방법과 (2) 평형점을 공개하는 방법을 제시한다. 8개의 다양한 데이터 집합의 실험적인 결과는 제시한 방법론이 노이즈에 강건한 내부의 점을 활용해 기존의 차분 프라이버시 서포트 벡터 머신보다 성능을 높이고, 차분 프라이버시가 적용되기 어려운 작은 데이터셋에도 활용될 수 있다는 기술임을 보여준다.

주요어: 차분 프라이버시, 머신 러닝, 서포트 벡터 분류, 산업공학

학번: 2020-26472

감사의 글

서울대학교 산업공학과의 모든 식구들께 감사드립니다.