Statistica Sinica Preprint No: SS-2025-0083	
Title	Estimation and Inference for Density-convoluted Support Vector Machine with Streaming Data
Manuscript ID	SS-2025-0083
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202025.0083
Complete List of Authors	Haochen Rao, Xu Guo, Heng Lian and Haobo Qi
Corresponding Authors	Haobo Qi
E-mails	haobo4869@bnu.edu.cn

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

Estimation and Inference for Density-convoluted Support Vector Machine with Streaming Data

Haochen Rao[1], Xu Guo*[1], Heng Lian[2] and Haobo Qi*[1]

[1]*Beijing Normal University and* [2]*City University of Hong Kong*

Abstract: This paper investigates estimation and inference problems for the support vector machine coefficients with high-dimensional streaming data. To handle the non-smooth hinge loss, the density convolution technique is adopted. We first propose an online lasso estimator that can be obtained by optimizing a surrogate loss function. Instead of using complete historical data, the surrogate loss function uses a renewable quadratic form to approximate historical information. As a result, the estimation procedure only requires newly arrived data and limited historical information, which can be updated in an online manner. We derive the theoretical error bounds of the proposed online lasso estimator under mild conditions. To eliminate the inherent bias of the lasso estimator, we further propose an online debiased lasso estimator and construct a valid inference procedure. We establish the asymptotic normality of the debiased estimator. To numerically compute the proposed online lasso estimators, we consider the proximal gradient descent algorithm, which is feasible for high-dimensional models and is computationally efficient. Extensive simulation studies and real-data analysis are conducted to illustrate the finite-sample performance of the proposed methods.

Key words and phrases: Debiased lasso estimator, Online learning, Streaming data, Support vector machine.

*Corresponding authors: Xu Guo and Haobo Qi. Email addresses: xustat12@bnu.edu.cn and haobo4869@bnu.edu.cn.

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

1. Introduction

With the rapid advancement of information technology, the volume of data generated is increasing steadily and the analysis of large-scale data has received much attention within the field of statistical analysis (Wang et al., 2021; Zhu et al., 2021). In recent years, there has been a growing interest in the analysis of streaming data. Analogously to the flow of water streams, streaming data refers to a data collection mechanism in which observations are collected sequentially and continuously over time. For instance, log files from commerce purchases and financial transactions are updated continuously while the data monitors are making real-time decisions (Akter and Wamba, 2016; Das et al., 2018). The streaming data has a wide range of applications covering the field of web applications (Jiang et al., 2018), infectious disease surveillance (Choi et al., 2016; Samaras et al., 2020), mobile health consortia (Shameer et al., 2016; Kraft et al., 2020), and many others. The shift towards real-time analytics is driven by the need for timely insights and the ability to make data-driven decisions quickly.

Despite its widespread application in real practice, the statistical analysis of streaming data suffers from several challenges. First, since the streaming data arrives sequentially, a direct analysis based on the entire dataset becomes infeasible. Second, due to limited storage resources, only a finite and overwritable set of statistics can be stored during the whole learning process. Consequently, traditional offline statistical methods, which highly rely on access to the entire static dataset for analysis, become ineffective. Therefore, it is necessary to develop novel statistical methods that can update the previous results with only newly arrived data and limited historical information.

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

To this end, researchers have proposed various methods to modify the offline estimation and inference procedures into an online-aware manner. Previous efforts can be mainly categorized into two directions. The literature in the first direction considers performing statistical estimation and inference in an updating manner. For instance, Schifano et al. (2016) developed online estimating algorithms and statistical inference procedures for linear models and estimating equations. Luo and Song (2020) proposed an incremental optimization framework for generalized linear models. Wu et al. (2021) studied online updating for survival analysis under the Cox proportional hazards models. Based on convolution smoothing (Fernandes et al., 2021), Jiang and Yu (2022) investigated parameter estimation and variable selection for the low-dimensional quantile regression model. Sun et al. (2023) also considered the low-dimensional quantile regression model with streaming data. For high-dimensional data, Han et al. (2021) proposed an online debiased lasso estimator for the linear regression model. Luo et al. (2023) further introduced an online statistical inference approach for high-dimensional generalized linear models with streaming data. Peng and Wang (2024) investigated online statistical inference for the high-dimensional quantile regression model. For other recent studies about streaming data, see also Ma et al. (2023); Quan and Lin (2024); Jiang et al. (2024).

Another research direction focuses on online optimization algorithms. Among them, the first-order optimization algorithms have drawn abundant attention due to their iterative nature and computational efficiency. Take the stochastic gradient descent (SGD) algorithm (Fang et al., 2018; Zhu et al., 2023; Han et al., 2024) as an example.

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

When a new batch of data arrives, the SGD algorithm computes the gradient with respect to the current data and then updates the model estimates. The main advantage of the SGD algorithm is that it only requires calculating a gradient vector and storing the historical estimators for point estimation. However, the SGD algorithm also faces the problem of a slow convergence rate and learning rate misspecification. Moreover, it still needs to store the cumulative sample covariance matrix of the historical data for inference. Therefore, several improvements have been proposed (Toulis et al., 2014; Bordes et al., 2009; Duchi et al., 2011). Besides the first-order online algorithms, several high-order methods such as the online quasi-Newton algorithm (Schraudolph et al., 2007; Bordes et al., 2009) and online Newton step (Hazan et al., 2007) have been altered for streaming data. However, these methods mainly focus on the numerical convergence analysis, while the statistical properties and inferences of the obtained estimator have not received much attention.

The support vector machine (SVM) (Cortes and Vapnik, 1995) is a widely used and highly powerful algorithm for binary classification, which has been extensively studied by the past literature; see for example Blanchard et al. (2008); Koo et al. (2008); Park et al. (2012); Peng et al. (2016); Lian and Fan (2018); Wang et al. (2019); Zhang et al. (2022). The standard SVM can also be called the ℓ_2-norm SVM since its objective function is the hinge loss plus an ℓ_2-penalty. In modern classification problems, however, we often encounter the challenge of a great deal of predictive features. For example, it is important to build a classifier for disease diagnosis with thousands of candidate genes, while only dozens of instances are available for genetic study. For such a high-

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

dimensional setting, Zhu et al. (2004) considered replacing the ℓ_2-norm penalty with the ℓ_1-norm penalty, which is known as the ℓ_1-norm SVM. Compared to the standard SVM, the ℓ_1-norm SVM is an effective improvement, which can perform classification and variable selection simultaneously. Peng et al. (2016) established an error bound for ℓ_1-norm SVM in ultra-high dimension. Besides, another challenge of the SVM algorithm is that the hinge loss is not smooth, making the sparse penalized SVM computationally intensive. To tackle this important issue, Wang et al. (2023) proposed a novel density-convoluted SVM (DCSVM), which embeds a second-order differentiable loss function via kernel smoothing. They further showed that the DCSVM outperforms the traditional SVM under sparse high-dimensional settings. Moreover, this idea has been extended to the convolution support matrix machine (Chen and Chen, 2024) and tensor support vector machine (Wang et al., 2024). A similar convolution smoothing technique has been applied by Wang et al. (2025) to the online updating estimation for ℓ_2-norm SVM with low-dimensional data. However, valid estimation and inference for DCSVM remain to be studied when the data are sequential and high-dimensional. The good theoretical and empirical performance of DCSVM motivates us to further explore its estimation and inference problems in a high-dimensional online setting.

In this paper, we investigate the estimation and inference problems of the DCSVM with high-dimensional streaming data. With the help of the density-convolution technique, we propose an online lasso estimator for the DCSVM. Under mild conditions, we derive the ℓ_1-norm and ℓ_2-norm error bounds of the online lasso estimator. However, the lasso estimator does not have a tractable limiting distribution due to its inherent

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

bias. To address this issue, an online debiased lasso estimator (Van de Geer et al., 2014; Zhang and Zhang, 2014) is further studied for performing valid statistical inference. Under mild regularity conditions, we derive the asymptotic normality of the proposed online debiased estimator. Actually even in the offline setting, a debiased lasso estimator is not developed and investigated for the DCSVM. For our procedures, only the current data batch and limited historical information are required for both the estimation and inference procedures. Furthermore, we consider using the proximal-gradient-descent algorithm to compute the proposed two online estimators, which is computationally efficient.

The rest of the paper is organized as follows. In Section 2.1, we introduce the basic setting and the DCSVM. Section 2.2 presents the online lasso estimator. Section 2.3 introduces the online debiased lasso estimator. Section 3 derives the theoretical properties of the proposed online lasso estimator and the online debiased lasso estimator. Section 4 provides several simulation studies to demonstrate the finite sample performance of the proposed method. The proposed method is applied to real-time financial data in Section 5. Conclusions and discussions are given in Section 6. The supplementary material provides detailed proofs of our theoretical results, technical lemmas used in the proof, and additional theoretical results.

Notations. For a vector $\boldsymbol{a} = (a_1, \ldots, a_p)^\top \in \mathbb{R}^p$, denote $\|\boldsymbol{a}\|_0 = \sum_{j=1}^p I(a_j \neq 0)$, $\|\boldsymbol{a}\|_r = (\sum_{j=1}^p |a_j|^r)^{1/r}$ for $r \geq 1$, and $\|\boldsymbol{a}\|_\infty = \max_{j=1,\ldots,p} |a_j|$. For a real matrix $\mathbf{A} = \{a_{ij}\}_{i,j}^p \in \mathbb{R}^{p\times p}$, let $\tau_{\max}(\mathbf{A})$ and $\tau_{\min}(\mathbf{A})$ be the maximum and the minimum eigenvalues of $\mathbf{A}$, respectively. Define the elementwise max-norm $\|\mathbf{A}\|_{\max} = \max_{i,j} |a_{ij}|$,

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

the elementwise ℓ_1-norm $\|\mathbf{A}\|_1 = \sum_{i,j} |a_{ij}|$, the spectral norm $\|\mathbf{A}\|_2 = \sqrt{\tau_{\max}(\mathbf{A}^\top \mathbf{A})}$ and the matrix L_1-norm $\|\mathbf{A}\|_{L_1} = \max_j \sum_i |a_{ij}|$. For two sequences of non-negative numbers a_n and b_n, we denote $a_n = o(b_n)$ if $\lim_{n\to\infty} a_n/b_n = 0$, and $a_n = O(b_n)$ or $a_n \lesssim b_n$ if $|a_n| \leq C b_n$ for some positive constant $C \geq 0$ independent of n, $a_n \gtrsim b_n$ is similarly defined, and $a_n \asymp b_n$ if $a_n = O(b_n)$ as well as $b_n = O(a_n)$.

2. Methodology

In this section, we describe our online learning methods of DCSVM for streaming data. We first describe the DCSVM as the basic classifier. Based on DCSVM, the online lasso estimator for the model parameters is then constructed. Furthermore, we propose an online debiasing procedure to eliminate the bias of the lasso estimator and construct valid inferences.

2.1 Density-convoluted Support Vector Machine

Suppose that $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ are the independently and identically distributed $(i.i.d)$ observations collected from an unknown distribution $P(\mathbf{x}, y)$. Here $y_i \in \{-1, 1\}$ is the class label and $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top \in \mathbb{R}^p$ is the associated p-dimensional feature vector. Then, for a binary classification problem, the standard soft-margin linear SVM model (Cortes and Vapnik, 1995) is established as

$$\min \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 \qquad \text{s.t.} \quad 1 - y_i(\mathbf{x}_i^\top \boldsymbol{\beta}) \leq \zeta_i,$$
$$\zeta_i \geq 0, \ \sum_{i=1}^n \zeta_i \leq C, \quad i = 1, \ldots, n$$

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

where $\zeta_i, i = 1, \dots n$ are slack variables with tolerance C and $\boldsymbol{\beta} \in \mathbb{R}^p$ is the model parameter. The constrained optimization problem above can be rewritten into the following Lagrangian multiplier form as

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ \frac{1}{n}\sum_{i=1}^{n} L(y_i \mathbf{x}_i^\top \boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_2^2 \right\},$$

where $L(\cdot)$ is the hinge loss function $L(u) = \max\{0, 1-u\}$ (Vapnik, 1996), and λ is a tuning regularization parameter. To solve the minimization problem above, one should start with the population hinge loss $\mathbb{E}\{L(y\mathbf{x}^\top\boldsymbol{\beta})\}$. First of all, the target parameter is the minimizer

$$\boldsymbol{\beta}^0 = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \mathbb{E}\{L(y\mathbf{x}^\top\boldsymbol{\beta})\}. \tag{2.1}$$

Moreover, as shown in Koo et al. (2008), the gradient vector $\boldsymbol{S}(\boldsymbol{\beta}) \in \mathbb{R}^p$ and the Hessian matrix $\mathbf{H}(\boldsymbol{\beta}) \in \mathbb{R}^{p\times p}$ of the population hinge loss can be calculated as, respectively,

$$\begin{aligned} \boldsymbol{S}(\boldsymbol{\beta}) &= -\mathbb{E}\{\mathbb{I}(y\mathbf{x}^\top\boldsymbol{\beta} \le 1)\mathbf{x}y\}, \\ \mathbf{H}(\boldsymbol{\beta}) &= \mathbb{E}\{\delta(1 - y\mathbf{x}^\top\boldsymbol{\beta})\mathbf{x}\mathbf{x}^\top\}, \end{aligned} \tag{2.2}$$

where $\mathbb{I}(\cdot)$ is the indicator function and $\delta(\cdot)$ is the Dirac delta function. In order to obtain $\widehat{\boldsymbol{\beta}}$, we need to find appropriate estimators for $\boldsymbol{S}(\boldsymbol{\beta})$ and $\mathbf{H}(\boldsymbol{\beta})$. Specifically, $\boldsymbol{S}(\boldsymbol{\beta})$ can be directly estimated by its sample analog, that is $-n^{-1}\sum_{i=1}^{n}\mathbb{I}(y_i\mathbf{x}_i^\top\boldsymbol{\beta} \le 1)\mathbf{x}_i y_i$. While for $\mathbf{H}(\boldsymbol{\beta})$, note that the Dirac delta function involved in $\mathbf{H}(\boldsymbol{\beta})$ is either 0 or ∞, and thus the sample moment estimators are infeasible. Therefore, constructing proper

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

estimators for $\mathbf{H}(\boldsymbol{\beta})$ becomes challenging. To address this critical issue, Wang et al. (2023) recently adopted the idea of smoothing the hinge loss $L(\cdot)$ by substituting a kernel density estimator for the measure in the calculation of expectation. Specifically, they considered using the following density-convoluted loss

$$L_h(t) = \int_{-\infty}^{\infty} L(u) \frac{1}{h} K\left(\frac{u-t}{h}\right) \mathrm{d}u, \tag{2.3}$$

where $K(\cdot)$ is a kernel function and $h > 0$ is the bandwidth. Based on this idea, the ℓ_1-norm density-convoluted SVM (DCSVM) is constructed as

$$\widehat{\boldsymbol{\beta}}_h = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} L_h(y_i \mathbf{x}_i^\top \boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1 \right\}. \tag{2.4}$$

The true parameter of DCSVM is thereby defined as the minimizer of the population convoluted loss, that is

$$\boldsymbol{\beta}^{0h} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \mathbb{E}\{L_h(y\mathbf{x}^\top \boldsymbol{\beta})\}. \tag{2.5}$$

As a new classifier, the DCSVM possesses good theoretical and empirical performance. Thus we build our methodology based on the DCSVM.

Note that $L_h(\cdot)$ in (2.3) is convex and second-order differentiable. The first and second derivatives can be computed as

$$L_h^{'}(t) = \int_{-\infty}^{\frac{1-t}{h}} K(u)\mathrm{d}u, \qquad L_h^{''}(t) = \frac{1}{h} K\Big(\frac{1-t}{h}\Big).$$

We then naturally define the smoothed population gradient $\boldsymbol{S}_h(\boldsymbol{\beta}) := \mathbb{E}\{L_h^{'}(y\mathbf{x}^\top\boldsymbol{\beta})y\mathbf{x}\}$ and the Hessian matrix $\mathbf{H}_h(\boldsymbol{\beta}) := \mathbb{E}\{L_h^{''}(y\mathbf{x}^\top\boldsymbol{\beta})\mathbf{x}\mathbf{x}^\top\}$, respectively. Furthermore, given a random sample $\mathcal{D} = \{y_i, \mathbf{x}_i\}_{i=1}^n$, the sample analog of $\boldsymbol{S}_h(\boldsymbol{\beta})$ and $\mathbf{H}_h(\boldsymbol{\beta})$ can be

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

calculated as follows,

$$\widehat{\boldsymbol{S}}_h(\boldsymbol{\beta};\mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} L_h'(y_i\mathbf{x}_i^\top\boldsymbol{\beta})y_i\mathbf{x}_i = -\frac{1}{n}\sum_{i=1}^{n}\int_{-\infty}^{\frac{1-y_i\mathbf{x}_i^\top\boldsymbol{\beta}}{h}} K(u)\mathrm{d}u\cdot y_i\mathbf{x}_i, \tag{2.6}$$

$$\widehat{\mathbf{H}}_h(\boldsymbol{\beta};\mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} L_h''(y_i\mathbf{x}_i^\top\boldsymbol{\beta})\mathbf{x}_i\mathbf{x}_i^\top = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K\Big(\frac{1-y_i\mathbf{x}_i^\top\boldsymbol{\beta}}{h}\Big)\mathbf{x}_i\mathbf{x}_i^\top. \tag{2.7}$$

With the help of estimators (2.6) and (2.7), efficient offline algorithms can be developed to solve the optimization problem in (2.4), such as Newton's method. However, the historical data cannot be preserved due to limited storage under the online setting, making it invalid to estimate the model parameters based on the complete data. Then how to construct a new online-aware estimator for streaming data becomes a problem of interest.

2.2 Online Lasso Estimator

Consider a standard streaming data setup, where there are a total of $b \geq 2$ data batches, denoted by $\mathcal{D}_1,\ldots,\mathcal{D}_b$, arriving in a sequential manner. Let the batch size be $n_l = |\mathcal{D}_l|$ for $l = 1,\ldots,b$, and the cumulative sample size be $N_l = \sum_{s=1}^{l} n_s$. Denote $\mathcal{D}_l = \{\mathbf{x}_{li}, y_{li}\}_{i=1}^{n_l}$, where $\mathbf{x}_{li}$ is the p-dimensional feature vector and y_{li} is the binary response. Further suppose that $\log(p) = O(n_1^r), r \in (0, 1/4]$, meaning that the dimensionality of covariates is in exponential order of the sample size in the first batch. The rationale behind this requirement is that we expect our procedure to have good performance on the first batch. As discussed in Section 3, estimation errors will pass through batches and accumulate. Therefore, we need enough units on the first batch to ensure the "initial error" is small. A similar condition was imposed by Luo et al. (2023); Xie et al. (2025). Moreover, let $n_l \asymp (N_l/l)$ for $l = 1,\ldots,b$, indicating our data streams have a

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

balanced allocation of the batch sizes. This condition helps to control the estimation errors in the subsequent batches. In practice, it is unnecessary for every batch to have the same size. Schifano et al. (2016) required a similar condition. They assumed that $\lim_{n_l\to\infty}(n_l/N_l) = C_l$ for some constants $C_l \in (0,1)$.

Denote the loss function in the l-th batch as $L_{h_l}(\boldsymbol{\beta};\mathcal{D}_l) = n_l^{-1}\sum_{i=1}^{n_l} L_{h_l}(y_{li}, \mathbf{x}_{li}^\top\boldsymbol{\beta})$. Let $\widehat{\boldsymbol{S}}_{h_l}(\boldsymbol{\beta};\mathcal{D}_l)$ and $\widehat{\mathbf{H}}_{h_l}(\boldsymbol{\beta};\mathcal{D}_l)$ be the corresponding sample gradient and the sample Hessian matrix in the l-th batch, respectively. Furthermore, let $\widetilde{\mathcal{D}}_b = \{\mathcal{D}_1,\ldots,\mathcal{D}_b\}$ denote the cumulative dataset up to the b-th batch. For high-dimensional data, recall the offline ℓ_1-norm DCSVM estimator in (2.4), which is rewritten as

$$\widehat{\boldsymbol{\beta}}^*_{h_b} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\left\{\frac{1}{N_b}\sum_{l=1}^{b} n_l L_{h_l}(\boldsymbol{\beta};\mathcal{D}_l) + \lambda_b\|\boldsymbol{\beta}\|_1\right\}. \tag{2.8}$$

Note that we cannot obtain $\widetilde{\mathcal{D}}_b$ as a whole for streaming data, thus the offline estimator $\widehat{\boldsymbol{\beta}}^*_{h_b}$ in (2.8) is inaccessible. To this end, we propose an online version of ℓ_1-norm DCSVM. To illustrate the idea, we start with the simplest case when $b = 2$. Firstly, we obtain the initial estimator through ℓ_1-norm DCSVM on $\mathcal{D}_1$ as

$$\widehat{\boldsymbol{\beta}}^1_{h_1} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\left\{L_{h_1}(\boldsymbol{\beta};\mathcal{D}_1) + \lambda_1\|\boldsymbol{\beta}\|_1\right\}. \tag{2.9}$$

When $\mathcal{D}_2$ arrives after $\mathcal{D}_1$, we aim to renew the estimator from $\widehat{\boldsymbol{\beta}}^1_{h_1}$ only based on $\mathcal{D}_2$ and limited historical information, which is free of the raw data $\mathcal{D}_1$. The key issue then is how to reconstruct the raw data loss $L_{h_1}(\boldsymbol{\beta};\mathcal{D}_1)$ so that it becomes accessible under limited storage. Specifically, we take the second-order Taylor expansion of $L_{h_1}(\boldsymbol{\beta};\mathcal{D}_1)$

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

around $\widehat{\boldsymbol{\beta}}_{h_1}^1$:

$$\begin{aligned} L_{h_1}(\boldsymbol{\beta};\mathcal{D}_1) =& L_{h_1}(\widehat{\boldsymbol{\beta}}_{h_1}^1;\mathcal{D}_1) + \widehat{\boldsymbol{S}}_{h_1}(\widehat{\boldsymbol{\beta}}_{h_1}^1;\mathcal{D}_1)^\top(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_{h_1}^1) \\ &+ \frac{1}{2}(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_{h_1}^1)^\top\widehat{\mathbf{H}}_{h_1}(\widehat{\boldsymbol{\beta}}_{h_1}^1;\mathcal{D}_1)(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_{h_1}^1) + o(\|\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_{h_1}^1\|_2^2) \\ \approx& L_{h_1}(\widehat{\boldsymbol{\beta}}_{h_1}^1;\mathcal{D}_1) + \frac{1}{2}(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_{h_1}^1)^\top\widehat{\mathbf{H}}_{h_1}(\widehat{\boldsymbol{\beta}}_{h_1}^1;\mathcal{D}_1)(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_{h_1}^1). \end{aligned} \tag{2.10}$$

To see the feasibility of the approximation (2.10), we give an intuitive explanation below. Notice that by the KKT condition, $\widehat{\boldsymbol{S}}_{h_1}(\widehat{\boldsymbol{\beta}}_{h_1}^1;\mathcal{D}_1) = -\lambda_1\boldsymbol{\kappa}_1$, where $\boldsymbol{\kappa}_1$ is a subgradient of $\|\cdot\|_1$ at $\widehat{\boldsymbol{\beta}}_{h_1}^1$. By Hölder's inequality, $|\lambda_1\boldsymbol{\kappa}_1^\top(\widehat{\boldsymbol{\beta}}_{h_1}^1 - \boldsymbol{\beta})| \le \lambda_1\|\boldsymbol{\kappa}_1\|_\infty\|\widehat{\boldsymbol{\beta}}_{h_1}^1-\boldsymbol{\beta}\|_1 = O\{\sqrt{\log(p)/n_1}\|\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_{h_1}^1\|_1\}$ given $\lambda_1 \asymp \sqrt{\log(p)/n_1}$. This implies that $|\widehat{\boldsymbol{S}}_h(\widehat{\boldsymbol{\beta}}_{h_1}^1;\mathcal{D}_1)^\top(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_{h_1}^1)|$ is asymptotically negligible relative to the quadratic term. Moreover, the remainder term $o(\|\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_{h_1}^1\|_2^2)$ is also asymptotically negligible. Since $L_{h_1}(\widehat{\boldsymbol{\beta}}_{h_1}^1,\mathcal{D}_1)$ is a constant and can be ignored for optimization, the online updating loss function can be written as

$$\mathcal{L}_2(\boldsymbol{\beta}) = \frac{1}{N_2}\left\{n_1\frac{1}{2}(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_{h_1}^1)^\top\widehat{\mathbf{H}}_{h_1}(\widehat{\boldsymbol{\beta}}_{h_1}^1;\mathcal{D}_1)(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_{h_1}^1) + n_2 L_{h_2}(\boldsymbol{\beta};\mathcal{D}_2)\right\},$$

and the online renewable estimator is then defined as

$$\widehat{\boldsymbol{\beta}}_{h_2}^2 = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\left\{\mathcal{L}_2(\boldsymbol{\beta}) + \lambda_2\|\boldsymbol{\beta}\|_1\right\}. \tag{2.11}$$

To solve the optimization (2.11) we only require the summary statistics $\{\widehat{\boldsymbol{\beta}}_{h_1}^1, \widehat{\mathbf{H}}_h(\widehat{\boldsymbol{\beta}}_{h_1}^1;\mathcal{D}_1)\}$ and the current data $\mathcal{D}_2$, which is available for streaming data. Similar to the analysis above, at the time $\mathcal{D}_b$ arrives, the renewable estimator $\widehat{\boldsymbol{\beta}}_{h_b}^b$ is given by

$$\widehat{\boldsymbol{\beta}}_{h_b}^b = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\left[\frac{1}{N_b}\left\{\frac{1}{2}(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_{h_{b-1}}^{b-1})^\top\widehat{\mathbf{Q}}_{b-1}(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}_{h_{b-1}}^{b-1}) + n_b L_{h_b}(\boldsymbol{\beta};\mathcal{D}_b)\right\} + \lambda_b\|\boldsymbol{\beta}\|_1\right], \tag{2.12}$$

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

where $\widehat{\mathbf{Q}}_{b-1} = \sum_{l=1}^{b-1} n_l \widehat{\mathbf{H}}_{h_l}(\widehat{\boldsymbol{\beta}}_{h_l}^l; \mathcal{D}_l)$ denotes the aggregated Hessian matrix. Obviously, only the last estimator $\widehat{\boldsymbol{\beta}}_{h_{b-1}}^{b-1}$, the previous Hessian matrix $\widehat{\mathbf{H}}_{h_l}(\widehat{\boldsymbol{\beta}}_{h_l}^l; \mathcal{D}_l), l = 1, \ldots, b-1$ and the present batch $\mathcal{D}_b$ are involved. Therefore, the above estimation procedure can be conducted for streaming data.

It is worth noting that the tuning parameter λ_b should be carefully designed during the online estimation procedure, where we adopt the "rolling-original-recalibration" (ROR) scheme. Suppose we choose the tuning parameter from a pre-specified candidate set $S_\lambda = \{\lambda_1^*, \ldots, \lambda_K^*\}$. On the l-th batch $\mathcal{D}_l$, for each $k \in \{1, \ldots, K\}$, denote $\widehat{\boldsymbol{\beta}}_{h_l}^l(\lambda_k^*)$ as the estimator calculated using $\lambda_l = \lambda_k^*$. Specifically, λ_1 in (2.9) can be selected through offline cross-validation (CV) based on data batch $\mathcal{D}_1$. At time b, regard the previous cumulative data $\widetilde{\mathcal{D}}_{b-1}$ as the training set that generates the online estimator $\widehat{\boldsymbol{\beta}}_{h_{b-1}}^{b-1}$ while the current batch $\mathcal{D}_b$ as the testing set. We then determine λ_b by minimizing the predicted error based on previous estimator $\widehat{\boldsymbol{\beta}}_{h_{b-1}}^{b-1}$:

$$\lambda_b = \arg \min_{k \in \{1, \ldots, K\}} L_{h_b}\left\{\widehat{\boldsymbol{\beta}}_{h_{b-1}}^{b-1}(\lambda_k^*); \mathcal{D}_b\right\}. \tag{2.13}$$

As a result, combining the current data batch $\mathcal{D}_b$ with the former summary statistics $\{\widehat{\boldsymbol{\beta}}_{h_{b-1}}^{b-1}, \widehat{\mathbf{Q}}_{b-1}\}$, one can solve the optimization (2.12). It is noteworthy that the approximated loss function in (2.12), denoted as $\mathcal{L}_b(\boldsymbol{\beta})$, is second-order differentiable about $\boldsymbol{\beta}$ and $\|.\|_1$ is a closed proper convex penalty. Therefore, we employ the proximal-gradient-descent (PGD) algorithm (Beck and Teboulle, 2009), which is computationally efficient. Specifically, given initial value $\widehat{\boldsymbol{\beta}}_b^{(0)}$, the t-th time of PGD algorithm computes:

$$\widehat{\boldsymbol{\beta}}_b^{(t)} = \mathbf{prox}_{\eta_t \lambda_b \|\cdot\|_1}\left\{\widehat{\boldsymbol{\beta}}_b^{(t-1)} - \eta_t \nabla \mathcal{L}_b(\widehat{\boldsymbol{\beta}}_b^{(t-1)})\right\}, \quad t \geq 1, \tag{2.14}$$

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

where η_t is the step size, $\nabla\mathcal{L}_b(\boldsymbol{\beta}) = \big\{\widehat{\mathbf{Q}}_{b-1}(\boldsymbol{\beta}-\widehat{\boldsymbol{\beta}}^{b-1}_{h_{b-1}})+n_b\widehat{\boldsymbol{S}}_{h_b}(\boldsymbol{\beta};\mathcal{D}_b)\big\}/N_b$ is the gradient of $\mathcal{L}_b$ and $\mathbf{prox}_{\eta_t\lambda_b\|\cdot\|_1}(u) = \mathbf{sgn}(u)(|u| - \eta_t\lambda_b)$ is the soft-thresholding operator.

2.3 Online Debiased Lasso Estimator

As shown by Knight and Fu (2000), the lasso-type estimator is biased due to the ℓ_1-norm regularization and one cannot obtain its exact limiting distribution. Therefore, it is difficult to construct valid inference directly using the ℓ_1-norm DCSVM estimator in (2.8). To circumvent this issue, we introduce a debiased lasso estimator based on the lasso estimator in (2.8) as follows:

$$\widehat{\boldsymbol{\beta}}^{*,de}_{h_b} = \widehat{\boldsymbol{\beta}}^{*}_{h_b} - \widehat{\boldsymbol{\Omega}}_b^{-1}\frac{1}{N_b}\sum_{l=1}^{b} n_l\widehat{\boldsymbol{S}}_{h_b}(\widehat{\boldsymbol{\beta}}^{*}_{b};\mathcal{D}_l), \tag{2.15}$$

where $\widehat{\boldsymbol{\Omega}}_b = \sum_{l=1}^{b} n_l\widehat{\mathbf{H}}_{h_b}(\widehat{\boldsymbol{\beta}}^{*}_{h_b};\mathcal{D}_l)$, $\widehat{\boldsymbol{S}}_{h_l}$ and $\widehat{\mathbf{H}}_{h_l}$ are defined in (2.6) and (2.7). Under a high-dimensional setting, $\widehat{\boldsymbol{\Omega}}_b$ is probably rank deficient and singular, thus the standard inverse $\widehat{\boldsymbol{\Omega}}_b^{-1}$ is not a good estimator. To this end, one may obtain the approximate precision matrix $\widehat{\boldsymbol{\Lambda}}_b$ by solving an optimization problem originally proposed by Cai et al. (2011),

$$\widehat{\boldsymbol{\Lambda}}_b = \arg\min_{\boldsymbol{\Lambda}_b\in\mathbb{R}^p}\|\boldsymbol{\Lambda}_b\|_1, \quad \text{s.t.} \quad \big\|\boldsymbol{\Lambda}_b\widehat{\boldsymbol{\Omega}}_b/N_b - \mathbf{I}\big\|_{\max} \le C\zeta_{N_b}, \tag{2.16}$$

for some tuning parameter $\zeta_{N_b} \to 0$ as $N_b \to \infty$.

Similar to the previous analysis, the offline debiased lasso estimator $\widehat{\boldsymbol{\beta}}^{*,de}_{h_b}$ is also infeasible for streaming data. To define an online debiased lasso estimator, we first replace $\widehat{\boldsymbol{\Omega}}_b$ with the aggregated Hessian matrix $\widehat{\mathbf{Q}}_b$ in (2.12), and similarly calculate

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

the approximate inverse $\widehat{\mathbf{\Theta}}_b$ by solving the following optimization,

$$\widehat{\mathbf{\Theta}}_b = \arg\min_{\mathbf{\Theta}_b \in \mathbb{R}^p} \|\mathbf{\Theta}_b\|_1, \quad \text{s.t.} \quad \left\|\mathbf{\Theta}_b \widehat{\mathbf{Q}}_b / N_b - \mathbf{I}\right\|_{\max} \leq C\omega_{N_b}, \tag{2.17}$$

for some tuning parameter $\omega_{N_b} \to 0$ as $N_b \to \infty$. From the optimization (2.17), $\widehat{\mathbf{\Theta}}_b$ can be obtained via previous summary statistics $\widehat{\mathbf{Q}}_{b-1}$ and current data batch $\mathcal{D}_b$. Then, given the online updated lasso estimator $\widehat{\boldsymbol{\beta}}^b_{h_b}$ solved by optimization (2.12), we propose the following online debiased ℓ_1-norm DCSVM estimator

$$\widehat{\boldsymbol{\beta}}^{b,de}_{h_b} = \widehat{\boldsymbol{\beta}}^b_{h_b} - \widehat{\mathbf{\Theta}}_b \frac{1}{N_b} \sum_{l=1}^{b} n_l \widehat{\boldsymbol{S}}_{h_l}(\widehat{\boldsymbol{\beta}}^l_{h_l}; \mathcal{D}_l) + \widehat{\mathbf{\Theta}}_b \frac{1}{N_b} \sum_{l=1}^{b} n_l \widehat{\mathbf{H}}_{h_l}(\widehat{\boldsymbol{\beta}}^l_{h_l}; \mathcal{D}_l)(\widehat{\boldsymbol{\beta}}^l_{h_l} - \widehat{\boldsymbol{\beta}}^b_{h_b}). \tag{2.18}$$

The online debiased estimator (2.18) has two error correction terms. The first term is the bias from the lasso penalty, which is inherited from the offline debiased estimator (2.15). When $b = 1$, the online estimator (2.18) is equivalent to the offline estimator (2.15). The second term is the error stemming from the approximation of the $\widehat{\boldsymbol{S}}_{h_b}(\boldsymbol{\beta}^0; \mathcal{D}_l)$ by the summary statistic $\widehat{\boldsymbol{S}}_{h_l}(\widehat{\boldsymbol{\beta}}^l_{h_l}; \mathcal{D}_l)$. This correction term is engaged to reduce the approximation error arising from the online update. Moreover, define the following summary statistics

$$\begin{aligned} &(\boldsymbol{W}^{(1)}_b)_{p\times 1} = \sum_{l=1}^{b} n_l \widehat{\boldsymbol{S}}_h(\widehat{\boldsymbol{\beta}}^l_{h_l}; \mathcal{D}_l), \quad (\boldsymbol{W}^{(2)}_b)_{p\times 1} = \sum_{l=1}^{b} n_l \widehat{\mathbf{H}}_h(\widehat{\boldsymbol{\beta}}^l_{h_l}; \mathcal{D}_l)\widehat{\boldsymbol{\beta}}^l_{h_l}, \\ &\text{and} \quad (\widehat{\mathbf{Q}}_b)_{p\times p} = \sum_{l=1}^{b} n_l \widehat{\mathbf{H}}_h(\widehat{\boldsymbol{\beta}}^l_{h_l}; \mathcal{D}_l). \end{aligned} \tag{2.19}$$

Then, the proposed debiased estimator (2.18) aligns with the online setting framework, as it only requires the current data batch $\mathcal{D}_b$ and the summary statistics in (2.19) instead of the entire cumulative data $\widetilde{\mathcal{D}}_b$. Moreover, the storage cost of summary statistics in (2.19) is $O(p^2)$, which is free of batch number b and total size N_b. Thus,

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

the storage space is unchanged as the data batches arrive in succession and the total size increases. To summarize the proposed online debiased lasso procedure, we provide the pseudocode for the detailed computation process in the following Algorithm 1.

3. Theoretical Results

In this section, we establish some theoretical properties for the proposed online estimation and inference procedures. Before introducing the theoretical results, we first state some prerequisite assumptions. Define the signal set $S_0 = \{j : \beta_j^0 \neq 0,\ 1 \leq j \leq p\}$ and the sparsity level $s_0 = \#S_0$. Denote the conditional density of $\mathbf{x}$ given $y = 1$ and $y = -1$ as $f(\cdot)$ and $g(\cdot)$, respectively. For any $j \in \{1, \ldots, p\}$, let $\mathbf{x}_{-j}$ denote the subvector of $\mathbf{x}$ without x_j and let $f(\cdot \mid \mathbf{x}_{-j})$ be the density function of x_j given $\mathbf{x}_{-j}$ and conditioned on $y = 1$. Similarly, let $g(\cdot \mid \mathbf{x}_{-j})$ be the density function of x_j given $\mathbf{x}_{-j}$ under $y = -1$. Define $f'(x_j \mid \mathbf{x}_{-d}) := \mathrm{d}f(x_j \mid \mathbf{x}_{-d})/\mathrm{d}x$ and $g'(x_j \mid \mathbf{x}_{-d}) := \mathrm{d}g(x_j \mid \mathbf{x}_{-d})/\mathrm{d}x$. For any function $q : \mathbb{R} \to \mathbb{R}$, define $B(q) = \sup_{x \in \mathbb{R}} \max\{q(x), |x|q(x), x^2 q(x)\}$.

Assumption 1. Assume that $\|\mathbf{x}\|_\infty \leq u_{\mathbf{x}}$ for some positive constant $u_{\mathbf{x}}$.

Assumption 2. There exists a positive constant M such that

$$0 < M^{-1} < \tau_{\min}\{\mathbf{H}(\boldsymbol{\beta}^0)\} \leq \tau_{\max}\{\mathbf{H}(\boldsymbol{\beta}^0)\} < M.$$

Assumption 3. Assume that $\boldsymbol{\beta}^0$ is unique and there exists $d \in \{1, \ldots, p\}$ such that the d-th element of $\boldsymbol{\beta}^0$ satisfies $(\beta_d^0)^{-1} \leq c$ for some constant $c > 0$.

Assumption 4. Assume that $f(\cdot)$ and $g(\cdot)$ are continuous and have finite second moments. In addition, $\max\{B(f(\cdot \mid \mathbf{x}_{-d})), B(g(\cdot \mid \mathbf{x}_{-d})), B(f'(\cdot \mid \mathbf{x}_{-d})), B(g'(\cdot \mid \mathbf{x}_{-d}))\} \leq$

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

Algorithm 1: Online debiased lasso procedure for DCSVM

Input: Kernel function $K(\cdot)$, the bandwidth h, a candidate set S_λ;

Receive the first data batch $\mathcal{D}_1$;

Compute $\{\widehat{\boldsymbol{\beta}}^1_{h_1}(\lambda), \lambda \in S_\lambda\}$ by offline DCSVM and select λ_1 through CV;

Record $\{\widehat{\boldsymbol{\beta}}^1_{h_1}(\lambda), \lambda \in S_\lambda\}$ and summary statistics $\{\widehat{\mathbf{Q}}_1, \boldsymbol{W}_1^{(1)}, \boldsymbol{W}_1^{(2)}\}$ as in (2.19);

Calculate $\widehat{\boldsymbol{\Theta}}_1$ through (2.17) and compute $\widehat{\boldsymbol{\beta}}^{1,de}_{h_1}$ via (2.15) and $\widehat{\mathbf{G}}_1$ via (3.20).

for $b = 2, 3, \ldots$ **do**

  Receive the streaming dataset $\mathcal{D}_b$;

  Load the summary statistics $\{\widehat{\mathbf{Q}}_{b-1}, \boldsymbol{W}_{b-1}^{(1)}, \boldsymbol{W}_{b-1}^{(2)}\}$;

  For each $\lambda \in S_\lambda$, compute $\widehat{\boldsymbol{\beta}}^b_{h_b}(\lambda)$ by PGD algorithm given in (2.14) and record $\{\widehat{\boldsymbol{\beta}}^b_{h_b}(\lambda), \lambda \in S_\lambda\}$;

  Determine λ_b based on minimum prediction error via (2.13);

  Update and store summary statistics $\{\widehat{\mathbf{Q}}_b, \boldsymbol{W}_b^{(1)}, \boldsymbol{W}_b^{(2)}\}$ as in (2.19);

  Update $\widehat{\boldsymbol{\beta}}^b_{h_b}$ based on summary statistic $\widehat{\mathbf{Q}}_b$ by (2.17);

  Compute $\widehat{\boldsymbol{\beta}}^{b,de}_{h_b}$ via (2.18) and $\widehat{\mathbf{G}}_b$ via (3.20);

  Release dataset $\mathcal{D}_b$ from the memory.

end

Output: Debiased estimators $\{\widehat{\boldsymbol{\beta}}^{b,de}_{h_b}, \widehat{\mathbf{G}}_b\}, b = 1, 2, \ldots$

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

C for some constant $C > 0$.

Assumption 5. The kernel function $K(\cdot)$ is twice differentiable with upper-bounded first and second derivatives $K'(\cdot)$ and $K''(\cdot)$. Moreover, $K(\cdot)$ satisfies: (i). $K(u) = K(-u)$, $\forall u \in \mathbb{R}$; (ii). $\int_{\mathbb{R}} K(u)\mathrm{d}u{=}1$; (iii). $\kappa_q = \int_{\mathbb{R}} |u|^q K(u)\mathrm{d}u < \infty$, $q = 1, 2$; (iv). $\kappa_{\max} = \sup_{u\in\mathbb{R}} K(u) < \infty$; (v). $\int_{\mathbb{R}}\{K'(u)\}^2\mathrm{d}u < \infty$.

Remark 1. Assumption 1 imposes a bounded design for covariates $\mathbf{x}$, which is common in the context of high-dimensional SVM. See for instance Peng et al. (2016); Lian and Fan (2018). Assumption 2 indicates that $\mathbf{H}(\boldsymbol{\beta}^0)$ has bounded eigenvalues, which is similar to Assumption (C0) in Wang et al. (2019). Assumption 3 is fundamental in the context of SVM, see Lemma 4 and Lemma 5 in Koo et al. (2008). Assumption 4 imposes some technical conditions on the conditional density function. Similar assumptions are made in Rybak et al. (2025). Assumption 5 imposes some general conditions on the kernel function, which is also assumed by Tan et al. (2022). For instance, the Gaussian kernel or the Epanechnikov kernel satisfies the assumption.

3.1 Smoothing Bias and Offline Estimation Error

First, we provide the following lemma that characterizes the smoothing bias by the density convolution. We then establish the offline estimation error between the DCSVM estimator (2.4) and the true parameter of SVM (2.1).

Lemma 1. *Under Assumption 1-4, we have*

$$\|\boldsymbol{S}_h(\boldsymbol{\beta}^0)\|_2 \leq Ch^2\kappa_2(1+\mu_1)(\beta_d^0)^{-2}$$

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

for some constant $C > 0$. *Here* $\mu_1 = \sup_{\|\boldsymbol{\delta}\|_2=1} \mathbb{E}\big(|\boldsymbol{\delta}^\top \mathbf{x}|\big) < \infty$.

Lemma 1 establishes the upper bound for the smooth gradient valued at $\boldsymbol{\beta}^0$. A similar result has been demonstrated by Lemma 18 in Rybak et al. (2025).

Theorem 1. *Under Assumption 1-5, assume that* $s_0 \log(p) = o(n)$. *If we select* $h \lesssim \{s_0 \log(p)/n\}^{1/4}$ *and* $\lambda = O(\{\log(p)/n\}^{1/2})$, *the offline estimator defined in* (2.4) *satisfies*

$$\|\widehat{\boldsymbol{\beta}}_h - \boldsymbol{\beta}^0\|_2 = O_P\big[\{s_0 \log(p)/n\}^{1/2}\big],$$
$$\|\widehat{\boldsymbol{\beta}}_h - \boldsymbol{\beta}^0\|_1 = O_P\big[s_0\{\log(p)/n\}^{1/2}\big].$$

Theorem 1 establishes the estimation error bounds for the full-sample estimator $\widehat{\boldsymbol{\beta}}_h$ with respect to the parameter $\boldsymbol{\beta}^0$. Instead, in Wang et al. (2023), they established the estimation error bounds for the full-sample estimator $\widehat{\boldsymbol{\beta}}_h$ with respect to the parameter $\boldsymbol{\beta}^{0h}$.

3.2 Estimation Error of Online Lasso Estimator

To start with, we show that a batch-wise choice of tuning parameter λ_l can be lower-bounded by the l_∞-norm of the gradient, which further provides a theoretical guarantee on the good performance of $\widehat{\boldsymbol{\beta}}^l_{h_l}$.

Lemma 2 (Choice of tuning parameter λ_l)**.** *Suppose that Assumption 1 holds. If* $\lambda_l = c_0\sqrt{\log(p)/N_l}$ *for some constant* $c_0 > 0$, *we have, for any* $l = 1, \ldots, b$

$$P\left\{\|\widehat{\boldsymbol{S}}_{h_l}(\boldsymbol{\beta}^0; \mathcal{D}_l) - \boldsymbol{S}_{h_l}(\boldsymbol{\beta}^0)\|_\infty \le \theta_l^{-1}\lambda_l\right\} \ge 1 - 2p^{1-c_0^2/(2u_{\mathbf{x}}^2)},$$

where $\theta_l = (n_l/N_l)^{-1/2}$.

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

Lemma 2 suggests an appropriate order $\lambda_l = O(\sqrt{\log(p)/N_l})$. By choosing a tuning parameter λ_l, the concentration of the loss gradient evaluated at the true parameter is well-controlled with a large probability. Next, we give out the estimation error bounds of the online lasso estimators.

Theorem 2. *Under Assumption 1-5, for $l = 1, \ldots, b$, let the tuning parameter $\lambda_l = O\{\sqrt{\log(p)/N_l}\}$ and the bandwidth $h_l = O[\{s_0 \log(p)/N_l\}^{1/4}]$. Further suppose that $s_0^9 \log(p) b^2 \{1 + \log(N_b/n_1)\}^2 = o(N_b)$ and $b = o\{\log(N_b)\}$. Then, the renewable estimator $\widehat{\boldsymbol{\beta}}_{h_b}^b$ in (2.12) satisfies the following error bounds:*

$$\|\widehat{\boldsymbol{\beta}}_{h_b}^b - \boldsymbol{\beta}^0\|_1 \leq c_b^{(1)} s_0 \lambda_b,$$

$$\|\widehat{\boldsymbol{\beta}}_{h_b}^b - \boldsymbol{\beta}^0\|_2 \leq c_b^{(2)} s_0^{1/2} \lambda_b,$$

with probability $1 - p^{-C}$, for some positive constant C. Here the constants $c_b^{(1)}$ and $c_b^{(2)}$ do not rely on the sample size N_b.

Theorem 2 provides the error bounds and the convergence rates for the online lasso estimator $\widehat{\boldsymbol{\beta}}_{h_b}^b$ in both ℓ_1-norm and ℓ_2-norm. For any batch, we know from Theorem 2 that $c_b^{(1)}$ and $c_b^{(2)}$ are constants independent of N_b. A key observation is that $\widehat{\boldsymbol{\beta}}_{h_b}^b$ is updated based on the former estimator $\widehat{\boldsymbol{\beta}}_{h_{b-1}}^{b-1}$, which carries the previous estimation error into the subsequent estimators and leads to error accumulation. As a result, it is unavoidable that constants $c_b^{(1)}$ and $c_b^{(2)}$ rely on b. To this end, $b = o(\log(N_b))$ is imposed to control these constants well.

By Theorem 1, an offline estimator with full samples has the same convergence rates as the online estimator. However, the sparsity condition in the offline case is

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

$s_0 \log(p) = o(N_b)$. In contrast, we impose a more stringent condition for sparsity s_0 in Theorem 2 due to the more complicated theoretical analysis involving differences between Hessian matrices. See Lemma **??** in the supplementary for details. Similar conditions were imposed in online learning of smoothed quantile regression models (Xie et al., 2025). Actually, Xie et al. (2025) assumed $\{s_0 \log(p)\}^{(3-\gamma)/(1-\gamma)} \leq n_1$ with $\gamma \in [3/4, 1)$ for smoothed quantile regression models.

Remark 2. In Wang et al. (2023), the DCSVM is regarded as a new classifier itself. Hence, their theory is established on the population density-convoluted hinge loss and the bandwidth h is selected from a finite set of constants. In contrast, we focus on the minimizer $\boldsymbol{\beta}^0$ of the original hinge loss here following the line of Wang et al. (2019), Wang et al. (2025), and Rybak et al. (2025). The main reason for our viewpoint is that, besides making better classification, we also aim to make inference for the parametric vector. On the other hand, our theoretical results can also be obtained for the DCSVM with some modifications. For instance, we could derive the online estimation error bounds $\|\widehat{\boldsymbol{\beta}}_h^b - \boldsymbol{\beta}^{0h}\|_1 \lesssim s_0\sqrt{\log(p)/N_b}$ and $\|\widehat{\boldsymbol{\beta}}_h^b - \boldsymbol{\beta}^{0h}\|_2 \lesssim \sqrt{s_0 \log(p)/N_b}$ which are similar with Theorem 2 under mild conditions. The supplementary material provides all the theoretical results and proof of online estimation and inference results for $\boldsymbol{\beta}^{0h}$.

3.3 Asymptotic Properties of Online Debiased Lasso Estimator

We first present an additional assumption for the inverse of the Hessian matrix.

Assumption 6. Denote $\boldsymbol{\Theta}_0 = \{\mathbf{H}(\boldsymbol{\beta}^0)\}^{-1}$, where $\mathbf{H}(\boldsymbol{\beta})$ is the population Hessian matrix of SVM defined in (2.2). Assume that there exists some $C_0 > 0$ such that

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

$\|\mathbf{\Theta}_0\|_{L_1} \leq C_0$. In addition, let $\Theta_{0,ij}$ be the (i,j)-th element of $\mathbf{\Theta}_0$. Assume that $\mathbf{\Theta}_0$ is row-wise sparse that $\max_{1\leq i\leq p}\sum_{j=1}^{p}|\Theta_{0,ij}|^q < C_r$ for $0 \leq q < 1$ and some $C_r > 0$. Here C_0 and C_r are bounded away from 0 and allowed to grow with N_b.

Remark 3. Assumption 6 imposes the sparsity of the inverse Hessian matrix $\mathbf{\Theta}_0$ both in the sense of L_1-norm and the matrix row space. Similar assumptions on the population Hessian matrix were made in Cai et al. (2011), also for high-dimensional inference on convoluted quantile regression (Yan et al., 2023) and convoluted rank regression (Cai et al., 2025). Given these sparse conditions, we establish the error bound for the estimator $\widehat{\mathbf{\Theta}}_b$.

Lemma 3. *Suppose the conditions in Theorem 2 and Assumption 6 hold. Then we have*

$$\left\|\mathbf{\Theta}_0\widehat{\mathbf{Q}}_b/N_b - \mathbf{I}\right\|_{\max} \leq C_0C_1\omega_{N_b}, \quad \left\|\widehat{\mathbf{\Theta}}_b\widehat{\mathbf{Q}}_b/N_b - \mathbf{I}\right\|_{\max} \leq C_0C_2\omega_{N_b}, \quad \|\widehat{\mathbf{\Theta}}_b\|_{L_1} \leq C_0$$

with probability at least $1 - p^{-C_3}$*, for some positive constants* C_1, C_2, C_3*. Here*

$$\begin{aligned}\omega_{N_b} =& s_0\left(\frac{\log(p)\left[(b-1)\{1+\log(N_{b-1}/n_1)\}\right]^{1/2}}{N_{b-1}h_{\min}^{3/2}} + \frac{(b-1)\log^{3/2}(p)}{N_{b-1}h_{\min}^2\sqrt{n_1}} + \sqrt{\frac{\log(p)}{N_{b-1}}}\right)\\ &+ \frac{s_0^2\log(p)\{1+\log(N_{b-1}/n_1)\}}{h_{\min}^3N_{b-1}} + \sqrt{\frac{\log(p)}{N_bh_{\min}^2}} + h_{\max},\end{aligned}$$

where $h_{\min} = \min\{h_1,\ldots,h_b\}$ *and* $h_{\max} = \max\{h_1,\ldots,h_b\}$.

Notice that ω_{N_b} here is chosen according to the upper bound for the max-norm of matrices in Lemma **??** in Supplementary **??**. Hence, with the proper choice of ω_{N_b} suggested in Lemma 3, the optimization problem in (2.17) has a feasible solution $\widehat{\mathbf{\Theta}}_b$ with a high probability.

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

Lemma 4. *Assume the conditions in Lemma 3 hold. With probability tending to 1, we have*

$$\|\widehat{\boldsymbol{\Theta}}_b - \boldsymbol{\Theta}_0\|_{L_1} \leq 14C_r(\omega_{N_b}\|\boldsymbol{\Theta}_0\|_{L_1})^{1-q} \leq 14C_r C_0^{1-q}\omega_{N_b}^{1-q}.$$

Lemma 4 provides the error bound for the inverse Hessian matrix estimator $\widehat{\boldsymbol{\Theta}}_b$ in terms of L_1-norm. Based on this result, we then establish the asymptotic behavior of the proposed debiased estimator in (2.18).

Theorem 3. *Suppose conditions in Theorem 2 and Lemma 3 hold. Then, for sufficiently large N_b, under the scaling conditions $C_r^4 C_0^{4-4q} s_0^{13-9q}\log^{3-q}(p)b^{1-q}\{1+\log(N_b/n_1)\}^{1-q} = o(N_b^{1-q})$, we have*

$$\sqrt{N_b}\big(\widehat{\boldsymbol{\beta}}_{h_b}^{b,de} - \boldsymbol{\beta}^0\big) = \boldsymbol{Z}_b + \boldsymbol{U}_b,$$
$$\boldsymbol{Z}_b = -\frac{1}{\sqrt{N_b}}\boldsymbol{\Theta}_0\sum_{l=1}^{b} n_l\big\{\widehat{\boldsymbol{S}}_{h_l}(\boldsymbol{\beta}^0;\mathcal{D}_l) - \boldsymbol{S}_{h_l}(\boldsymbol{\beta}^0)\big\}, \quad \|\boldsymbol{U}_b\|_\infty = o_p(1).$$

Theorem 3 indicates that the asymptotic representation of $\sqrt{N_b}\big(\widehat{\boldsymbol{\beta}}_{h_b}^{b,de} - \boldsymbol{\beta}^0\big)$ can be divided into two terms. The major term $\boldsymbol{Z}_b$ is the linear combination of independent individuals across $\widetilde{\mathcal{D}}_b = \{\mathcal{D}_1, \ldots, \mathcal{D}_b\}$. While the remainder $\boldsymbol{U}_b$ vanishes as N_b goes to infinity. Further, based on Theorem 3, the asymptotic covariance matrix of $\sqrt{N_b}(\widehat{\boldsymbol{\beta}}_{h_b}^{b,de} - \boldsymbol{\beta}^0)$ can be estimated by

$$\widehat{\mathbf{G}}_b = \frac{1}{N_b}\widehat{\boldsymbol{\Theta}}_b^\top\left[\sum_{l=1}^{b}\sum_{i=1}^{n_l}\mathbf{x}_{li}\mathbf{x}_{li}^\top\left\{\int_{-\infty}^{\frac{1-y_{li}\mathbf{x}_{li}^\top\widehat{\boldsymbol{\beta}}_{h_l}^{l}}{h}} K(u)\mathrm{d}u\right\}^2\right]\widehat{\boldsymbol{\Theta}}_b, \tag{3.20}$$

which can also be updated accordingly under the online setting. Consequently, given the significance level $\alpha \in (0,1)$, we can construct a $(1-\alpha)\cdot 100\%$ confidence interval

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

for some coefficient $\beta_{0,j}$ of interest as

$$\mathbf{CI}_j = \left(\widehat{\beta}^{de}_{b,j} - \widehat{g}_{b,j} Z_{1-\alpha/2}/\sqrt{N_b}, \widehat{\beta}^{de}_{b,j} + \widehat{g}_{b,j} Z_{1-\alpha/2}/\sqrt{N_b}\right),$$

where $\widehat{g}_{b,j}$ is the j-th diagonal element of $\widehat{\mathbf{G}}_b$ and $Z_{1-\alpha/2}$ is the upper-$\alpha/2$ quantile of the standard normal distribution. In addition, we give the following proposition for the asymptotic normality of the offline debiased estimator in (2.15).

Remark 4. Theorem 3 provides the linear representation for estimating $\boldsymbol{\beta}^0$. When the target parameter changes to $\boldsymbol{\beta}^{0h}$, we can also derive that $\sqrt{N_b}(\widehat{\boldsymbol{\beta}}^{b,de}_h - \boldsymbol{\beta}^{0h}) = \boldsymbol{M}_b + \boldsymbol{R}_b$, where $\boldsymbol{M}_b = -\{\mathbf{H}_h(\boldsymbol{\beta}^{0h})\}^{-1} \sum_{l=1}^{b} n_l \{\widehat{\boldsymbol{S}}_{h_l}(\boldsymbol{\beta}^{0h}; \mathcal{D}_l) - \boldsymbol{S}_{h_l}(\boldsymbol{\beta}^{0h})\}/\sqrt{N_b}$ and $\|\boldsymbol{R}_b\|_\infty = o_p(1)$, given that $C_r C_0^{1-q} s_0^{2-q} \log^{5/2-2q}(p)/h^{2-2q} = o(N_b^{1/2-q/2})$.

Proposition 1. *Under conditions* $C_r^4 C_0^{4-4q} s_0^{11-9q} \log^{3-q}(p) = o(N_b^{1-q})$, *we have*

$$\sqrt{N_b}(\widehat{\boldsymbol{\beta}}^{*,de}_{h_b} - \boldsymbol{\beta}^0) = \boldsymbol{Z}^*_b + \boldsymbol{U}^*_b,$$

$$\boldsymbol{Z}^*_b = -\frac{1}{\sqrt{N_b}} \boldsymbol{\Theta}_0 \sum_{l=1}^{b} n_l \big\{\widehat{\boldsymbol{S}}_{h_b}(\boldsymbol{\beta}^0; \mathcal{D}_l) - \boldsymbol{S}_{h_b}(\boldsymbol{\beta}^0)\big\}, \quad \|\boldsymbol{U}^*_b\|_\infty = o_p(1).$$

Similarly, Proposition 1 shows that the offline debiased estimator $\widehat{\boldsymbol{\beta}}^{*,de}_{h_b}$ has asymptotic normality. Valid confidence intervals and hypothesis testing are thereby feasible.

Remark 5. The linear representation in Proposition 1 can also be derived for $\boldsymbol{\beta}^{0h}$ that $\sqrt{N_b}(\widehat{\boldsymbol{\beta}}^{*,de}_h - \boldsymbol{\beta}^{0h}) = -\{\mathbf{H}_h(\boldsymbol{\beta}^{0h})\}^{-1} \sum_{l=1}^{b} n_l \{\widehat{\boldsymbol{S}}_{h_l}(\boldsymbol{\beta}^{0h}; \mathcal{D}_l) - \boldsymbol{S}_{h_l}(\boldsymbol{\beta}^{0h})\}/\sqrt{N_b} + \boldsymbol{R}^*_b$ with $\|\boldsymbol{R}^*_b\|_\infty = o_p(1)$, under the condition $C_r C_0^{1-q} s_0^{2-q} \log^{5/2-2q}(p)/h^{2-2q} = o(N_b^{1/2-q/2})$.

4. Simulations

In this section, we conduct several numerical experiments to illustrate the finite sample performances of the proposed online lasso estimator and online debiased lasso estimator

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

for the DCSVM.

4.1 Model Settings

We first introduce our model setting. We consider the data generation processes similar to Wang et al. (2023). Specifically, assume the binary response $y \in \{-1, 1\}$ and the two classes are balanced, that is $P(y = 1) = P(y = -1) = 1/2$. Moreover, assume the conditional distribution $\mathbf{x}|y = 1$ is drawn from $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\mathbf{x}|y = -1$ is drawn from $N_p(-\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\mu} = (0.8, 0.8, 0.8, 0.8, 0.8, 0, \ldots, 0)^\top$. The covariance matrix is block-diagonal with structure

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}^C_{5\times 5} & \mathbf{0}_{5\times(p-s)} \\ \mathbf{0}_{(p-5)\times 5} & \boldsymbol{\Sigma}^{UC}_{(p-5)\times(p-5)} \end{pmatrix},$$

According to Zhang et al. (2016), the corresponding Bayesian rule is $\mathbf{sgn}(2\boldsymbol{\mu}^\top\boldsymbol{\Sigma}^{-1}\mathbf{x})$. Based on these settings, we consider the following three models:

- **Model 1**

  Both $\boldsymbol{\Sigma}^C$ and $\boldsymbol{\Sigma}^{UC}$ have diagonal elements 1 and off-diagonal elements 0.2. The Bayes rule is $\mathbf{sgn}(0.89\mathbf{x}_1 + 0.89\mathbf{x}_2 + 0.89\mathbf{x}_3 + 0.89\mathbf{x}_4 + 0.89\mathbf{x}_5)$.

- **Model 2**

  $(\boldsymbol{\Sigma}^C)_{i,j} = (0.5)^{|i-j|}$, $1 \le i, j \le 5$ and $(\boldsymbol{\Sigma}^{UC})_{k,l} = (0.5)^{|k-l|}$, $1 \le k, l \le p-5$. The Bayes rule is $\mathbf{sgn}(1.07\mathbf{x}_1 + 0.53\mathbf{x}_2 + 0.53\mathbf{x}_3 + 0.53\mathbf{x}_4 + 1.07\mathbf{x}_5)$.

- **Model 3**

  $(\boldsymbol{\Sigma}^C)_{i,j} = (0.7)^{|i-j|}$, $1 \le i, j \le 5$ and $(\boldsymbol{\Sigma}^{UC})_{k,l} = (0.7)^{|k-l|}$, $1 \le k, l \le p-5$. The Bayes rule is $\mathbf{sgn}(0.94\mathbf{x}_1 + 0.28\mathbf{x}_2 + 0.28\mathbf{x}_3 + 0.28\mathbf{x}_4 + 0.94\mathbf{x}_5)$.

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

To simulate a data stream, a total of N_b samples are generated sequentially as $\mathcal{D}_1, \ldots, \mathcal{D}_b$ under each of the subsequent models. Moreover, we choose batches with equal size, that is $n_l = N_b/b$ for $l = 1, \ldots, b$. Then, the proposed online-aware methods are conducted on the given batches. The results of estimation error and debiased lasso inference are presented in Section 4.2 and Section 4.3, respectively.

4.2 Online Lasso Estimation Error

We first focus on the estimation for high-dimensional sparse parameters of DCSVM. For each model, set $(N_b, b, p, n_l) = (2000, 20, 400, 100)$ and $(6000, 30, 1000, 200)$. According to the conditions in our theory, we set the tuning parameter $\lambda_l = C_\lambda \sqrt{\log(p)/N_l}$. Here the parameter C_λ is selected by the ROR scheme in (2.13) from the candidate set $S_C = \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5\}$. Since the sparsity s_0 is generally unknown, we let $h_l = c_h * \{\log(p)/N_l\}^{1/4}$ with constant $c_h > 0$ to be tuned, guided by Theorem 2. Preliminary experiments have been implemented to find a proper range for c_h. We compute and compare the ℓ_2-norm estimation errors under different values of c_h, as displayed in Figure **??** in Section **??** of the supplementary. Under all scenarios, the ℓ_2-errors reach the bottom when $c_h \in [2, 3]$. Therefore, our studies select c_h from $\{2, 2.2, 2.4, 2.6, 2.8, 3\}$ by the ROR scheme. Finally, according to Assumption 5, we use the Gaussian kernel function.

For comparison, we consider the following methods: (i). the proposed online ℓ_1-DCSVM lasso estimator (2.12), abbreviated as "on-l1-DCSVM"; (ii). the offline DCSVM lasso estimator with full data at the terminal time b, abbreviated as "off-l1-

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

DCSVM"; (iii). the offline SVM lasso estimator at time b, abbreviated as "off-l1-SVM". The offline ℓ_1-DCSVM is coupled with Gaussian kernel function and trained using the R package `dcsvm`; the offline ℓ_1-SVM is fitted using the R package `penalizedSVM`. For both the offline methods, the regularized parameters are tuned via five-fold cross-validation.

Similar to Peng et al. (2016), we evaluate the performance of the proposed method and the two competitors by the misclassification rate, variable selection ability and estimation error. Specifically, we calculate:

- **Test error**: The misclassification error rate on testing data $\widetilde{\mathcal{D}}_b^*$. Here $\widetilde{\mathcal{D}}_b^*$ is an independent copy of $\widetilde{\mathcal{D}}_b$ with equal size and the same underlying distribution. Additionally, we report the corresponding standard errors in the brackets.
- **Signal**: The number of nonzero coefficients $\{\widehat{\beta}_j \neq 0, j \in S_0\}$ with $S_0 = \{1, 2, 3, 4, 5\}$.
- **Noise**: The number of nonzero coefficients $\{\widehat{\beta}_j \neq 0, j \in S_0^c\}$ with $S_0^c = \{1, 2..., p\}/S_0$.
- ℓ_2 **error**: $\|\widehat{\boldsymbol{\beta}}_{h_l}^l - \boldsymbol{\beta}^0\|_2$ on each data batch $\mathcal{D}_l$ for the online ℓ_1-DCSVM, $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0\|_2$ on $\widetilde{\mathcal{D}}_b$ for offline methods.
- **Run time**: The computation cost of each implementation, including all the time spent on tuning and training the models.

The Monte-Carlo simulation results over 200 iterations are shown in Table 1. We conclude the test error of online ℓ_1-DCSVM is as low as that of the offline methods, indicating the good performance of online ℓ_1-DCSVM as a classifier. The standard

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

errors are small, thus the results are stable and credible. Moreover, the signals can be fully selected by the online ℓ_1-DCSVM, while the noise variables are a little more than those of the offline DCSVM. This sacrifice is reasonable due to the tougher requirement for estimation accuracy in the online setting. The run time of the online algorithm is the lowest compared to the two offline benchmarks, indicating the advantages of our method on computational efficiency.

Further, we compute the ℓ_2-norm estimation error of online ℓ_1-DCSVM as the batches arrive sequentially and compare it with the two offline methods with full data at time b. The results for **Models 1-3** are summarized in Figures **??**-**??** in Section **??** of the supplementary, respectively. From the figures, it is clear that as the batches arrive, the ℓ_2-norm estimation errors of online ℓ_1-DCSVM reduce gradually and meet the estimation errors of the two offline methods. As N_l grows, the errors become lower. These patterns match our theoretical claims in Theorem 2.

4.3 Online Debiased Lasso Inference

Shifting our focus to the uncertainty qualification, we construct confidence intervals based on the proposed online debiased DCSVM. We consider the same data generation process and **Models 1-3** in section 4.2 with $(N, b, p, n_l) = (1200, 10, 200, 120)$ and $(2000, 10, 400, 200)$. We evaluate the performance of our methods by the empirical coverage probabilities and the average length of confidence intervals for individual

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

Table 1: Simulation results of estimation performance. Index "TE" stands for the test error (in percentage), index "Sig-Noi" is the pair of the signal and the noise, and index "Time" represents run time (in seconds).

Model 1						
(N_b, b, p, n_s)	$(2000, 20, 400, 100)$			$(6000, 30, 1000, 200)$		
Method	TE (%)	Sig-Noi	Time	TE (%)	Sig-Noi	Time
online l1-DCSVM	12.43(2.3e-3)	(5.00,1.97)	20.99	9.57(1.7e-3)	(5.00,2.16)	247.75
offline l1-DCSVM	12.24(8.8e-4)	(5.00,0.00)	140.63	9.36(3.2e-4)	(5.00,0.00)	1120.39
offline l1-SVM	12.25(1.2e-3)	(5.00,1.49)	3031.12	9.38(4.8e-4)	(5.00,1.86)	42574.07
Model 2						
(N_b, b, p, n_s)	$(2000, 20, 400, 100)$			$(6000, 30, 1000, 200)$		
Method	TE (%)	Sig-Noi	Time	TE (%)	Sig-Noi	Time
online l1-DCSVM	18.18(3.1e-3)	(5.00,2.60)	15.21	11.28(1.1e-3)	(5.00,3.47)	238.67
offline l1-DCSVM	17.92(1.1e-3)	(5.00,0.00)	155.97	11.19(2.3e-4)	(5.00,0.00)	1476.22
offline l1-SVM	18.01(1.3e-3)	(5.00,1.82)	3079.49	11.21(4.5e-4)	(5.00,2.87)	46433.84
Model 3						
(N_b, b, p, n_s)	$(2000, 20, 400, 100)$			$(6000, 30, 1000, 200)$		
Method	TE (%)	Sig-Noi	Time	TE (%)	Sig-Noi	Time
online l1-DCSVM	18.12(2.1e-3)	(4.94,1.11)	16.78	15.23(1.7e-3)	(5.00,2.15)	251.73
offline l1-DCSVM	18.11(1.2e-3)	(5.00,0.00)	115.85	15.07(3.4e-4)	(5.00,0.00)	1625.16
offline l1-SVM	18.11(2.5e-3)	(5.00,1.07)	3284.43	15.11(5.1e-4)	(5.00,1.83)	44865.23

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

coefficients across all covariates, covariates on S_0 and those on S_0^c, that is,

$$\mathbf{CP} = \sum_{j=1}^{p} I(\beta_{0,j} \in \mathbf{CI}_j)/p, \qquad \mathbf{AL} = \sum_{j=1}^{p} (2\widehat{g}_{b,j} Z_{1-\alpha/2}/\sqrt{N_b})/p,$$

$$\mathbf{CP}_{S_0} = \sum_{j\in S_0} I(\beta_{0,j} \in \mathbf{CI}_j)/s_0, \qquad \mathbf{AL}_{S_0} = \sum_{j\in S_0} (2\widehat{g}_{b,j} Z_{1-\alpha/2}/\sqrt{N_b})/s_0,$$

$$\mathbf{CP}_{S_0^c} = \sum_{j\in S_0^c} I(\beta_{0,j} \in \mathbf{CI}_j)/(p-s_0), \qquad \mathbf{AL}_{S_0^c} = \sum_{j\in S_0^c} (2\widehat{g}_{b,j} Z_{1-\alpha/2}/\sqrt{N_b})/(p-s_0).$$

For comparison, we consider the following methods: (i) the proposed online debiased DCSVM lasso estimator (2.18), abbreviated as "online-deb"; (ii) the offline debiased DCSVM lasso estimator (2.15) with full data at time b, abbreviated as "offline-deb".

The results are summarized in Table 2. The empirical coverage probabilities are nominal across all settings. Additionally, the confidence intervals of the online debiased method provide coverage probabilities nearly as high as the offline methods. Moreover, the average lengths of the online method are close to the offline method. In a word, the performance of the proposed online method approaches the offline method with the full benchmark, which indicates the inferential accuracy under online updating. These results indicate the advantages of the proposed online debiased method for the sparse classification model with high-dimensional streaming data.

5. Real Data Analysis

In this section, we apply our proposed method to identify financial distress. The data we used here is available at `https://www.kaggle.com/datasets/shebrahimi/financial-distress`. It collects the financial distress data of a group of companies, which varies from 1 to 14 time points. The financial distress index can be regarded as

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

Table 2: The coverage probability and average length of the confidence intervals.

Method	(N_b, b, p, n_l)	**CP**	**CP**$_{S_0}$	**CP**$_{S_0^c}$	**AL**	**AL**$_{S_0}$	**AL**$_{S_0^c}$
		Model 1					
online-deb	(1200, 10, 200, 120)	0.948	0.940	0.948	0.160	0.142	0.160
offline-deb		0.957	0.950	0.957	0.156	0.140	0.157
online-deb	(2000, 10, 400, 200)	0.947	0.900	0.948	0.140	0.124	0.140
offline-deb		0.951	0.910	0.952	0.136	0.120	0.136
		Model 2					
online-deb	(1200, 10, 200, 120)	0.932	0.920	0.932	0.155	0.153	0.155
offline-deb		0.937	0.920	0.937	0.151	0.148	0.151
online-deb	(2000, 10, 400, 200)	0.936	0.880	0.937	0.142	0.141	0.142
offline-deb		0.942	0.900	0.943	0.139	0.138	0.139
		Model 3					
online-deb	(1200, 10, 200, 120)	0.946	0.940	0.948	0.173	0.169	0.173
offline-deb		0.948	0.940	0.949	0.165	0.161	0.165
online-deb	(2000, 10, 400, 200)	0.945	0.910	0.945	0.161	0.163	0.161
offline-deb		0.948	0.920	0.948	0.158	0.160	0.158

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

the response variable, where the value -1 stands for healthy and the value 1 stands for financially distressed. Meanwhile, there are a total of 82 associated covariate variables that consist of both financial and non-financial characteristics of the sampled companies. Our goal in this study is to select and make inferences for the variables that potentially affect the companies' financial distress.

The dataset consists of $N_b = 3672$ records in total. Each record contains 300 covariates, including the 82 original covariates, the 82 squares of the original covariates and 136 interaction terms (the products of 17 pairs of the original covariates). To use our online estimation procedure, the data was split into $b = 14$ batches in chronological order. The minimal batch size is 242 and the maximal batch size is 280. To identify the influential variables, we aim to test: $H_{0,j} : \beta_{0,j} = 0$ for $j = 1, \ldots, p$. The tuning parameters λ_l and ω_l are determined by the ROR scheme in the same way in simulation studies. Under the prespecified level $\alpha = 0.05$, we observe that 37 variables are significant in the online framework, and the associated p-values of the 14 batches are presented in Figure 1. It can be seen from this figure that most variables are more significant and reach relative stability as data are collected continuously. This example further confirms that our online estimation and inference procedures have fairly good performance in streaming financial data.

6. Conclusion

In this paper, we provide a comprehensive exploration of estimation and inference of support vector machines with streaming data under the high-dimensional regime. The

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

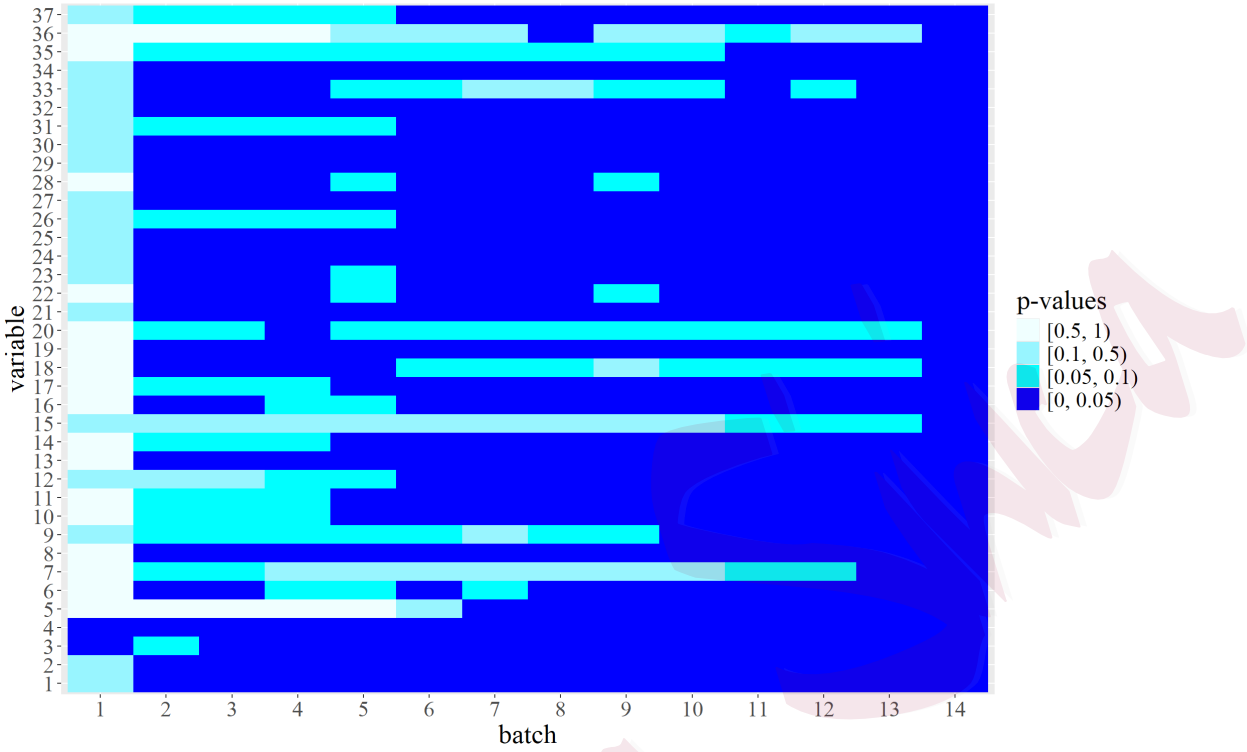

Figure 1: results of p-values for financial distress data

proposed online methods, built upon density-convolution, can be achieved through only the current data batch and the summary statistics of historical data but free of the entire data. Two procedures are studied: the online ℓ_1-regularized estimation and the online debiased estimation, which are capable of making valid inferences for the parameters in binary classification models. Moreover, our theory demonstrates that these methods reach the same statistical accuracy as the offline benchmark with full data. Simulation studies and the real data application further verify the feasibility of our methods.

There are several issues deserving more concern. First, we only consider the i.i.d. data with a fixed model parameter in our framework, while heterogeneity is not tackled.

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

The heterogeneity can be either from data or from model structure. Second, the DCSVM focuses on the linear transformation of covariates, while the non-parametric effects require further exploration. Furthermore, the idea of density convolution can be easily generalized to other non-smooth objective functions and our framework in this paper can be further generalized. Lastly, although our method as with many other existing works (Luo and Song, 2020; Han et al., 2021; Jiang and Yu, 2022; Xie et al., 2025; Jiang et al., 2024; Wang et al., 2025) do not need to store raw data, in the high-dimensional case where $p > N_b$, the storage cost $O(p^2)$ of our methods is indeed larger than that of the raw data $O(N_b p)$. The online stochastic gradient descent (SGD) might be an alternative method to avoid the Hessian matrix. Compared to the proposed second-order method, SGD requires many more steps to converge. We leave space here for further research.

Acknowledgment

We would like to thank the Editor, the Associate Editor, and the two anonymous reviewers for their valuable comments and constructive suggestions, which led to significant improvements in the paper. Guo's research is supported by the National Key R & D Program of China (grant No. 2023YFA1011100), National Natural Science Foundation of China (grant No. 12322112), and the Fundamental Research Funds for the Central Universities. Qi's research is supported by the National Natural Science Foundation of China (grant No. 12401386), and the Fundamental Research Funds for the Central Universities.

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

Supplementary Materials

The online Supplementary Materials contains the proof of theoretical results in Section 3, as well as additional theoretical and simulation results.

References

Akter, S. and S. F. Wamba (2016). Big data analytics in e-commerce: a systematic review and agenda for future research. *Electronic Markets 26*, 173–194.

Beck, A. and M. Teboulle (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *Society for Industrial and Applied Mathematics 2*(1), 183–202.

Blanchard, G., O. Bousquet, and P. Massart (2008). Statistical performance of support vector machines. *The Annals of Statistics 36*, 489–531.

Bordes, A., L. Bottou, and P. Gallinari (2009). Sgd-qn: Careful quasi-newton stochastic gradient descent. *Journal of Machine Learning Research 10*, 1737–1754.

Cai, L., X. Guo, H. Lian, and L. Zhu (2025). Statistical inference for high-dimensional convoluted rank regression. *Journal of the American Statistical Association 120*(552), 1–25.

Cai, T., W. Liu, and X. Luo (2011). A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association 106*(494), 594–607.

Chen, B. and C. Chen (2024). Convoluted support matrix machine in high dimensions. *Statistica Sinica*.

Choi, J., Y. Cho, E. Shim, and H. Woo (2016). Web-based infectious disease surveillance systems and public health perspectives: a systematic review. *BMC Public Health 16*(1), 1238.

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine Learning 20*(3), 273–297.

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

Das, S., R. K. Behera, M. Kumar, and S. K. Rath (2018). Real-time sentiment analysis of twitter streaming data for stock prediction. *Procedia Computer Science 132*, 956–964. International Conference on Computational Intelligence and Data Science.

Duchi, J., E. Hazan, and Y. Singer (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research 12*, 2121–2159.

Fang, Y., J. Xu, and L. Yang (2018). Online bootstrap confidence intervals for the stochastic gradient descent estimator. *Journal of Machine Learning Research 19*, 1–21.

Fernandes, M., E. Guerre, and E. Horta (2021). Smoothing quantile regressions. *Journal of Business & Economic Statistics 39*(1), 338–357.

Han, R., L. Luo, Y. Lin, and J. Huang (2021). Online debiased lasso for streaming data. *arXiv preprint arXiv:2106.05925*.

Han, R., L. Luo, Y. Lin, and J. Huang (2024). Online inference with debiased stochastic gradient descent. *Biometrika 111*, 93–108.

Hazan, E., A. Agarwal, and S. Kale (2007). Logarithmic regret algorithms for online convex optimization. *Machine Learning 69*, 169–192.

Jiang, R., L. Liang, and K. Yu (2024). Renewable huber estimation method for streaming datasets. *Electronic Journal of Statistics 18*(1), 674–705.

Jiang, R. and K. Yu (2022). Renewable quantile regression for streaming data sets. *Neurocomputing 508*, 208–224.

Jiang, T., J. Yang, C. Yu, and Y. Sang (2018). A clickstream data analysis of the differences between visiting behaviors of desktop and mobile users. *Data and Information Management 2*, 130–140.

Knight, K. and W. Fu (2000). Asymptotics for lasso-type estimators. *The Annals of Statistics 28*(5), 1356–

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

1378.

Koo, J.-Y., Y. Lee, Y. Kim, and C. Park (2008). A bahadur representation of the linear support vector machine. *Journal of Machine Learning Research 9*, 1343–1368.

Kraft, R., F. Birk, M. Reichert, A. Deshpande, W. Schlee, B. Langguth, H. Baumeister, T. Probst, M. Spiliopoulou, and R. Pryss (2020). Efficient processing of geospatial mhealth data using a scalable crowdsensing platform. *Sensors 20*(12), 3456.

Lian, H. and Z. Fan (2018). Divide-and-conquer for debiased l_1-norm support vector machine in ultra-high dimensions. *Journal of Machine Learning Research 18*, 1–26.

Luo, L., R. Han, Y. Lin, and J. Huang (2023). Statistical inference in high-dimensional generalized linear models with streaming data. *Electronic Journal of Statistics 17*, 3443–3471.

Luo, L. and P. X.-K. Song (2020). Renewable estimation and incremental inference in generalized linear models with streaming data sets. *Journal of Royal Statistical Society: Series B (Statistical Methodology) 82*(1), 69–97.

Ma, X., L. Lin, and Y. Gai (2023). A general framework of online updating variable selection for generalized linear models with streaming datasets. *Journal of Statistical Computation and Simulation 93*(3), 325–340.

Park, C., K.-R. Kim, R. Myung, and J.-Y. Koo (2012). Oracle properties of scad-penalized support vector machine. *Journal of Statistical Planning and Inference 142*, 2257–2270.

Peng, B., L. Wang, and Y. Wu (2016). An error bound for l_1-norm support vector machine coefficients in ultra-high dimension. *Journal of Machine Learning Research 17*, 1–26.

Peng, Y. and L. Wang (2024). Two-stage online debiased lasso estimation and inference for high-dimensional quantile regression with streaming data. *Journal of Systems Science and Complexity 37*(3), 1251–1270.

Quan, M. and Z. Lin (2024). Optimal one-pass nonparametric estimation under memory constraint. *Journal*

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

of the American Statistical Association 119, 285–296.

Rybak, J., H. Battey, and W.-X. Zhou (2025). On inference for the support vector machine. *Journal of Machine Learning Research 26*(85), 1–54.

Samaras, L., E. García-Barriocanal, and M.-A. Sicilia (2020). Chapter 2 - syndromic surveillance using web data: a systematic review. In M. D. Lytras and A. Sarirete (Eds.), *Innovation in Health Informatics*, Next Gen Tech Driven Personalized Med & Smart Healthcare. Academic Press.

Schifano, E. D., J. Wu, C. Wang, J. Yan, and M.-H. Chen (2016). Online updating of statistical inference in the big data setting. *Technometrics 58*(3), 393–403.

Schraudolph, N. N., J. Yu, and S. G¨unter (2007). A stochastic quasi-newton method for online convex optimization. *Proceedings of Machine Learning Research 2*, 436–443.

Shameer, K., M. A. Badgeley, R. Miotto, B. S. Glicksberg, J. W. Morgan, and J. T. Dudley (2016). Translational bioinformatics in the era of real-time biomedical, health care and wellness data streams. *Briefings in Bioinformatics 18*(1), 105–124.

Sun, X., H. Wang, C. Cai, M. Yao, and K. Wang (2023). Online renewable smooth quantile regression. *Computational Statistics and Data Analysis 185*, 107781.

Tan, K. M., L. Wang, and W.-X. Zhou (2022). High-dimensional quantile regression: Convolution smoothing and concave regularization. *Journal of the Royal Statistical Society Series B: Statistical Methodology 84*(1), 205–233.

Toulis, P., J. Rennie, and E. M. Airoldi (2014). Statistical analysis of stochastic gradient methods for generalized linear models. *Proceedings of Machine Learning Research 32*, 667–675.

Van de Geer, S., P. Bühlmann, Y. Ritov, and R. Dezeure (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics 42*(3), 1166–1202.

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

Vapnik, V. (1996). *The Nature of Statistical Learning Theory*. Springer.

Wang, B., L. Zhou, Y. Gu, and H. Zou (2023). Density-convoluted support vector machines for high-dimensional classification. *IEEE transactions on Information Theory 69*(4), 2523–2536.

Wang, B., L. Zhou, J. Yang, and Q. Mai (2024). Density-convoluted tensor support vector machines. *Statistics and Its Interface 17*, 231–247.

Wang, F., Y. Zhu, D. Huang, H. Qi, and H. Wang (2021). Distributed one-step upgraded estimation for non-uniformly and non-randomly distributed data. *Computational Statistics & Data Analysis 162*, 107265.

Wang, K., X. Meng, and X. Sun (2025). Convolution smoothing and online updating estimation for support vector machine. *Test 34*, 288–323.

Wang, X., Z. Yang, X. Chen, and W. Liu (2019). Distributed inference for linear support vector machine. *Journal of Machine Learning Research 20*(113), 1–41.

Wu, J., M.-H. Chen, E. D. Schifano, and J. Yan (2021). Online updating of survival analysis. *Journal of Computational and Graphical Statistics 30*(4), 1209–1223.

Xie, J., X. Yan, B. Jiang, and L. Kong (2025). Statistical inference for smoothed quantile regression with streaming data. *Journal of Econometrics 249*, 105924.

Yan, Y., X. Wang, and R. Zhang (2023). Confidence intervals and hypothesis testing for high-dimensional quantile regression: Convolution smoothing and debiasing. *Journal of Machine Learning Research 24*, 1–49.

Zhang, C.-H. and S. S. Zhang (2014). Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society Series B: Statistical Methodology 76*(1), 217–242.

Zhang, X., Y. Wu, L. Wang, and R. Li (2016). Variable selection for support vector machines in moderately high dimensions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 78*(1), 53–76.

Statistica Sinica: Preprint
doi:10.5705/ss.202025.0083

Zhang, Y., Y.-Y. Zhao, and H. Lian (2022). Statistical rates of convergence for functional partially linear support vector machines for classification. *Journal of Machine Learning Research 23*(156), 1–24.

Zhu, J., S. Rosset, T. Hastie, and R. Tibshirani (2004). 1-norm support vector machines. *Advances in Neural Information Processing Systems 16*(1), 49–56.

Zhu, W., X. Chen, and W. B. Wu (2023). Online covariance matrix estimation in stochastic gradient descent. *Journal of the American Statistical Association 118*, 393–404.

Zhu, X., F. Li, and H. Wang (2021). Least squares approximation for a distributed system. *Journal of Computational and Graphical Statistics 30*(4), 1004–1018.

School of Statistics, Beijing Normal University, Beijing, China

E-mail: haochenrao@mail.bnu.edu.cn

School of Statistics, Beijing Normal University, Beijing, China

E-mail: xustat@bnu.edu.cn

Department of Mathematics, City University of Hong Kong, Hong Kong, China

E-mail: henglian@cityu.edu.hk

School of Statistics, Beijing Normal University, Beijing, China

E-mail: haobo4869@bnu.edu.cn