Statistica Sinica Preprint No: SS-2024-0194	
Title	Convoluted Support Matrix Machine in High Dimensions
Manuscript ID	SS-2024-0194
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202024.0194
Complete List of Authors	Bingzhen Chen and Canyi Chen
Corresponding Authors	Canyi Chen
E-mails	canyic@umich.edu
Notice: Accepted author version.	

Convoluted Support Matrix Machine in High Dimensions

Bingzhen Chen and Canyi Chen

Hangzhou Dianzi University and University of Michigan

Abstract: The Support Vector Machine (SVM) has been effective in various discrimination problems. Recently, there has been growing interest in extending the traditional vector-based SVM to accommodate structured matrix inputs. However, the nonsmooth hinge loss poses significant challenges for both theoretical and computational development. To address these issues, we propose a convex smoothing procedure for the hinge loss. Additionally, we introduce an elastic-net type penalty to handle high-dimensional matrix inputs. Our approach surpasses the standard SVM for discrimination involving high-dimensional matrix inputs. The proposed method provably achieves an optimal statistical convergence rate, and the smooth, convex loss function enables the development of a highly efficient optimization algorithm. This algorithm features a fast linear convergence rate and a simple implementation. Extensive simulations and an electroencephalography application demonstrate the method's superiority in classification accuracy and computational efficiency.

Key words and phrases: Linear support vector machines, asymptotic theory, convolution-type smoothing, high-dimensional matrix regression.

1. Introduction

Since its inception by Boser et al. (1992) and Vapnik (2000), the Support Vector Machine (SVM) has become a key tool for discrimination problems in a broad range of applications, including pattern recognition, computer vision, and disease diagnosis (Bishop, 2006). The statistical properties of SVM have been well-explored (Steinwart and Scovel, 2007; Eberts and Steinwart, 2013), with recent work by Cui et al. (2022) providing explicit error rates for nonlinear SVM.

Despite its success, applying standard SVM to real-world problems presents significant challenges. The first challenge is that standard SVM is primarily designed for vector inputs, while many practical discrimination tasks involve more structured inputs, such as matrices. In applications like computer vision or medical diagnosis, images are typically represented as matrices where each pixel corresponds to a matrix entry. Reshaping these matrices into vectors can disrupt the inherent structural information and produce very high-dimensional vectors, leading to suboptimal performance.

This issue is exacerbated by the second challenge: high-dimensional data. In the era of big data, rapid technological advances have spawned massive datasets characterized by large sample sizes and high dimensionality. For example, in electroencephalogram (EEG) classification problems, the

dimensions of EEG data are comparable to the sample size. Ignoring structural information can render standard SVM computationally infeasible. In high-dimensional settings, a reasonable assumption to capture the intrinsic dimension is the low rankness of the true parameter matrix, commonly imposed in the literature (Zhou and Li, 2014). To address this issue, Luo et al. (2015); Zheng et al. (2018) proposed the penalized support matrix machine (SMM) to handle matrix inputs and demonstrated performance improvements over the standard SVM. The statistical properties of the penalized SMM are further investigated in Xu et al. (2024).

However, their methods face a third challenge, which is computational efficiency. The *doubly* nonsmooth structure in the nuclear norm penalized SMM can degrade the performance of popular optimization algorithms such as the interior point algorithm (Ferris and Munson, 2002) and the Alternating Direction Method of Multipliers (ADMM) algorithm (Luo et al., 2015), especially when the sample size and dimension are both large. For example, the ADMM algorithm proposed by Xu et al. (2024) demands inverting a $pq \times pq$ matrix in each iteration, where $p \times q$ is the dimension of matrix inputs. From the perspective of iterative complexity, Nesterov (2005); Beck and Teboulle (2012) points out that the non-differentiable loss function can induce higher iterative complexity than a smooth loss function.

This paper aims to enhance the computational efficiency of the penalized SMM for binary classification through a novel convex smoothing and provide the convergence rate under the general random design setting. We make three new contributions to the literature. First, a convolution-type smoothing of the hinge loss is suggested to avoid the complexity associated with the nonsmooth hinge loss. The new loss function is convex and smooth, allowing for the development of an efficient optimization algorithm for solving the penalized SMM. Second, an elastic-net type penalty consisting of the squared Frobenius and nuclear norm terms is suggested for estimating the true low-rank coefficients matrix. The nuclear norm term in the penalty constrains the singular values of the estimated coefficients matrix to zero, thereby inducing a low-rank estimate. The regularized estimate holds the convergence rate $\{(p+q)r/n\}^{1/2}$, where n is the sample size, and r is the rank of the true coefficients matrix. This rate coincides with that of the least squares matrix regression (Fan et al., 2020) and is sharper than that of the nuclear-norm penalized SMM in Xu et al. (2024). Third, we develop an efficient proximal ADMM algorithm for solving the resulting optimization problem. The algorithm converges linearly with a simple implementation.

The theoretical development of the linear SVM has a relatively short history. The standard linear SVM, first proposed by Boser et al. (1992)

and Vapnik (2000), is usually referred to as the ℓ_2-norm SVM because its objective has the form of hinge loss plus an ℓ_2 penalty. Koo et al. (2008) derived a Bahadur representation for the standard linear SVM under fixed dimensions, laying the foundation for many subsequent works. In the high-dimensional regime,Wang et al. (2006); Zhu et al. (2003); Peng et al. (2016) proposed using the ℓ_1-norm penalty (i.e., the lasso) instead of the ℓ_2-norm penalty for classification and variable selection simultaneously. Park et al. (2012) and Peng et al. (2016) further considered the SCAD penalty. However, these works focus solely on SVM with vector inputs, and only some exist for SMM. Xu et al. (2024) and Luo et al. (2015) developed ADMM algorithms for the penalized SMM and derived an explicit error bound. However, because of the nonsmoothness of the hinge loss function, such ADMM algorithms may have low computational efficiency and are not scalable for high-dimensional inputs.

In addition to SVM, various methods exist to address discrimination problems. These methods span from classical parametric approaches, such as Fisher's linear discriminant analysis and logistic regression, to more advanced techniques like distance-weighted discrimination (Marron et al., 2007; Egashira et al., 2021, DWD) and Boosting (Friedman et al., 2000); see also Fan et al. (2020) for a comprehensive review. The literature also delves

into multi-category SVM, exploring different strategies such as the multi-category hinge loss (Doğan et al., 2016; Egashira, 2024), one-versus-one and one-versus-rest methods, as well as error-correcting codes (Wang and Zou, 2019). In cases where the optimal Bayesian discrimination rule is nonlinear, kernel methods are frequently used to map features into high-dimensional or even infinite-dimensional spaces, facilitating nonlinear extensions of these techniques (Vedaldi and Zisserman, 2012). The generic convolution-type smoothing technique can be employed to improve the smoothness of the loss functions associated with these methods, potentially leading to significant computational gains.

To harness the nonsmoothness of the loss function, Horowitz (1998) proposed to smooth the indicator part of the quantile regression (QR) function by the survival function of a kernel. This smoothing technique has been used in various QR-related problems, as seen in Galvao and Kato (2016); Chen et al. (2019). Wang et al. (2019) employed it to tackle the standard SVM. However, this smoothing technique gains smoothness at the expense of convexity, which may lead to optimization issues. To maintain convexity and provide statistical guarantees, Fernandes et al. (2021) proposed a new smoothing technique for solving quantile regression. Tan et al. (2021) further studied the smoothed quantile regression under high-dimensional

settings and showed that the statistical properties of quantile regression are maintained after smoothing. Wang et al. (2022) used this smoothing technique to study the penalized SVM under high-dimensional settings and gave statistical guarantees. Our work focuses on matrix inputs; hence, the technical proofs and the development of the optimization algorithm are different from theirs.

We organize the rest of the paper as follows. In Section 2, we present the penalized smoothed SMM. Section 3 establishes its statistical convergence rate. We devise an efficient proximal ADMM algorithm in Section 4. Extensive simulations and an application to an EEG dataset in Section 5 demonstrate the competitive performance of our method. We conclude the paper with a brief discussion in Section 6. All the technical proofs are relegated to the online Supplementary Material.

We use the following notations in the subsequent illustration. We use the standard asymptotic notation. Denote by $\odot$ the standard Hadamard product. For a vector $\mathbf{v} = (V_1, \ldots, V_p)^\top \in \mathbb{R}^p$ and $b \in \mathbb{N}_+$, we use $|\mathbf{v}|_b \stackrel{\text{def}}{=} (\sum_{j=1}^p V_j^b)^{1/b}$ denote its ℓ_b norm. For any matrix $\mathbf{A} = (A_{jk})$ of size $p \times q$, its nuclear norm is the sum of its singular values $\|\mathbf{A}\|_* \stackrel{\text{def}}{=} \sum_{j=1}^{\min(p,q)} \sigma_j(\mathbf{A})$, where $\sigma_1(\mathbf{A}) \geq \ldots \geq \sigma_{\min(p,q)}(\mathbf{A})$ is the ordered singular values of $\mathbf{A}$. Let $\Lambda_{\min}(\mathbf{A})$ and $\Lambda_{\max}(\mathbf{A})$ be the smallest and largest nonzero singular values of $\mathbf{A}$. In addition, the

Frobenius norm is $\|\mathbf{A}\|_F = (\sum_{j,k} A_{jk}^2)^{1/2} = \{\sum_{j=1}^{\min(p,q)} \sigma_j^2(\mathbf{A})\}^{1/2}$, and the spectral norm is $\|\mathbf{A}\| = \sigma_1(\mathbf{A})$. Let $\mathrm{vec}(\mathbf{A})$ be the vectorized operator that transforms the matrix $\mathbf{A}$ into a column vector by vertically stacking the columns of the matrix. Its inverse operation $\mathrm{reshape}(\mathbf{v}, p, q)$ transforms a vector $\mathbf{v} \in \mathbb{R}^{pq}$ into a matrix in $\mathbb{R}^{p\times q}$ columnwisely. We use $\mathrm{diag}(\mathbf{x})$ to denote the diagonal matrix with the vector $\mathbf{x}$ as its main diagonal entries.

2. Low-Rank Convoluted Support Matrix Machine

2.1 Penalized support matrix machine

This subsection introduces the penalized SMM as a method for two-class discrimination problems. Let $\{(Y_i, \mathbf{X}_i)\}_{i=1}^n$ be the independent and identically distributed (i.i.d.) data sampled from some distribution $\mathcal{D}(Y, \mathbf{X})$, where $Y \in \{-1, 1\}$ is the output label and $\mathbf{X} \in \mathbb{R}^{p\times q}$ is the input matrix. To handle the matrix inputs, one may consider extending the conventional SVM (Boser et al., 1992; Vapnik, 2000) to find a hyperplane that separates the two classes of data points via optimizing

$$\min_{a\in\mathbb{R}^1, \mathbf{A}\in\mathbb{R}^{p\times q}} \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}[Y_i\{\mathrm{tr}(\mathbf{X}_i^\top \mathbf{A}) + a\}] + \lambda_0\|\mathbf{A}\|_F^2,$$

where $\mathcal{L}(u) = (1-u)_+ = \max(1-u, 0)$ is the usual hinge loss, and $\lambda > 0$ is a tuning parameter. By the fact that $\mathrm{tr}(\mathbf{X}_i^\top \mathbf{A}) = \{\mathrm{vec}(\mathbf{X}_i)\}^\top \mathrm{vec}(\mathbf{A})$ and $\|\mathbf{A}\|_F = |\mathrm{vec}(\mathbf{A})|_2$, the above formulation is indeed equivalent to the conventional SVM formulation and does not consider the structure of the coefficients matrix. However, the true coefficients matrix happens to be low-rank in many applications, for example, phase retrieval, EEG data analysis, and 1-bit matrix completion. To fully exploit the intrinsic low-rankness, Luo et al. (2015) suggested a new method called *penalized support matrix machine*, which finds such a hyperplane by optimizing

$$\min_{a\in\mathbb{R}^1, \mathbf{A}\in\mathbb{R}^{p\times q}} \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}[Y_i\{\mathrm{tr}(\mathbf{X}_i^\top \mathbf{A}) + a\}] + \lambda_0\|\mathbf{A}\|_F^2 + \lambda\|\mathbf{A}\|_*, \tag{2.1}$$

where $\lambda_0 > 0$ and $\lambda > 0$ are tuning parameters.

The nuclear norm is the best convex approximation of $\mathrm{rank}(\mathbf{A})$ over the unit ball of matrices, which favors the optimization and analysis of statistical properties (Zhou and Li, 2014). When $\lambda_0 = 0$, Xu et al. (2024) thoroughly investigate the statistical properties of problem (2.1). The loss+penalty formulation of SVM has been widely adopted for studying the theoretical properties of SVM, with examples including the ℓ_1 penalty in Zhu et al. (2003); Peng et al. (2016), the SCAD penalty in Park et al. (2012); Peng

et al. (2016) and the nuclear norm penalty in Xu et al. (2024).

However, the nonsmooth hinge loss $\mathcal{L}(\cdot)$ obstacles the theoretical and computational development. The statistical properties of (2.1) remain largely vague in the literature. Due to the non-differentiability of the hinge loss function, existing optimization algorithms, for example, the ADMM algorithms in Xu et al. (2024), and interior point algorithm (Ferris and Munson, 2002), may not scale well with high-dimensional inputs. Theoretically, a major obstacle is that the hinge loss is piecewise linear, so its "curvature energy" is concentrated in a single point. This contrasts with many popular loss functions considered in the statistical literature, such as the square, logistic, or Huber loss, which are simultaneously convex and smooth. Therefore, a proper smoothing procedure that yields smoothness and convexity is essential for the success of the proposed framework.

2.2 Convolution-type smoothing procedure

We then suggest a convolution-type smoothing procedure for the hinge loss function. The resulting new loss features convexity and smoothness that greatly facilitate theoretical and computational advances. To begin with, we define a new random variable $U = Y\{\mathrm{tr}(\mathbf{X}^\top\mathbf{A}) + a\}$ and let $F(u; a, \mathbf{A})$ be its cumulative distribution function (cdf). We can then write the population

version of (2.1) as

$$E(\mathcal{L}[Y\{\mathrm{tr}(\mathbf{X}^\top\mathbf{A})+a\}]) = \int_{-\infty}^{\infty}\mathcal{L}(t)\mathrm{d}F(t;a,\mathbf{A}).$$

If the cdf $F(\cdot;a,\mathbf{A})$ is smooth enough, one may expect that $E(\mathcal{L}[Y\{\mathrm{tr}(\mathbf{X}^\top\mathbf{A})+a\}])$ is at least twice differentiable and convex. For each $(a,\mathbf{A})\in\mathbb{R}^1\times\mathbb{R}^{p\times q}$, let $\widehat{F}(t;a,\mathbf{A}) = 1/n\sum_{i=1}^n I[Y_i\{\mathrm{tr}(\mathbf{X}_i^\top\mathbf{A})+a\}\le t]$ be the empirical cdf based on i.i.d. realization of U, where $I(\cdot)$ is the indicator function. Then we can express the unpenalized objective function in (2.1) as

$$\int_{-\infty}^{\infty}\mathcal{L}(t)\mathrm{d}\widehat{F}(t;a,\mathbf{A}).$$

Unfortunately, the empirical cdf is discontinuous, which makes the loss in (2.1) hold the same degree of smoothness as the hinge loss $\mathcal{L}(\cdot)$. This observation motivates us to use an alternative smooth estimate $\widetilde{F}(\cdot;a,\mathbf{A})$ for the cdf.

In particular, we suggest using the Rosenblatt-Parzen kernel density estimate for the cdf

$$\widetilde{F}(t;a,\mathbf{A}) = \int_{-\infty}^{t}\frac{1}{nh}\sum_{i=1}^{n}K\left[\frac{u-Y_i\{\mathrm{tr}(\mathbf{X}_i^\top\mathbf{A})+a\}}{h}\right]\mathrm{d}u,$$

where $K: \mathbb{R} \to [0, \infty)$ is a smooth kernel function fulfilling $K(-u) = K(u)$, $\forall u \in \mathbb{R}$, $\int_{-\infty}^{\infty} K(u)\mathrm{d}u = 1$ and $\int_{-\infty}^{\infty} |u|K(u)\mathrm{d}u < \infty$, and $h > 0$ is the bandwidth. Replacing $\widehat{F}$ with $\widetilde{F}$ leads to a new loss function

$$\begin{aligned}\int_{-\infty}^{\infty} \mathcal{L}(t)\mathrm{d}\widetilde{F}(t; a, \mathbf{A}) &= \frac{1}{n}\sum_{i=1}^{n}\int_{-\infty}^{\infty}\mathcal{L}(t)\frac{1}{h}\sum_{i=1}^{n}K\left[\frac{t - Y_i\{\mathrm{tr}(\mathbf{X}_i^\top\mathbf{A}) + a\}}{h}\right]\mathrm{d}t \\ &\stackrel{\mathrm{def}}{=} \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_h[Y\{\mathrm{tr}(\mathbf{X}^\top\mathbf{A}) + a\}],\end{aligned}$$

where $\mathcal{L}_h(t) = \int_{-\infty}^{\infty}(1-u)_+h^{-1}K\{(u-t)/h\}\mathrm{d}u$. With our construction, the new smoothed hinge loss function $\mathcal{L}_h(\cdot)$ is a convex and smooth function. Also, it satisfies the relation $\mathcal{L}_h = \mathcal{L} * K_h$ where $K_h(u) = h^{-1}K(u/h)$ and the operator "$*$" stands for convolution.

We list the new smoothed hinge loss function with several commonly used kernel functions.

(i) (Uniform kernel) For the uniform kernel $K(u) = 1/2I(|u| \le 1)$, which is the density function of the uniform distribution on $[-1, 1]$, the resulting smoothed hinge loss is $\mathcal{L}_h^U(v) = (1-v)I(v \le 1-h) + (1+h-v)^2/(4h)I(1-h < v \le 1+h)$.

(ii) (Laplacian kernel) For the Laplacian kernel $K(u) = \exp(-|u|)/2$, we have $\mathcal{L}_h^L(v) = [1 + h/2\exp\{(v-1)/h\} - v]I(v < 1) + h/2\exp\{(1-v)/h\}I(v \ge 1)$.

(iii) (Logistic kernel) For the logistic kernel $K(u) = \exp(-u)/\{1+\exp(-u)\}^2$, the resulting smoothed hinge loss is $\mathcal{L}_h^{Logit} = -v + h\log\{\exp(1/h) + \exp(v/h)\}$.

(iv) (Gaussian kernel) For the Gaussian kernel $K(u) = (2\pi)^{-1/2}\exp(-u^2/2)$, the resulting smoothed hinge loss is $\mathcal{L}_h^G(v) = (1-v)\Phi\{(1-v)/h\} + h(2\pi)^{-1/2}\exp\left\{-(1-v)^2/(2h^2)\right\}$, where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

(v) (Epanechnikov kernel) For the Epanechnikov kernel $K(u) = 3/4 \cdot (1-u^2)I(-1 \le u \le 1)$, the resulting smoothed hinge loss is $\mathcal{L}_h^E(v) = (1-v)I(v \le 1-h) + [(1-v+h)^3\{3h-(1-v)\}/(16h^3)]I(1-h < v \le 1+h)$.

While our novel classifier is developed from a statistical perspective, its computational value is noteworthy because it effectively addresses the nonsmoothness inherent in the original hinge loss. The technique of smoothing nonsmooth problems can be traced back to the concept of mollification, as described in Friedrichs (1944), and has been extensively explored in optimization literature (e.g., Rubinstein (1983)). Recently, this method has garnered increasing attention from the statistical community (He et al., 2021; Fernandes et al., 2021; Tan et al., 2021; Wang et al., 2022). Rosset and Zhu (2007) and Wang et al. (2008) introduced a Huberized smoothing approxima-

tion for the hinge loss, aimed at computing an elastic-net penalized support vector machine via a smoothed optimization method. The proposed loss function belongs to the general framework of convolution-type smoothing, with K as the general uniform kernel. The theoretical framework developed in this work applies to the specific case of the proposed loss function.

Replacing the hinge loss in (2.1) with our new smoothed hinge loss gives

$$(\widehat{a}, \widehat{\mathbf{A}}) = \underset{(a,\mathbf{A})\in\mathbb{R}^1\times\mathbb{R}^{p\times q}}{\arg\min} \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_h\big[Y_i\{\mathrm{tr}(\mathbf{X}_i^\top \mathbf{A}) + a\}\big] + \lambda_0\|\mathbf{A}\|_F^2 + \lambda\|\mathbf{A}\|_*. \quad (2.2)$$

When $\lambda_0 = 0$, the problem above reduces to

$$(\breve{a}, \breve{\mathbf{A}}) = \underset{(a,\mathbf{A})\in\mathbb{R}^1\times\mathbb{R}^{p\times q}}{\arg\min} \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_h\big[Y_i\{\mathrm{tr}(\mathbf{X}_i^\top \mathbf{A}) + a\}\big] + \lambda\|\mathbf{A}\|_*. \quad (2.3)$$

We regard the classifiers from the above problems as new classifiers. By the convention in high-dimensional statistics, we refer to the estimate in (2.2) as elastic-net convoluted SMM and refer to the estimate in (2.3) as low-rank convoluted SMM. The bandwidth h is used for indexing the new classifier.

In this work, we mainly focus on the Gaussian kernel and the Epanechnikov kernel in convoluted SMM. Intuitively, h should be small such that the convoluted SMM is very close to the SMM. According to the kernel density estimate theory, the optimal rate for h is $O(n^{-1/5})$. So, we adopt $h = Cn^{-1/5}$

in our implementation, where C is some numerical constant within the range $[0.25, 3]$.

3. Statistical Theory

Let $(a^*, \mathbf{A}^*) \stackrel{\text{def}}{=} \arg\min_{a,\mathbf{A}} E(\mathcal{L}_h[Y\{\text{tr}(\mathbf{X}^\top\mathbf{A}) + a\}])$. In this work, we aim to obtain a low-rank estimate assuming that $\mathbf{A}^*$ is truly low-rank. Let r be the rank of $\mathbf{A}^*$. We allow $p = p_n$, $q = q_n$ and $r = r_n$ to diverge with n, and we assume $r_n \geq 1$ and $\min(p_n, q_n)$ goes to infinity as n goes to infinity. For convenience, we still use p, q, and r when no confusion is caused.

We list the assumptions required in our theoretical development. Let $\mathbf{x}_i \stackrel{\text{def}}{=} \text{vec}(\mathbf{X}_i)$. The following assumption concerns the random design.

(A1) The zero-mean predictor $\mathbf{x}_i$ is sub-exponential, i.e., for some $m_0 > 0$,

$$\sup_{\mathbf{a}\in\mathbb{R}^{pq}:\,|\mathbf{a}|_2\leq 1} \text{pr}(|\mathbf{a}^\top\mathbf{x}_i| > t) \leq 2\exp(-t/m_0) \text{ for any } t \geq 0.$$

Before making further assumptions, we need additional notations and definitions. Suppose the truncated singular value decomposition (SVD) of $\mathbf{A}^*$ is $\mathbf{A}^* = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ with $\mathbf{U} \in \mathbb{R}^{p\times r}$ and $\mathbf{V} \in \mathbb{R}^{q\times r}$ being orthonormal matrices. Let $\mathcal{M} \stackrel{\text{def}}{=} \{\mathbf{A} \in \mathbb{R}^{p\times q} : \text{row}(\mathbf{A}) \subseteq \mathbf{V}, \text{col}(\mathbf{A}) \subseteq \mathbf{U}\}$ and $\mathcal{N} \stackrel{\text{def}}{=} \{\mathbf{A} \in \mathbb{R}^{p\times q} : \text{row}(\mathbf{A}) \perp \mathbf{V}, \text{col}(\mathbf{A}) \perp \mathbf{U}\}$. Let $\Pi_{\mathcal{N}} : \mathbb{R}^{p\times q} \to \mathbb{R}^{p\times q}$ be the projection of a $p \times q$ matrix onto $\mathcal{N}$ under norm $\|\cdot\|_F$. For any matrix $\mathbf{\Delta} \in \mathbb{R}^{p\times q}$,

let $\mathbf{\Delta}_{r^c} \overset{\text{def}}{=} \mathbf{\Pi}_{\mathcal{N}}\mathbf{\Delta}$ and $\mathbf{\Delta}_r = \mathbf{\Delta} - \mathbf{\Delta}_{r^c}$. We define the restricted cone set $\mathcal{A} \overset{\text{def}}{=} \{(\delta, \mathbf{\Delta}) \in \mathbb{R} \times \mathbb{R}^{p\times q} : \|\mathbf{\Delta}_{r^c}\|_* \le 3\|\mathbf{\Delta}_r\|_* + |\delta|\}$. Such cone sets have been widely considered in the literature on high-dimensional statistics; see Fan et al. (2020). Define $\widetilde{\mathbf{x}}_i \overset{\text{def}}{=} (1, \mathbf{x}_i^\top)^\top$, and let $\mathbf{I}(a, \mathbf{A}) \overset{\text{def}}{=} E\big[\mathcal{L}_h''\{Y(\mathrm{tr}(\mathbf{X}^\top\mathbf{A}) + a)\}\widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^\top\big]$ be the Hessian/information matrix of the population loss.

(A2) There exists a constant $\kappa > 0$ such that for sufficiently large n,

$$\min_{(\delta,\mathbf{\Delta})\in\mathcal{A}:\ \delta^2+\|\mathbf{\Delta}\|_F^2=O\{(p+q)r/n\}} \Lambda_{\min}\{\mathbf{I}(a^* + \delta, \mathbf{A}^* + \mathbf{\Delta})\} \ge \kappa.$$

Assumption (A1) concerns the distribution of the predictors, which relaxes the classical condition that the components of $\mathbf{X}$ are bounded random variables and $\mathbf{x}$ is sub-Gaussian (Xu et al., 2024). Assumption (A2) is a *locally* restricted eigenvalue (RE) type condition, which is commonly adopted in the high-dimensional statistics literature (Fan et al., 2020).

Theorem 1. Suppose Assumptions (A1) and (A2) hold and $(p+q)r/n = o(1)$. Choose the tuning parameters such that $8\lambda_0\|\mathbf{A}^*\| \le \lambda$. Then, there exists a sufficiently large constant $c_0 > 0$ such that with the choice $\lambda = c_0\{(p+q)/n\}^{1/2}$, the elastic-net penalized convoluted SMM estimate $(\widehat{a}, \widehat{\mathbf{A}})$ satisfies $|\widehat{a} - a^*|^2 + \|\widehat{\mathbf{A}} - \mathbf{A}^*\|_F^2 = O_p\{(p+q)r/n\}$.

Two remarks in order. First, Theorem 1 implies that the elastic-net

convoluted SMM estimate attains a shaper convergence rate than the nuclear-norm-penalized SMM proposed by Xu et al. (2024). Concurrently, the elastic-net convoluted SMM gains better computational efficiency than the nuclear-norm-penalized SMM thanks to the smoothness and convexity of the new loss function, as we will show in Sections 4 and 5. Second, Theorem 1 shows the advantage of considering the matrix structure. Ignoring the structure of the matrix and simply treating the vectorized version will put one in the face of a pq-dimensional vector. Without any further structure on the resulting vector, the classical convergence rate of the least squares estimate will be $(pq/n)^{1/2}$. To ensure a consistent estimate, such a vectorized operator will hence require $pq = o(n)$, which is a much more stringent condition than ours, $(p+q)r = o(n)$ when r is typically much smaller order of $\min(p, q)$.

As a method utilizing a convex and smooth loss function, the DWD is arguably an improved classifier compared to SVM. SVM may experience reduced generalizability in high-dimensional, low-sample-size (HDLSS) data contexts, as highlighted by Marron et al. (2007), largely due to the data-piling phenomenon. This issue arises when support vectors accumulate along the boundaries of the margin when projected onto the normal vector of the separating hyperplane. Consequently, minor noise artifacts within the specific data realization can degrade classification performance. To address

this, the DWD method proposed by Marron et al. (2007) mitigates this limitation by building the separating hyperplane with all data vectors, which by contrast, are not always used in SVM. However, Egashira et al. (2021) found that DWD may encounter significant bias under data heterogeneity (imbalanced sample sizes or heterogeneous covariances). In response, Egashira et al. (2021) introduced a novel bias correction method to ensure classification consistency under data heterogeneity. Similar challenges have been identified and addressed in the context of classical SVMs (Egashira, 2024). This current study explores a different high-dimensional setting where the dimensions p and q approach infinity at a rate of $(p+q)r/n = o(1)$, with a particular focus on addressing the challenges associated with the nonsmooth hinge loss used in penalized SMMs. We hypothesize that the proposed method may also encounter bias in the HDLSS scenario. Future research could explore bias correction techniques or weighted approaches to manage data heterogeneity.

4. Optimization Method

In this section, we develop an efficient proximal ADMM algorithm for solving problems (2.3) and (2.2), which holds a fast linear convergence rate. Because problem (2.3) is a special case of problem (2.2), we focus on elaborating on

the algorithm for solving problem (2.2).

4.1 Proximal ADMM

For ease of the presentation, let $\mathbb{X} \overset{\text{def}}{=} (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{pq\times n}$, $\mathbf{y} \overset{\text{def}}{=} (Y_1, \ldots, Y_n)^\top \in \mathbb{R}^n$, $r_i = Y_i\{\mathrm{tr}(\mathbf{X}_i^\top \mathbf{A}) + a\}$, $\mathbf{r} \overset{\text{def}}{=} (r_1, \ldots, r_n)^\top$ and $f(\mathbf{r}) \overset{\text{def}}{=} n^{-1}\sum_{i=1}^n \mathcal{L}_h(r_i)$. By convexity, problem (2.2) is equivalent to a constrained optimization problem:

$$\min_{a\in\mathbb{R}^1, \mathbf{A}\in\mathbb{R}^{p\times q}, \mathbf{r}\in\mathbb{R}^n} f(\mathbf{r}) + \lambda_0\|\mathbf{A}\|_F^2 + \lambda\|\mathbf{A}\|_*, \quad \text{s.t.} \quad \mathbf{r} = \mathbf{y}\odot\big(\mathbb{X}^\top\mathrm{vec}(\mathbf{A}) + a\mathbf{1}_n\big).$$

Following the classic ADMM theory (Boyd, 2010), we can construct its augmented Lagrangian with a penalty parameter $\tau > 0$,

$$\mathcal{L}_\tau(a, \mathbf{A}, \mathbf{r}, \mathbf{u}) \overset{\text{def}}{=} f(\mathbf{r}) + \lambda_0\|\mathbf{A}\|_F^2 + \lambda\|\mathbf{A}\|_* - \big\langle \mathbf{u}, \mathbf{r} - \mathbf{y}\odot\big(\mathbb{X}^\top\mathrm{vec}(\mathbf{A}) + a\mathbf{1}_n\big)\big\rangle + \frac{\tau}{2}\big|\mathbf{r} - \mathbf{y}\odot\big(\mathbb{X}^\top\mathrm{vec}(\mathbf{A}) + a\mathbf{1}_n\big)\big|_2^2,$$

where $\mathbf{u} \in \mathbb{R}^n$ is the Lagrangian multiplier. For $k \geq 1$, let a_{k-1}, $\mathbf{A}_{k-1}$, $\mathbf{r}_{k-1}$ and $\mathbf{u}_{k-1}$ be the iterate after the $(k-1)$th iteration of the algorithm, and denote by $\mathbf{a}_k = \mathrm{vec}(\mathbf{A}_k)$. Then, updates in the kth iteration of the ADMM

for solving (2.2) are,

$$\begin{aligned}
\mathbf{A}_k &= \underset{\mathbf{A}\in\mathbb{R}^{p\times q}}{\arg\min}\ \lambda_0\|\mathbf{A}\|_F^2+\lambda\|\mathbf{A}\|_*+\left\langle \mathbf{u}_{k-1}, \mathbf{y}\odot\left\{\mathbb{X}^\top \mathrm{vec}(\mathbf{A})\right\}\right\rangle \\
&\quad +\frac{\tau}{2}\left|\mathbf{r}_{k-1}-\mathbf{y}\odot\left\{\mathbb{X}^\top\mathrm{vec}(\mathbf{A})+a_{k-1}\mathbf{1}_n\right\}\right|_2^2, \qquad (4.4a)\\
a_k &= \left[\mathbf{y}^\top(\mathbf{r}_{k-1}-\mathbf{u}_{k-1}/\tau)-\mathbf{a}_k^\top\mathbb{X}\mathbf{1}_n\right]/n,\\
\mathbf{r}_k &= \underset{\mathbf{r}\in\mathbb{R}^n}{\arg\min}\ f(\mathbf{r})-\langle \mathbf{u}_{k-1},\mathbf{r}\rangle+\frac{\tau}{2}\left|\mathbf{r}-\mathbf{y}\odot\left(\mathbb{X}^\top\mathbf{a}_k+a_k\mathbf{1}_n\right)\right|_2^2, \qquad (4.4b)\\
\mathbf{u}_k &= \mathbf{u}_{k-1}-\tau\left[\mathbf{r}_k-\mathbf{y}\odot\left(\mathbb{X}^\top\mathbf{a}_k+a_k\mathbf{1}_n\right)\right].
\end{aligned}$$

We directly give formulation for updates (4.4a) and (4.4b) with detailed derivations relegated to Section D of the Supplementary Material. Let $\mathbf{G}_{k-1}\overset{\text{def}}{=}\mathrm{reshape}\big(\mathbb{X}[\mathbf{y}\odot(\mathbf{r}_{k-1}-\mathbf{u}_{k-1}/\tau)-a_{k-1}\mathbf{1}_n]+\mathbf{S}\mathbf{a}_{k-1},p,q\big)/\eta$ with singular value decomposition being $\mathbf{G}_{k-1}=\mathbf{U}\mathrm{diag}(\boldsymbol{\sigma})\mathbf{V}^\top$, where $\mathbf{S}=\eta\mathbf{I}_{pq}-2\lambda_0/\tau\mathbf{I}_{pq}-\mathbb{X}\mathbb{X}^\top$, and $\eta\geq 2\lambda_0/\tau+\Lambda_{\max}\big(\mathbb{X}\mathbb{X}^\top\big)$. Write $\boldsymbol{\sigma}=(\sigma_1,\ldots,\sigma_{\min(p,q)})^\top$. By taking quadratic majorization with a proximal term, $\mathbf{A}_k$ in (4.4a) can be approximately updated as $\mathbf{A}_k=\mathbf{P}_\tau(\mathbf{G}_{k-1},\lambda/(\tau\eta))\overset{\text{def}}{=}\mathbf{U}\mathrm{diag}(\mathbf{g})\mathbf{V}^\top$, where $\mathbf{g}=(g_1,\ldots,g_{\min(p,q)})^\top$ and $g_i=\max\{\sigma_i-\lambda/(\tau\eta),0\}$. We suggest taking Newton–Raphson iterations for solving (4.4b), which features fast convergence starting from a good initial point. The diagonal structure of the Hessian matrix in (4.4b) enables the efficient Newton–Raphson iterations.

We summarize the above proximal ADMM algorithm for solving (2.2)

in Algorithm 1. Theorem 2 demonstrates its linear convergence rate.

Algorithm 1 Proximal ADMM for solving penalized convoluted support matrix machine

Input: Data $(\mathbb{X}, \mathbf{y})$, tolerance tol > 0, maximum number of iterations $M \in \mathbb{N}_+$, $\lambda_0 \geq 0$, $\lambda > 0$, $\tau > 0$, $\eta \geq 2\lambda_0/\tau + \Lambda_{\max}(\mathbb{X}\mathbb{X}^\top)$, initials a_0, $\mathbf{A}_0$, $\mathbf{r}_0$, $\mathbf{u}_0$.
for $k = 1, 2, \ldots, M$ **do**
  Compute $\mathbf{A}_k = \mathbf{P}_\tau(\mathbf{G}_{k-1}, \lambda/(\tau\eta))$, where
  $\mathbf{G}_{k-1} = \text{reshape}\big(\mathbb{X}[\mathbf{y} \odot (\mathbf{r}_{k-1} - \mathbf{u}_{k-1}/\tau) - a_{k-1}\mathbf{1}_n] + \mathbf{S}\mathbf{a}_{k-1}, p, q\big)/\eta$;
  Compute $a_k = \big[\mathbf{y}^\top(\mathbf{r}_{k-1} - \mathbf{u}_{k-1}/\tau) - \mathbf{a}_k^\top\mathbb{X}\mathbf{1}_n\big]/n$;
  Compute $\mathbf{r}_k = \underset{\mathbf{r}\in\mathbb{R}^n}{\arg\min}\ f(\mathbf{r}) - \langle \mathbf{u}_{k-1}, \mathbf{r}\rangle + \tau/2\big|\mathbf{r} - \mathbf{y} \odot (\mathbb{X}^\top\mathbf{a}_k + a_k\mathbf{1}_n)\big|_2^2$;
  Compute $\mathbf{u}_k = \mathbf{u}_{k-1} - \tau\big[\mathbf{r}_k - \mathbf{y} \odot (\mathbb{X}^\top\mathbf{a}_k + a_k\mathbf{1}_n)\big]$;
  if the stopping criterion in (E.7) of the Supplementary Material is satisfied or $k = M$ **then**
    break
  end if
end for

Theorem 2. The sequence of iterates $\{(a_k, \text{vec}(\mathbf{A}_k)^\top)^\top\}$ generated by Algorithm 1 converges linearly to an optimal solution of (2.2), $\{(\widehat{a}, \text{vec}(\widehat{\mathbf{A}})^\top)^\top\}$, provided the penalty parameter τ is sufficiently small.

Thanks to the superior linear convergence rate of Algorithm 1, we can show that, to achieve the desired prefixed precision $\epsilon > 0$, the required computational complexity is of order $O\{npq\log(1/\epsilon)\}$ for matrix inputs and reduces to $O\{np\log(1/\epsilon)\}$ for vector inputs with $q = 1$. Accordingly, for vector inputs, our proposed Algorithm 1 demonstrates superior computational efficiency compared to the DWD method, which also utilizes a smooth and convex loss function but is typically solved via second-order cone

programming (SOCP). The current best-known computational complexity of SOCP is $O\{(n+p)^{2.37}\log(1/\epsilon)\}$, with additional polynomial terms involving $\log\{(n+p)/\epsilon\}$ (Wei and Ye, 2023). Implementation details, including the stopping rule, the choice of the penalty parameter τ, and the computational complexity analysis, are relegated to Section E of the Supplementary Material.

5. Numerical Studies

In this section, we present a thorough empirical evaluation of our proposed method through both simulated and real-world data analysis. The code utilized in the experiments was written in MATLAB (Version R2019a) and executed on a desktop computer equipped with an Intel® Core™ i5-12500 (3.00 GHz) CPU and with 16 GB of RAM.

5.1 Simulations

The data generation process is as follows. We generate the labels $\{Y_i\}_{i=1}^n$ from a Bernoulli distribution with $\text{pr}(Y_i = 1) = \text{pr}(Y_i = -1) = 0.5$. When $Y_i = 1$, the entries of the covariate matrix $\mathbf{X}_i \in \mathbb{R}^{p\times q}$ are independently generated from a normal distribution with mean matrix $\mathbf{U} = (u_{jj'})_{p\times q}$ and variance 1. On the other hand, when $Y_i = -1$, entries of $\mathbf{X}_i$ are independently

drawn from a normal distribution with mean $d\mathbf{U}$ and variance 1, where d is a scalar. We consider two choices of $\mathbf{U}$. In Example 1, we set the entries of matrix $\mathbf{U}$ as $u_{1j'} = 0.1j'$ for $1 \le j' \le 5$, $u_{2j'} = 0.2j' - 0.1$ for $1 \le j' \le 5$ and $u_{jj'} = 0$ otherwise. In Example 2, we set $u_{jj} = 0.1j$ and $u_{j(j+1)} = 0.2j - 0.1$ for $1 \le j \le 5$ and $u_{jj'} = 0$ otherwise. The ranks of $\mathbf{U}$ equal to 2 and 5 in Examples 1 & 2, respectively. For our simulations, we consider $d \in \{-1.5, -1, -0.5\}$. A smaller d leads to a larger discrepancy between the two classes.

We consider the following classifiers. (i) LRCSMM-G: low-rank CSMM with Gaussian kernel; LRCSMM-E: low-rank CSMM with Epanechnikov kernel; EnetCSMM-G: elastic-net CSMM with Gaussian kernel; EnetCSMM-E: elastic-net CSMM with Epanechnikov kernel. (ii) EnetSMM-H: elastic-net SMM with hinge loss and elastic-net penalty proposed in Goldstein et al. (2014). (iii) LRSMM-H: low-rank SMM with hinge loss and nuclear-norm penalty proposed in Xu et al. (2024). (iv) RMV-logistic: matrix variate logistic regression with ridge penalty studied in Hung et al. (2013).

For a fair comparison, we select the best tuning parameters for all the classifiers by validation. Take our proposal for example. We choose the best tuple of (h, λ_0, λ) that minimizes the prediction error on an independently generated validation set of size n. We vary h in $\{0.25, 0.26, 0.27, 0.28, 0.29, 0.3\} \times$

$n^{-1/5}$ and λ_0 in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$. Theorem 1 implies that the theoretically optimal tuning parameter λ is given by $c_0\{(p+q)/n\}^{1/2}$ for an unknown $c_0 > 0$. This guides us vary the constant c_0 within a a user-defined interval $[\lambda_{\min}, \lambda_{\max}]$. Specifically, we construct a sequence of candidate values for λ as $\{\{(p+q)/n\}^{1/2}\lambda_{\max}^{(100-k)/99}\lambda_{\min}^{(k-1)/99} \mid k = 1, \cdots, 100\}$, which are spaced evenly on a logarithmic scale over the range $[\lambda_{\min}, \lambda_{\max}]$. In our numerical studies, we set $\lambda_{\max} = 50$ and $\lambda_{\min} = \lambda_{\max}/10$, which demonstrated competitive performance. Define $\widetilde{a} \stackrel{\text{def}}{=} \arg\min_{a\in\mathbb{R}^1} \frac{1}{n}\sum_{i=1}^n \mathcal{L}_h(aY_i)$. Another common data-driven strategy involves setting $\lambda_{\max} = \sigma_1(\sum_{i=1}^n \nabla_r \mathcal{L}_h(\widetilde{a}Y_i)Y_i\mathbf{X}_i)/n$, for which the corresponding solution of $\mathbf{A}$ in (2.2) is exactly zero by the Karush–Kuhn–Tucker condition.

Two metrics are used for measuring the performance of different classifiers. The first is the prediction error on an independently generated test set of size n. The second is the runtime (in seconds) an algorithm requires to optimize with fixed parameters.

Section G of the Supplementary Material studies the impact of different kernels and penalties on our proposed CSMM. In view of Section G of the Supplementary Material, we can see that the CSMM with Gaussian kernel performs slightly better than CSMM with Epanechnikov kernel. Thus, we only consider CSMM with Gaussian kernel in the following numerical

studies. We set $p = 50$, $q = 100$, and $n = 500$. The numerical results are shown in Table 1. We make the following four observations. First, both LRCSMM-G and EnetCSMM-G attain the best prediction error in almost all cases. Second, although having the least runtimes, RMV-logistic delivers the worst prediction error among all classifiers in almost all cases. This is because RMV-logistic did not learn the intrinsic low-rank structure in a data-driven manner. Third, LRSMM-H, the closest rival in terms of prediction error, has a much longer runtime than our proposals. This is mainly because a computationally extensive matrix inversion in $\mathbb{R}^{pq\times pq}$ is needed in each iteration of the ADMM algorithm used for estimating LRSMM-H proposed by Xu et al. (2024). Fourth, although the runtime of EnetSMM-H is comparable to our proposals, its prediction performance is not satisfactory. Summarizing the above observations, we can conclude that our proposals are competitive regarding both the prediction error and runtime.

5.2 EEG alcoholism data analysis

In this section, we use CSMM with Gaussian kernel to analyze the electroencephalograph (EEG) alcoholism dataset. The dataset is available in UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml/`

Table 1: The prediction error (in percentages) and runtime (in seconds) of low-rank, elastic-net convoluted SMM with Gaussian kernel, low-rank SMM with hinge loss, elastic-net SMM with hinge loss, and ridge matrix variate logistic regression. Under each simulation setting, the method with the lowest prediction error is marked by a black box. All the results are averaged over 50 independent runs.

	LRCSMM-G		EnetCSMM-G		EnetSMM-H		LRSMM-H		RMV-logistic	
d	err (%)	time	err (%)	time	err (%)	time	err (%)	time	err (%)	time
Example 1										
-0.5	18.60	1.33	18.60	1.44	27.40	1.59	**17.4**	5.38	20.40	0.44
-1	**8.00**	1.21	**8.00**	1.26	11.40	1.45	8.80	5.15	11.20	0.38
-1.5	**3.20**	1.44	**3.20**	1.73	**3.20**	1.47	3.80	3.57	5.80	0.42
Example 2										
-0.5	**35.40**	1.18	35.60	1.45	42.60	1.46	36.50	5.23	38.80	1.08
-1	23.80	1.29	23.40	1.14	33.80	1.51	**22.60**	5.74	35.60	1.11
-1.5	**14.80**	1.05	**14.80**	1.16	20.00	1.46	23.20	2.25	22.60	1.15

`datasets/EEG+Database`). The dataset consists of 122 individuals, of which 77 individuals belong to the group of alcoholism ($Y_i = 1$), and the remaining individuals are in the control group ($Y_i = -1$). Each individual completed 120 trials under three types of stimuli: single stimulus, two matched stimuli, and two unmatched stimuli. In each trial, 64 channels of electrodes were placed at different locations of the scalp, and the voltage values at 256 time points were recorded, which resulted in a 256×64 covariate matrix. In this study, we focus on the data under the single stimulus and averaged all 120 trials for each individual. The data analysis aims to determine which group an individual belongs to based on its covariate.

According to the ratio of the number of two groups, we randomly select

two-thirds of the data as the training set and the rest as the test set. We use 10-fold cross-validation to determine the tuning parameters. We report the mean prediction error (in percentages), median estimated rank, and runtime (in seconds) among 50 independent runs in Table 2.

There are 320 (256+64) parameters to be estimated in RMV-logistic model but only 122 samples in the EEG alcoholism data. Thus it can not be used to analyze EEG alcoholism data directly. To deal with this issue, Hung et al. (2013) applied the generalized low-rank approximations of matrices to reduce the dimensionality of $\mathbf{X}_i$ by transforming $\mathbf{X}_i$ into $\widetilde{\mathbf{X}}_i = \mathbf{U}^\top \mathbf{X}_i \mathbf{V}$, where $\mathbf{U} \in \mathbb{R}^{p\times p_0}$, $\mathbf{V} \in \mathbb{R}^{q\times q_0}$, with $p_0 < p$, $q_0 < q$ and $p_0 + q_0 < n$. This process can be viewed as denoising the data. Although leading to a lower misclassification error from their numerical results, such a process brings some new troubles. For example, how to guarantee full preservation of all relevant information and determine the values of p_0 and q_0. The primary goal of this data analysis is to compare the CSMM with the classical SMM. Thus, we did not present the numerical result of RMV-logistic model.

From the numerical results in Table 2, it can be seen that the CSMM with Gaussian kernel proposed in this paper performs much better than SMM with hinge loss in terms of median rank and misclassification error. The EnetSMM-H takes much less runtime than our proposals. But its

misclassification error is about 27.59% and higher than our proposals.

Table 2: The prediction error (in percentages), median rank, and runtime (in seconds) of low-rank, elastic-net convoluted SMM with Gaussian kernel, low-rank SMM with hinge loss, and elastic-net SMM with hinge loss. The method with the lowest prediction error is marked by a black box. All the results are averaged over 50 independent runs.

	LRCSMM-G	EnetCSMM-G	EnetSMM-H	LRSMM-H
median rank	**2.00**	**2.00**	9.00	7.00
error (%)	21.95	**21.81**	27.59	28.71
runtime	17.55	17.67	1.41	40.44

6. Conclusion

In this paper, we developed a new classifier called elastic-net convoluted SMM for handling high-dimensional matrix inputs. Such a convolution-type smoothing procedure turns the nonsmooth hinge loss function into a new smooth and convex loss function, which favors statistical and computational advances. Statistically, we have rigorously shown elastic-net CSMM attains a sharper convergence rate $O_p[\{(p+q)r/n\}^{1/2}]$ compared to the classic competitor (Xu et al., 2024). Of note, the theoretical conditions imposed in this paper are more general than those in Xu et al. (2024). From a computational viewpoint, the convexity and smoothness of the new loss enable us to devise an efficient proximal ADMM algorithm for solving the penalized convoluted SMM that is much more scalable to large-scale

datasets. Tensor data are also increasingly common with applications such as color image and social network classification. We give a primary formulation in Section F of the Supplementary Material for extension to tensor inputs. Future work may involve developing efficient and globally convergent algorithms for our tensor extension. Another potential direction for extension is to consider multi-category discrimination problems (Zou et al., 2008; Wang and Zou, 2019) and debiased version in the HDLSS setting (Egashira, 2024). However, the technical details for these extensions need to be carefully worked out, and we leave it to future works.

Acknowledgments

Canyi Chen is the corresponding author. The research of Bingzhen Chen is supported by the Natural Science Foundation of Cangzhou (221001007D) and the Scientific Research Foundation of Hangzhou Dianzi University (KYS155623054).

References

Beck, A. and M. Teboulle (2012). Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization 22*(2), 557–580.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning* (First ed.). Information

Science and Statistics. New York: Springer.

Boser, B. E., I. M. Guyon, and V. N. Vapnik (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory - COLT '92*, Pittsburgh, Pennsylvania, United States, pp. 144–152. ACM Press.

Boyd, S. (2010). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning 3*(1), 1–122.

Chen, X., W. Liu, and Y. Zhang (2019, December). Quantile regression under memory constraint. *Annals of Statistics 47*(6), 3244–3273.

Cui, H., B. Loureiro, F. Krzakala, and L. Zdeborová (2022, November). Generalization error rates in kernel regression: The crossover from the noiseless to noisy regime. *Journal of Statistical Mechanics: Theory and Experiment 2022*(11), 114004.

Doğan, Ü., T. Glasmachers, and C. Igel (2016). A unified view on multi-class support vector classification. *Journal of Machine Learning Research 17*(45), 1–32.

Eberts, M. and I. Steinwart (2013, January). Optimal regression rates for SVMs using Gaussian kernels. *Electronic Journal of Statistics 7*, 1–42.

Egashira, K. (2024). Asymptotic properties of multiclass support vector machine under high dimensional settings. *Communications in Statistics-Simulation and Computation 53*(4), 1991–2005.

Egashira, K., K. Yata, and M. Aoshima (2021). Asymptotic properties of distance-weighted

discrimination and its bias correction for high-dimension, low-sample-size data. *Japanese Journal of Statistics and Data Science 4*(2), 821–840.

Fan, J., R. Li, C.-H. Zhang, and H. Zou (2020). *Statistical Foundation of Data Science* (First ed.). CRC Data Science Series. Boca Raton: CRC Press.

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

Ferris, M. C. and T. S. Munson (2002, January). Interior-Point Methods for Massive Support Vector Machines. *SIAM Journal on Optimization 13*(3), 783–804.

Friedman, J., T. Hastie, and R. Tibshirani (2000). Additive logistic regression: A statistical view of boosting (With discussion and a rejoinder by the authors). *The Annals of Statistics 28*(2), 337–407.

Friedrichs, K. O. (1944). The Identity of Weak and Strong Extensions of Differential Operators. *Transactions of the American Mathematical Society 55*(1), 132–151.

Galvao, A. F. and K. Kato (2016, July). Smoothed quantile regression for panel data. *Journal of Econometrics 193*(1), 92–112.

Goldstein, T., B. O'Donoghue, S. Setzer, and R. Baraniuk (2014). Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences 7*, 1588–1623.

He, X., X. Pan, K. M. Tan, and W.-X. Zhou (2021, August). Smoothed quantile regression with large-scale inference. *Journal of Econometrics*, S0304407621001950.

Horowitz, J. L. (1998). Bootstrap Methods for Median Regression Models. *Econometrica 66*(6), 1327.

Hung, Hung, Chen-Chien, and Wang (2013). Matrix variate logistic regression model with application to eeg data. *Biostatistics 14*(1), 189–202.

Koo, J.-Y., Y. Lee, Y. Kim, and C. Park (2008, July). A Bahadur Representation of the Linear Support Vector Machine. *Journal of Machine Learning Research 9*, 1343–1368.

Luo, L., Y. Xie, Z. Zhang, and W.-J. Li (2015, June). Support Matrix Machines. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 938–947. PMLR.

Marron, J. S., M. J. Todd, and J. Ahn (2007). Distance-weighted discrimination. *Journal of the American Statistical Association 102*(480), 1267–1271.

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Mathematical programming 103*, 127–152.

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*(8), 2257–2270.

Peng, B., L. Wang, and Y. Wu (2016). An error bound for l1-norm support vector machine coefficients in ultra-high dimension. *Journal of Machine Learning Research 17*(233), 1–26.

Rosset, S. and J. Zhu (2007, July). Piecewise linear regularized solution paths. *The Annals of Statistics 35*(3), 1012–1030.

Rubinstein, R. Y. (1983, February). Smoothed Functionals in Stochastic Optimization. *Mathe-*

matics of Operations Research 8(1), 26–33.

Steinwart, I. and C. Scovel (2007, April). Fast rates for support vector machines using Gaussian kernels. *The Annals of Statistics 35*(2), 575–607.

Tan, K. M., L. Wang, and W.-X. Zhou (2021, September). High-Dimensional Quantile Regression: Convolution Smoothing and Concave Regularization.

Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory* (Second ed.). New York, NY: Springer New York.

Vedaldi, A. and A. Zisserman (2012, March). Efficient Additive Kernels via Explicit Feature Maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*(3), 480–492.

Wang, B., L. Zhou, Y. Gu, and H. Zou (2022). Density-Convoluted Support Vector Machines for High-Dimensional Classification. *IEEE Transactions on Information Theory 69*(4), 2523–2536.

Wang, B. and H. Zou (2019, July). A Multicategory Kernel Distance Weighted Discrimination Method for Multiclass Classification. *Technometrics 61*(3), 396–408.

Wang, L., J. Zhu, and H. Zou (2006). The Doubly Regularized Support Vector Machine. *Statistica Sinica 16*(2), 589–615.

Wang, L., J. Zhu, and H. Zou (2008, February). Hybrid huberized support vector machines for microarray classification and gene selection. *Bioinformatics 24*(3), 412–419.

Wang, X., Z. Yang, X. Chen, and W. Liu (2019). Distributed inference for linear support vector

machine. *Journal of Machine Learning Research 20*, 113:1–113:41.

Wei, M. and G. Ye (2023). Solving Second-Order Cone Programs Deterministically in Matrix Multiplication Time. `https://shorturl.at/l1Q4G`.

Xu, W., J. Liu, and H. Lian (2024). Distributed Estimation of Support Vector Machines for Matrix Data. *IEEE Transactions on Neural Networks and Learning Systems 35*(5), 6643–6653.

Zheng, Q., F. Zhu, J. Qin, and P.-A. Heng (2018, September). Multiclass Support Matrix Machine for Single Trial EEG Classification. *Neurocomputing 275*, 869–880.

Zhou, H. and L. Li (2014). Regularized matrix regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76*(2), 463–483.

Zhu, J., S. Rosset, R. Tibshirani, and T. Hastie (2003). 1-norm Support Vector Machines. In *Advances in Neural Information Processing Systems*, Volume 16. MIT Press.

Zou, H., J. Zhu, and T. Hastie (2008). New multicategory boosting algorithms based on multicategory Fisher-consistent losses. *The Annals of Applied Statistics 2*(4), 1290–1306.

School of Economics, Hangzhou Dianzi University, Hangzhou 310018, China

E-mail: chenbingzhen6026@163.com

Department of Biostatistics, School of Public Health, University of Michigan, MI 48109, USA

E-mail: canyic@umich.edu