Statistica Sinica Preprint No: SS-2025-0109	
Title	Communication-Efficient Estimation of Regularized Smoothed Support Tensor Machine
Manuscript ID	SS-2025-0109
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202025.0109
Complete List of Authors	Zihao Song, Lei Wang, Riquan Zhang and Weihua Zhao
Corresponding Authors	Weihua Zhao
E-mails	zhaowhstat@163.com
Notice: Accepted author version.	

Communication-Efficient Estimation of Regularized Smoothed Support Tensor Machine

Zihao Song[1], Lei Wang[2], Riquan Zhang[3], Weihua Zhao[1]

1. Nantong University, 2. Nankai University

3.Shanghai University of International Business and Economics

Abstract: Tensor analysis methods are becoming increasingly prevalent across various scientific applications, including neuroscience and signal processing. Existing tensor discrimination models often rely on decomposition techniques such as CANDECOMP/PARAFAC and Tucker decomposition. However, these methods typically require unfolding of tensors into matrices, which may compromise their intrinsic structural information. This article harnesses the recently introduced concept of tubal rank to present a smoothed support tensor machine with tubal nuclear norm regularization. The statistical properties of the resulting estimator are established, and the framework is extended to a distributed setting. Within this paradigm, a communication-efficient regularized estimator is introduced, which only needs access to local data from the first machine and gradient information from other local machines. Furthermore, the convergence rate of this distributed estimator is derived. By exploiting the well-defined properties of the tubal nuclear norm, we provide theoretical guarantees for low-rank structure recovery. To compute the estimator, an alternating

Corresponding author: Weihua Zhao, School of Mathematics and Statistics, Nantong University, Jiangsu Nantong, 226019, China, E-mail, zhaowhstat@163.com

minimization algorithm is developed, and its global convergence properties are analyzed. Lastly, extensive simulations are carried out to validate the proposed method, and its practical utility is demonstrated in an application involving data from invasive ductal carcinoma.

Key words and phrases: Support tensor machine; Kernel density smoothing; Low tubal rank; Distributed estimator; Tubal nuclear norm.

1. Introduction

With the rapid development of modern science and technology, data is increasingly collected in the form of multidimensional arrays. Tensors, as a natural representation of high-dimensional data, have garnered growing attention. For example, fMRI brain images (Gandy et al., 2011) are prototypical order-3 tensors of voxels. A common and simplistic approach for tensor data analysis is reshaping multidimensional arrays into vectors or matrices. However, this not only destroys intrinsic structural information but also introduces computational inefficiencies and suboptimal performance in downstream applications. Hence, the complex multidimensional nature of tensors poses significant challenges for classification tasks. Among these existing discrimination methods, support vector machines (SVMs) (Vapnik, 2013; Lian and Fan, 2018; Xu et al., 2024; Koo et al., 2008; Wang et al., 2019) have achieved remarkable success in numerous classification problems. The core of standard SVM lies in the prin-

ciple of maximizing the margin of separation between classes within a dataset, thereby enhancing generalization performance. Although the statistical properties of SVMs have been extensively studied in the literature, they are designed for vector or matrix input, not tensor data. Therefore, it is very important to develop SVMs for the tensor input, i.e., support tensor machines (STMs).

Various tensor decomposition approaches have led to the development of various STMs. For example, Hao et al. (2013) proposed a rank-one tensor factorization using multilinear algebra. Based on the Tucker decomposition (Kolda and Bader, 2009), Kotsia and Patras (2011) adopted a weight parameter strategy to better preserve the inherent structural information, while Zeng et al. (2017) employed genetic algorithm for the contraction of tensor input. Chen et al. (2019) proposed a support tensor train machine based on the tensor train (TT) decomposition (Oseledets, 2011). Furthermore, combining the CANDECOMP/PARAFAC (CP) decomposition (Kolda and Bader, 2009) and the TT decomposition, Kour et al. (2023) proposed a structure-preserving STM. Notably, most of these existing STMs often lack the guidance of statistical properties. In addition, none of them considers efficient estimation of STM in the distributed setting, which is precisely the core motivation of this work.

A fundamental constraint shared by all these existing approaches is their reliance on matricization techniques, which require the unfolding of the high-

dimensional tensors into matrices. Unfortunately, direct unfolding approaches inevitably destroy the intrinsic structural information of tensors, often leading to suboptimal results. To address this limitation, Kilmer and Martin (2011) introduced the tensor-tensor product (t-product) and corresponding t-product singular value decomposition (t-SVD), which extends the standard matrix operations and preserves the appropriate algebraic structures for the tensors. Due to its superior performance on preserving structural information, the t-product framework has been successfully applied to multiple domains, including image processing (Lu et al., 2018), tensor regression analysis (Roy and Michailidis, 2022), and many other areas. Building upon these advances, our primary goal is to apply the t-product to STM, thereby developing a novel tensor classification methodology with enhanced structural preservation capabilities. Concurrently, the growth of sample size of the tensor data presents significant computational challenges. Centralized processing becomes increasingly impractical due to prohibitive communication costs, data privacy concerns, and storage limitations. Consider, for instance, a distributed sensor network scenario where transmitting all local sensor data to a central machine would impose substantial communication overhead and storage burdens. To mitigate these challenges, Jordan et al. (2019) introduced a communication-efficient surrogate likelihood (CSL) method for distributed frameworks. Inspired by this approach, our secondary goal is

to develop a distributed STM estimator that maintains computational efficiency while preserving statistical accuracy.

In this paper, we aim to present the statistical performance of the STM based on the t-product to retain the original structural information and the tubal nuclear norm (t-TNN) to reduce the number of estimated parameters, and develop an implementable estimation algorithm with convergence guarantee. To the best of our knowledge, these problems have not previously been investigated in the context of STMs. The main contributions of this article are summarized as follows.

(1) We propose a novel STM for tensor classification that leverages t-TNN to induce a low-rank structure in the tensor parameter, thereby reducing the effective number of coefficients. To address the analytical and computational challenges posed by the non-smooth hinge loss in classical SVMs, the kernel smoothing technique is applied and we obtain the low tubal rank regularized smoothed support tensor machine (RSSTM) estimator. The estimation error bound of the proposed estimator and the convergence properties based on the alternating minimization algorithm are given. Moreover, Theorem 2 proves that the proposed estimator enjoys the low rank of order $O(r)$, where r denotes the true rank. As far as we know, this is the first work in the STMs literature to provide the low-rank guarantee. More importantly, the decomposability property of the t-TNN is given in the

Appendix, which is different from Roy and Michailidis (2022). It should be pointed out that, by applying the circulate operator on the tensor, Roy and Michailidis (2022) actually used the decomposability property of the matrix nuclear norm rather than t-TNN.

(2) For massive tensor data collected and stored in the distributed environment, we further construct a new type of communication-efficient surrogate hinge loss, which only requires individual-level data from a local machine and summary statistics from other machines, and then develop a distributed RSSTM estimator. However, it is nontrivial to establish the theoretical results, since CSL approach requires the loss function to be at least thrice differentiable while our derived loss is at most twice differentiable. Therefore, some different statistical tools based on empirical process are applied to derive the convergence rate. We prove that the proposed distributed estimator enjoys low-rank property (see Theorem 5) and it has the same convergence rate as the central estimator.

The rest of the paper is organized as follows. In Section 2, the RSSTM estimator and its estimation algorithm are built. The convergence guarantee and statistical properties are developed. In Section 3, we introduce the distributed RSSTM estimator based on the CSL method. The statistical rate and a bound of the estimated rank are also investigated. Section 4 presents some numerical

results to study the finite-sample performance. An application of invasive ductal carcinoma data is illustrated in Section 5. Section 6 concludes this paper with some discussions.

Notation. In the following, vectors are denoted with lowercase boldface letter $\boldsymbol{a}$, matrices with boldface capital letters $\boldsymbol{A}$ and order-3 tensors with boldface Euler script letters $\boldsymbol{\mathcal{A}}$. We use $\|\cdot\|_F$, $\|\cdot\|_*$ and $\|\cdot\|_{op}$ to indicate the Frobenius, nuclear and operator norms for matrix, respectively. For brevity, the k-th frontal slice of $\boldsymbol{\mathcal{A}}$ is denoted as $\boldsymbol{\mathcal{A}}^{(k)}$. The inner product between $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ is $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle = \mathrm{vec}^{\mathsf{T}}(\boldsymbol{\mathcal{A}})\mathrm{vec}(\boldsymbol{\mathcal{B}})$, where $\mathrm{vec}(\cdot)$ stacks the elements of the tensor into a vector. Denote $\|\boldsymbol{\mathcal{A}}\|_F = \sqrt{\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{A}} \rangle}$ as the Frobenius norm and $\|\boldsymbol{\mathcal{A}}\|_* = \sum_{i=1}^{r} \boldsymbol{\mathcal{S}}(i,i,1)$ as the tubal nuclear norm for the tensor, where $\boldsymbol{\mathcal{S}}$ is the singular value tensor of $\boldsymbol{\mathcal{A}}$ (see Appendix S1 for more details). Denote $a \vee b = \max\{a, b\}$ and $a \wedge b = \min\{a, b\}$ for $a, b \in \mathbb{R}$.

2. Smoothed STM with low tubal-rank regularization

2.1 Model and estimation

For a binary classification study, the independent and identically distributed (i.i.d.) observations $\{(y_i, \boldsymbol{\mathcal{X}}_i)\}_{i=1}^n$ are drawn from the joint distribution $(y, \boldsymbol{\mathcal{X}})$. Here, the response variable $y_i \in \{-1, 1\}$ represents the class label and $\boldsymbol{\mathcal{X}}_i \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is the corresponding predictor tensor. We consider the estimation of

dimension-compatible tensor coefficient $\boldsymbol{\mathcal{B}}$ of interest and the intercept term β by,

$$\min_{\beta,\boldsymbol{\mathcal{B}}} \sum_{i=1}^{n} L(y_i(\beta + \langle \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}}_i \rangle)) \tag{2.1}$$

where $L(t) = (1-t)_+$ is hinge loss and the tensor coefficient $\boldsymbol{\mathcal{B}}$ is used to capture the relationship between label responses and tensor covariates. Motivated by l_1-norm SVM (Peng et al., 2016), to reduce the large number of tensor coefficients and capture the baseline effect within different classes, we impose the t-TNN regularization $\|\boldsymbol{\mathcal{B}}\|_*$ on $\boldsymbol{\mathcal{B}}$. Thus, (2.1) can be rewritten as

$$\min_{\beta,\boldsymbol{\mathcal{B}}} \sum_{i=1}^{n} L(y_i(\beta + \langle \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}}_i \rangle)) + n\lambda \|\boldsymbol{\mathcal{B}}\|_*,$$

where λ is a positive tuning parameter. As the l_1-norm encourages sparse solutions, the t-TNN penalty yields tensor estimators with sparse singular values, thus inducing low tubal rank. In fact, we can regard a order-3 tensor of size $I_1 \times I_2 \times I_3$ as a matrix of size $I_1 \times I_2$ with each element being a tube in the third dimension. It implies that the tubal rank of a order-3 tensor is analogous to the rank of a matrix, and the tubal rank reduces to the matrix rank when $I_3 = 1$. Therefore, in addition to reduce the number of parameters, the low tubal rank property implies a type of dependence, namely t-linear dependence (Kilmer et al., 2013), among different slices. A similar conclusion can be found

in the matrix regularization, i.e., the low rank of a matrix characterizes linear dependence between columns and rows. In other words, the low tubal rank tensor coefficient $\boldsymbol{\mathcal{B}}$ could seize the baseline effects between classes in dataset.

Furthermore, to avoid the problems of asymptotic analysis and computation caused by the non-smooth hinge loss $L(\cdot)$, we adopt a smoothed function

$$L_h(t) = L \circledast K_h(t),$$

proposed by Wang et al. (2019, 2023) for SVM, to replace the original hinge loss, where operation $\circledast$ is convolution and $K_h(t) = \frac{1}{h}K(\frac{t}{h})$ with the smooth kernel function $K(\cdot)$ and positive bandwidth h. Hence, the proposed low tubal rank regularized smoothed support tensor machine (RSSTM) estimator is formulated as

$$(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}}) := \underset{\beta, \boldsymbol{\mathcal{B}}}{\operatorname{argmin}} \ f(\beta, \boldsymbol{\mathcal{B}}) = \sum_{i=1}^{n} L_h(y_i(\beta + \langle \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}}_i \rangle)) + n\lambda \|\boldsymbol{\mathcal{B}}\|_*. \tag{2.2}$$

In this work, we mainly employ the Gaussian kernel and Epanechnikov kernel for RSSTM and then obtain the corresponding smoothed loss functions $L_h^G(t)$ and $L_h^E(t)$, respectively. It should be noted that there exist other kernel functions which can be employed. Here, we use the widely used Gaussian kernel and Epanechnikov kernel as examples for clearer illustration, and there is indeed

little difference between them. By direct calculation, the more explicit forms are as follows,

$$L_h^G(t) = (1-t)\mathbf{\Phi}\Big(\frac{1-t}{h}\Big) + \frac{h}{\sqrt{2\pi}}\exp\Big\{-\frac{(1-t)^2}{2h^2}\Big\},$$

$$L_h^E(t) = (1-t)\cdot I\{t \le 1-h\} + 0\cdot I\{t \ge 1+h\} + \frac{(1-t+h)^3(3h-1+t)}{16h^3}\cdot I\{1-h < t \le 1+h\},$$

where $\mathbf{\Phi}(\cdot)$ denotes the cumulative distribution function of Gaussian distribution and $I\{\cdot\}$ is the indicator function. For the density estimator, the optimal rate of bandwidth h is $O(n^{-1/5})$. Hence, we set $h = \eta n^{-1/5}$ in our implementation with a positive constant $\eta \in (0, 3)$.

To obtain the RSSTM estimator $(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}})$, we develop an implementable estimation algorithm by the alternating minimization (AM) method. It is obvious that the objective function (2.2) is jointly convex, but it is non-smooth induced by the t-TNN. Therefore, we adopt a proximal mapping operator for $\boldsymbol{\mathcal{B}}$, i.e., the t-SVT operator, which is widely used in various applications of tensor learning (Lu et al., 2018; Roy and Michailidis, 2022). Without loss of generality, we uniformly use $L_h(\cdot)$ to represent the loss function smoothed by Gaussian kernel or Epanechnikov kernel.

Updating $\boldsymbol{\mathcal{B}}$: Let $u_i = y_i(\beta + \langle \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}}_i \rangle)$. Given the k-th iterative estimates β^k

and $\boldsymbol{\mathcal{B}}^k$, we consider the quadratic majorization function of $f(\beta, \boldsymbol{\mathcal{B}})$ with respect to $\boldsymbol{\mathcal{B}}$,

$$\widehat{f}(\beta^k, \boldsymbol{\mathcal{B}}) = \frac{1}{n}\sum_{i=1}^{n} L_h(u_i^k) + \frac{1}{n}\sum_{i=1}^{n} L_h'(u_i^k) y_i \langle \boldsymbol{\mathcal{B}} - \boldsymbol{\mathcal{B}}^k, \boldsymbol{\mathcal{X}}_i \rangle + \frac{\mu}{2}\|\boldsymbol{\mathcal{B}} - \boldsymbol{\mathcal{B}}^k\|_F^2 + \lambda\|\boldsymbol{\mathcal{B}}\|_*,$$

where $u_i^k = y_i(\beta^k + \langle \boldsymbol{\mathcal{B}}^k, \boldsymbol{\mathcal{X}}_i \rangle)$ and $\mu > 0$. The minimizer of $\widehat{f}(\beta^k, \boldsymbol{\mathcal{B}})$ is unique and enjoys a closed-form through t-SVT operator (see Theorem 4.2 in Lu et al. (2018) for example), obtained by

$$\boldsymbol{\mathcal{B}}^{k+1} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{S}}_{\frac{\lambda}{\mu}} * \boldsymbol{\mathcal{V}}^{\mathsf{T}}, \tag{2.3}$$

where $\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{V}}$ are the results of t-SVD of $\boldsymbol{\mathcal{B}}^k - \frac{1}{n\mu}\sum_{i=1}^{n} L_h'(u_i^k) y_i \boldsymbol{\mathcal{X}}_i$, notation $*$ denotes t-product and $\boldsymbol{\mathcal{S}}_\tau = \text{ifft}((\widetilde{\boldsymbol{S}} - \tau)_+, [], 3)$. The $\text{ifft}(\cdot)$ is the inverse Fast Fourier Transform using MATLAB command, $\widetilde{\boldsymbol{S}}$ is defined by (S1.1) of Appendix S1, and a_+ denotes the positive part of a, i.e., $a_+ = \max(t, 0)$.

Updating β: To solve the subproblem of β, we resort to the gradient descent method benefited from the smoothed and convex loss function L_h. Given $\boldsymbol{\mathcal{B}}^{k+1}$, the $(k+1)$-th iteration of β with step size ρ is written as

$$\beta^{k+1} = \beta^k - \frac{\rho}{n}\sum_{i=1}^{n} L_h'(y_i(\beta^k + \langle \boldsymbol{\mathcal{B}}^{k+1}, \boldsymbol{\mathcal{X}}_i \rangle)) y_i. \tag{2.4}$$

After some simple algebraic computation, the first-order derivative of L_h^G and L_h^E are $L_h^{G\prime}(t) = -\mathbf{\Phi}\left(\frac{1-t}{h}\right)$ and $L_h^{E\prime}(t) = -1 \cdot I\{t \leq 1-h\} - \frac{(1-t+h)^2(2h-1+t)}{4h^3} \cdot I\{1-h < t \leq 1+h\} + 0 \cdot I\{t \geq 1+h\}$, which are Lipschitz continuous with Lipschitz constants $C_{L_h^G} = \frac{1}{\sqrt{2\pi}h}$ and $C_{L_h^E} = \frac{3}{4h}$, respectively. Note that $|L_h^{G\prime}| \leq 1$ and $|L_h^{E\prime}| \leq 1$ hold, which will be used in the following analysis. We summarize the estimation procedure in Algorithm 1. The main overhead of estimation procedure lies in the per-iteration of $\boldsymbol{\mathcal{B}}^{k+1}$, which requires computing the fast Fourier transform with the time complexity $\boldsymbol{\mathcal{O}}(I_1 I_2 I_3 \log I_3)$ and SVD of $I_1 \times I_2$ matrices with the time complexity $\boldsymbol{\mathcal{O}}((I_1 \vee I_2)(I_1 \wedge I_2)^2 I_3)$. Hence, the main computation cost for each iteration is $\boldsymbol{\mathcal{O}}(I_1 I_2 I_3 \log I_3 + (I_1 \vee I_2)(I_1 \wedge I_2)^2 I_3)$.

Algorithm 1 Alternating minimization algorithm for RSSTM estimator.

Input: Initial value $(\boldsymbol{\mathcal{B}}^0, \beta^0)$, data $\{(y_i, \boldsymbol{\mathcal{X}}_i)\}_{i=1}^n$, λ, ϵ_{tol} and K_{iter}.
while $k \leq K_{iter}$ **do**
  Update $\boldsymbol{\mathcal{B}}^{k+1}$ by Equation (2.3)
  Update β^{k+1} by Equation (2.4)
  if $\|\boldsymbol{\mathcal{B}}^{k+1} - \boldsymbol{\mathcal{B}}^k\|_F \leq \epsilon_{tol}$ **then**
    break
  end if
  $k = k + 1$
end while
Output: $\widehat{\boldsymbol{\mathcal{B}}}$ and $\widehat{\beta}$.

2.2 Theoretical results

In this section, we first establish the non-asymptotic upper error bound of the RSSTM estimator with the corresponding statistical rate, which is a technical extension of Raskutti et al. (2019), and then prove that low rank is guaranteed with high probability, which has not been studied in the previous literature. Finally, the global convergence properties of the proposed algorithm are investigated. Beforehand, we should demonstrate the decomposability of t-TNN. The proofs of the following results are given in Appendix S2. Although the analysis focuses on the order-3 tensor in this paper, it is worth mentioning that our results can be easily extended to high-order tensors; see Appendix S5.

From Theorem S1.1 in the Appendix S1, the t-TNN is decomposable with the subspaces defined as follows. For any tensors $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ with tubal rank $r \leq I_1 \wedge I_2$. By skinny t-SVD, it yields $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}^{\mathsf{H}}$ where $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{I_1 \times r \times I_3}, \boldsymbol{\mathcal{V}} \in \mathbb{R}^{r \times I_2 \times I_3}$ denote the left and right orthogonal tensors. Anchoring tensor $\boldsymbol{\mathcal{B}}$, define

$$\mathscr{U} = \left\{\boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{M}} + \boldsymbol{\mathcal{N}} * \boldsymbol{\mathcal{V}}^{\mathsf{H}} : \boldsymbol{\mathcal{M}} \in \mathbb{R}^{r \times I_2 \times I_3}, \boldsymbol{\mathcal{N}} \in \mathbb{R}^{I_1 \times r \times I_3}\right\},$$

$$\mathscr{B} = \left\{\boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{U}}^{\mathsf{H}} * \boldsymbol{\mathcal{Z}} + \boldsymbol{\mathcal{Z}} * \boldsymbol{\mathcal{V}} * \boldsymbol{\mathcal{V}}^{\mathsf{H}} - \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{U}}^{\mathsf{H}} * \boldsymbol{\mathcal{Z}} * \boldsymbol{\mathcal{V}} * \boldsymbol{\mathcal{V}}^{\mathsf{H}} : \boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}\right\},$$

where $\mathscr{B} \subseteq \mathscr{U}$ holds clearly. Similar to Raskutti et al. (2019); Negahban et al.

(2010), t-TNN is $(\mathscr{U}, \mathscr{B})$-decomposable, that is,

$$\|\boldsymbol{\mathcal{A}} + \boldsymbol{\mathcal{B}}\|_* = \|\boldsymbol{\mathcal{A}}\|_* + \|\boldsymbol{\mathcal{B}}\|_*, \forall \boldsymbol{\mathcal{A}} \in \mathscr{U}^\perp, \boldsymbol{\mathcal{B}} \in \mathscr{B}, \tag{2.5}$$

where $\mathscr{U}^\perp$ indicates the orthogonal complement space of $\mathscr{U}$. The detailed proof of the above assertions is delegate to Appendix S1. It is remarkable that the decomposable property (2.5) holds for any low tubal rank tensor $\boldsymbol{\mathcal{B}}$, by defining the above subspaces dependent on the left (right) orthogonal tensor. That is, the decomposable property of t-TNN is an universal result. Notably, since orthogonal tensors $\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{V}}$ are t-SVD results of $\boldsymbol{\mathcal{B}}$, the decomposable property works only for the specific tensor $\boldsymbol{\mathcal{B}}$. For readers who are interested in decomposable regularization and/or t-TNN, please also refer to Raskutti et al. (2019); Negahban et al. (2010); Lu et al. (2018); Kilmer et al. (2008); Koltchinskii et al. (2011), among others. Furthermore, we introduce some additional notations and assumptions to derive our results. Given any tensor $\boldsymbol{\mathcal{D}}$, let $\boldsymbol{\mathcal{D}}^\perp$ be the projection of $\boldsymbol{\mathcal{D}}$ on $\mathscr{U}^\perp$ measured by the Frobenius norm and $\boldsymbol{\mathcal{D}}^0 = \boldsymbol{\mathcal{D}} - \boldsymbol{\mathcal{D}}^\perp$. The dual norm of t-TNN is denoted as $\|\cdot\|$ and its definition is given in the Appendix.

Assumption 1. Let $(\beta^*, \boldsymbol{\mathcal{B}}^*) \in \operatorname{argmin}_{\beta, \boldsymbol{\mathcal{B}}} \mathbb{E}[L_h(y(\beta + \langle \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}} \rangle))]$ be the true parameter values and the tubal rank of $\boldsymbol{\mathcal{B}}^*$ is bounded by r.

Assumption 2. $\boldsymbol{x} = \operatorname{vec}(\boldsymbol{\mathcal{X}})$ is a sub-Gaussian variable, and for some posi-

tive constants $c, C < \infty$, $c \leq \Lambda_{\min}(\mathbb{E}[(1, \boldsymbol{x}^{\mathsf{T}})^{\mathsf{T}}(1, \boldsymbol{x}^{\mathsf{T}})]) \leq \Lambda_{\max}(\mathbb{E}[(1, \boldsymbol{x}^{\mathsf{T}})^{\mathsf{T}}(1, \boldsymbol{x}^{\mathsf{T}})]) \leq C$, where $\Lambda_{\min}$ and $\Lambda_{\max}$ denote the minimum and maximum eigenvalues.

Assumption 3. $n\lambda \geq C\max\{\mathbb{E}[\|\sum_i \sigma_i \boldsymbol{\mathcal{X}}_i\|], \mathbb{E}[\|\sum_i L_h'(u_i^*) y_i \boldsymbol{\mathcal{X}}_i\|], \sqrt{n\log n}\}$ with a sufficiently large constant $C > 0$, where $L_h'(u_i^*) \in [-1, 1]$ with $u_i^* = y_i(\beta^* + \langle \boldsymbol{\mathcal{B}}^*, \boldsymbol{\mathcal{X}}_i \rangle)$ and σ_i being i.i.d. Rademacher random variables.

Assumption 4. For any small enough $t > 0$, define $\mathscr{Q} := \{(\delta, \boldsymbol{\mathcal{D}}) : \|\boldsymbol{\mathcal{D}}^{\perp}\|_* \leq 3\|\boldsymbol{\mathcal{D}}^0\|_* + |\delta|, \delta^2 + \mathbb{E}[\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{D}} \rangle^2] = t^2\}$. For $\forall (\delta, \boldsymbol{\mathcal{D}}) \in \mathscr{Q}$, $\mathbb{E}[L_h(y(\beta + \delta + \langle \boldsymbol{\mathcal{B}}^* + \boldsymbol{\mathcal{D}}, \boldsymbol{\mathcal{X}} \rangle))] - \mathbb{E}[L_h(y(\beta + \langle \boldsymbol{\mathcal{B}}^*, \boldsymbol{\mathcal{X}} \rangle))] \geq Ct^2$.

Remark 1. *Assumption 1 is a regularity condition to impose the low tubal rank structure. In addition, the true parameter enjoys the skinny t-SVD,* $\boldsymbol{\mathcal{B}}^* = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}^{\mathsf{H}}$*, such that* $\boldsymbol{\mathcal{B}}^* \in \mathscr{B}$ *where the subspace* $\mathscr{B}$ *is defined by* $\boldsymbol{\mathcal{U}}$ *and* $\boldsymbol{\mathcal{V}}$*. Assumption 2 imposes a restricted eigenvalue (RE) type of condition to establish the Frobenius-type error bound for the t-TNN regularized estimator. In Assumption 3, a high-level condition is imposed on* λ*. In Lemma S2.3 of Appendix S2, we will show that* $\mathbb{E}[\|\sum_i \sigma_i \boldsymbol{\mathcal{X}}_i\|]$ *and* $\mathbb{E}[\|\sum_i L_h'(u_i^*) y_i \boldsymbol{\mathcal{X}}_i\|]$ *are all bounded by* $C\sqrt{n(I_1 I_3 \vee I_2 I_3)}$ *for some constants* $C > 0$*. Assumption 4, similar with Bernstein condition (Geoffrey et al., 2020), is a strong local convexity condition for*

the expected loss.

For the sake of brevity, let C be a generic positive constant, unless otherwise stated, whose value may be different at different places, even on the same line.

Theorem 1 *Under **Assumptions** 1-4, assume the condition for the constant M in Lemma S2.1 holds, with probability $1 - Ce^{-Cn\lambda/M} - CnP(\|\boldsymbol{\mathcal{X}}\| > M)$,*

$$\mathbb{E}[(\widehat{\beta} - \beta^* + \langle \boldsymbol{\mathcal{X}}, \widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^* \rangle)^2] \leq C\lambda^2 r,$$

$$(\widehat{\beta} - \beta^*)^2 + \|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_F^2 \leq C\lambda^2 r,$$

and

$$\|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_* \leq C\lambda r,$$

where the expectation in the first equation above is over the distribution of $\boldsymbol{\mathcal{X}}$ which is an independent copy of the sample data, and the probability $P(\|\boldsymbol{\mathcal{X}}\| > M) \leq Ce^{-CM^2}$ (see Lemma S2.4).

Corollary 1 *Under **Assumptions** 1-4, taking $M \asymp \sqrt{(I_1 I_3 \vee I_2 I_3)\log n}$ and $\lambda \asymp \sqrt{\frac{I_1 I_3 \vee I_2 I_3}{n}}$, Theorem 1 yields*

$$(\widehat{\beta} - \beta^*)^2 + \|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_F^2 \leq O_p\Big(\frac{r(I_1 I_3 \vee I_2 I_3)}{n}\Big).$$

Remark 2. *Lian (2021) used the SNN-type regularization (Huang et al., 2015)*

and derived the convergence rate as $O_p(r(I_1I_3 \vee I_2I_3 \vee I_1I_2)/n)$. *As we expected, it is slower than our rate, especially when* $I_3 \leq I_1 \wedge I_2$, *which is common in the color image data and some other real-world applications. In fact, the SNN requires unfolding tensors into matrices, which leads to massive overhead and loss of internal structure information, while the t-TNN directly seizes low-rank information in the tensor form to preserve the intrinsic high-dimensional structure information. To the best of our knowledge, this work is the first attempt to investigate the statistical rate of the t-TNN-type regularized estimator, and it is also a direct evidence to illustrate that t-TNN is more efficient than SNN from a statistical perspective.*

Theorem 2 *Under* ***Assumptions*** *1-4, with probability approaching one as* n *goes to infinity, we have*

$$\mathrm{rank_t}(\widehat{\boldsymbol{\mathcal{B}}}) \leq Cr.$$

Theorem 2 shows that the RSSTM estimator has a low rank of order $O(r)$, which gives the rigorous theoretical guarantee on the low rank property. This is also the first attempt to obtain low rank guarantee of STMs, compared with the existing literature.

2.3 Convergence properties of proposed algorithm

We next establish the global convergence of Algorithm 1. The Lemma 1 investigates the sufficient descent property of the sequence generated by our estimation algorithm, Theorem 3 studies the local convergence, and finally Corollary 2 states the global convergence of our estimation procedure through the convexity of L_h.

Lemma 1 *Denote* $\{(\boldsymbol{\mathcal{B}}^k, \beta^k)\}_{k=1}$ *as a sequence generated by Algorithm 1, then we have*

$$\begin{cases} f(\beta^k, \boldsymbol{\mathcal{B}}^{k+1}) + \gamma_1 \|\boldsymbol{\mathcal{B}}^{k+1} - \boldsymbol{\mathcal{B}}^k\|_F^2 \leq f(\beta^k, \boldsymbol{\mathcal{B}}^k), & (2.6) \\ f(\beta^{k+1}, \boldsymbol{\mathcal{B}}^{k+1}) + \gamma_2 |\beta^{k+1} - \beta^k|^2 \leq f(\beta^k, \boldsymbol{\mathcal{B}}^{k+1}). & (2.7) \end{cases}$$

where $\gamma_1 = \frac{\mu - C_{L_h}}{2}$, $\gamma_2 = \frac{2 - \rho C_{L_h}}{2\rho}$ *and* C_{L_h} *is the Lipschitz constant of the smoothed loss* L_h.

Remark 3. *In Lemma 1, the sufficient descent inequality holds only under the latent conditions* $\gamma_1 > 0$ *and* $\gamma_2 > 0$. *Hence, it implies that we should require* $\mu > C_{L_h}$ *and* $0 < \rho < 2/C_{L_h}$.

For brevity, let $\boldsymbol{\mathcal{W}} := (\beta, \boldsymbol{\mathcal{B}})$ and $f(\boldsymbol{\mathcal{W}}) := f(\beta, \boldsymbol{\mathcal{B}})$. Theorem 3 states that our estimation algorithm enjoys local convergence.

Theorem 3 *Denote* $\{\boldsymbol{\mathcal{W}}^k\}_{k=1}$ *as a sequence generated by Algorithm 1, the fol-*

lowing assertions hold,

(1) $f(\boldsymbol{\mathcal{W}}^k)$ *is monotonically decreasing, i.e.,* $f(\boldsymbol{\mathcal{W}}^{k+1}) + \gamma\|\boldsymbol{\mathcal{W}}^{k+1} - \boldsymbol{\mathcal{W}}^k\|_F^2 \leq f(\boldsymbol{\mathcal{W}}^k)$ *with* $0 < \gamma \leq \gamma_1 \wedge \gamma_2$*;*

(2) $\{\boldsymbol{\mathcal{W}}^k\}$ *is a bounded sequence. In addition,* $\sum_{k=1}^{\infty} \|\boldsymbol{\mathcal{W}}^{k+1} - \boldsymbol{\mathcal{W}}^k\|_F^2 < \infty$ *and* $\lim_{k\to+\infty}(\boldsymbol{\mathcal{W}}^{k+1} - \boldsymbol{\mathcal{W}}^k) = 0$.

(3) Any accumulation point $\boldsymbol{\mathcal{W}}^\star$ *of* $\{\boldsymbol{\mathcal{W}}^k\}$ *is a critical point of* $f(\boldsymbol{\mathcal{W}})$.

Corollary 2 *Due to the convexity of* $L_h(\cdot)$*, according to Theorem 3, the Algorithm 1 eventually enjoys global convergence. That is, the whole sequence* $\{\boldsymbol{\mathcal{W}}^k\}$ *converges to the critical point of* $f(\boldsymbol{\mathcal{W}})$ *and satisfies* $\sum_{k=0}^{+\infty} \|\boldsymbol{\mathcal{W}}^{k+1} - \boldsymbol{\mathcal{W}}^k\|_F < +\infty$.

3. Communication-efficient distributed RSSTM

In this section, we consider the distributed setting. For ease of exposition, we assume that the entire dataset $\{(y_i, \boldsymbol{\mathcal{X}}_i)\}_{i=1}^N$ of the total sample size N is stored on m different machines independently and identically. Without loss of generality, assume that all local machines have the same sample size n and $N = m \cdot n$. To be specific, for $j = 1, \cdots, m$, the n observations in the j-th machine are denoted by $\mathcal{D}_j = \{(y_i, \boldsymbol{\mathcal{X}}_i)\}_{i\in\mathcal{I}_j}$, where $\{\mathcal{I}_j\}_{j=1}^m$ are disjoint

index sets with $|\mathcal{I}_j| = n$ and $\cup_{j=1}^{m}\mathcal{I}_j = \{1, \cdots, N\}$. In the distributed setting, due to privacy reasons and transmission cost, it is infeasible to analysis all data on a single machine to estimate the unknown parameter $(\beta, \boldsymbol{\mathcal{B}})$ by empirical hinge loss $(1/N)\sum_{i=1}^{N} L_h(y_i(\beta + \langle \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}}_i\rangle))$, which requires communication cost $\mathcal{O}(mnI_1I_2I_3)$.

3.1 Model and estimation

Motivated by the CSL method (Jordan et al., 2019), we propose a novel distributed estimator based on the communication-efficient surrogate RSSTM loss function, which only uses the data from the first machine and the gradient statistics from other machines. Here, the first machine displays the role of the central machine and the proposed method only needs the total communication cost $\mathcal{O}(mI_1I_2I_3)$. However, Jordan et al. (2019) requires that the loss function is at least thrice differentiable for the theoretical properties, which cannot be applied to our smoothed hinge loss directly. Therefore, we apply empirical process techniques to address these problems to establish theoretical results. With the above notation, the global and the j-th local smoothed hinge loss functions are given as

$$Q(\beta, \boldsymbol{\mathcal{B}}) = \frac{1}{N}\sum_{i=1}^{N} L_h(y_i(\beta+\langle \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}}_i\rangle)), \;\; Q_j(\beta, \boldsymbol{\mathcal{B}}) = \frac{1}{n}\sum_{i\in\mathcal{I}_j} L_h(y_i(\beta+\langle \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}}_i\rangle)).$$

Adapted from the CSL framework, with an initial estimator $(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}})$, we define the surrogate loss function $\breve{Q}(\beta, \boldsymbol{\mathcal{B}})$ by utilizing global first-order information and local higher-order information as follows:

$$\begin{aligned}\breve{Q}(\beta, \boldsymbol{\mathcal{B}}) = Q_1(\beta, \boldsymbol{\mathcal{B}}) - \beta(\nabla_\beta Q_1(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}}) - \nabla_\beta Q(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}})) \\ - \langle \boldsymbol{\mathcal{B}}, \nabla_{\boldsymbol{\mathcal{B}}} Q_1(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}}) - \nabla_{\boldsymbol{\mathcal{B}}} Q(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}}) \rangle,\end{aligned}$$

where $\nabla_\beta Q_1(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}}) = (1/n)\sum_{i\in\mathcal{I}_1} L_h'(y_i(\widehat{\beta} + \langle \widehat{\boldsymbol{\mathcal{B}}}, \boldsymbol{\mathcal{X}}_i\rangle))y_i$, $\nabla_{\boldsymbol{\mathcal{B}}} Q_1(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}}) = (1/n)\sum_{i\in\mathcal{I}_1} L_h'(y_i(\widehat{\beta} + \langle \widehat{\boldsymbol{\mathcal{B}}}, \boldsymbol{\mathcal{X}}_i\rangle))y_i\boldsymbol{\mathcal{X}}_i$, and $\nabla_\beta Q(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}})$ and $\nabla_{\boldsymbol{\mathcal{B}}} Q(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}})$ are similarly derived using the whole data. The distributed RSSTM estimator is defined as

$$(\breve{\beta}, \breve{\boldsymbol{\mathcal{B}}}) := \underset{\beta, \boldsymbol{\mathcal{B}}}{\operatorname{argmin}}\, \breve{Q}(\beta, \boldsymbol{\mathcal{B}}) + \lambda\|\boldsymbol{\mathcal{B}}\|_*, \tag{3.8}$$

where λ is a tuning parameter. We adapt the AM method for estimating $(\beta, \boldsymbol{\mathcal{B}})$ in the distributed framework. Similarly, the main technique of our algorithm is to construct the quadratic majorization function which locally majorizes the surrogate loss function $\breve{Q}(\beta, \boldsymbol{\mathcal{B}})$.

Updating $\boldsymbol{\mathcal{B}}$: At the $(k+1)$-th iteration, with the previous k-th estimates $(\beta^k, \boldsymbol{\mathcal{B}}^k)$, we define the majorization function

$$F(\beta^k, \boldsymbol{\mathcal{B}}) = \breve{Q}(\beta^k, \boldsymbol{\mathcal{B}}^k) + \langle \nabla_{\boldsymbol{\mathcal{B}}} \breve{Q}(\beta^k, \boldsymbol{\mathcal{B}}^k), \boldsymbol{\mathcal{B}} - \boldsymbol{\mathcal{B}}^k \rangle + \frac{\mu}{2}\|\boldsymbol{\mathcal{B}} - \boldsymbol{\mathcal{B}}^k\|_F^2 + \lambda\|\boldsymbol{\mathcal{B}}\|_*.$$

The isotropic form of $F(\beta^k, \boldsymbol{\mathcal{B}})$ enjoys a simple analytic solution via t-SVT operator and $\boldsymbol{\mathcal{B}}^{k+1}$ takes the form,

$$\boldsymbol{\mathcal{B}}^{k+1} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{S}}_{\frac{\lambda}{\mu}} * \boldsymbol{\mathcal{V}}^{\mathsf{T}}, \tag{3.9}$$

where $\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{V}}$ are results of t-SVD of $\boldsymbol{\mathcal{B}}^k - \frac{1}{\mu}\nabla_{\boldsymbol{\mathcal{B}}}\breve{Q}(\beta^k, \boldsymbol{\mathcal{B}}^k)$.

Updating β: With the step size ρ, by gradient descent method, the iteration equation of β^{k+1} can be written as

$$\beta^{k+1} = \beta^k - \rho\nabla_{\beta}\breve{Q}(\beta^k, \boldsymbol{\mathcal{B}}^{k+1}). \tag{3.10}$$

See Algorithm 2 for the detailed implementation and pseudocode. The main time complexity of per-iteration is $\mathcal{O}(I_1 I_2 I_3 \log I_3 + (I_1 \vee I_2)(I_1 \wedge I_2)^2 I_3)$. By a similar argument in Section 2.2, the distributed estimation algorithm also has the global convergence property; see the following Corollary 3.

Corollary 3 *The whole sequence* $\{\boldsymbol{\mathcal{W}}^k\}$ *generated by Algorithm 2 converges to the critical point of* $\breve{Q}(\beta, \boldsymbol{\mathcal{B}})$ *and satisfies* $\sum_{k=0}^{+\infty} \|\boldsymbol{\mathcal{W}}^{k+1} - \boldsymbol{\mathcal{W}}^k\|_F < +\infty$, *where* $\boldsymbol{\mathcal{W}}^k := (\beta^k, \boldsymbol{\mathcal{B}}^k)$.

Algorithm 2 Distributed learning for RSSTM estimator.

Require: Local data $\{(y_i, \boldsymbol{\mathcal{X}}_i)\}_{i\in\mathcal{I}_j}$, λ, ϵ_{tol} and K_{iter}.
Compute the initial estimate $(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}})$ by Algorithm 1 and transmit them to all machines.
Calculate the local gradient $\nabla Q_j(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}})$ for $j = 1, \cdots, m$ and transmit them to the first machine.
Compute the global gradient $\nabla Q(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}})$ at the first machine.
while $k \leq K_{iter}$ **do**
    Update $\boldsymbol{\mathcal{B}}^{k+1}$ by Equation (3.9)
    Update β^{k+1} by Equation (3.10)
    if $\|\boldsymbol{\mathcal{B}}^{k+1} - \boldsymbol{\mathcal{B}}^{k}\|_F \leq \epsilon_{tol}$ **then**
        break
    end if
    $k = k + 1$
end while
Output: $\breve{\boldsymbol{\mathcal{B}}}$ and $\breve{\beta}$.

3.2 Statistical properties of distributed RSSTM estimator

Next, we establish the non-asymptotic theoretical results for the proposed distributed RSSTM estimator. Since the smoothed hinge loss is not thrice differentiable and the complex intrinsic structure of the tensor covariate, we cannot directly employ the proof techniques of Jordan et al. (2019). In addition, the definition of surrogate loss $\breve{Q}(\beta, \boldsymbol{\mathcal{B}})$ only involves local data on the first machine, which poses a technical challenge for deriving error bounds with respect to the global data via Lemma S2.1. To address these difficulties, the key technical we adopted lies in Bernstein's inequality and covering argument to obtain a more refined bound in Lemma S3.1 of the Appendix S3. Under the distributed

framework, the following additional assumptions are needed to impose the restrictions on the initial estimator. The rate c_n suggests that the initial estimator $(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}})$ should not be too bad, and it also works to evaluating the effect of initial estimator on the statistical rate of our proposed distributed estimator.

Assumption 5. $|\widehat{\beta} - \beta^*| + \|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_* \leq c_n := Cr\sqrt{(I_1 I_3 \vee I_2 I_3)/n}$ and $\mathrm{rank_t}(\widehat{\boldsymbol{\mathcal{B}}}) \leq Cr$.

Remark 4. *For any initial estimator we used, the following theoretical results holds as long as it satisfies Assumption 5. In fact, a natural choice of* $(\widehat{\beta}, \widehat{\boldsymbol{\mathcal{B}}})$ *can be obtained from* (2.2) *on the first machine, since it does satisfy the above assumption by Theorem 1 and Theorem 2. Moreover, if* $\mathrm{rank_t}(\widehat{\boldsymbol{\mathcal{B}}}) \leq Cr$, *the Assumption 5 can be replaced by* $|\widehat{\beta} - \beta^*| + \|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_F \leq C\sqrt{r(I_1 I_3 \vee I_2 I_3)/n}$ *since* $\|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_* \leq C\sqrt{r}\|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_F$.

The following theorem presents the statistical convergence rate of the distributed RSSTM estimator.

Theorem 4 *Under* ***Assumption*** *1-5, taking*

$$\lambda \asymp \sqrt{\frac{(I_1 I_3 \vee I_2 I_3)\log N}{N}} + \frac{r(I_1 I_3 \vee I_2 I_3)}{n}\sqrt{\log n} + \frac{r^2 (I_1 I_3 \vee I_2 I_3)^{5/2}(\log n)^2}{n^{3/2}}, \tag{3.11}$$

then with the probability at least $1 - n^{-C}$*, we have*

$$\mathbb{E}[(\breve{\beta} - \beta^* + \langle \boldsymbol{\mathcal{X}}, \breve{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^* \rangle)^2] \leq a_n^2,$$

$$(\breve{\beta} - \beta^*)^2 + \|\breve{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_F^2 \leq a_n^2,$$

and

$$\|\breve{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_* \leq C\sqrt{r}a_n,$$

where $a_n = C\Big\{\lambda\sqrt{r} + \frac{b_n^3 r^{3/2} I_1 I_2 I_3^2 \log n}{n}\Big\}$ *and* $b_n = C\sqrt{(I_1 I_3 \vee I_2 I_3)\log n}$.

In our analysis, c_n is used to bound the operator norm of the gradient variable of $\breve{Q}$, which directly determines the value of λ. Therefore, the estimation accuracy above (a_n) is closely related to the convergence rate c_n of the initial estimator in Assumption 5. For instance, the sharper c_n, the sharper error bound on distributed estimator. In addition, our result can also be extended to other estimators with a different c_n. With an additional condition on the sample size, the following Corollary 4, which is an immediate result of Theorem 4, shows that the first term of λ can become the dominant term.

Corollary 4 *Under the assumptions of Theorem 4 and assume that*

$$\frac{r(I_1 I_3 \vee I_2 I_3)\sqrt{\log n}}{n} + \frac{r^2 (I_1 I_3 \vee I_2 I_3)^{5/2} (\log n)^2}{n^{3/2}} = O\Big(\sqrt{\frac{(I_1 I_3 \vee I_2 I_3)\log N}{N}}\Big), \tag{3.12}$$

and

$$\frac{b_n^3 r^{3/2} I_1 I_2 I_3^2 \log n}{n} = O\Big(\sqrt{\frac{r(I_1 I_3 \vee I_2 I_3)\log N}{N}}\Big), \tag{3.13}$$

then we have

$$(\breve{\beta} - \beta^*)^2 + \|\breve{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_F^2 \leq O_p\Big(\frac{r(I_1 I_3 \vee I_2 I_3)\log N}{N}\Big).$$

With some positive constant C, conditions (3.12) *and* (3.13) *are equivalent to a constraint for m as $m \leq CN^{1/2}/\log^2 N$.*

Remark 5. *Note that the Condition* (3.12) *can dominate the first term in the definition of λ in* (3.11). *A similar constraint is also imposed on a_n by* (3.13). *Eventually, the above two conditions are equivalent to impose a constraint on the number of machines. It suggests that the number of machines m should not be too large or the local sample size n cannot be too small. To further illustrate the requirement of distributed setting, we simplify* (3.13) *as a condition similar to Wang et al. (2025), that is, $N/\log N \leq Cn^2(I_1 I_3 \vee I_2 I_3)^{-2} r^{-2} \log^{-5} n$, which suffices to make our distributed estimator achieving the same rate as the central estimators that use the full data. For example, for fixed n and r, N (also m) should be smaller as I_1, I_2, I_3 increases. Note that, there is however no free lunch, without setting any conditions on m, n, N, we may not obtain the desired convergence rate.*

Next, we again illustrate the low rank of the distributed RSSTM estimator.

Theorem 5 *Under the same assumptions in Theorem 4, with probability approaching one as n goes to infinity,, we have*

$$\mathrm{rank_t}(\check{\boldsymbol{\mathcal{B}}}) \leq Cr.$$

From Theorem 5, the distributed RSSTM estimator also enjoys the low rank of order $O(r)$ with theoretical guarantee. Empirically, compared with the local estimator, the estimated rank of our distributed estimator is close to the true rank (see the following simulations), since it involves the gradient statistics of other machines.

4. Simulation

In this section, we first verify the statistical rate and convergence behavior of the proposed RSSTM estimator in Corollary 1 through simulations in Section 4.1. Then, we further investigate the performance of the distributed RSSTM estimators in Section 4.2. Due to page limitation, some simulations results are delegated to the Appendix S4.

4.1 RSSTM

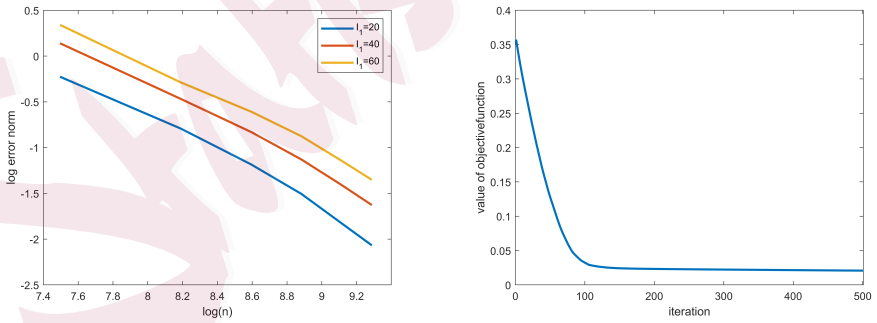
In the first simulation, we consider the matrix observations by setting $I_1 = I_2 = 20, 40, 60$ and $I_3 = 1$, since how to get the ground-truth tensor parameter is still undeveloped. The response variables are generated by $P(y_i = 1) = P(y_i = -1) = 0.5$ for $i = 1, \cdots, n$ and the sample size in two classes is balanced. If $y_i = 1$, the elements of the covariate $\boldsymbol{\mathcal{X}}_i$ are independently generated from a normal distribution with the mean matrix $\boldsymbol{A} = (a_{ij})_{I_1 \times I_2}$ and unit variance. If $y_i = -1$, the entries are independently generated from a normal distribution with mean matrix $-0.5\boldsymbol{A}$ and variance 1. Thus, by Koo et al. (2008), we have

$$\boldsymbol{\mathcal{B}}^* = 2\boldsymbol{A}/\{1.5\mathrm{tr}(\boldsymbol{A}^\top\boldsymbol{A}) + 2b\sqrt{\mathrm{tr}(\boldsymbol{A}^\top\boldsymbol{A})}\}, \ \ \beta^* = -\mathrm{tr}(\boldsymbol{A}^\top\boldsymbol{\mathcal{B}}^*)/4,$$

where b is obtained by $\phi(b) = \sqrt{\mathrm{tr}(\boldsymbol{A}^\top\boldsymbol{A})}\boldsymbol{\Phi}(b)$ with $\phi(\cdot)$ being the density function of $\mathcal{N}(0, 1)$. We set $n = 1800, 3600, 5400, 7200, 9000, 10800$, and $r = 2$ for $\boldsymbol{\mathcal{B}}^*$ by imposing the mean matrix $\boldsymbol{A}$: $a_{1j} = 0.1j$, $a_{2j} = 0.2j - 1$ for $1 \le j \le 5$ and $a_{ij} = 0$ otherwise. The tuning parameter tuple of (h, λ) is selected by five-fold cross-validation with the ranges $\eta \in (0, 3)$ and $\lambda \in [2^{-3}, 2^3]\sqrt{(I_1I_3 \vee I_2I_3)/n}$.

The estimation error is evaluated by the logarithmic Frobenius norm $\log(|\widehat{\beta} - \beta^*| + \|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_F)$ and Figure 1(a) shows the averaged estimation errors by 50

independent replications. It is easy to see that the slope of the curve is approximately $n^{-1/2}$ in agreement with the order given in Corollary 1. Moreover, the intercept is also consistent with the order of I_1 in our result. For instance, the difference between the blue and red lines should be around (log(40)-log(20))/2=0.35, and the difference between the red and yellow lines should be almost (log(60)-log(40))/2=0.2, which are in line with the plot. To visualize the convergence behavior of our algorithm, we plot the convergence curve of Algorithm 1 with certain trade-off parameters under setting $I_1 = 20$ and $n = 3600$. As illustrated in Figure 1(b), the value of objective function strictly decreases. Since the convergence behavior, and the finite-sample performance in terms of the slope of curve and intercept term are similar for both the Gaussian kernel and the Epanechnikov kernel, we only report the result obtained by the Epanechnikov kernel in this example.



(a) (b)

Figure 1: (a) Log error versus $\log(n)$ for various dimension I_1. (b) Convergence process of the proposed algorithm under the setting $I_1 = 20$, $n = 10800$.

4.2 Distributed RSSTM

We next verify the classification performance and rank estimation of distributed RSSTM by tensor observation of size $I \times I \times 3$. The response variable and the corresponding covariate are generated in similar way as in Sec 4.1, expect the mean tensor $\boldsymbol{\mathcal{A}}$ with tubal rank $r = 2, 5$ to be generated by following two scenarios:

1) $\forall k$, $a_{1jk} = 0.1j$, $a_{2jk} = 0.2j - 1$ for $1 \leq j \leq 5$ and $a_{ijk} = 0$ otherwise;

2) $\forall k$, $a_{ijk} = 0.1i$, $a_{i(i+1)k}) = 0.2i - 1$ for $1 \leq i \leq 5$ and $a_{ijk} = 0$ otherwise.

Moreover, if $y_i = -1$, the entries of $\boldsymbol{\mathcal{X}}_i$ are independently sampled from a normal distribution with mean tensor $d\boldsymbol{\mathcal{A}}$ and unit variance. In all simulations, we consider $I = 20$, $d = -0.5, -1$ and 10000 samples. The prediction error and estimated tubal rank of the initial and distributed estimator based on 50 independent runs. For ease of illustration, the compared estimators are indicated with follows:

(1) Sub-RSSTM (Sub): the subsample RSSTM estimator obtained by the first machine;

(2) Ave-RSSTM (Ave): the averaged RSSTM estimator which computes the local RSSTM estimators on each local machine and combines them via taking the average;

(3) CSL-RSSTM (CSL): the proposed distributed estimator based on CSL method.

Example 1: We consider the fixed sample size $N = 10000$ and vary the number of machines $m = 1, 5, 10, 20$. The averaged rank estimation and the averaged prediction errors for the initial estimator and distributed estimator are recorded in Table 1 and Table S4.1, respectively. (1) For any fixed m, Sub-RSSTM obtains the highest prediction errors since it only uses local data on the first machine, while it is improved by Ave-RSSTM by taking average. Our proposed CSL-RSSTM yields the best results. (2) As m increases, all errors are reduced. However, compared with Sub-RSSTM and Ave-RSSTM, the proposed CSL-RSSTM significantly reduces errors. (3) The estimated rank of CSL-RSSTM is closer to r than other estimators,implying that the proposed distributed estimator could improve the accuracy of the rank estimation.

Example 2: We consider the fixed local sample size $n = 1500$ and vary the number of machines $m = 1, 5, 10, 20$. Table S4.2 demonstrates the averaged prediction errors. The results also show that the prediction errors obtained by CSL-RSSTM decrease with m and are lower than the errors of Ave-RSSTM and Sub-RSSTM. Similarly, Ave-RSSTM performs better than Sub-RSSTM.

Example 3: We consider $n = 1500$, $m = 10$ and vary the dimensions $I = 20, 40, 60, 100, 200$. The averaged prediction errors and the results of rank estimation are reported in Tables S4.3 and S4.4, respectively. It can be seen that

prediction errors increase with I, which is consistent with the theoretical result in Corollary 4. In addition, the CSL-RSSTM performs better than the others. Moreover, we can see that the estimated ranks of CSL-RSSTM are near the true value in addition to the large dimension $I = 200$.

Table 1: Rank estimation of different estimators with $N = 10000$ and different values of m under the setting $r = 2, d = -0.5$.

m	CSL		Ave		Sub	
	Gaussian	Epanechnikov	Gaussian	Epanechnikov	Gaussian	Epanechnikov
1	2.16 (0.3004)	2.18 (0.5343)	2.16 (0.3004)	2.18 (0.5343)	2.16 (0.3004)	2.18 (0.5343)
5	1.90 (0.2958)	1.84 (0.4229)	2.23 (0.3219)	2.19 (0.4919)	2.28 (0.2873)	2.36 (0.3739)
10	2.12 (0.5159)	2.02 (0.4690)	2.15 (0.4479)	2.13 (0.5257)	2.54 (0.2943)	2.40 (0.2857)
20	2.32 (0.5078)	2.16 (0.5045)	2.17 (0.5643)	2.24 (0.5381)	2.80 (0.4082)	2.72 (0.3690)

5. Application

In this section, we apply our proposed method to an image dataset for invasive ductal carcinoma (IDC), consisting of 198,738 negative samples and 78,783 positive samples. Each colorful image, of size $50 \times 50 \times 3$, denotes the cell smear of breast tissue. For the sake of storage, we randomly select 10000 negative samples and 10000 positive samples. The proportion of the training and testing set is 9:1 for the purpose of comparison. Moreover, the numbers of negative and

positive samples are balanced in both the training and the testing sets. We set $m = 1, 5, 10, 15, 20$. Figure 2 demonstrates the prediction errors of various estimators increase with the number of machines, while the CSL-based estimator performs well. Table 2 reports the averaged running time and the estimated ranks. Note that the central machine spends expensive computational overhead resulting in the time cost. With increasing number of machines, the time cost of Sub estimator decreases, as expected. We can see that the running time of Ave-RSSTM is around m times than that of Sub-RSSTM, and the CSL estimator is close to the Sub estimator from the respect of computational time. It is clear that the CSL estimator can obtain a higher classification accuracy with few computation time. Since the rank of the true parameter is unknown, we regard the estimated rank by the central machine as the ground-truth rank. It is clear that the rank estimation result of CSL is more close to that of the central estimator.

6. Discussion

In this paper, we consider the statistical rate of the proposed RSSTM estimator, as well as its distributed counterpart via the CSL method. An implementable alternating minimization algorithm is developed with its convergence analysis to obtain the estimator. With some conditions on the number of local machines, the convergence rate of the distributed estimator is consistent with that of the central

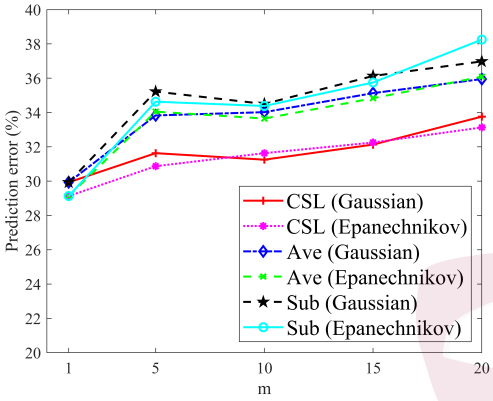


Figure 2: The average prediction error changes with m based on 50 random partitions.

Table 2: Average running time (seconds) and rank estimation with different values of m based on 50 random partitions.

m		CSL		Ave		Sub	
		Gaussian	Epanechnikov	Gaussian	Epanechnikov	Gaussian	Epanechnikov
1	running time	244.97	241.93	244.97	241.93	244.97	241.93
	rank estimation	2.22 (0.5954)	1.98 (0.8061)	2.22 (0.5954)	1.98 (0.8061)	2.22 (0.5954)	1.98 (0.8061)
5	running time	60.23	50.71	237.08	231.91	51.22	46.6
	rank estimation	2.88 (0.7067)	2.02 (0.8351)	4.42 (0.7661)	4.50 (0.5741)	1.76 (0.6914)	1.38 (0.5688)
10	running time	30.18	30.8	227.51	220.85	25.27	23.09
	rank estimation	2.02 (0.6927)	2.28 (0.4725)	5.66 (0.7994)	5.82 (0.6381)	1.42 (0.7767)	1.62 (0.8167)
15	running time	20.05	19.09	213.72	210.43	17.63	14.33
	rank estimation	2.72 (0.7285)	2.12 (0.6467)	5.08 (0.8412)	5.96 (0.5494)	1.54 (0.6397)	1.64 (0.6335)
20	running time	16.88	15.44	209.94	200.08	12.76	10.71
	rank estimation	2.66 (0.5269)	2.14 (0.4197)	6.64 (0.7531)	6.76 (0.5586)	3.88 (0.4746)	2.24 (0.7492)

estimator. A series of numerical experiments illustrates that the proposed distributed estimator performs well. Our proposed method can be easily extended to some sparse regularization (such as $l_{2,1}$ penalty) STMs according to decomposability and convexity, which is an interesting topic in the tensor classification problem.

There are some issues to be further addressed. The theoretical results established in this work only focus on the linear STM while there does exist some nonlinear case in practice. To this end, we could consider a nonlinear tensor function in a reproducing kernel Hilbert space instead of a tensor inner product. Thus, it is of interest that we study the statistical properties of the general kernel-based STM. On the other hand, in some datasets such as genetics data, it is cheap to obtain the covariates compared to the corresponding expensive labels. Another promising direction is leveraging unlabeled data to enhance the performance of STM. Additionally, to better capture the idiosyncratic effects between the score y and the predictor $\boldsymbol{\mathcal{X}}$, we can consider to decompose coefficient tensor $\boldsymbol{\mathcal{B}}$ as $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{S}}$ with a low-rank tensor $\boldsymbol{\mathcal{L}}$ and a structured sparsity tensor $\boldsymbol{\mathcal{S}}$. How to establish the theoretical grantee for both $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{S}}$ by the tensor incoherence condition under the general loss function is a challenging problem. Moreover, some concave techniques (Tan et al., 2021; Wang et al., 2017) can be applied to penalize the singular values of the tensor $\boldsymbol{\mathcal{B}}$ to alleviate the non-

negligible bias induced by the low rank. However, it requires more technical details for the above topics and we leave these in future work.

Supplementary Materials

The preliminaries of the tensor-tensor product (t-product), the proofs of the theorems, and some results of simulations are contained in the Supplementary Materials.

Acknowledgements

This work was supported in part by the National Social Science Fund (22BTJ025), the National Natural Science Fund (12271272,12371272), the Basic Research Project of Shanghai Science and Technology Commission (22JC1400800) and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX24_3622). All authors contributed equally to this work.

References

Chen, C., K. Batselier, C.-Y. Ko, and N. Wong (2019). A support tensor train machine. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8.

Gandy, S., B. Recht, and I. Yamada (2011). Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Probl. 27*, 025010.

Geoffrey, C., G. Lecué, and M. Lerasle (2020). Robust statistical learning with lipschitz and convex loss functions. *Probab. Theory Related Fields 176*, 897–940.

Hao, Z., L. He, B. Chen, and X. Yang (2013). A linear support higher-order tensor machine for classification. *IEEE Trans. Image Process. 22*(7), 2911–2920.

Huang, B., C. Mu, D. Goldfarb, and J. Wright (2015). Provable models for robust low-rank tensor completion. *Pacific J. Optimization. 11*, 339–364.

Jordan, M. I., J. D. Lee, and Y. Yang (2019). Communication-efficient distributed statistical inference. *J. Am. Stat. Assoc. 114*(526), 668–681.

Kilmer, M. and C. Martin (2011). Factorization strategies for third-order tensors. *Linear Algebra Appl. 435*, 641–658.

Kilmer, M. E., K. Braman, N. Hao, and R. C. Hoover (2013). Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM J. Matrix Anal. Appl. 34*(1), 148–172.

Kilmer, M. E., C. D. Martin, and L. Perrone (2008). A third-order generalization of the matrix svd as a product of third-order tensors. *Tufts University, Department of Computer Science, Tech. Rep. TR-2008-4*.

Kolda, T. G. and B. W. Bader (2009). Tensor decompositions and applications. *SIAM Rev. 51*(3), 455–500.

Koltchinskii, V., K. Lounici, and A. B. Tsybakov (2011). Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *Ann. Stat. 39*(5), 2302 – 2329.

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

Kotsia, I. and I. Patras (2011). Support tucker machines. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 633–640.

Kour, K., S. Dolgov, M. Stoll, and P. Benner (2023). Efficient structure-preserving support tensor train machine. *J. Mach. Learn. Res. 24*(4), 1–22.

Lian, H. (2021). Learning rate for convex support tensor machines. *IEEE Trans. Neural Netw. Learn. Syst. 32*(8), 3755–3760.

Lian, H. and Z. Fan (2018). Divide-and-conquer for debiased l_1-norm support vector machine in ultra-high dimensions. *J. Mach. Learn. Res. 18*(182), 1–26.

Lu, C., J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan (2018). Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Trans. Pattern Anal. Mach. Intell. 42*(4), 925–938.

Negahban, S., P. Ravikumar, M. Wainwright, and B. Yu (2010). A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statist. Sci. 27*, 538–557.

Oseledets, I. (2011). Tensor-train decomposition. *SIAM J. Scientific Computing 33*, 2295–2317.

Peng, B., L. Wang, and Y. Wu (2016). An error bound for l1-norm support vector machine coefficients in ultra-high dimension. *J. Mach. Learn. Res. 17*(1), 8279–8304.

Raskutti, G., M. Yuan, and H. Chen (2019). Convex regularization for high-dimensional multiresponse tensor regression. *Ann. Stat. 47*(3), 1554 – 1584.

Roy, S. and G. Michailidis (2022). Regularized high dimension low tubal-rank tensor regression. *Electron. J. Stat. 16*, 2683–2723.

Tan, K. M., L. Wang, and W. X. Zhou (2021). High-dimensional quantile regression: Convolution smoothing and concave regularization. *J. R. Stat. Soc. B. 84*(1), 205–233.

Vapnik, V. (2013). *The Nature of Statistical Learning Theory*. Springer: New York, NY,USA.

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

Wang, L., X. Zhang, and Q. Gu (2017). A unified computational and statistical framework for nonconvex low-rank matrix estimation. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Volume 54 of *Proceedings of Machine Learning Research*, pp. 981–990. PMLR.

Wang, X., Z. Yang, X. Chen, and W. Liu (2019). Distributed inference for linear support vector machine. *J. Mach. Learn. Res. 20*(113), 1–41.

Wang, Y., W. Lu, L. Wang, Z. Zhu, H. Lin, and H. Lian (2025). Regularized adaptive huber matrix regression and distributed learning. *Stat. Sinica 35*, 919–937.

Xu, W., J. Liu, and H. Lian (2024). Distributed estimation of support vector machines for matrix data. *IEEE Trans. Neural Netw. Learn. Syst. 35*(5), 6643–6653.

Zeng, D., S. Wang, Y. Shen, and C. Shi (2017). A ga-based feature selection and parameter optimization for support tucker machine. *Procedia Comput. Sci. 111*, 17–23.

Zihao Song

School of Mathematics and Statistics, Nantong University, Jiangsu Nantong, 226019, China

E-mail: zihaosongntu@126.com

Lei Wang

School of Statistics and Data Science, KLMDASR, LEBPS and LPMC, Nankai University, Tianjin, 300071, China

E-mail: lwangstat@nankai.edu.cn

Riquan Zhang

School of Statistics and Information, Shanghai University of International Business and Economics, Shanghai, 201620, China

E-mail: rqzhang@suibe.edu.cn

Weihua Zhao

School of Mathematics and Statistics, Nantong University, Jiangsu Nantong, 226019, China

E-mail: zhaowhstat@163.com