Lei GUAN, Tao Sun, Lin-bo QIAO, Zhi-hui YANG, Dong-sheng LI, Ke-shi GE, Xi-cheng LU, 2020. An efficient parallel and distributed solution to nonconvex penalized linear SVMs. *Frontiers of Information Technology & Electronic Engineering*, 21(4):587-603. https://doi.org/10.1631/FITEE.1800566

An efficient parallel and distributed solution to nonconvex penalized linear SVMs

Key words: Linear classification; Support vector machine (SVM); Nonconvex penalty; Alternating direction method of multipliers (ADMM); Parallel algorithm

Corresponding author: Dong-sheng LI
E-mail: dsli@nudt.edu.cn
ORCID: https://orcid.org/0000-0001-9743-2034

Motivation

1. Nonconvex penalized SVMs are hard to solve due to their nondifferentiability, nonconvexity, and nonsmoothness.

2. Existing solutions to the nonconvex penalized SVMs typically solve this problem in a serial fashion, which are unable to fully use the parallel computing power of modern multi-core machines.

3. Many real-world data are usually stored in a distributed manner, which urgently calls for a parallel and distributed solution to nonconvex penalized SVMs.

Main idea

1. Reasonable derivations are performed to transform the nondifferentiable, nonconvex, and nonsmooth problem to an equivalent optimization objective that can be solved by applying the consensus ADMM procedure.

2. The parallel algorithm handles the Cholesky factorization of a large matrix by performing Cholesky factorization of small matrices in parallel.

3. Detailed convergence analysis is provided.

4. The proposed parallel algorithm has low time complexity.

Method

Objective:

$$\min_{\{\boldsymbol{w},b\}} \frac{1}{n}\sum_{i=1}^{n}\left[1-y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)\right]_+ + P_\lambda(\boldsymbol{w})$$

Table 1 Summarization of nonconvex regularizers

Name	$p_\lambda(w_j)$
LSP	$\lambda \log\left(1+|w_j|/\theta\right)$
SCAD	$\begin{cases} \lambda|w_j|, & |w_j| \le \lambda, \\ \dfrac{-w_j^2+2\theta\lambda|w_j|-\lambda^2}{2(\theta-1)}, & \lambda < |w_j| \le \theta\lambda, \\ \dfrac{(\theta+1)\lambda^2}{2}, & |w_j| > \theta\lambda. \end{cases}$
MCP	$\begin{cases} \lambda|w_j| - w_j^2/(2\theta), & |w_j| \le \theta\lambda, \\ \theta\lambda^2/2, & |w_j| > \theta\lambda. \end{cases}$
Capped-ℓ_1	$\lambda \min\left(|w_j|,\theta\right)$

$P_\lambda(\boldsymbol{w}) = \sum_{j=1}^{d} p_\lambda(w_j)$. Here, $\theta > 2$ for the SCAD regularizer and $\theta > 0$ for the other regularizers

Method (Cont'd)

Reformulation:

$$\begin{aligned} \min_{\{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{s},\boldsymbol{z}\}} \quad & \frac{1}{n}1_n^{\mathrm{T}}\boldsymbol{\xi} + P_\lambda(\boldsymbol{z}) \\ \text{s.t.} \quad & \boldsymbol{w} = \boldsymbol{z}, \\ & \boldsymbol{Y}(\boldsymbol{X}\boldsymbol{w} + b1_n) + \xi = s + 1_n, \\ & \boldsymbol{\xi} \succeq 0_n, s \succeq 0_n \end{aligned}$$

Scaled-form surrogate augmented Lagrangian function:

$$\begin{aligned} & \mathcal{L}(\boldsymbol{w}_i, b_i, \boldsymbol{z}, \boldsymbol{\xi}_i, \boldsymbol{s}_i, \boldsymbol{u}_i, \boldsymbol{v}_i) \\ = \; & \mathcal{L}_0(\boldsymbol{w}_i, b_i, \boldsymbol{z}, \boldsymbol{\xi}_i, \boldsymbol{s}_i, \boldsymbol{\gamma}_i, \boldsymbol{\tau}_i) + \frac{\rho_1}{2}||\boldsymbol{w}_i - \boldsymbol{z}||_2^2 \\ & + \frac{\rho_2}{2}||\boldsymbol{H}_i\boldsymbol{w}_i + b_i\boldsymbol{y}_i + \boldsymbol{\xi}_i - \boldsymbol{s}_i - 1_{n_i}||_2^2 \\ = \; & P_\lambda(\boldsymbol{z}) + \Big[\frac{1}{n}1_{n_i}^{\mathrm{T}}\boldsymbol{\xi}_i + \frac{\rho_1}{2}||\boldsymbol{w}_i - \boldsymbol{z} + \boldsymbol{u}_i||_2^2 \\ & + \frac{\rho_2}{2}||\boldsymbol{H}_i\boldsymbol{w}_i + b_i\boldsymbol{y}_i + \boldsymbol{\xi}_i - \boldsymbol{s}_i - 1_{n_i} + \boldsymbol{v}_i||_2^2\Big] \\ & + \text{constant} \end{aligned}$$

Method (Cont'd)

ADMM iteration:

$$\boldsymbol{w}_i^{(k+1)} = \arg\min_{\boldsymbol{w}_i} \mathcal{L}\left(\boldsymbol{w}_i, b_i^{(k)}, \boldsymbol{z}^{(k)}, \boldsymbol{\xi}_i^{(k)}, \boldsymbol{s}_i^{(k)}, \boldsymbol{u}_i^{(k)}, \boldsymbol{v}_i^{(k)}\right), \tag{7}$$

$$b_i^{(k+1)} = \arg\min_{b_i} \mathcal{L}\left(\boldsymbol{w}_i^{(k+1)}, b_i, \boldsymbol{z}^{(k)}, \boldsymbol{\xi}_i^{(k)}, \boldsymbol{s}_i^{(k)}, \boldsymbol{u}_i^{(k)}, \boldsymbol{v}_i^{(k)}\right), \tag{8}$$

$$\boldsymbol{z}^{(k+1)} = \arg\min_{\boldsymbol{z}} \mathcal{L}\left(\boldsymbol{w}_i^{(k+1)}, b_i^{(k+1)}, \boldsymbol{z}, \boldsymbol{\xi}_i^{(k)}, \boldsymbol{s}_i^{(k)}, \boldsymbol{u}_i^{(k)}, \boldsymbol{v}_i^{(k)}\right), \tag{9}$$

Method (Cont'd)

$$\boldsymbol{\xi}_i^{(k+1)} = \underset{\boldsymbol{\xi}_i \succeq \mathbf{0}_{n_i}}{\arg\min}\ \mathcal{L}\left(\boldsymbol{w}_i^{(k+1)}, b_i^{(k+1)}, \boldsymbol{z}^{(k+1)}, \boldsymbol{\xi}_i, \boldsymbol{s}_i^{(k)}, \boldsymbol{u}_i^{(k)}, \boldsymbol{v}_i^{(k)}\right), \tag{10}$$

$$\boldsymbol{s}_i^{(k+1)} = \underset{\mathbf{s}_i \succeq \mathbf{0}_{n_i}}{\arg\min}\ \mathcal{L}\left(\boldsymbol{w}_i^{(k+1)}, b_i^{(k+1)}, \boldsymbol{z}^{(k+1)}, \boldsymbol{\xi}_i^{(k+1)}, \boldsymbol{s}_i, \boldsymbol{u}_i^{(k)}, \boldsymbol{v}_i^{(k)}\right), \tag{11}$$

$$\boldsymbol{u}_i^{(k+1)} = \boldsymbol{u}_i^{(k)} + \left(\boldsymbol{w}_i^{(k+1)} - \boldsymbol{z}^{(k+1)}\right), \tag{12}$$

$$\boldsymbol{v}_i^{(k+1)} = \boldsymbol{v}_i^{(k)} + \left(\boldsymbol{\xi}_i^{(k+1)} - \boldsymbol{s}_i^{(k+1)} + \boldsymbol{H}_i \boldsymbol{w}_i^{(k+1)} + b_i \boldsymbol{y}_i - \mathbf{1}_{n_i}\right). \tag{13}$$

Method (Cont'd)

Optimization of computation:

$$\boldsymbol{w}_i^{(k+1)} = (\rho_1 \boldsymbol{I}_d + \rho_2 \boldsymbol{H}_i^{\mathrm{T}} \boldsymbol{H}_i)^{-1} \Big[\rho_1 (\boldsymbol{z}^{(k)} - \boldsymbol{u}_i^{(k)}) + \rho_2 \boldsymbol{H}_i^{\mathrm{T}} \left(\boldsymbol{s}_i^{(k)} + 1_{n_i} - \boldsymbol{\xi}_i^{(k)} - \boldsymbol{v}_i^{(k)} - b_i^{(k)} \boldsymbol{y}_i \right) \Big],$$

Reduce to

$$\boldsymbol{w}_i^{(k+1)} = \begin{cases} \boldsymbol{U}_i^{-1}(\boldsymbol{L}_i^{-1} \boldsymbol{f}_i^{(k)}), & \text{if } n_i \geq d, \\ \dfrac{\boldsymbol{f}_i^{(k)}}{\rho} - \dfrac{(\boldsymbol{H}_i^{\mathrm{T}}(\boldsymbol{U}_i^{-1}(\boldsymbol{L}_i^{-1}(\boldsymbol{H}_i \boldsymbol{f}_i^{(k)}))))}{\rho^2}, & \text{otherwise,} \end{cases}$$

where $\boldsymbol{L}_i$ and $\boldsymbol{U}_i$ are the Cholesky factorization of $(\rho \boldsymbol{I}_d + \boldsymbol{H}_i^{\mathrm{T}} \boldsymbol{H}_i)$ if $n_i \geq d$, and the Cholesky factorization of $\left(\boldsymbol{I}_{n_i} + \frac{1}{\rho} \boldsymbol{H}_i \boldsymbol{H}_i^{\mathrm{T}} \right)$ otherwise.

Method (Cont'd)

Optimization of synchronization:

$$\text{AllReduce } \psi_i^{(k+1)} \text{ and obtain } \sum_{i=1}^{K} \psi_i^{(k+1)}$$

$$\text{AllReduce } \sum_{j=1}^{n_i} \xi_{ij}^{(k+1)}, \text{obtain } \sum_{i=1}^{K} \sum_{j=1}^{n_i} \xi_{ij}^{(k+1)}$$

Reduce to

$$\text{Merge } \psi_i^{(k+1)} \text{ and } \sum_{j=1}^{n_i} \xi_{ij}^{(k+1)}$$
$$\text{AllReduce } \psi_i^{(k+1)} \text{ and } \sum_{j=1}^{n_i} \xi_{ij}^{(k+1)}$$

Major results

1. Comparison with the serial algorithm

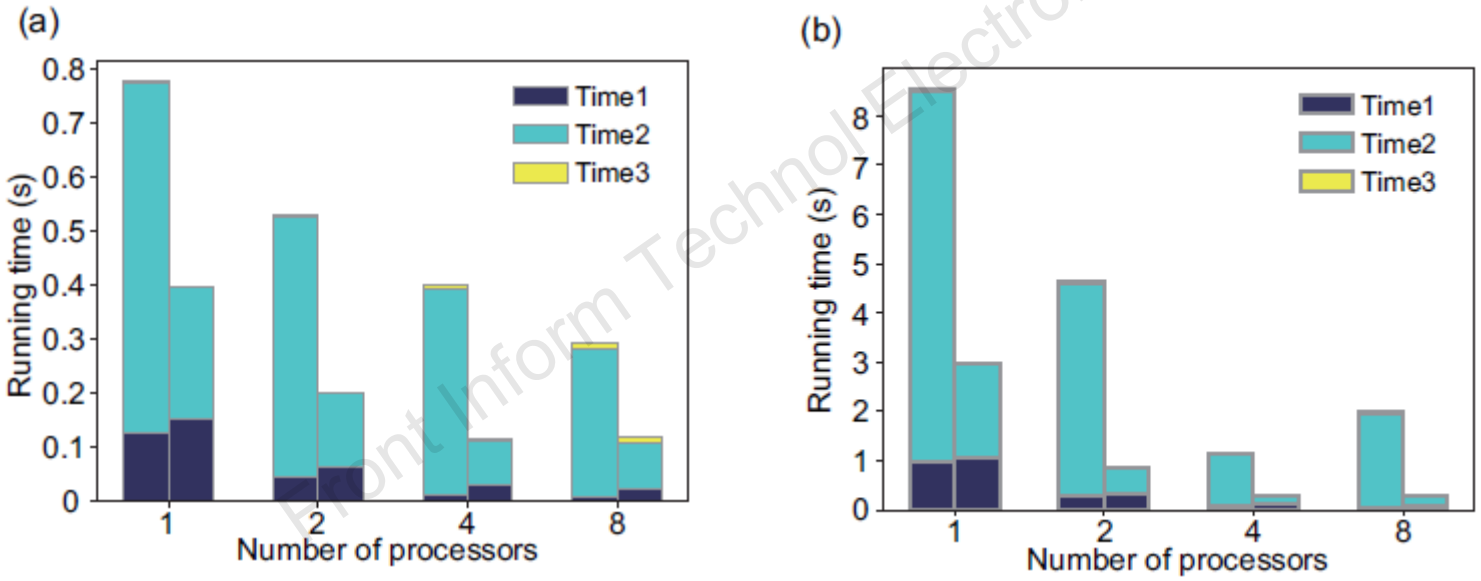

Fig. 1 Performance breakdown: (a) a2a; (b) mushrooms. Time1 denotes the pre-computation time; time2 refers to the iteration time; time3 stands for the synchronization time. All variables were measured in CPU seconds

Major results (Cont'd)

2. Evaluation results on news20 and rcv1 datasets

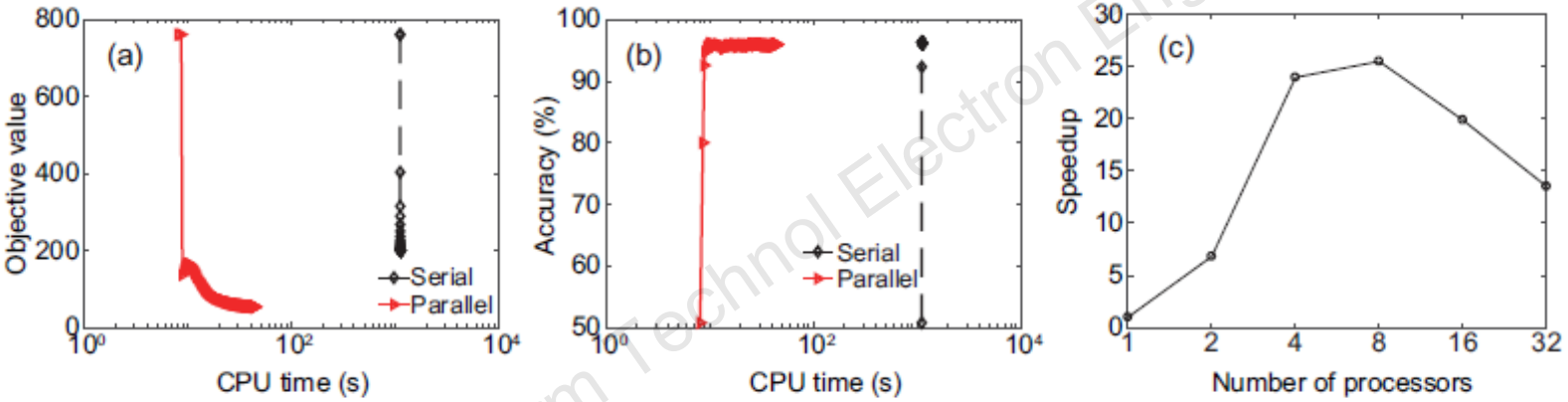

Fig. 2 Evaluation results on the new20 dataset: (a) objective; (b) accuracy; (c) speedup

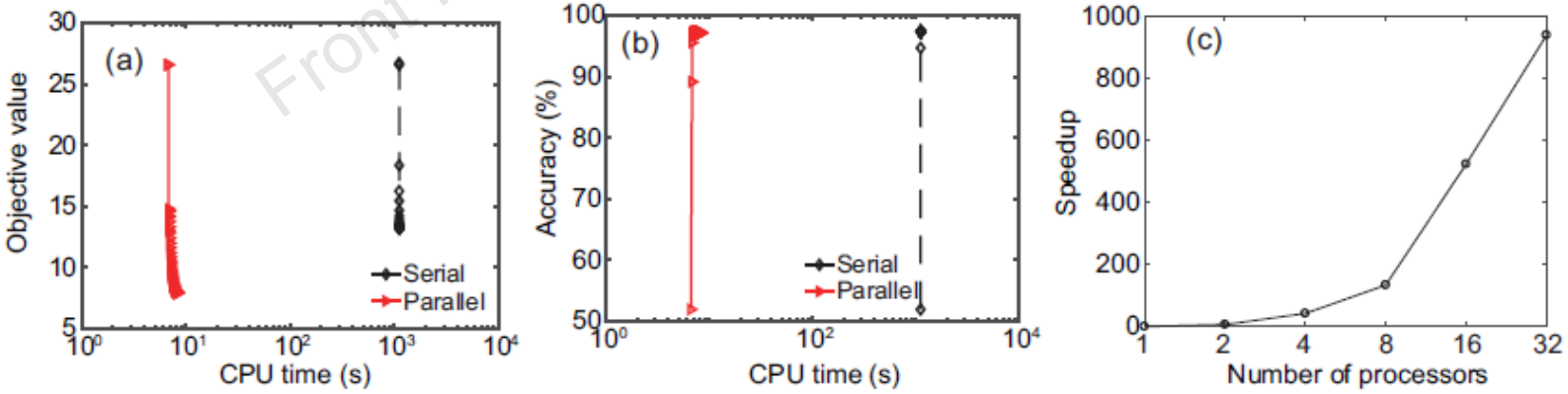

Fig. 3 Evaluation results on the rcv1 dataset: (a) objective; (b) accuracy; (c) speedup

Major results (Cont'd)

3. Breakdown results on news20 and rcv1 datasets

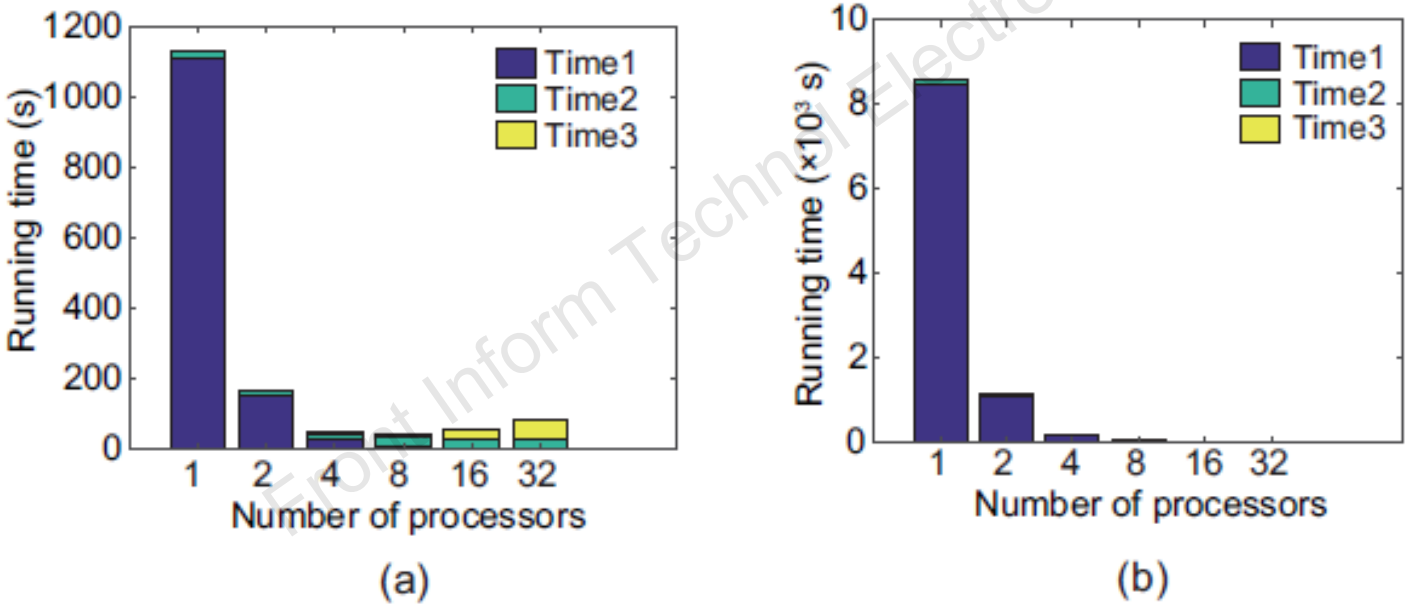

Fig. 4 Breakdown results for the running time: (a) news20; (b) rcv1

Conclusions

1. We have proposed, analyzed, and evaluated an efficient algorithm in a parallel and distributed environment.

2. Many efficient schemes have been adopted to decrease the computation and synchronization cost.

3. The convergence of the proposed algorithm is guaranteed.