

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:

$$\min_{\{w, b\}} \frac{1}{n} \sum_{i=1}^n [1 - y_i(w^T x_i + b)]_+ + P_\lambda(w)$$

Table 1 Summarization of nonconvex regularizers

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

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

Method (Cont'd)

Reformulation:

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

Scaled-form surrogate augmented Lagrangian function:

$$\begin{aligned} & \mathcal{L}(w_i, b_i, z, \xi_i, s_i, u_i, v_i) \\ = & \mathcal{L}_0(w_i, b_i, z, \xi_i, s_i, \gamma_i, \tau_i) + \frac{\rho_1}{2} \|w_i - z\|_2^2 \\ & + \frac{\rho_2}{2} \|H_i w_i + b_i y_i + \xi_i - s_i - \mathbf{1}_{n_i}\|_2^2 \\ = & P_\lambda(z) + \left[\frac{1}{n} \mathbf{1}_{n_i}^T \xi_i + \frac{\rho_1}{2} \|w_i - z + u_i\|_2^2 \right. \\ & \left. + \frac{\rho_2}{2} \|H_i w_i + b_i y_i + \xi_i - s_i - \mathbf{1}_{n_i} + v_i\|_2^2 \right] \\ & + \text{constant.} \end{aligned}$$

Method (Cont'd)

ADMM iteration:

$$w_i^{(k+1)} = \arg \min_{w_i} \mathcal{L} \left(w_i, b_i^{(k)}, z^{(k)}, \xi_i^{(k)}, s_i^{(k)}, \right. \\ \left. u_i^{(k)}, v_i^{(k)} \right), \quad (7)$$

$$b_i^{(k+1)} = \arg \min_{b_i} \mathcal{L} \left(w_i^{(k+1)}, b_i, z^{(k)}, \xi_i^{(k)}, s_i^{(k)}, \right. \\ \left. u_i^{(k)}, v_i^{(k)} \right), \quad (8)$$

$$z^{(k+1)} = \arg \min_z \mathcal{L} \left(w_i^{(k+1)}, b_i^{(k+1)}, z, \xi_i^{(k)}, s_i^{(k)}, \right. \\ \left. u_i^{(k)}, v_i^{(k)} \right), \quad (9)$$

Method (Cont'd)

$$\xi_i^{(k+1)} = \arg \min_{\xi_i \succeq 0_{n_i}} \mathcal{L} \left(w_i^{(k+1)}, b_i^{(k+1)}, z^{(k+1)}, \xi_i, s_i^{(k)}, \right. \\ \left. u_i^{(k)}, v_i^{(k)} \right), \quad (10)$$

$$s_i^{(k+1)} = \arg \min_{s_i \succeq 0_{n_i}} \mathcal{L} \left(w_i^{(k+1)}, b_i^{(k+1)}, z^{(k+1)}, \xi_i^{(k+1)}, \right. \\ \left. s_i, u_i^{(k)}, v_i^{(k)} \right), \quad (11)$$

$$u_i^{(k+1)} = u_i^{(k)} + \left(w_i^{(k+1)} - z^{(k+1)} \right), \quad (12)$$

$$v_i^{(k+1)} = v_i^{(k)} + \left(\xi_i^{(k+1)} - s_i^{(k+1)} + H_i w_i^{(k+1)} \right. \\ \left. + b_i y_i - 1_{n_i} \right). \quad (13)$$

Method (Cont'd)

Optimization of computation:

$$w_i^{(k+1)} = (\rho_1 I_d + \rho_2 H_i^T H_i)^{-1} \left[\rho_1 (z^{(k)} - u_i^{(k)}) + \rho_2 H_i^T \left(s_i^{(k)} + 1_{n_i} - \xi_i^{(k)} - v_i^{(k)} - b_i^{(k)} y_i \right) \right],$$

Reduce to

$$w_i^{(k+1)} = \begin{cases} U_i^{-1}(L_i^{-1} f_i^{(k)}), & \text{if } n_i \geq d, \\ \frac{f_i^{(k)}}{\rho} - \frac{(H_i^T (U_i^{-1}(L_i^{-1}(H_i f_i^{(k)}))))}{\rho^2}, & \text{otherwise,} \end{cases}$$

where L_i and U_i are the Cholesky factorization of $(\rho I_d + H_i^T H_i)$ if $n_i \geq d$, and the Cholesky factorization of $\left(I_{n_i} + \frac{1}{\rho} H_i H_i^T \right)$ otherwise.

Method (Cont'd)

Optimization of synchronization:

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

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

Reduce to

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

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

Major results

1. Comparison with the serial algorithm

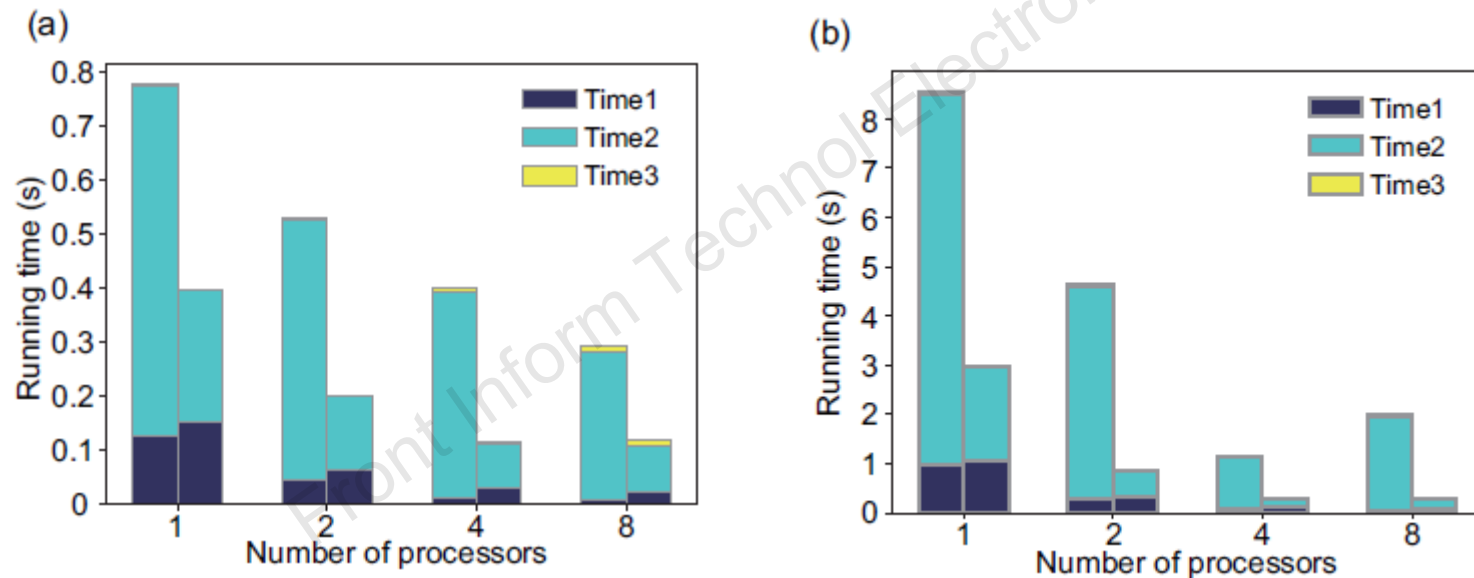


Fig. 1 Performance breakdown: (a) a2a; (b) mushrooms. Time1 denotes the pre-computation time; time2 refers to the iteration time; time3 stands for the synchronization time. All variables were measured in CPU seconds

Major results (Cont'd)

2. Evaluation results on news20 and rcv1 datasets

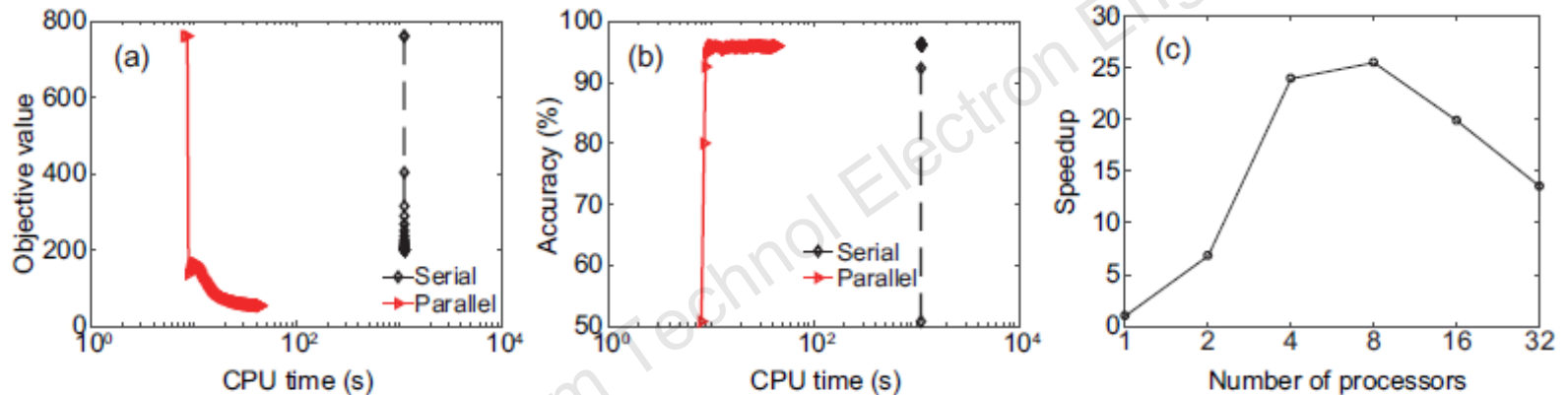


Fig. 2 Evaluation results on the new20 dataset: (a) objective; (b) accuracy; (c) speedup

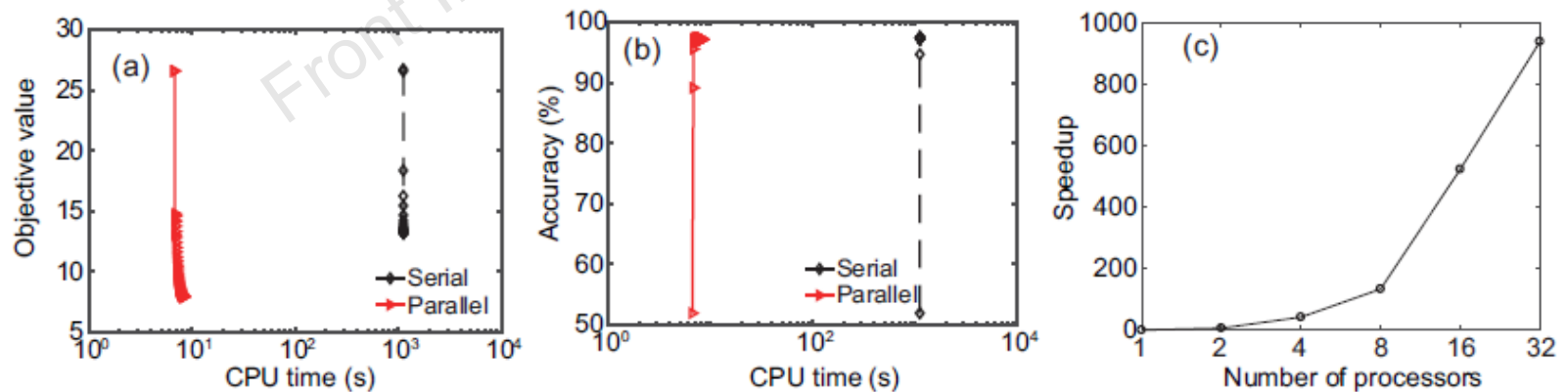


Fig. 3 Evaluation results on the rcv1 dataset: (a) objective; (b) accuracy; (c) speedup

Major results (Cont'd)

3. Breakdown results on news20 and rcv1 datasets

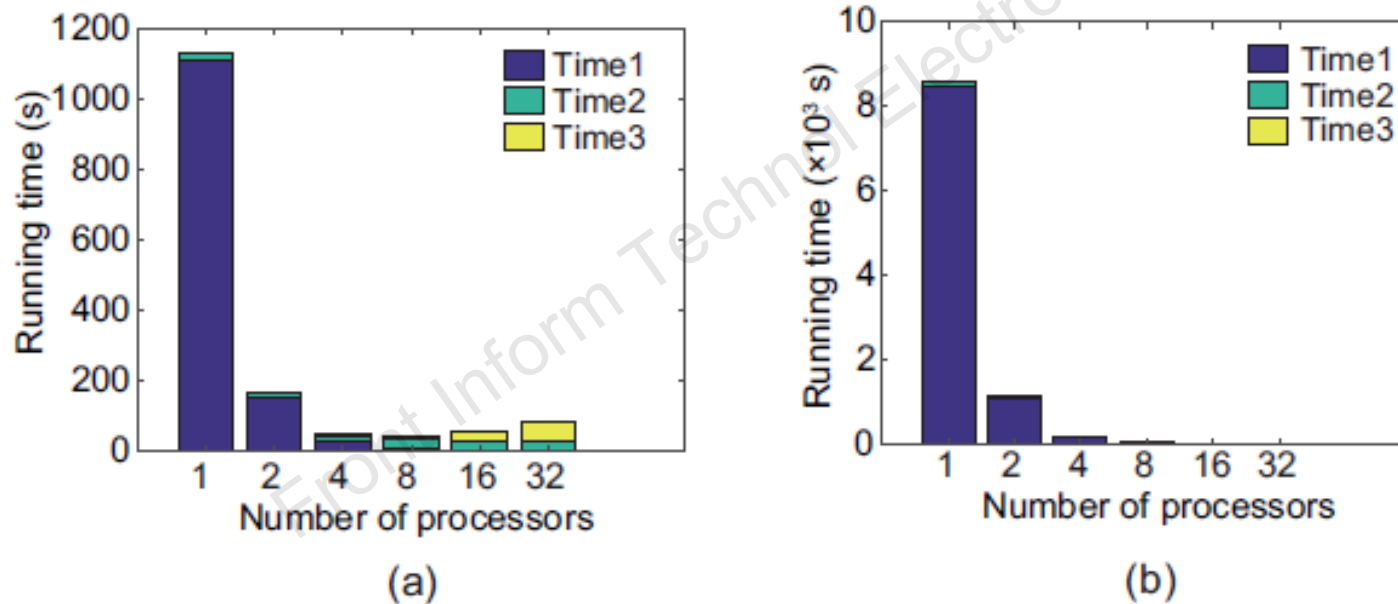


Fig. 4 Breakdown results for the running time: (a) news20; (b) rcv1

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.