

A Dual-Augmented Block Minimization Framework for Learning with Limited Memory

Ian E.H. Yen¹, Shan-Wei Lin², Shou-De Lin²

¹University of Texas at Austin. ²National Taiwan University

Abstract

- In this work, we consider **Empirical Risk Minimization (ERM)** when data size is larger than the memory capacity of machines.
- State-of-the-art **batch algorithms** become slow due to I/O.
- Online algorithms** converge slowly (especially for non-smooth regularizer), while existing **distributed approach** requires data to fit into memory of several machines.
- We propose a **Block Minimization** framework that generalizes (Yu. et al. 2010) for SVM to that for any convex ERM, which can be integrated with **any convex optimization solver** to achieve global fast convergence in limited-memory condition.

Regularized Empirical Risk Minimization (ERM)

Given a data set $\mathcal{D} = \{(\Phi_n, \mathbf{y}_n)\}_{n=1}^N$, the ERM estimates model through

$$\min_{\mathbf{w} \in \mathbb{R}^d} F(\mathbf{w}) = R(\mathbf{w}) + \sum_{n=1}^N L_n(\Phi_n \mathbf{w}) \quad (1)$$

- $\mathbf{w} \in \mathbb{R}^d$ are parameters to be estimated, Φ_n is $p \times d$ feature matrix of n -th sample, and $L_n(\cdot)$, $R(\cdot)$ are loss function and regularizer.

Examples

- Multiclass Classification:** ($p = |\mathcal{Y}|$, where \mathcal{Y} :label set).
Logistic loss: $L_n(\xi) = \log(\sum_{k \in \mathcal{Y}} \exp(\xi_k)) - \xi_{y_n}$.
Hinge loss: $L_n(\xi) = \max_{k \in \mathcal{Y}} (1 - \delta_{k, y_n} + \xi_k - \xi_{y_n})$.
- Multitask Regression:** ($p = K$, where $K = \#$ tasks)
Square loss: $L_n(\xi) = \frac{1}{2} \|\xi - \mathbf{y}_n\|^2$.
- Others:** Ranking, Matrix Completion, Structured Learning, Clustering etc..
- Regularizers:** L2 norm $\lambda \|\mathbf{w}\|^2$, L1 norm $\lambda \|\mathbf{w}\|_1$, Group norm $\lambda \|W\|_g$, Nuclear norm $\lambda \|W\|_*$, and etc.

Strong Convexity & Smoothness

- A function $f(x)$ is **strongly convex** iff it is lower bounded by a simple quadratic function

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) + \frac{m}{2} \|\mathbf{x} - \mathbf{y}\|^2 \quad (2)$$

for some constant $m > 0$ and $\forall \mathbf{x}, \mathbf{y} \in \text{dom}(f)$.

- A function $f(x)$ is **smooth** iff it is upper bounded by a simple quadratic function

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) + \frac{M}{2} \|\mathbf{x} - \mathbf{y}\|^2 \quad (3)$$

for some constant $0 \leq M < \infty$ and $\forall \mathbf{x}, \mathbf{y} \in \text{dom}(f)$.

- Theorem 1:** A convex function $f(\cdot)$ is smooth with parameter M if and only if its convex conjugate $f^*(\cdot)$ is strongly convex with parameter $m = 1/M$.

Dual Form

The dual of ERM problem (1) is of the form

$$\min_{\alpha_n \in \mathbb{R}^p} G(\alpha) = R^* \left(- \sum_{n=1}^N \Phi_n^T \alpha_n \right) + \sum_{n=1}^N L_n^*(\alpha_n). \quad (4)$$

- Block Coordinate Descent on (4) guarantees convergence only for **smooth** $R^*(\cdot)$ (**strongly convex** $R(\cdot)$), which does not hold for most of regularizers.
- Use **Proximal Minimization** to ensure convergence for any convex ERM.

Dual-Augmented Block Minimization

- The Dual-Augmented Lagrangian method (or equivalently, Primal Proximal Minimization) solves a series of **augmented sub-problems**

$$\hat{\mathbf{w}}^{t+1} = \arg \min_{\mathbf{w}} F(\mathbf{w}) + \frac{1}{2\eta_t} \|\mathbf{w} - \hat{\mathbf{w}}^t\|^2, \quad (5)$$

which, by Theorem 1, has a dual problem of **smooth** $\tilde{R}^*(\cdot)$ since the augmented regularizer $\tilde{R}(\mathbf{w})$ is **strongly convex**.

- Let $\mathcal{L}(\mathbf{w}, \alpha)$ be the Lagrangian of (5). Our algorithm performs **Block-Coordinate Descent** on dual of (5), which minimizes block of variables α_B via

$$\begin{aligned} \max_{\alpha_B} \min_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \alpha) &= \min_{\mathbf{w}} \max_{\alpha_B} \mathcal{L}(\mathbf{w}, \alpha) \\ &= \min_{\mathbf{w}} R(\mathbf{w}) + \sum_{n \in B} L_n(\Phi_n \mathbf{w}) + \mu_B^T \mathbf{w} + \frac{1}{2\eta_t} \|\mathbf{w} - \hat{\mathbf{w}}^t\|^2, \end{aligned} \quad (6)$$

which requires only data in block B and can be solved via **any solver designed for (1)**, where vector μ_B memorizes historical gradient given by data not in B :

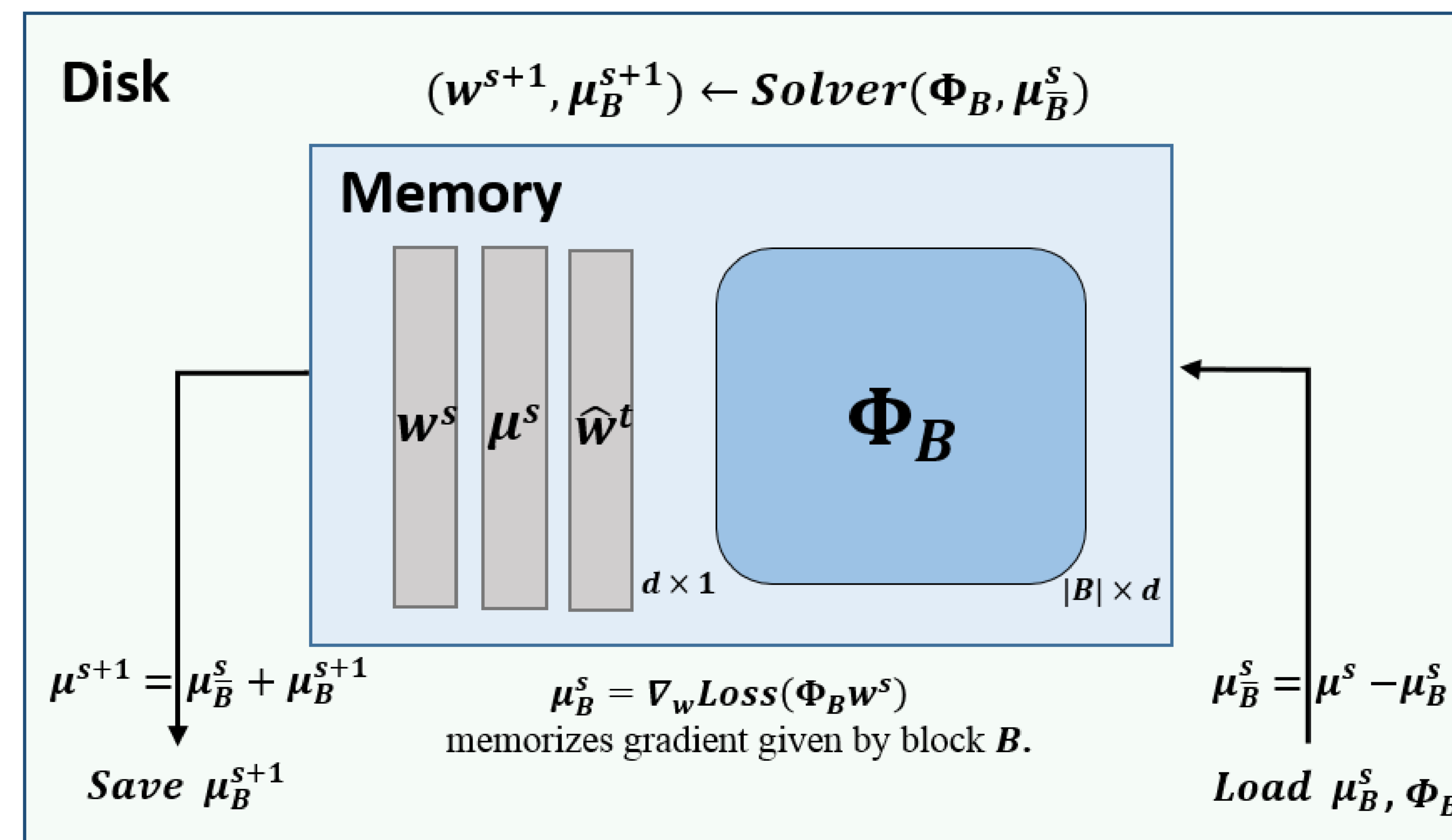
$$\mu_B = \sum_{n \notin B} \Phi_n^T \alpha_n = \sum_{n=1}^N \Phi_n^T \alpha_n - \sum_{n \in B} \Phi_n^T \alpha_n = \mu - \mu_B \quad (7)$$

- After solving block sub-problem (6), we obtain new α_B^* and μ_B^* via

$$\alpha_B^* = \nabla_{\xi_B} \text{Loss}(\xi_B^* = \Phi_B \mathbf{w}^*) \quad \mu_B^* = \Phi_B^T \alpha_B^* = \nabla_{\mathbf{w}} \text{Loss}(\Phi_B \mathbf{w}^*). \quad (8)$$

- Only one of α_B or μ_B needs to be maintained. If $d > |B|p$, **maintaining** α_B is cheaper; otherwise, **maintaining** μ_B is more space-efficient.

Algorithmic Framework



Dual-Augmented Block Minimization Algorithm

- Split data \mathcal{D} into blocks B_1, B_2, \dots, B_K .
- Initialize $\hat{\mathbf{w}}^0 = \mathbf{0}$, $\mu^0 = \mathbf{0}$.
- for** $t = 0, 1, \dots$ (outer iteration) **do**
 - for** $s = 0, 1, \dots, S$ **do**
 - Draw B uniformly from B_1, B_2, \dots, B_K .
 - Load \mathcal{D}_B , μ_B^s (or α_B^s) into memory.
 - Solve (6) to obtain \mathbf{w}^* .
 - Maintain μ_B^{s+1} (or α_B^{s+1}) through relation (8).
 - Maintain $\mu^{s+1} = \mu_B^s + \mu_B^{s+1}$.
 - Save μ_B^{s+1} (or α_B^{s+1}) out of memory.
 - end for**
- $\hat{\mathbf{w}}^{t+1} = \mathbf{w}^*(\alpha^S)$.
- end for**

Convergence of Block Minimization

- The dual of (5) takes the form

$$\min_{\alpha_n \in \mathbb{R}^p} \tilde{R}^* \left(- \sum_{n=1}^N \Phi_n^T \alpha_n \right) + \sum_{n=1}^N L_n^*(\alpha_n) \quad (9)$$

where $\tilde{R}^*(\cdot)$ is the convex conjugate of $\tilde{R}(\mathbf{w}) = R(\mathbf{w}) + \frac{1}{2\eta_t} \|\mathbf{w} - \hat{\mathbf{w}}^t\|^2$.

- Since $\tilde{R}(\mathbf{w})$ is **strongly convex** with parameter $m = 1/\eta_t$, the convex conjugate $\tilde{R}^*(\cdot)$ is **smooth** with parameter $M = \eta_t$ according to Theorem 1.

- The augmented dual (9) is composite of a **convex, smooth** function plus a **convex, block-separable** function, for which BCD has guaranteed convergence to optimum. In particular, with probability $1 - \rho$

$$\tilde{F}^*(\alpha^s) - \tilde{F}^* \leq \epsilon, \quad \text{for } s \geq \beta K \log \left(\frac{\tilde{F}^*(\alpha^0) - \tilde{F}^*}{\rho \epsilon} \right) \quad (10)$$

for some constant $\beta > 0$ if (i) $L_n(\cdot)$ is smooth, or (ii) $L_n(\cdot)$ is polyhedral and $R(\cdot)$ is also polyhedral or smooth. Otherwise, for any convex $L_n(\cdot)$, $R(\cdot)$,

$$\tilde{F}^*(\alpha^s) - \tilde{F}^* \leq \epsilon, \quad \text{for } s \geq \frac{cK}{\epsilon} \log \left(\frac{\tilde{F}^*(\alpha^0) - \tilde{F}^*}{\rho \epsilon} \right), \quad (11)$$

with some constant $c > 0$ and probability $1 - \rho$.

Convergence of Overall Procedure

- The sequence $\{\hat{\mathbf{w}}^t\}_{t=1}^{\infty}$ produced by Proximal Minimization (5) with $\eta_t = \eta$ and radius of initial level set \mathcal{R} has

$$F(\hat{\mathbf{w}}^{t+1}) - F \leq \epsilon, \quad \text{for } t \geq \tau \log \left(\frac{\omega}{\epsilon} \right). \quad (12)$$

for some constant $\tau, \omega > 0$ if both $L_n(\cdot)$ and $R(\cdot)$ are (i) strictly convex and smooth or (ii) polyhedral. Otherwise, for any convex $F(\mathbf{w})$ we have

$$F(\hat{\mathbf{w}}^{t+1}) - F \leq \mathcal{R}^2 / (2\eta t). \quad (13)$$

- Due to non-expansiveness of proximal operator, we show that solving sub-problem (5) with tolerance ϵ/t suffices for convergence to ϵ overall precision where t is the number of outer iterations required by (12), (13).

- The **overall procedure** requires $O(K \log(1/\epsilon) \log(t/\epsilon)) = O(K \log^2(1/\epsilon))$ block minimization steps if $L_n(\cdot)$, $R(\cdot)$ are **strictly convex and smooth**, or **polyhedral**. Otherwise, we need $O(K(1/\epsilon) \log(t/\epsilon)) = O(\frac{K}{\epsilon} \log(1/\epsilon))$ block minimization steps as long as $L_n(\cdot)$ is **smooth**.

Experiments

Data	#train	#test	dimension	#non-zeros	Memory (GB)	Block
webspam	315,000	31,500	680,714	1,174,704,031	20.7	2.07
rcv1	202,420	20,242	7,951,176	656,977,694	12.0	2.20
year-pred	463,715	51,630	2,000	927,893,715	13.7	1.38
E2006	16,087	3,308	30,000	8,088,636	8.08	0.80

