Supplement of “ Limited-memory Common-directions Method for Distributed Optimization and its Application on Empirical Risk Minimization”

Ching-pei Lee[*] Po-Wei Wang[†] Weizhu Chen[‡] Chih-Jen Lin[§]

I Introduction

In this document, we present more experimental results, and detailed proofs for the theorems stated in the main paper.

II Detailed derivations for (4.37)

We show how to compute $P_k^T \boldsymbol{u}_{k-1}$ efficiently. From (2.15), we have

$$\boldsymbol{\psi}^T \boldsymbol{u}_{k-1} = \theta_{k-1} \boldsymbol{\psi}^T P_{k-1} \boldsymbol{t}_{k-1}, \forall \boldsymbol{\psi} \in \mathbf{R}^n.$$

Therefore,

$$\begin{aligned}&\left(P_k^T \boldsymbol{u}_{k-1}\right)_i \\ =&\begin{cases} \theta_{k-1} M^{k-1}_{i+2,:} \boldsymbol{t}_{k-1} & \text{if } i < 2m-1 \\ \theta_{k-1}^2 \boldsymbol{t}_{k-1}^T M^{k-1} \boldsymbol{t}_{k-1} & \text{if } i = 2m-1 \\ \boldsymbol{u}_{k-1}^T \boldsymbol{s}_{k-1} = \left(P_k^T \boldsymbol{s}_{k-1}\right)_{2m-1} & \text{if } i = 2m \\ \boldsymbol{u}_{k-1}^T \nabla f(\boldsymbol{w}_k) = (P_k^T \nabla f(\boldsymbol{w}_k))_{2m-1} & \text{if } i = 2m+1 \end{cases},\end{aligned}$$

For $i < 2m-1$, we have

$$\begin{aligned}&(P_k^T \boldsymbol{u}_{k-1})_i \\ =& \theta_{k-1} (P_k^T P_{k-1} \boldsymbol{t}_{k-1})_i \\ =& \theta_{k-1} M^k_{i+2,:} \boldsymbol{t}_{k-1}.\end{aligned}$$

For $i = 2m-1$, from (2.15) we have

$$\begin{aligned}&(P_k^T \boldsymbol{u}_{k-1})_i \\ =& \theta_{k-1} \boldsymbol{u}_{k-1}^T P_{k-1} \boldsymbol{t}_{k-1} \\ =& \theta_{k-1} \boldsymbol{t}_{k-1}^T P_{k-1}^T P_{k-1} \boldsymbol{t}_{k-1} \\ =& \theta_{k-1} \boldsymbol{t}_{k-1}^T M^{k-1} \boldsymbol{t}_{k-1}.\end{aligned}$$

III Proof of Theorem 3.1

The solution of (3.16) also solves the following linear system.

$$P_k^T H_k P_k \boldsymbol{t} = -P_k^T \nabla f(\boldsymbol{w}_k), \tag{III.1}$$

where $P_k \equiv [\boldsymbol{q}_1, \dots, \boldsymbol{q}_m]$. If $\boldsymbol{q}_j$ satisfies (3.18), then the right-hand side of (III.1) is not all zero, hence $\boldsymbol{t} \neq \mathbf{0}$ and $P_k \boldsymbol{t} \neq \mathbf{0}$. Therefore, from (III.1), we have

(III.2)
$$-\boldsymbol{p}_k^T \nabla f(\boldsymbol{w}_k) = (P_k \boldsymbol{t})^T H_k P_k \boldsymbol{t} \geq M_2 \|P_k \boldsymbol{t}\|^2 = M_2 \|\boldsymbol{p}_k\|^2.$$

We then have from Assumption 1 and (III.2) that

$$\begin{aligned}f(\boldsymbol{w}_k + \theta_k \boldsymbol{p}_k) &\leq f(\boldsymbol{w}_k) + \theta_k \nabla f(\boldsymbol{w}_k)^T \boldsymbol{p}_k + \theta_k^2 \frac{\rho}{2} \|\boldsymbol{p}_k\|^2 \\ &\leq f(\boldsymbol{w}_k) + \theta_k \nabla f(\boldsymbol{w}_k)^T \boldsymbol{p}_k (1 - \frac{\rho \theta_k}{2M_2}).\end{aligned}$$

From (III.2), $\nabla f(\boldsymbol{w}_k)^T \boldsymbol{p}_k < 0$. Therefore, when

$$1 - \frac{\rho \theta_k}{2M_2} \geq c_1,$$

(2.14) is satisfied. Thus, we have that the backtracking line search gets a step size θ_k such that

$$\theta_k \geq \min\left(1, \frac{2\beta\left(1-c_1\right) M_2}{\rho}\right). \tag{III.3}$$

Therefore, the backtracking line search procedure takes at most $\min\left(0, \lceil \log_\beta (2\beta(1-c_1)(M_2/\rho)) \rceil\right)$ steps.

IV Proof of Theorem 3.2

The j-th equation in the linear system (III.1) is

$$\boldsymbol{p}_k^T H_k \boldsymbol{q}_j = -\nabla f(\boldsymbol{w}_k)^T \boldsymbol{q}_j. \tag{IV.1}$$

By (3.17), (3.18), and (IV.1),

$$\begin{aligned}\|\boldsymbol{p}_k\| \|\boldsymbol{q}_j\| &\geq |\frac{1}{M_1} (\boldsymbol{p}_k)^T H_k \boldsymbol{q}_j| \\ &= |\frac{1}{M_1} \nabla f(\boldsymbol{w}_k)^T \boldsymbol{q}_j| \geq \|\nabla f(\boldsymbol{w}_k)\| \|\boldsymbol{q}_j\| \frac{\delta}{M_1}.\end{aligned}$$

Therefore,

$$\|\boldsymbol{p}_k\| \geq \frac{\delta}{M_1} \|\nabla f(\boldsymbol{w}_k)\|. \tag{IV.2}$$

Combining (III.2) and (IV.2), we can establish the following result.

$$-\frac{\boldsymbol{p}_k^T \nabla f(\boldsymbol{w}_k)}{\|\boldsymbol{p}_k\| \|\nabla f(\boldsymbol{w}_k)\|} \geq \frac{M_2 \|\boldsymbol{p}_k\|^2}{\|\boldsymbol{p}_k\| \|\nabla f(\boldsymbol{w}_k)\|} \geq \frac{\delta M_2}{M_1}.$$

[*] University of Wisconsin-Madison. ching-pei@cs.wisc.edu
[†] Carnegie Mellon University. poweiw@cs.cmu.edu
[‡] Microsoft. wzchen@microsoft.com
[§] National Taiwan University. cjlin@csie.ntu.edu.tw

Now to show the convergence, by (2.14), (III.2), and (IV.2), we have

$$\begin{aligned} f(\boldsymbol{w}_{k+1}) &\leq f(\boldsymbol{w}_k) + \theta_k c_1 \nabla f(\boldsymbol{w}_k)^T \boldsymbol{p}_k \\ &\leq f(\boldsymbol{w}_k) - \frac{M_2 \delta^2 \theta_k c_1}{M_1^2} \|\nabla f(\boldsymbol{w}_k)\|^2. \end{aligned}$$

From (III.3), we can replace θ_k in the above result with some positive constant κ and get

$$f(\boldsymbol{w}_{k+1}) \leq f(\boldsymbol{w}_k) - \frac{M_2 \delta^2 \kappa c_1}{M_1^2} \|\nabla f(\boldsymbol{w}_k)\|^2. \tag{IV.3}$$

Thus, summing (IV.3) from $\boldsymbol{w}_0$ to $\boldsymbol{w}_k$, we get

$$\begin{aligned} \sum_{j=0}^{k} \frac{M_2 \delta^2 \kappa c_1}{M_1^2} \|\nabla f(\boldsymbol{w}_j)\|^2 &\leq f(\boldsymbol{w}_0) - f(\boldsymbol{w}_{k+1}) \\ &\leq f(\boldsymbol{w}_0) - f^*, \end{aligned} \tag{IV.4}$$

where f^* is the minimal objective value of f. Consequently,

$$\begin{aligned} \min_{0 \leq j \leq k} \|\nabla f(\boldsymbol{w}_j)\|^2 &\leq \frac{1}{k+1} \sum_{j=0}^{k} \|\nabla f(\boldsymbol{w}_j)\|^2 \\ &\leq \frac{1}{k+1} \frac{M_1^2}{M_2 \delta^2 \kappa c_1} (f(\boldsymbol{w}_0) - f^*) \end{aligned}$$

and (3.19) follows. Note that since

$$\sum_{j=0}^{k} \|\nabla f(\boldsymbol{w}_j)\|^2$$

is bounded, we have that $\|\nabla f(\boldsymbol{w}_k)\|$ converges to zero as k approaches infinity.

Now consider that f satisfies (3.20). Deducting f^* from both sides of (IV.3) and combining it with (3.20), we get for all k

$$f(\boldsymbol{w}_{k+1}) - f^* \leq (1 - \frac{2\sigma M_2 \delta^2 \kappa c_1}{M_1^2})(f(\boldsymbol{w}_k) - f^*), \tag{IV.5}$$

and thus to get $f(\boldsymbol{w}_k) - f^* \leq \epsilon$, it takes $O(\log(1/\epsilon))$ iterations. Note that our assumptions give $c_1 > 0$, and $\sigma M_2 / M_1^2 > 0$. Therefore the coefficient in the right-hand side of (IV.5) is smaller than 1.

V Proof of Corollary 3.1

We note that in Algorithm 1, because the gradient itself is included in the directions, (3.18) is satisfied with $\delta = 1$. Moreover, (3.17) is satisfied by $(M_1, M_2) = (\rho, \sigma)$ from Assumption 1 and the strong convexity. Thus from (III.3),

$$\theta_k \geq \min\left(1, \frac{2\beta\left(1 - c_1\right)\sigma}{\rho}\right) \geq \frac{\beta(1 - c_1)\sigma}{\rho}.$$

Note that in the second inequality, we utilized the fact that $\beta, 1 - c_1, \sigma/\rho \in [0, 1]$. Therefore, the backtracking line search procedure takes at most $\lceil \log_\beta(\beta(1 - c_1)(\sigma/\rho)) \rceil$ steps, and we can take

$$\kappa = \frac{\beta(1 - c_1)\sigma}{\rho}. \tag{V.1}$$

Now to show the linear convergence, the Polyak-Łojasiewicz condition (3.20) holds with σ from the strong convexity by [1, Theorem 2.1.10] and noting that $\nabla f(\boldsymbol{w}^*) = \mathbf{0}$. Therefore, by (V.1), (IV.5) becomes

$$f(\boldsymbol{w}_{k+1}) - f^* \leq (1 - \frac{2\beta c_1 (1 - c_1) \sigma^3}{\rho^3})(f(\boldsymbol{w}_k) - f^*),$$

and the required iterations to reach an ϵ-accurate solution is therefore

$$k \geq \frac{\log\left(f\left(\boldsymbol{w}_0\right) - f^*\right) + \log\left(\frac{1}{\epsilon}\right)}{\log\left(1/\left(1 - \frac{2\beta c_1 (1 - c_1)\sigma^3}{\rho^3}\right)\right)}.$$

VI Convergence for Regularized Neural Networks

In this section, we show that the framework discussed in Section 3 applies to the algorithm for the L2-regularized nonconvex neural network problem considered in [2], with a minor condition in solving the linear system below in (VI.3), and therefore the result of Theorem 3.2 provides a convergence rate guarantee for their algorithm.

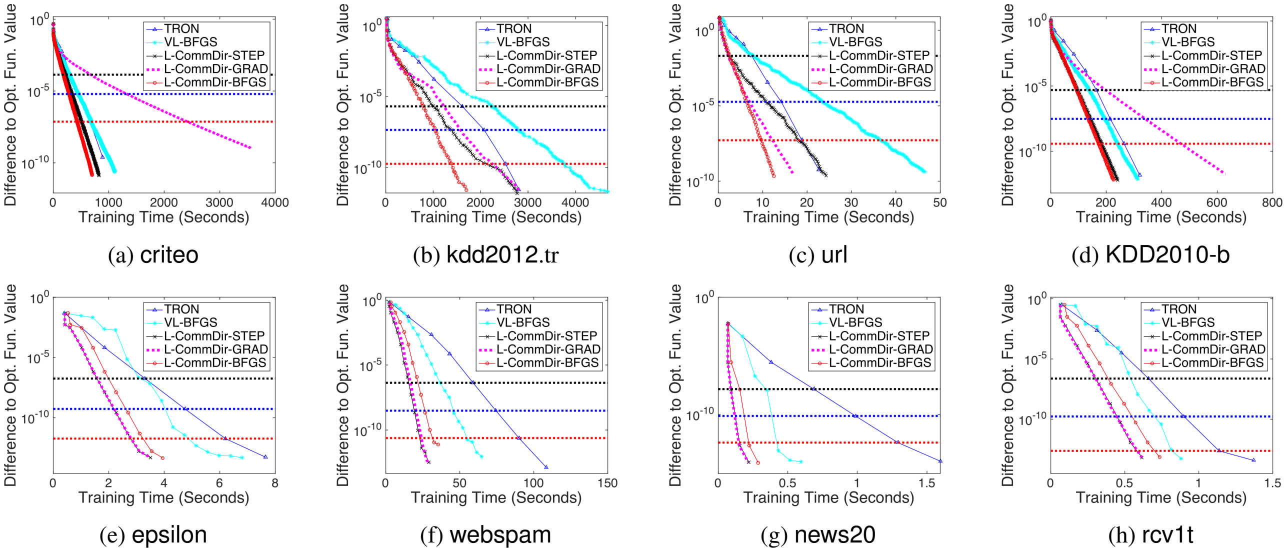
The neural network problem considered in [2] is of the following form.

$$\min_{\boldsymbol{\theta}} \quad g(\boldsymbol{\theta}) \equiv \frac{1}{2}\|\boldsymbol{\theta}\|^2 + C \sum_{i=1}^{l} \xi(\boldsymbol{\theta}; \boldsymbol{x}_i, y_i), \tag{VI.1}$$

where $\boldsymbol{\theta}$ is the collection of all weights between two adjacent layers of the neural network denoted as a vector, $C > 0$ is a parameter specified by users, $\{(y_i, \boldsymbol{x}_i)\}, i = 1, \ldots, l$ are the training instances, and ξ is nonnegative and twice continuously differentiable with respect to $\boldsymbol{\theta}$. Therefore, g is twice differentiable. Because ξ is nonnegative and a descent method is used, given any initial point $\boldsymbol{\theta}_0$, the sequence of points generated by their algorithm is confined in the level set

$$\{\boldsymbol{\theta} \mid \frac{1}{2}\|\boldsymbol{\theta}\|^2 \leq \frac{1}{2}\|\boldsymbol{\theta}_0\|^2 + C \sum_{i=1}^{l} \xi(\boldsymbol{\theta}_0; \boldsymbol{x}_i, y_i)\}, \tag{VI.2}$$

which is compact. Therefore, as a continuous function, $\|\nabla^2 g(\boldsymbol{\theta})\|$ is upper-bounded in this compact set, which is equivalent to that the $\|\nabla g(\boldsymbol{\theta}_1) - \nabla g(\boldsymbol{\theta}_2)\| / \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|$ is upper bounded by some finite value. Thus Assumption 1 is satisfied.



(a) criteo (b) kdd2012.tr (c) url (d) KDD2010-b

(e) epsilon (f) webspam (g) news20 (h) rcv1t

Figure (I): Comparison of different algorithms with $C = 10^{-3}$. We show training time v.s. relative difference to the optimal function value. The horizontal lines indicate when the algorithm is terminated in practice by marking the stopping condition of TRON in MPI-LIBLINEAR: $\|\nabla f(\boldsymbol{w})\| \leq \epsilon \frac{\min(\#y_i=1,\#y_i=-1)}{l} \|\nabla f(\mathbf{0})\|$, with $\epsilon = 10^{-2}$ (default), 10^{-3}, and 10^{-4}.

Recall from (2.2) that [2] combines two directions $\boldsymbol{d}_k$ and $\boldsymbol{p}_{k-1}^{\text{SN}}$, where $\boldsymbol{d}_k$ is the stochastic Newton direction at the current iterate $\boldsymbol{\theta}_k$. We will show that the direction $\boldsymbol{d}_k$ satisfies (3.18). The linear system solved in [2] to obtain $\boldsymbol{d}_k$ is

$$\text{(VI.3)} \qquad (G_{S_k} + \lambda_k I)\boldsymbol{d}_k = -\nabla f(\boldsymbol{\theta}_k),$$

where S_k is a random subset of instances, G_{S_k} is a subsampled Gauss-Newton matrix corresponding to S_k at $\boldsymbol{\theta}_k$, $\lambda_k > 0$ is a damping factor, and I is the identity matrix. The needed condition here is that λ_k is upper- and lower-bounded by some positive constants. If $\lambda_k \geq \lambda > 0$ always holds, then with the positive semi-definiteness of G_{S_k}, we see that

$$\text{(VI.4)} \quad -\boldsymbol{d}_k^T \nabla f(\boldsymbol{\theta}_k) = \boldsymbol{d}_k^T (G_{S_k} + \lambda_k I)\boldsymbol{d}_k \geq \lambda \|\boldsymbol{d}_k\|^2.$$

On the other hand, because $\boldsymbol{\theta}_k$ is in the compact set (VI.2) and G_{S_k} is a continuous function of $\boldsymbol{\theta}_k$ under a fixed S_k, if λ_k is upper bounded, since there are only finite choices of S_k as a subset of a finite set, there exists $\gamma > 0$ such that

$$\text{(VI.5)} \quad \|\boldsymbol{d}_k\| = \|(G_{S_k} + \lambda_k I)^{-1} \nabla f(\boldsymbol{\theta}_k)\| \geq \gamma \|\nabla f(\boldsymbol{\theta}_k)\|.$$

From (VI.4) and (VI.5),

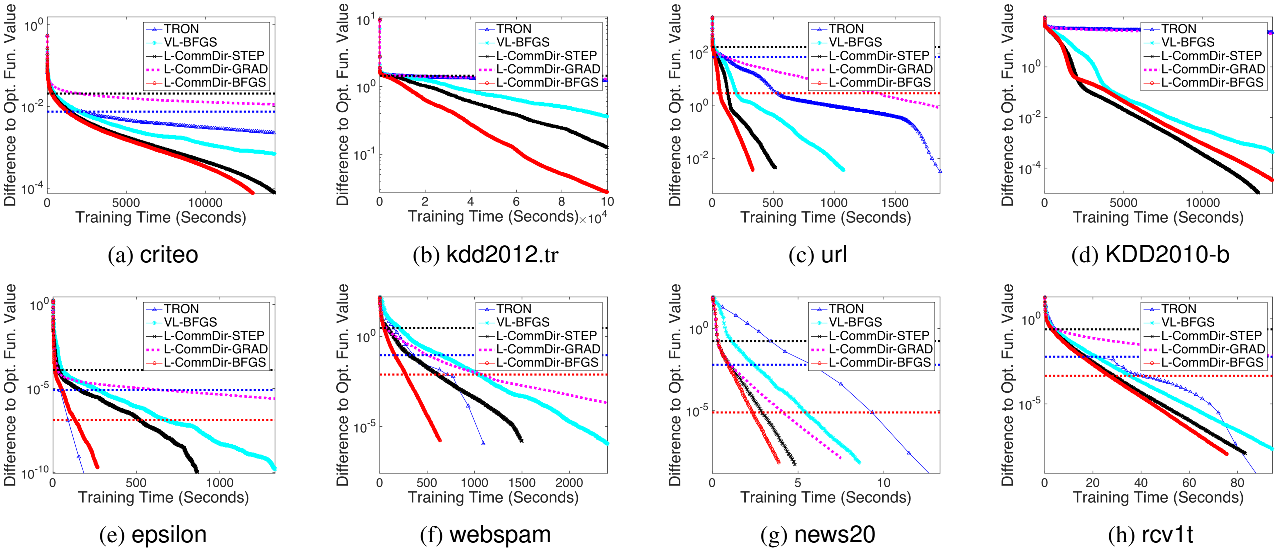
$$\left| \boldsymbol{d}_k^T \nabla f(\boldsymbol{\theta}_k) \right| \geq \lambda \|\boldsymbol{d}_k\|^2 \geq \lambda\gamma \|\boldsymbol{d}_k\| \|\nabla f(\boldsymbol{\theta}_k)\|,$$

and hence (3.18) holds. Therefore all conditions of Theorem 3.2 are satisfied. Because the iterates lie in a compact set, there are convergent sub-sequences of the iterates, and every limit point of the iterates is stationary by Theorem 3.2.

VII More Experiments

We present more experimental results that are not included in the main paper in this section. We consider the same experiment environment, and the same problem being solved. We present the results using different values of C to see the relative efficiency when the problems become more difficult or easier. The result of $C = 10^{-3}$ is shown in Figure (I), and the result of $C = 1,000$ is shown in Figure (II). For $C = 10^{-3}$, the problems are easier to solve. We observe that L-CommDir-BFGS is faster than existing methods on all data sets, and L-CommDir-Step outperforms state of the art on all data sets but url, but is still competitive on url. For $C = 1,000$, L-CommDir-BFGS is the fastest on most data sets, and the only exception is KDD2010-b, on which L-CommDir-BFGS is slightly slower than L-CommDir-Step, but faster than other methods. On the other hand, L-CommDir-Step is slower than TRON on webspam but faster than existing methods on other data sets. These results show that our method is highly efficient by using (2.11) or (2.12) to decide P_k. The other choice, L-CommDir-Grad, is obviously inferior for most cases.

We also modify from TRON to obtain a line-search truncated Newton solver to compare with our method. This line-search truncated Newton method is denoted by NEWTON in the results in Figures (III)-(V). Results show that NEWTON is consistently the fastest on criteo for all choices of C, but L-CommDir-BFGS and L-CommDir-Step are faster in most other cases.



(a) criteo

(b) kdd2012.tr

(c) url

(d) KDD2010-b

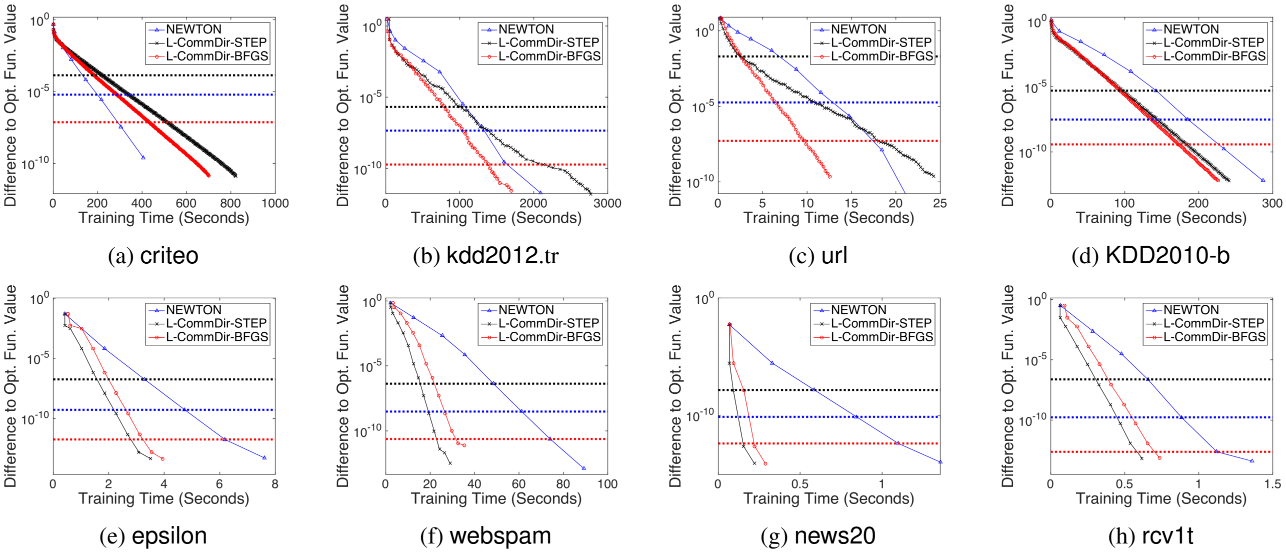
(e) epsilon

(f) webspam

(g) news20

(h) rcv1t

Figure (II): Comparison of different algorithms with $C = 1,000$. We show training time v.s. relative difference to the optimal function value. The horizontal lines indicate when the algorithm is terminated in practice by marking the stopping condition of TRON in MPI-LIBLINEAR: $\|\nabla f(\boldsymbol{w})\| \leq \epsilon \frac{\min(\#y_i=1,\#y_i=-1)}{l} \|\nabla f(\mathbf{0})\|$, with $\epsilon = 10^{-2}$ (default), 10^{-3}, and 10^{-4}.



(a) criteo

(b) kdd2012.tr

(c) url

(d) KDD2010-b

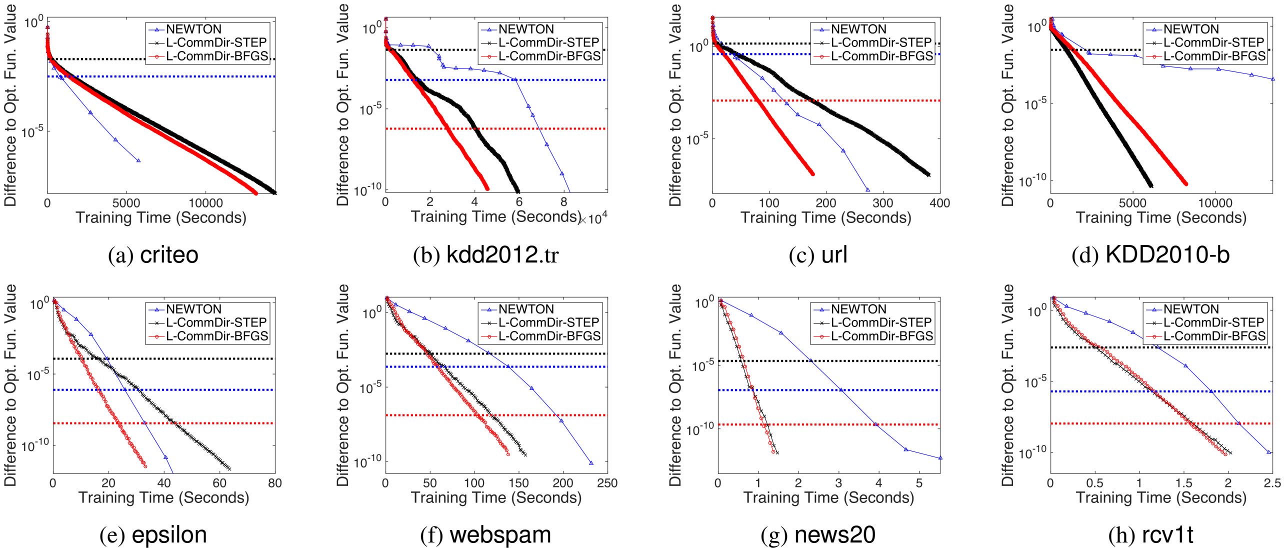
(e) epsilon

(f) webspam

(g) news20

(h) rcv1t

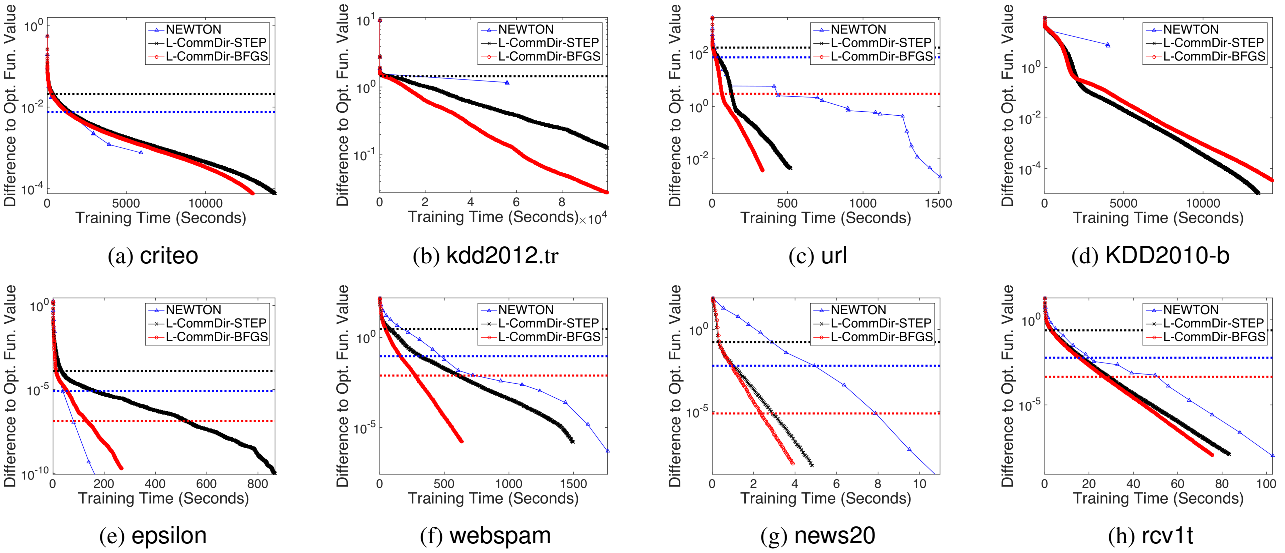
Figure (III): Comparison with line-search truncated Newton with $C = 10^{-3}$. We show training time v.s. relative difference to the optimal function value. The horizontal lines indicate when the algorithm is terminated in practice by marking the stopping condition of TRON in MPI-LIBLINEAR: $\|\nabla f(\boldsymbol{w})\| \leq \epsilon \frac{\min(\#y_i=1,\#y_i=-1)}{l} \|\nabla f(\mathbf{0})\|$, with $\epsilon = 10^{-2}$ (default), 10^{-3}, and 10^{-4}.



(a) criteo (b) kdd2012.tr (c) url (d) KDD2010-b

(e) epsilon (f) webspam (g) news20 (h) rcv1t

Figure (IV): Comparison with line-search truncated Newton with $C = 1$. We show training time v.s. relative difference to the optimal function value. The horizontal lines indicate when the algorithm is terminated in practice by marking the stopping condition of TRON in MPI-LIBLINEAR: $\|\nabla f(\boldsymbol{w})\| \leq \epsilon \frac{\min(\#y_i=1, \#y_i=-1)}{l} \|\nabla f(\mathbf{0})\|$, with $\epsilon = 10^{-2}$ (default), 10^{-3}, and 10^{-4}.



(a) criteo (b) kdd2012.tr (c) url (d) KDD2010-b

(e) epsilon (f) webspam (g) news20 (h) rcv1t

Figure (V): Comparison with line-search truncated Newton with $C = 1,000$. We show training time v.s. relative difference to the optimal function value. The horizontal lines indicate when the algorithm is terminated in practice by marking the stopping condition of TRON in MPI-LIBLINEAR: $\|\nabla f(\boldsymbol{w})\| \leq \epsilon \frac{\min(\#y_i=1, \#y_i=-1)}{l} \|\nabla f(\mathbf{0})\|$, with $\epsilon = 10^{-2}$ (default), 10^{-3}, and 10^{-4}.

References

[1] Y. E. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2003.

[2] C.-C. Wang, C.-H. Huang, and C.-J. Lin. Subsampled Hessian Newton methods for supervised learning. *Neural Comput.*, 27:1766–1795, 2015.