Limited-memory Common-directions Method With Subsampled Newton Directions for Large-scale Linear Classification

Jui-Nan Yen
National Taiwan University
juinanyen@gmail.com

Chih-Jen Lin
National Taiwan University
cjlin@csie.ntu.edu.tw

***Abstract*—The common-directions method is an optimization method recently proposed to utilize second-order information. It is especially efficient on large-scale linear classification problems, and it is competitive with state-of-the-art optimization methods like BFGS, LBFGS, and Nesterov's accelerated gradient method. The main idea of the method is to minimize the local quadratic approximation within the selected subspace. Regarding the selection of the subspace, the original authors only focused on the span of current and past gradient directions. In this work, we analyze the impact of subspace selection, and point out that the lack of direction diversity can be a potential weakness for using gradients as directions. To address this problem, we propose the use of subsampled Newton directions, which always possess diversity unless they are already close to the true Newton direction. Our experiments on large-scale linear classification problems show that our proposed methods are generally better than subsampled Newton methods and the original common-directions method.**

I. Introduction

The common-directions method was proposed by Wang et al. [1] as an interpolation between first- and second-order methods for regularized empirical risk minimization problems. The main idea of the method is to minimize the local quadratic approximation within the selected subspace. Their experiments on large-scale linear classification problems show that it is competitive with state-of-the-art optimization methods like BFGS [2] and Nesterov's accelerated gradient method [3].

The limited-memory version of the common-directions method was then developed by Lee et al. [4]. Their theoretical results show that it has global linear convergence for convex problems and converges to stationary points for non-convex problems. A similar method called the subspace Newton method was later proposed by Gower et al. [5].

Regarding the selection of the subspace, Gower et al. [5] simply use some randomly chosen vectors. On the other hand, inspired by the heavy-ball method and the BFGS method, Wang et al. [1] and Lee et al. [4] considered the span of current and past gradient directions.

In this work, we assume the loss function to be twice-differentiable, Lipschitz smooth, and strictly convex. We then analyze the impact of subspace selection, and point out that the lack of direction diversity can be a potential weakness for using gradients as directions. To address this problem, we propose the use of subsampled Newton directions [6], which always possess diversity unless they are already close to the true Newton direction. Our experiments on large-scale linear classification problems show that our proposed methods are generally better than subsampled Newton methods and the original common-directions method.

The paper is organized as follows. In Section II, we introduce the common-directions method. In Section III, we analyze the impact of subspace selection and point out the lack of direction diversity can be a potential weakness for the original common-directions method. In Section IV, we propose to use subsampled Newton directions with the common-directions method, which does not possess the same weakness. We discuss the convergence of our proposed method in Section V. We put some other algorithmic considerations in Section VI. Empirical comparisons are conducted in Section VII. Finally, Section VIII concludes our work.

We put the code for our experiments and the additional experiment results at https://www.csie.ntu.edu.tw/~cjlin/papers/commdir_subsampled.

II. Review of Linear Classification and the Common-Directions Method

Given a set of training instances $(y_i, \mathbf{x}_i), i = 1, \dots, l$, where y_i is a label and $\mathbf{x}_i \in R^n$ is a feature vector, a supervised learning problem can be formulated as the following regularized empirical risk minimization problem

$$\min_{\mathbf{w}} f(\mathbf{w}) \equiv \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l} \xi(\mathbf{w}; \mathbf{x}_i, y_i), \tag{1}$$

where $\mathbf{w}^T\mathbf{w}/2$ is the L2-regularization term, ξ is a loss function parametrized by a weight vector $\mathbf{w} \in R^n$, and $C > 0$ is a parameter to balance the two terms.

In this work, we assume ξ to be twice-differentiable, Lipschitz smooth, and strictly convex. In particular, we consider the logistic loss

$$\xi_{\text{LR}} = \log(1 + \exp(-y\mathbf{w}^T\mathbf{x}))$$

for large-scale linear classification problems, where the number of instances l and/or the number of features n are large.

The common-directions method was proposed by Wang et al. [1] as an interpolation between first- and second-order methods for solving (1). The limited-memory version was later developed by Lee et al. [4], which is the focus of this work.

Let

$$q_k(\mathbf{s}) \equiv \frac{1}{2}\mathbf{s}^T H_k \mathbf{s} + \mathbf{g}_k^T \mathbf{s} \approx f(\mathbf{w}_k + \mathbf{s}) - f(\mathbf{w}_k)$$

be the quadratic approximation at the current iterate $\mathbf{w}_k$, where $\mathbf{g}_k \equiv \nabla f(\mathbf{w}_k)$ and $H_k \equiv \nabla^2 f(\mathbf{w}_k)$. The common-directions method first chooses a set of directions

$$P_k = [\mathbf{p}_1, \ldots, \mathbf{p}_m],$$

and then computes the update direction

$$\mathbf{u}_k = P_k \mathbf{t}_k,$$

where $\mathbf{t}_k$ is the solution of

$$\min_{\mathbf{t}} q_k(P_k \mathbf{t}) = \frac{1}{2}(P_k \mathbf{t})^T H_k (P_k \mathbf{t}) + \mathbf{g}_k^T (P_k \mathbf{t}). \tag{2}$$

After a suitable step size α_k was decided by line search, the next iterate is then computed by

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k \mathbf{u}_k.$$

To solve (2), we consider its first-order condition,

$$(P_k^T H_k P_k)\mathbf{t}_k + P_k^T \mathbf{g}_k = \mathbf{0}. \tag{3}$$

After we compute and store $P_k^T H_k P_k$ and $P_k^T \mathbf{g}_k$, we can then solve the linear system (3) in $O(m^3)$.

To make the computation of $P_k^T H_k P_k$ more efficient, Lee et al. [4] showed that the convergence results will still hold if we replace the Hessian matrix H_k with any positive definite matrix B_k.

Furthermore, Lee et al. [4] showed that for linear classification problems, where ξ in (1) can be represented as a function of $\mathbf{w}^T \mathbf{x}$, we can compute $P_k^T H_k P_k$ exactly with efficiency if P_k consists of a search direction $\tilde{\mathbf{s}}_k$ and $m-1$ past directions.

The search direction $\tilde{\mathbf{s}}_k$ can be the subsampled Newton direction as in this work or the gradient $\mathbf{g}_k$ as proposed by Wang et al. [1]. For the $m-1$ past directions, they can be past search and/or past update directions. In this work, we use the **Mixed** strategy proposed by Lee et al. [4], where half of the past directions are past search directions and the other half are past update directions.

III. Weakness for Using Gradients as Directions

In this section, we analyze the original common-directions method and point out that the lack of direction diversity can be a potential weakness for using gradients as directions. We put the proofs of the theorems in the appendix.

A. Notation

For convenience, we first define some notations which will be used in our analysis. We use

$$\mathbf{s}_k = -H_k^{-1}\mathbf{g}_k \tag{4}$$

to denote the Newton direction, which is the minimizer of $q_k(\mathbf{s})$. Furthermore, we omit the subscript k when there is no confusion. In particular,

$$\mathbf{g}_k \to \mathbf{g}, H_k \to H, \mathbf{s}_k \to \mathbf{s}.$$

We use the math operator

$$\operatorname*{mean}_i[\ ]$$

to denote taking the average over i, where i belongs to a finite set.

For each iteration with $\mathbf{g} \neq \mathbf{0}$, we define

$$\nu(\alpha, \mathbf{u}) \equiv \frac{q_k(\alpha \mathbf{u})}{q_k(\mathbf{s})} \tag{5}$$

and

$$\mu(\mathbf{u}) \equiv \max_{\alpha} \nu(\alpha, \mathbf{u}) = \frac{\min_\alpha q_k(\alpha \mathbf{u})}{q_k(\mathbf{s})} \tag{6}$$

to measure the strength of an arbitrary direction $\mathbf{u}$ against the Newton direction $\mathbf{s}$. Since $q_k(\mathbf{0}) = 0$, we always have

$$\min_{\alpha} q_k(\alpha \mathbf{u}) \leq 0.$$

Furthermore, since $\mathbf{s}$ is the minimizer of q_k, which is strongly convex, and we have $\mathbf{g} \neq \mathbf{0}$, it follows that

$$q_k(\mathbf{s}) = -\frac{1}{2}\mathbf{g}^T H^{-1} \mathbf{g} < 0 \quad \text{and} \quad 0 \leq \mu(\mathbf{u}) \leq 1.$$

We generalize μ to a set of directions P_k as

$$\mu(P_k) \equiv \frac{\min_{\mathbf{t}} q_k(P_k \mathbf{t})}{q_k(\mathbf{s})}. \tag{7}$$

For vectors $\mathbf{u}, \mathbf{v} \neq \mathbf{0}$ and a positive definite matrix A, we define

$$\tau_A(\mathbf{u}, \mathbf{v}) \equiv \frac{|\mathbf{v}^T A \mathbf{u}|}{\|A^{\frac{1}{2}}\mathbf{u}\|_2 \|A^{\frac{1}{2}}\mathbf{v}\|_2} \tag{8}$$

to measure their similarity. Due to the Cauchy inequality, we always have

$$0 \leq \tau_A(\mathbf{u}, \mathbf{v}) \leq 1.$$

Furthermore, since A is positive definite, we have $\tau_A(\mathbf{u}, \mathbf{v}) = 1$ if and only if $\mathbf{u} = \mathbf{v}$ up to a scale factor.

We also define $\mathbf{\Delta}_k = \mathbf{w}^* - \mathbf{w}_k$ and $H^* \equiv \nabla^2 f(\mathbf{w}^*)$ to prove some convergence properties, where $\mathbf{w}^*$ is the global minimum of f.

B. Interpretation for ν, μ, and τ

Wang et al. [7] prove that ν is strongly related to convergence under the following assumption.

Assumption 1: The Hessian matrix $\nabla^2 f(\mathbf{w})$ is Lipschitz continuous with parameter $\hat{L}$, i.e.,

$$\|\nabla^2 f(\mathbf{w}) - \nabla^2 f(\mathbf{w}')\|_2 \leq \hat{L}\|\mathbf{w} - \mathbf{w}'\|_2$$

and f is strongly convex.

More specifically, Lemma 9 of Wang et al. [7] indicates that for an arbitrary optimization method, if the update direction $\mathbf{u}$ and the step size α satisfy $\nu(\alpha, \mathbf{u}) > \bar{\nu}$ for every iteration, then $\mathbf{\Delta}_k^T H^* \mathbf{\Delta}_k$ converges linearly locally with rate $(1-\bar{\nu})/\bar{\nu}$.

To ensure $(1-\bar{\nu})/\bar{\nu} < 1$, one must have $\bar{\nu} > 1/2$. We improve their result and give the following theorem, which has a smaller convergence rate and only requires $\bar{\nu} > 0$.

Theorem 1: Let Assumption 1 hold and $\bar{\nu} \in (0, 1)$ be a fixed constant. If at every iteration, we have $\nu(\alpha, \mathbf{u}) \geq \bar{\nu}$, then $\mathbf{\Delta}_k^T H^* \mathbf{\Delta}_k$ converges linearly locally with rate $1 - \bar{\nu}$.

It is worth noticing that the purpose of this theorem is to show that ν and μ are good measures for the strength of our update directions. Our proposed method does not rely on this theorem to obtain convergence guarantees, so we do not require Assumption 1.

To connect μ and τ, we give the following proposition.

Proposition 1: For vector $\mathbf{u}$, Hessian H, Newton direction $\mathbf{s}$, we have $\tau_H(\mathbf{u}, \mathbf{s})^2 = \mu(\mathbf{u})$.
Therefore, a direction $\mathbf{u}$ more similar to the Newton direction $\mathbf{s}$ under τ_H leads to a larger $\mu(\mathbf{u})$, and by (6), this $\mathbf{u}$ should be a better direction.

C. Effectiveness of the Common-Directions Method

To analyze the performance of the common-directions method, we demonstrate some of its most important properties in the following theorem.

Theorem 2: Let $P_k = [\mathbf{p}_1, \ldots, \mathbf{p}_m]$ be m linearly independent directions. We have

$$\mu(P_k) \geq \max_i \mu(\mathbf{p}_i)$$

and

$$\mu(P_k) \geq \Gamma \operatorname*{mean}_i \mu(\mathbf{p}_i), \tag{9}$$

where

$$\Gamma = \frac{m}{1 + (m-1)\zeta}, \tag{10}$$

and

$$\zeta = \max_i \operatorname*{mean}_{j:j\neq i} \tau_H(\mathbf{p}_i, \mathbf{p}_j). \tag{11}$$

The first result simply states that the common-directions method should always perform better than any of its individual directions.

The second result gives a lower bound on the usefulness of the common-directions method. We find it hard to give a meaningful upper bound of $\mu(P_k)$ due to the following example. Let $\mathbf{u}$ be a direction orthogonal to $\mathbf{g}_k$. We have $\min_\alpha q_k(\alpha \mathbf{u}) = 0$, and thus $\mu(\mathbf{u}) = 0$. For $\mathbf{p}_1 = \mathbf{u} + \epsilon \mathbf{s}$ and $\mathbf{p}_2 = \mathbf{u} - \epsilon \mathbf{s}$, we have $\mu(\mathbf{p}_1 - \mathbf{p}_2) = \mu(\mathbf{s}) = 1$, while $\mu(\mathbf{p}_1)$ and $\mu(\mathbf{p}_2)$ can be arbitrarily small.

The second result states that there are three determining factors for the lower bound

1) The average strength of selected directions $\operatorname{mean}_i \mu(\mathbf{p}_i)$.
2) The number of directions m.
3) The similarity of the selected directions ζ.

Since in Theorem 2 we assume the directions to be linearly independent, and H is positive definite, we have

$$0 \leq \tau_H(\mathbf{p}_i, \mathbf{p}_j) < 1$$

for $j \neq i$. Thus, we always have

$$0 \leq \zeta < 1.$$

From (9), (10), and (11), one can see that the improvement of the common-directions method over the average strength of the directions becomes larger as the selected directions $[\mathbf{p}_1, \ldots, \mathbf{p}_m]$ become less similar to each other, resulting in a decrease in ζ.

Besides, if ζ does not change much, then Γ slowly increases as m becomes larger. In other words, when the number of directions increases, the common-directions method should perform better.

D. The Lack of Direction Diversity for Gradient Directions

From Theorem 2, we can see that for a fixed number of directions, the average strength of the selected directions and their similarity determine the performance of the common-directions method. Now we will show that under some cases, the lack of direction diversity for gradient directions can make them both weak and similar to each other, thus leading to a poor performance.

Assume that at some iteration, the combination of our selected directions $[\mathbf{p}_1, \ldots, \mathbf{p}_m]$ from past gradient and update directions is weak and gives us a very small update $\mathbf{u}$. Since we assume the gradient to be Lipschitz continuous, the change in the gradient will also be small after we apply our update.

Consequently, our new gradient direction $\mathbf{g}$, which is also our newly added search direction, will be very close to the previous gradient direction, and thus our next update will also be small. Repeating the above process for several iterations, our selected directions will now become not only weak but also very similar to each other. From Theorem 2, we can see that this will lead to a poor performance.

IV. Benefit of Subsampled Newton Directions

In this section, we introduce subsampled Newton directions and show that they cannot be both weak and similar to each other. Thus, we believe that subsampled Newton directions are better than gradient directions when used in the common-directions method.

A. Subsampled Newton Directions

From (4), one can see that the computation of the Newton direction requires the use of the full Hessian H_k. One can instead use the subsampled Hessian [6]

$$\tilde{H}_k \equiv I + \frac{Cl}{|S_k|} \sum_{i \in S_k} \nabla^2 \xi(\mathbf{w}; \mathbf{x}_i, y_i)$$

to approximate the true Hessian, where $S_k \subseteq \{1, \ldots, l\}$ is a training subset. We can then derive the subsampled Newton direction $\tilde{\mathbf{s}}_k$ by minimizing the subsampled quadratic approximation

$$\tilde{q}_k(\mathbf{s}) = \frac{1}{2}\mathbf{s}^T \tilde{H}_k \mathbf{s} + \mathbf{g}_k^T \mathbf{s}. \tag{12}$$

For large-scale problems, $-\tilde{H}_k^{-1}\mathbf{g}_k$, the exact minimizor of (12) could be too expensive to compute. Furthermore, the subsampled Hessian matrix $\tilde{H}_k \in R^{n \times n}$ may be too large to be stored. Thus, we would use the conjugate gradient (CG) method instead to approximately minimize (12). The conjugate gradient method is an iterative process which involves a sequence of Hessian-vector products. Past works such as Keerthi et al. [8] and Lin et al. [9] have shown that for linear classification problems, the special form of the

Hessian allows us to conduct Hessian-vector products without explicitly forming the matrix.

Similarly, we can conduct the conjugate gradient method to minimize (12) without forming the subsampled Hessian, as it shares a similar form with the full Hessian matrix.

When S_k is chosen uniformly and all the training samples $(y_i, \mathbf{x}_i)$ are from the same distribution, we have

$$E[\tilde{H}_k] = H_k.$$

However, one should notice that we have

$$E[-\tilde{H}_k^{-1}\mathbf{g}_k] \neq -H_k^{-1}\mathbf{g}_k,$$

which means the subsampled Newton direction is not an unbiased estimator of the Newton direction.

Our proposal is to use subsampled Newton directions in the common-directions method. Just as the gradient descent method is a special case of the original common-directions method, the subsampled Newton method [6] is a special case of our proposed method, where the number of directions used is one. Another special case of our proposed method is the work of Wang et al. [10], where the current subsampled Newton direction is combined with the previous update direction $\mathbf{u}_{k-1}$ to produce the current update direction $\mathbf{u}_k$.

B. Relation Between Strength and Similarity

To show that subsampled Newton directions cannot be both weak and similar to each other, we consider the case where $\mathbf{g}$ barely changes, as intuitively subsampled Newton directions will be very different when the gradient $\mathbf{g}$ changes a lot.

The intuition behind the use of subsampled Newton directions is that even though they are not unbiased estimators, they should still be very close to the Newton direction if multiple of them are close to each other, and the Newton direction is the strongest direction in terms of μ.

To show this, we prove the following theorem.

Theorem 3: Given subsampled Hessian $\bar{H}_i$ and $\bar{H}_j$, subsampled Newton directions $\bar{\mathbf{s}}_i = -\bar{H}_i^{-1}\mathbf{g}$ and $\bar{\mathbf{s}}_j = -\bar{H}_j^{-1}\mathbf{g}$, we have

$$\tau_{\bar{H}}(\bar{\mathbf{s}}_i, \bar{\mathbf{s}}_j) \leq \min\{\tau_{\bar{H}}(\bar{\mathbf{s}}_i, \bar{\mathbf{s}}), \tau_{\bar{H}}(\bar{\mathbf{s}}_j, \bar{\mathbf{s}})\} \tag{13}$$

for all $\bar{\mathbf{s}} = -\bar{H}^{-1}\mathbf{g}$, $\bar{H} = \beta\bar{H}_i + (1-\beta)\bar{H}_j$, $0 \leq \beta \leq 1$. This theorem states that $\bar{\mathbf{s}}_i$ and $\bar{\mathbf{s}}_j$ are both closer to $\bar{\mathbf{s}}$ than to each other. That is to say when $\bar{\mathbf{s}}_i \approx \bar{\mathbf{s}}_j$, we have

$$\bar{\mathbf{s}}_i \approx \bar{\mathbf{s}}_j \approx \bar{\mathbf{s}}.$$

This implies that for multiple subsampled Newton directions, where $\mathrm{mean}_i[\bar{H}_i] \approx H$, we should have

$$\bar{\mathbf{s}}_i \approx \mathbf{s}$$

if $\bar{\mathbf{s}}_i \approx \bar{\mathbf{s}}_j$ for every i, j.

This means the subsampled Newton directions should be strong whenever they are similar to each other. Therefore, they do not possess the same weakness as the gradient directions.

V. Convergence

To apply results in [4], we need some conditions.

Assumption 2: The objective f is Lipschitz smooth and strongly convex.

Assumption 3: For all k, at least one of the directions in P_k is a sufficient descent direction; see the explanation below.

Since we assume ξ to be Lipschitz smooth and strictly convex, and we adopt regularization, Assumption 2 holds. Additionally, the subsampled Newton direction $\tilde{\mathbf{s}}_k$ is always a sufficient descent direction. That is for all k, we have

$$\frac{-\mathbf{g}_k^T\tilde{\mathbf{s}}_k}{\|\mathbf{g}_k\|_2\|\tilde{\mathbf{s}}_k\|_2} = \frac{\mathbf{g}_k^T\tilde{H}_k^{-1}\mathbf{g}_k}{\|\mathbf{g}_k\|_2\|\tilde{H}_k^{-1}\mathbf{g}_k\|_2} \geq \delta > 0, \tag{14}$$

where δ is a fixed constant. Because by our design $\tilde{\mathbf{s}}_k$ is included in P_k, Assumption 3 also holds.

Furthermore, we adopt the backtracking line search, so the following theorem holds.

Theorem 4 (Lee et al. [4] Theorem 3.2): If Assumption 2 and Assumption 3 hold, and we use the solution of the common-directions method as the update direction $\mathbf{u}_k$ and adopt the backtracking line search, then the function value converges linearly.

This ensures our proposed method has global linear convergence.

VI. Other Algorithmic Considerations

To determine the maximum number of directions m, we propose the following heuristic: We select m such that the extra cost induced by the common-directions method is $O(\#\text{nnz})$, where $\#\text{nnz}$ is the number of non-zero elements in the data set. Since the cost to compute the gradient and the function value are both $\Theta(\#\text{nnz})$, this makes our computational cost comparable to a single iteration of most optimization methods.

The extra cost for the common-directions method is $O(m^3 + m^2 l + mn)$ time and $O(m^2 + ml + mn)$ space. As a result, we propose to choose

$$m = O(\sqrt{\#\text{nnz}/l}).$$

Under the assumption that $n = O(l)$, this makes both the extra time and space $O(\#\text{nnz})$. In our experiment, we pick

$$m = \begin{cases} \hat{m} & \text{if } \hat{m} \text{ is odd} \\ \hat{m}+1 & \text{otherwise,} \end{cases}$$

where $\hat{m} = \lfloor\sqrt{\#\text{nnz}/l}\rfloor$. In other words, we choose the closest odd number to $\sqrt{\#\text{nnz}/l}$, as the **Mixed** strategy mentioned in Section II requires the number of directions to be odd.

VII. Experiments

The binary classification data sets we used are listed in the supplementary material. All data sets except yahookr can be downloaded from the publicly available LIBSVM Data Sets.[1] We modify the publicly available software LIBLINEAR [11] to compute subsampled Newton directions and incorporate the

[1] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets

use of the common-directions method. To decide the step size, we use the backtracking line search method with the Armijo condition. That is to say, given $c, \beta \in (0, 1)$, we find the smallest nonnegative integer i such that the step size $\alpha_k = \beta^i$ satisfies

$$f(\mathbf{w}_k + \alpha_k \mathbf{u}_k) \leq f(\mathbf{w}_k) + c\alpha_k \mathbf{g}_k^T \mathbf{u}_k.$$

In our experiments, we use $c = 0.01$ and $\beta = 0.5$.

To compute the subsampled Newton directions, we first shuffle and partition the data set into fixed training subsets. We then use these subsets in a cyclic manner to form the subsampled Hessian matrices. This increases the locality for us to compute the subsampled Hessian vector product. We follow Byrd et al. [6] to limit the number of CG steps (#CG) for each iteration. For simplicity, we do not consider other more complex inner stopping conditions for the CG procedure.

We conduct a detailed investigation by checking the relationship between the running time and the following relative function-value reduction $(f(\mathbf{w}_k) - f(\mathbf{w}^*))/f(\mathbf{w}^*)$, where $\mathbf{w}^*$ is obtained by LIBLINEAR under a very strict stopping condition. LIBLINEAR uses the following stopping condition,

$$\|\nabla f(\mathbf{w}_k)\|_2 \leq \epsilon \frac{\min(\#\text{pos}, \#\text{neg})}{l} \|\nabla f(\mathbf{w}_0)\|_2, \qquad (15)$$

where l is the total number of instances, #pos and #neg are the numbers of positive and negative instances, $\mathbf{w}_0$ is the weight initialization, which is $\mathbf{0}$ in our setting, and ϵ is the specified tolerance. Horizontal lines in our figures show when (15) with tolerances 10^{-1}, 10^{-2} (default), and 10^{-3} (the bottom of the figure) are reached by LIBLINEAR; such information indicates when the training algorithm should stop.

Regarding the regularization parameter C, we consider $C = C_{\text{Best}} \times \{1, 64\}$, where C_{Best} for each data set is the value leading to the best cross validation accuracy. We only show the figures for $C = C_{\text{Best}}$ due to the space limit.

A. Comparison With Other Methods

In this section, we compare our proposed method with other related optimization methods. Specifically, we compare

- **SubNewtonMixed**: The **Mixed** strategy under the common-directions framework with subsampled Newton directions as search directions.
- **SubNewton**: Subsampled Newton methods without the common-directions framework.
- **GradientMixed**: The **Mixed** strategy under the common-directions framework with gradients as search directions, which is proposed by Lee et al. [4].
- **Newton**: The preconditioned full Newton solver [12] in LIBLINEAR.

Subsampled Newton directions are computed using 5% of the training data and the number of CG steps is set to be 20.

Due to the space limit, here we do not show the results of some optimization methods which seem to be less competitive in past comparisons. For the comparison between the original common-directions method and LBFGS [13], one can see the work of Lee et al. [4]. For the comparison between **Newton** and first-order methods like SAG [14] and SAGA [15], one can see the work of Galli et al. [12].

From Figure I, we can see that for $C = C_{\text{Best}}$, **SubNewtonMixed** is in general better than **SubNewton** and **GradientMixed**. The only exception is news20, which is a smaller data set. The results are similar for $C = 64C_{\text{Best}}$. This demonstrates the effectiveness of our proposed method.

From Figure I, we can also see that for $C = C_{\text{Best}}$, **SubNewtonMixed** performs better than **Newton**. However, we observe that for $C = 64C_{\text{Best}}$, **SubNewtonMixed** could perform slightly worse than **Newton** on sparse data sets like kdda and kddb. This is because when the data set is sparse and the choice of C is large, the problem is more ill-conditioned and the strength of subsampled Newton directions is weaker. The full Newton method can be useful in such cases.

To conclude, our proposed method is an improvement upon the original common-directions method. While it can be slower than **Newton** under specific settings, its overall performance is competitive across sparse and dense data sets and different choices of C.

VIII. Conclusions

In this work, we analyze the impact of subspace selection for the common-directions method, and we point out that the lack of direction diversity can be a potential weakness for using gradients as directions. To address this problem, we propose the use of subsampled Newton directions, which always possess diversity unless they are already close to the true Newton direction. Our experiments on large-scale linear classification problems show that our proposed methods are generally better than the original common-directions method.

References

[1] P.-W. Wang, C.-P. Lee, and C.-J. Lin. The common-directions method for regularized empirical risk minimization. *JMLR*, 20:1–49, 2019.

[2] J. E. Dennis Jr and J. J. Moré. Quasi-Newton methods, motivation and theory. *SIAM Review*, 19(1):46–89, 1977.

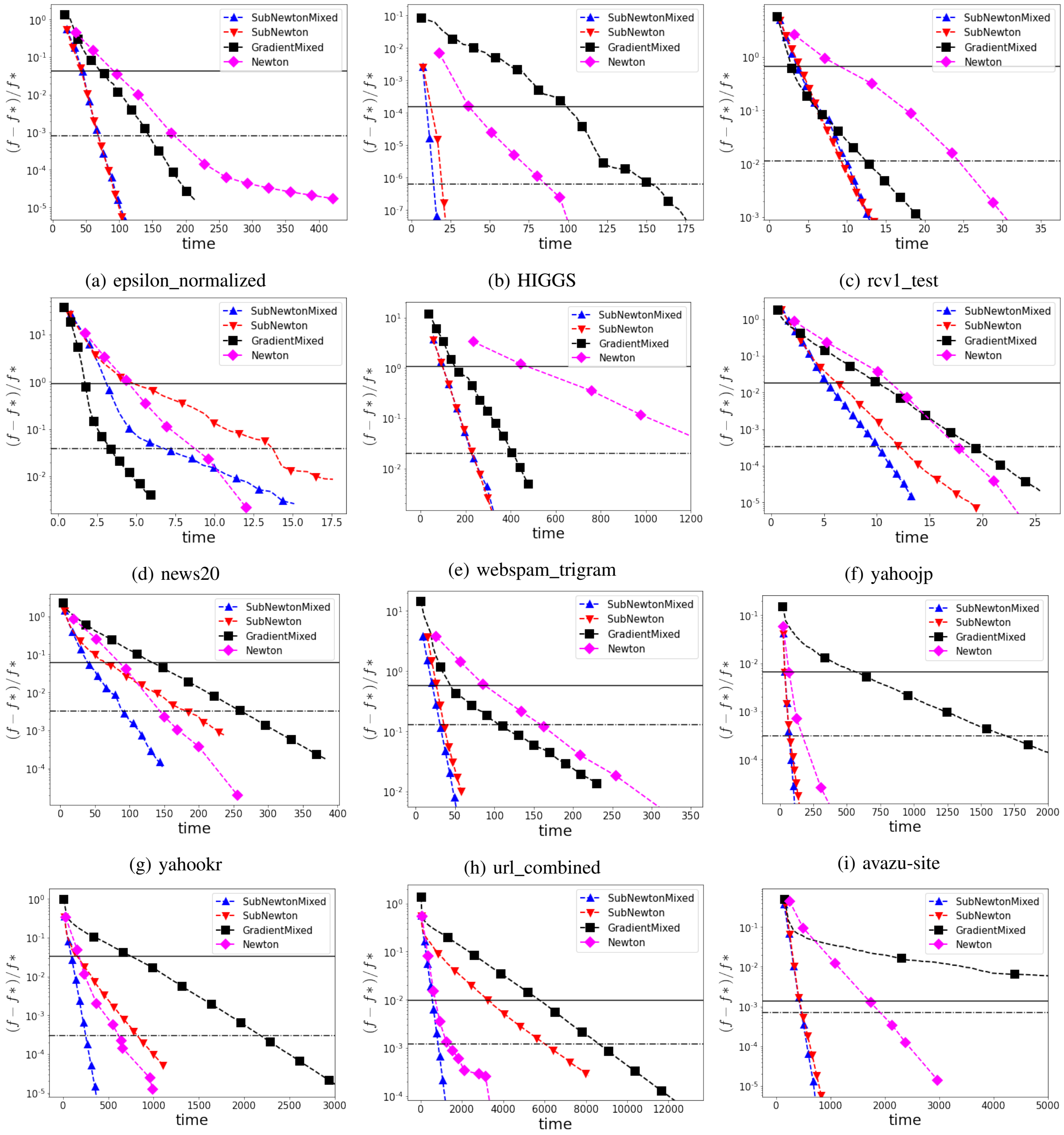
[3] Y. E. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27:372–376, 1983.

[4] C.-P. Lee, P.-W. Wang, and C.-J. Lin. Limited-memory common-directions method for large-scale optimization: convergence, parallelization, and distributed optimization, 2022. Under minor revision for Mathematical Programming Computation.

[5] R. Gower, D. Koralev, F. Lieder, and P. Richtárik. RSN: randomized subspace Newton. In *NIPS*, 2019.

[6] R. H. Byrd, G. M. Chin, W. Neveitt, and J. Nocedal. On the use of stochastic Hessian information in optimization methods for machine learning. *SIAM J. Optim.*, 21(3):977–995, 2011.

[7] S. Wang, F. Roosta-Khorasani, P. Xu, and M. W. Mahoney. GIANT: globally improved approximate Newton method for distributed optimization. In *NIPS*. 2018.



(a) epsilon_normalized

(b) HIGGS

(c) rcv1_test

(d) news20

(e) webspam_trigram

(f) yahoojp

(g) yahookr

(h) url_combined

(i) avazu-site

(j) kdda

(k) kddb

(l) kdd12

Fig. I: Training time of logistic regression with $C = C_{\text{Best}}$

[8] S. S. Keerthi and D. DeCoste. A modified finite Newton method for fast solution of large scale linear SVMs. *JMLR*, 6:341–361, 2005.

[9] C.-J. Lin, R. C. Weng, and S. S. Keerthi. Trust region Newton method for large-scale logistic regression. *JMLR*, 9:627–650, 2008.

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

[11] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: a library for large linear classification. *JMLR*, 9:1871–1874, 2008.

[12] L. Galli and C.-J. Lin. Truncated Newton methods for linear classification. *IEEE TNNLS*, 2021. To appear.

[13] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Program.*, 45(1):503–528, 1989.

[14] M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Math. Program.*, 162(1-2):83–112, 2017.

[15] A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, 2014.