VARIABLE METRIC PROXIMAL GRADIENT METHOD WITH DIAGONAL BARZILAI-BORWEIN STEPSIZE

Youngsuk Park[⋆] *Sauptik Dhar*[†] *Stephen Boyd*[⋆] *Mohak Shah* [†]

[⋆] Department of Electrical Engineering, Stanford University, Stanford, CA 94305.
[†] LG Sillicon Valley Lab, Santa Clara, CA 95050.

ABSTRACT

This paper proposes an adaptive metric selection strategy called *diagonal Barzilai-Borwein* (DBB) stepsize for the popular Variable Metric Proximal Gradient (VM-PG) algorithm [1, 2]. The proposed approach better captures the local geometry of the problem while keeping the per-step computation cost similar to the widely used scalar Barzilai-Borwein (BB) stepsize. We provide the theoretical convergence analysis for VM-PG using DBB stepsize. Finally, our empirical results show $\sim$ 10 - 40 % improvement in convergence times for the VM-PG using DBB compared to the BB stepsize for different machine learning problems on several datasets.

***Index Terms*—** proximal gradient, Barzilai-Borwein, diagonal metric, algorithm stepsize

1. INTRODUCTION

We tackle a convex optimization in the composite form

$$\underset{x\in\mathbf{R}^n}{\text{minimize}} \quad F(x) := f(x) + g(x), \tag{1}$$

where $x \in \mathbf{R}^n$ is the decision variable, $f : \mathbf{R}^n \to \mathbf{R}$ is convex and differentiable, and $g : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ is convex and can be non-differentiable. The structured form in (1) appears across a wide range of machine learning problems like classification, regression, matrix completion, graphical model, etc. [3, 4, 5, 6, 7, 8]. Proximal gradient (PG) methods have been widely used to solve optimization problems of the form (1). One popular variant of the proximal gradient algorithm is the *Variable Metric Proximal Gradient method* (VM-PG) [1, 2], provided in Algorithm 1.

Algorithm 1 Variable metric proximal gradient (VM-PG)

Given a starting point $x^0 \in \mathbf{R}^n$
repeat
  Update the metric $U^k \in \mathbf{S}^n_{++}$
  $y^{k+1} = x^k - (U^k)^{-1}\nabla f(x^k)$
  $x^{k+1} = \mathbf{prox}_{g,U^k}(y^{k+1})$
until stopping criterion $\left\|y^{k+1} - y^k\right\|_2 \le \epsilon_{\text{tol}}$ satisfied

Here, x^k is the k^{th} iterate, $U^k \in \mathbf{S}^n_{++}$ is a positive definite matrix, $\|z\|_{U^k} = \sqrt{z^T U^k z}$ is the U^k-norm at k^{th} iteration, and $\mathbf{prox}_{g,U^k}(\cdot) := \text{argmin}_x \left(g(x) + \frac{1}{2}\|\cdot - x\|^2_{U^k}\right)$ is the scaled proximal mapping of g relative to the metric induced by U^k norm.

Selection of the U^k metric can greatly impact the convergence behaviour of Algorithm 1. For example, setting $U_k = (\alpha^k)^{-1}I$ with scalar $\alpha^k \in \mathbf{R}$ results in the standard proximal gradient algorithm. Whereas, setting $U^k \approx \nabla^2 f(x^k)$ translates to the proximal (quasi) Newton method [9, 10]. These two families of algorithms have contrasting convergence behaviours. For example, proximal Newton-type methods provide fast convergence in terms of the overall iteration numbers, but incurs high per-step computation costs. On the other hand, proximal gradient methods enjoy low per-step computation costs, but can take a large number of iterations to converge. In this paper we attempt to utilize the best of both approaches and propose a new adaptive rule for the metric selection for VM-PG called the *Diagonal Borzilai-Borwein* (DBB) stepsize. Here are the summary of contributions.

- **Formulation**: We formulate a new rule for metric selection in Section 2. The proposed formulation of DBB (in eq. (5)) enjoys a closed form solution (see eq. (6)) and maintains similar per-step iteration cost ($O(n)$) as the scalar BB stepsize rule.
- **Convergence analysis**: We analyze the convergence of the VM-PG with the proposed DBB stepsize in Section 2.3.
- **Results**: Our empirical results show $\sim 10 - 40\%$ improvement of overall convergence times in favor of the proposed approach compared to scalar BB stepsize rule for different machine learning problems on several synthetic and real-world datasets in Section 3.

2. THE DIAGONAL BARZILAI-BORWEIN STEPSIZE (DBB)

The proposed DBB stepsize for VM-PG is motivated by the strength and weakness of the conventional BB stepsize [11] used for the proximal gradient algorithm. We first discuss the

BB stepsize in context of the proximal gradient algorithm and highlight its limitations. Next, we propose the new diagonal BB (DBB) stepsize for VM-PG to alleviate the limitations, and provide theoretical convergence guarantees for the overall algorithm.

2.1. Background: Barzilai and Borwein (BB) stepsize

The proximal gradient step in Algorithm 1 can be viewed as minimizing the overall function F where the differentiable part f is replaced by its second order approximation at x^k (w.r.t. some $U^k \in \mathbf{S}^n_{++}$) [12],

$$\mathbf{prox}_{g,U^k}(x^k - (U^k)^{-1}\nabla f(x^k)) =$$
$$\underset{x}{\operatorname{argmin}}\ g(x) + f(x^k) + \nabla f(x^k)^T(x - x^k) + \frac{1}{2}\left\|x - x^k\right\|^2_{U^k}.$$

This approximation suggests that $U^k = \nabla^2 f(x^k)$ would be a desirable choice, like the proximal Newton method [10]. However, using the Hessian typically incurs a high per-iteration cost. An alternative to that involves approximating the hessian using the secant condition,

$$U^k s^k \approx y^k, \tag{2}$$

where, $s^k = x^k - x^{k-1}$ and $y^k = \nabla f(x^k) - \nabla f(x^{k-1})$. The Barzilai and Borwein (BB) method [11] is one such popular approach that estimates a scalar approximation of the Hessian by setting $U^k = (\alpha^k)^{-1}I$ that best satisfies (2). The estimated scalar approximation is called the BB stepsize. Two of the most widely used BB stepsizes are,

$$\alpha^k_{\mathrm{BB1}} := \left\|s^k\right\|_2^2 / \langle s^k, y^k\rangle,$$
$$\alpha^k_{\mathrm{BB2}} := \langle s^k, y^k\rangle / \left\|y^k\right\|_2^2. \tag{3}$$

Note that $\alpha^k_{\mathrm{BB1}} \geq \alpha^k_{\mathrm{BB2}}$ holds due to Cauchy–Schwarz inequality. Recent research [13, 14] has shown that the performance of Algorithm 1 can be improved using a hybrid choice between these two stepsizes,

$$\alpha_{\mathrm{BB_{hybrid}}} = \begin{cases} \alpha^k_{\mathrm{BB2}} & \text{if } \alpha^k_{\mathrm{BB1}} < \delta\alpha^k_{\mathrm{BB2}} \\ \alpha^k_{\mathrm{BB1}} - \frac{1}{\delta}\alpha^k_{\mathrm{BB2}} & \text{otherwise} \end{cases} \tag{4}$$

here, the hyperparameter $\delta \in \mathbf{R}$ is typically chosen as 2. Also for cases when $\alpha^k_{\mathrm{BB}_{hybrid}} < 0$ (in (4)) we select the previous stepsize, i.e., $\alpha^k_{\mathrm{BB_{hybrid}}} = \alpha^{k-1}_{\mathrm{BB_{hybrid}}}$.

Such modifications are mainly designed to handle the instability in the (original) BB stepsize (3) for ill-conditioned f. However, the scalar BB may still be prone to inconsistencies. For example, in the case of a proximal mapping involving projection, the step (s^k) and gradient-change (y^k) directions can sometimes be close to being orthogonal. This causes degenerate scenarios with $\alpha_{\mathrm{BB1}} \to \infty$ or $\alpha_{\mathrm{BB2}} \to 0$. For such cases, the scalar estimates may significantly deviate from the secant condition (2), and hence the Hessian geometry. For the rest of the paper we refer to the Algorithm 1 using the BB scalar metric in (4) as PG (BB).

	PG (BB)	VM-PG (DBB)	Prox L-BFGS	Prox Newton
Metric	$O(n)$	$O(n)$	$O(n^2)$	$O(n^2)$
Forward	$O(n)$	$O(n)$	$O(n^2)$	$O(n^3)$

Table 1: Cost for computing metric U^k and forward step $(x^k - (U^k)^{-1}\nabla f(x^k))$.

2.2. Diagonal Barzilai and Borwein (DBB) stepsizes

To better capture the Hessian geometry of f, we propose a diagonal metric U^k at each iteration k computed as follows

$$\begin{aligned} \underset{u\in\mathbf{R}^n}{\text{minimize}} \quad & \left\|Us^k - y^k\right\|_2^2 + \mu\left\|U - U^{k-1}\right\|_F^2 \\ \text{subject to} \quad & (\alpha^k_{\mathrm{BB1}})^{-1}I \preceq U \preceq (\alpha^k_{\mathrm{BB2}})^{-1}I, \\ & U = \mathrm{Diag}(u). \end{aligned} \tag{5}$$

Here, the hyperparameter $\mu > 0$ controls the trade-off between satisfying the secant condition (2) and being consistent with the previous metric U^{k-1}. We select a large μ if we expect the Hessian not to change much over iterations. Otherwise, we select a small μ which simply serves as a numerical safeguard. Lastly, the diagonal elements are bounded by the (safeguarded) BB stepsizes in (3).

One advantage of the proposed formulation (5) is that it has a closed-form solution. That is, for $U^k = \mathrm{Diag}(u^k)$ and $u^k = [u^k_1, \ldots, u^k_n] \in \mathbf{R}^n$, the solution to (5) is given as,

$$u^k_i = \begin{cases} \frac{1}{\alpha^k_{\mathrm{BB1}}} & \frac{s^k_i y^k_i + \mu u^{k-1}_i}{(s^k_i)^2 + \mu} < \frac{1}{\alpha^k_{\mathrm{BB1}}} \\ \frac{1}{\alpha^k_{\mathrm{BB2}}} & \frac{s^k_i y^k_i + \mu u^{k-1}_i}{(s^k_i)^2 + \mu} > \frac{1}{\alpha^k_{\mathrm{BB2}}} \\ \frac{s^k_i y^k_i + \mu u^{k-1}_i}{(s^k_i)^2 + \mu} & \text{otherwise} \end{cases} \tag{6}$$

where, s^k_i and y^k_i are i^{th} elements of s^k and y^k respectively.

Note that, using this diagonal metric selection the VM-PG maintains similar per-iteration cost as the PG (BB) method (see Table 1), and still provides several advantages compared to a (scalar) BB stepsize discussed next. First, the DBB uses a diagonal structure rather than a single scalar element used in BB, and hence can better satisfy the secant condition (2). Second, it is robust to the degenerate case discussed in section 2.1, where $\langle s^k, y^k\rangle \approx 0$ (resulting $\alpha^k_{\mathrm{BB1}} \approx \infty$, $\alpha^k_{\mathrm{BB2}} \approx 0$). Now the metric updates can be safeguarded against the high residual of the secant condition $\|U^k s^k - y^k\|$ through a careful selection of $\mu > 0$. Also, u^k at each iteration is finite as long as $0 \leq u^{k-1} < \infty$ and $\mu > 0$. This in practice, makes VM-PG with diagonal BB stepsize numerically more stable than the PG (BB) algorithm. The effect of this $\mu > 0$ hyperparameter on the convergence of VM-PG with DBB stepsize has been extensively analyzed in a longer version of the paper [15].

2.3. Convergence of VM-PG with DBB

Similar to PG using BB stepsize, the VM-PG with DBB stepsize is not guaranteed to converge without a line search. Hence, we incorporate a line search strategy for the convergence guarantee. The overall algorithm is provided below.

Algorithm 2 VM-PG with diagonal BB metric

Given parameters $M_{\rm LS} \geq 1$, $\beta > 1$, $\mu > 0$, a starting point $x^0, x^1 \in \mathbf{R}^n$ and, an initial metric $U^0 \in \mathbf{S}^n_{++}$
repeat
  Compute $\alpha^k_{\rm BB1}$ and $\alpha^k_{\rm BB2}$ from (4)
  Initialize U^k from (6)
  Update $x^{k+1} := \mathbf{prox}_{g,U_k}(x^k - (U^k)^{-1}\nabla f(x^k))$
  repeat
    $U^k := \beta U^k$
    $x^{k+1} := \mathbf{prox}_{g,U_k}(x^k - (U^k)^{-1}\nabla f(x^k))$
  until line search criterion in (7) is satisfied
  return metric U^k and next iterate x^{k+1}
until stopping criterion satisfied

In literature several line-search strategies (with backtracking) have been proposed for theoretical convergence guarantees. In practice however, a non-monotonic line search typically provides low line search cost (per iteration), with better convergence results for VM-PG (and PG (BB)) algorithms [16, 17, 18, 14]. Basically, a non-monotonic line search allows the objective function $F(x)$ to increase between subsequent iterations, but results in an eventual decrease in its values. Here, given the current iterate x^k, an initial metric U^k from (6), and (a potential) next iterate x^{k+1}; the non-monotonic line search checks whether (U^k, x^{k+1}) satisfies the following criterion,

$$F(x^{k+1}) \leq \hat{F}^k - \frac{1}{2}\left\|x^{k+1} - x^k\right\|^2_{U^k}, \tag{7}$$

where, $M_{\rm LS} \geq 1$ is an integer line search parameter, and $\hat{F}^k = \max\{F(x^k), F(x^{k-1}), \ldots, F(x^{k-\min(M_{LS},k-1)}\})$. Then it backtracks by re-scaling the metric U^k by a factor of $\beta > 1$ until (7) is satisfied. Note that setting $M_{\rm LS} = 1$ corresponds to a monotonic line search.

Next, we provide the convergence analysis for Algorithm 2. We first start with definitions,

Definition 1 *A differentiable function $f : \mathbf{R}^n \to \mathbf{R}$ is L-smooth and m-strongly convex if $\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x-y\|_2$ holds and $\langle \nabla f(x) - \nabla f(y), x - y\rangle \geq m\|x-y\|_2^2$ holds for all $x, y \in \mathbf{R}^n$, respectively.*

Under assumption that f is L-smooth and $U^k > 0$, the following guarantee holds.

Theorem 1 *the VM-PG Algorithm* (2) *guarantees that $F(x^k)$ converges to the optimal value $F^\star$, i.e., $\lim_{k\to\infty} F(x^k) := F^\star$.*

In addition, our diagonal strategy in algorithm (2) ensures,

Theorem 2 *The VM-PG in Algorithm* (2)*, with monotonic line search (i.e. $M_{\rm LS} = 1$) satisfies,*

$$\min_{k=1,\ldots,K} \left\|G_{U^k}(x^k)\right\|^2_{(U^k)^{-1}} \leq \frac{2(F(x^0) - F^\star)}{K}$$

where $G_{U^k}(x^k) \in \nabla f(x^k) + \partial g(x^k - (U^k)^{-1}\nabla f(x^k))$ and $G_{U^k}(x^k) = 0$ iff $0 \in \partial F(x^k)$.
Further, if f is m-strongly convex, then

$$\left\|x^{k+1} - x^\star\right\|^2_{U^k} \leq (1 - \frac{m}{u^k_{\max}})\left\|x^k - x^\star\right\|^2_{U^k}$$

where $u^k_{\max} = \max_i u^k_i$.

All proofs are available in a longer version of this work [15]. Note that, although the convergence guarantees are provided for a monotonic line search ($M_{\rm LS} = 1$); in practice the non-monotonic alternatives ($M_{\rm LS} > 1$) provide better empirical results. Hence, we adopt it for our experiments.

3. EXPERIMENTS

For our empirical studies, we use two popular machine learning applications with structure $F(x) := f(x) + g(x)$.

3.1. Penalized linear and logistic regression

For $i = 1 \ldots N$ samples of $a^{(i)} \in \mathbf{R}^n$ and the associated label $b^{(i)}$. We solve,

$$\underset{x\in\mathbf{R}^n}{\text{minimize}} \quad \frac{1}{N}\sum_{i=1}^N l\left(x; a^{(i)}, b^{(i)}\right) + g(x).$$

Here, l is a loss function given as

- Least square (LS) loss, $l(\theta; a, b) = \left\|\theta^T a - b\right\|_2^2$,
- Logistic regression (LR), $l(\theta; a, b) = \log(1 + e^{-b\theta^T a})$.

For a penalty function $g(x)$, we use nonnegative constraint $g(x) = \mathbf{1}_{\{z|z\geq 0\}}(x)$ or lasso $g(x) = \lambda\|x\|_1$ with parameter $\lambda \in \mathbf{R}_+$.

3.2. Datasets and experimental settings

Synthetic dataset. For our synthetic dataset we use $N = 0.2n$ samples generated from $a^{(i)} \sim \mathcal{N}(0, \Sigma)$ with some random $\Sigma \in \mathbf{S}^n_{++}$. The labels $\{b^{(i)}\}_{i=1}^N$ are generated as follows,

- LS : $b^{(i)} = \left(a^{(i)}\right)^T x^\star + 0.2\, v$ where $v \sim \mathcal{N}(0, I)$.
- LR: $y = \sigma\left(\left(a^{(i)}\right)^T x^\star\right) + 0.2\, w$ where σ is sigmoid function $\sigma(z) = \log(1 + e^{-z})$ and $w \sim$ Unif(0, 1). Then take $b^{(i)} = 1$ if $y \geq 0.5$ or $b^{(i)} = -1$ otherwise.

$f(x)$ $g(x)$	LS nonneg.	LR nonneg.	LS lasso	LR lasso
PG (BB)	52.3 (1.22)	54.5 (1.71)	82.1 (2.09)	61.5 (2.24)
VM-PG (DBB)	46.15 (1.08)	**46.2 (1.27)**	84.9 (2.21)	**45.5 (1.13)**

Table 2: Avg. number of iterations (CPU times in sec) for PG (BB) and VM-PG (DBB) convergence on synthetic dataset.

$f(x)$	LS	LR	LS	LR
Data (N, n)	MNIST (240, 784)	MNIST (1250, 784)	CIFAR (625, 3072)	CIFAR (500, 3072)
PG (BB)	83 (2.24)	181 (5.52)	175 (5.7)	91 (4.42)
VM-PG (DBB)	78 (2.01)	**133 (3.83)**	181 (5.52)	**49(2.67)**

Table 3: Average iterations (CPU times in sec) for PG (BB) and VM-PG (DBB) convergence using ℓ_1 regularization for real-world datasets.

Real-world datasets. We use two real-world datasets. Handwritten digit recognition MNIST [19] and object recognition CIFAR [20]. We show the results using ℓ_1 -regularized losses on a smaller subset of the datasets, illustrating the advantage of the proposed approach under ill-conditioned settings. The results using the entire dataset show similar conclusions and is available in a longer version of the paper [15]. Here, we use the LS loss to estimate the labels ('0' - '9') and LR to classify between the classes ('1' vs. '5').

For all the experiments we use $\lambda = 10^{-2}$ and 10^{-4} for LS and LR respectively. For regression, the data matrix is centered at 0 and column-wise normalized to a unit ℓ_2 norm.

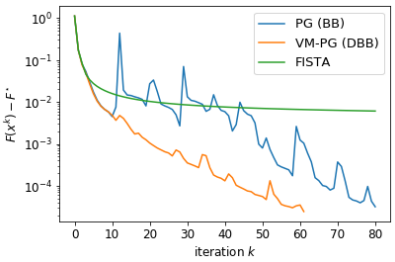
3.3. VM-PG algorithm parameters

For our experiments we fix $\mu = 10^{-6}$, $M_{\text{LS}} = 15$, $\beta = 2$, and the stopping error tolerance as $\epsilon_{\text{tol}} = 10^{-4}$ (for LS) and 10^{-2} (for LR) following [14]. A detailed analysis with varying values of the above parameters and their effects on the convergence of VM-PG (DBB) is available in [15].

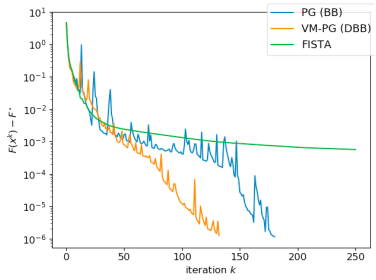
3.4. Results

Table 2 shows the total number of iterations (and CPU times in sec) for the convergence of the VM-PG (DBB) vs. PG (BB), averaged over 100 experimental runs for the synthetic data. And Table 3 shows the results for a sub-sampled (ill-conditioned) MNIST and CIFAR dataset. As seen from the Tables 2 and 3 the VM-PG (DBB) significantly outperforms the standard PG (BB) and provides $\sim 10 - 40\%$ improvement in the number of iteration to convergence for LR. Since the per-iteration cost for the VM-PG (DBB) is similar to PG (BB), this improvement in the number of iterations equivalently translates to the improvement in CPU times (in sec).

For additional analysis, we also provide the convergence curves in Fig. 1. Here, to simplify our discussion we only provide the curves for PG(BB), VM-PG(DBB) and the accelerated proximal gradient method (FISTA) (a non BB-type method) [21], using ℓ_1-regularized logistic regression for the synthetic and real-life (MNIST) dataset in Fig. 1a and Fig. 1b respectively. Note that, our proposed VM-PG (DBB) has been specifically designed targeting improvement over the PG (BB). The results for FISTA is added as an exemplar for state-of-art ℓ_1-regularized optimization problem solvers. Further, all the three methods require similar per iteration computational costs, i.e., $O(n)$ to compute and store metric, $O(mn)$ or $O(n^2)$ cost for gradient step, $O(n)$ for proximal step; of which the gradient steps are dominant. Hence, we only show the iteration counts in Fig. 1. These results will also hold for the overall CPU times (in sec). These show that the proposed VM-PG (DBB) outperforms both PG(BB) and FISTA for the current problem settings. A more detailed study on comparisons with other state-of-art methods like FISTA under different problem settings is an open research problem.



(a) ℓ_1 penalized LR for synthetic data with $N = 200, n = 1000$.



(b) ℓ_1 penalized LR for MNIST data with $N = 1250, n = 784$.

Fig. 1: Typical convergence behaviours of PG(BB) and VM-PG compared to Accelerated PG (FISTA).

4. CONCLUSION

This paper proposes a diagonal BB metric for the variable proximal gradient method. The proposed diagonal metric provides a better estimate of the ill-conditioned local Hessian compared to the standard scalar BB approach, resulting to faster convergence. Combined with a nonmonotonic line-search the overall algorithm is guaranteed to converge. Finally, for several machine learning applications with synthetic and real-world datasets, empirical results exhibit improved convergence behavior for the proposed methodology.

5. REFERENCES

[1] J Frédéric Bonnans, J Ch Gilbert, Claude Lemaréchal, and Claudia A. Sagastizábal, "A family of variable metric proximal methods," *Mathematical Programming*, vol. 68, no. 1-3, pp. 15–47, 1995.

[2] Lisandro A Parente, Pablo A Lotito, and Mikhail V Solodov, "A class of inexact variable metric proximal point algorithms," *SIAM Journal on Optimization*, vol. 19, no. 1, pp. 240–260, 2008.

[3] Emmanuel J Candès and Benjamin Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, pp. 717, 2009.

[4] Emmanuel J Candes and Yaniv Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, 2010.

[5] Martin J Wainwright, Michael I Jordan, et al., "Graphical models, exponential families, and variational inference," *Foundations and Trends® in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.

[6] Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis R Bach, "Proximal methods for sparse hierarchical dictionary learning.," in *ICML*. Citeseer, 2010, vol. 1, p. 2.

[7] Arthur Szlam, Karol Gregor, and Yann LeCun, "Fast approximations to structured sparse coding and applications to object classification," in *European Conference on Computer Vision*. Springer, 2012, pp. 200–213.

[8] Youngsuk Park, David Hallac, Stephen Boyd, and Jure Leskovec, "Learning the network structure of heterogeneous data via pairwise exponential markov random fields," in *Artificial Intelligence and Statistics*, 2017, pp. 1302–1310.

[9] Stephen Becker and Jalal Fadili, "A quasi-newton proximal splitting method," in *Advances in Neural Information Processing Systems*, 2012, pp. 2618–2626.

[10] Jason D Lee, Yuekai Sun, and Michael A Saunders, "Proximal newton-type methods for minimizing composite functions," *SIAM Journal on Optimization*, vol. 24, no. 3, pp. 1420–1443, 2014.

[11] Jonathan Barzilai and Jonathan M Borwein, "Two-point step size gradient methods," *IMA journal of numerical analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[12] Emilie Chouzenoux, Jean-Christophe Pesquet, and Audrey Repetti, "Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function," *Journal of Optimization Theory and Applications*, vol. 162, no. 1, pp. 107–132, 2014.

[13] Bin Zhou, Li Gao, and Yu-Hong Dai, "Gradient methods with adaptive step-sizes," *Computational Optimization and Applications*, vol. 35, no. 1, pp. 69–86, 2006.

[14] Tom Goldstein, Christoph Studer, and Richard Baraniuk, "A field guide to forward-backward splitting with a fasta implementation," *arXiv preprint arXiv:1411.3406*, 2014.

[15] Youngsuk Park, Sauptik Dhar, Stephen Boyd, and Mohak Shah, "Variable metric proximal gradient method with diagonal barzilai-borwein stepsize," *arXiv preprint arXiv:1910.07056*, 2019.

[16] Luigi Grippo, Francesco Lampariello, and Stephano Lucidi, "A nonmonotone line search technique for newton's method," *SIAM Journal on Numerical Analysis*, vol. 23, no. 4, pp. 707–716, 1986.

[17] Ernesto G Birgin, José Mario Martínez, and Marcos Raydan, "Nonmonotone spectral projected gradient methods on convex sets," *SIAM Journal on Optimization*, vol. 10, no. 4, pp. 1196–1211, 2000.

[18] Hongchao Zhang and William W Hager, "A nonmonotone line search technique and its application to unconstrained optimization," *SIAM journal on Optimization*, vol. 14, no. 4, pp. 1043–1056, 2004.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[21] Amir Beck and Marc Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.