

# OPTML 2017: Variable Metric Proximal Gradient Method with Diagonal Borzilai-Borwein Stepsize

**Youngsuk Park**

*Stanford University*

**Stephen Boyd**

*Stanford University*

**Sauptik Dhar**

*Bosch Center for Artificial Intelligence*

**Mohak Shah**

*Bosch Center for Artificial Intelligence*

youngsuk@stanford.edu

boyd@stanford.edu

sauptik.dhar@us.bosch.com

mohak.shah@us.bosch.com

## Abstract

We propose a diagonal metric selection for *variable metric proximal gradient method* (VM-PG). The proposed metric better captures the local geometry of the problem and provides improved convergence compared to the standard proximal gradient (PG) methods with Barzilai-Borwein (BB) stepsize selection. Further, we provide convergence guarantees for the proposed method and illustrate its advantages over traditional PG with (BB) through empirical results.

## 1 Introduction

We tackle a convex optimization problem in the composite form,

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

where,  $f$  is convex and differentiable under  $\mathbf{dom} f$  and  $g$  is convex and non-differentiable under  $\mathbf{dom} g$ . This structured form (1) appears across a wide range of machine learning problems like, classification, regression, matrix completion, nonnegative matrix factorization etc. Proximal gradient-type methods have been widely adopted for such non-differentiable problems. These methods offer several advantages like, low per-step computation costs, theoretical guarantees under mild conditions, and practical rules for stepsize selections [12, 5, 1, 15]. The generic form of the *variable metric proximal gradient method* [4, 13, 10] is provided in Algorithm (1).

---

**Algorithm 1** Variable metric proximal gradient (VM-PG)

---

**given** a starting point  $x^0 \in \mathbf{dom} f$

**repeat**

1. Update metric  $U_k$
2.  $y^{k+1} = x^k - U_k^{-1} \nabla f(x^k)$
3.  $x^{k+1} = \mathbf{prox}_{g, U_k}(y^{k+1})$

$$= \underset{x}{\operatorname{argmin}} \left( g(x) + \frac{1}{2} \|y^{k+1} - x\|_{U_k}^2 \right)$$

**until** stopping criterion  $\|y^{k+1} - y^k\|_2 \leq \epsilon_{\text{tol}}$  satisfied

---

Here  $x^k$  is the  $k$ th iterate,  $U_k \in \mathbf{S}_{++}^n$  is the positive definite metric at the  $k$ th iteration, and  $\|z\|_U = \sqrt{z^T U z}$  is  $U$ -norm, and  $\mathbf{prox}_{g, U}$  is the *scaled* proximal mapping of  $g$  relative to metric  $U$  [8].

**Special cases.** The Algorithm 1 transforms to the standard proximal gradient algorithm for  $U_k = \alpha_k^{-1} I$  [1] (with scalar stepsize  $\alpha_k$ ), and the proximal (quasi-)Newton method for  $U_k \approx H_k$ , (the Hessian at  $x^k$ ) [2, 9]. These special cases have their respective pros and cons. For example, proximal Newton-type methods provide fast convergence but worse per-step computation costs. On the other hand, proximal gradient methods have computationally attractive step updates, but exhibit relatively slower convergence behavior.

**Summary of contributions.** In this paper we propose a new *diagonal Borzilai-Borwein* (BB)-type metric for  $U_k$  (shown in (4)). The proposed metric exhibits better convergence behavior compared to scalar BB [11], typically used for proximal gradient methods. This is achieved through individual scaling of the coordinates resulting to a well-conditioned local geometry at each step. Further, the per-step computation cost and memory requirements is similar to the standard BB, and is much cheaper than proximal Newton methods. Finally, we provide the convergence guarantees and the empirical results in favor of the proposed metric selection. Discussions on several other computational benefits such as no matrix inversion, parallelization, and in most cases a closed-form solution have been omitted in this version of the paper.

## 2 Metric selection

The proximal gradient step can be viewed as minimizing the overall function  $F$  where the differentiable part  $f$  is approximated into its second order form (w.r.t  $U$ ) at  $x$  [3],

$$\text{prox}_{g,U}(x - U^{-1}\nabla f(x)) = \underset{y}{\text{argmin}} \quad g(y) + f(x) + \nabla f(x)^T(y - x) + \frac{1}{2} \|y - x\|_U^2.$$

This motivates setting  $U_k \approx H_k$  as a desirable choice. However, any such approximations should satisfy the *secant condition*, i.e.,

$$U_k s^k \approx y^k. \quad (2)$$

for the step  $s^k = x^k - x^{k-1}$  and gradient change  $y^k = \nabla g(x^k) - \nabla g(x^{k-1})$ .

**Barzilai and Borwein (BB).** is a very popular approach that finds the scalar curvature of the Hessian, (setting  $U_k = \alpha^k I$ ) by satisfying (2). Depending on the residual type used for (2), the two most widely used BB stepsizes are [15],

$$\text{(BB 1)} \quad \alpha_{SD}^k = \|s^k\|_2^2 / \langle s^k, y^k \rangle, \quad \text{(BB 2)} \quad \alpha_{MG}^k = \langle s^k, y^k \rangle / \|y^k\|_2^2. \quad (3)$$

A more practical choice is a hybrid between these two stepsizes, with additional safeguarding for numerical stability (see [15, 6] for details). However, one major caveat is that such selections may deviate from secant condition (2) too much when there is a huge discrepancy between the step ( $s^k$ ) and gradient-change ( $y^k$ ) directions.

**Diagonal BB metric.** To accommodate for such differences between the step and gradient-change directions we introduce a diagonal metric as provided below,

$$\begin{aligned} & \underset{u \in \mathbf{R}^n}{\text{minimize}} \quad \|Us^k - y^k\|_2^2 + \mu \|U - U_{k-1}\|_F^2 \\ & \text{subject to} \quad (M\alpha_{SD}^k)^{-1}I \preceq U \preceq (\alpha_{MG}^k/M)^{-1}I, \quad U = \text{Diag}(u) \end{aligned} \quad (4)$$

This proposed formulation strikes a balance between satisfying the secant rule (2) and being consistent with the previous metric. This balance is controlled by the user-defined parameter  $\mu$ . Further, there is an additional constraint to bound the metric  $U$  between the two BB stepsizes (shown in (3)) scaled by parameter  $M \geq 1$ . This problem (4) has a simple closed-form solution as shown below. For each coordinate  $i = 1, \dots, n$ , the diagonal element  $u^k$  of  $U_k$  is

$$u_i^k = \begin{cases} \frac{1}{M\alpha_{SD}^k} & \hat{u}_i < \frac{1}{M\alpha_{SD}^k} \\ \frac{M}{\alpha_{MG}^k} & \hat{u}_i > \frac{M}{\alpha_{MG}^k} \\ \hat{u}_i & \text{otherwise} \end{cases}, \quad \text{where} \quad \hat{u}_i = \frac{s_i y_i + \mu u_i^{k-1}}{s_i^2 + \mu}. \quad (5)$$

**Nonmonotone line-search.** Finally, we perform a nonmonotonic line-search similar to [7, 14]. Now, for a given current iterate  $x^k$  and (a potential) next iterate  $x^{k+1}$ , we check whether  $(x^k, x^{k+1})$  satisfies

$$f(x^{k+1}) < \hat{f}^k + \langle x^{k+1} - x^k, \nabla f(x^k) \rangle + \frac{1}{2} \|x^{k+1} - x^k\|_{U_k}^2 \quad (6)$$

where  $\hat{f}^k = \max\{f^{k-1}, f^{k-2}, \dots, f^{k-\min(M_{LS}, k)}\}$  and a positive integer  $M_{LS}$ . Else, we backtrack and scale the metric  $U_k$  by a factor  $\beta > 1$ . The algorithm is provided next.

---

**Algorithm 2** Subroutine: Update metric with nonmonotone line-search

---

given parameters  $M_{LS}, \beta > 1$ , iterates  $x^{k-1}, x^k$ , and gradients  $\nabla f(x^{k-1}), \nabla f(x^k)$

Compute  $\alpha_{SD}^k, \alpha_{MG}^k$  from (3)

Initialize metric  $U_k$  from (5)

$x^{k+1} := \mathbf{prox}_{g, U_k}(x^k - U_k^{-1} \nabla f(x^k))$

**repeat**

1.  $U_k := \beta U_k$

2.  $x^{k+1} := \mathbf{prox}_{g, U_k}(x^k - U_k^{-1} \nabla f(x^k))$

**until** line-search criterion (6) satisfied

**return** metric  $U_k$  and next iterate  $x^{k+1}$

---

Under this metric update 2, the algorithm 1 follows Theorem (2.1).

**Theorem 2.1** Assume the gradient  $\nabla f$  is  $L$ -Lipschitz. For the sequence of  $U_k \in \mathbf{S}_{++}^n$  generated by Algorithm 2 at each iteration,  $\lim_{k \rightarrow \infty} F(x^k) := F^*$ .

### 3 Experiments

We demonstrate the performance of the VM-PG with diagonal BB metric through several experiments. For each experiment, we consider different problem types shown below,

**Quadratic programming with nonnegative constraint.**

$$\underset{x \in \mathbf{R}^n}{\text{minimize}} \quad \frac{1}{2} x^T Q x + q^T x + p + \mathbf{1}_{\{z|z \geq 0\}}(x)$$

Here,  $Q \in \mathbf{S}_{++}^n$ . This is a generic problem typically seen in many machine learning applications like, hard-margin SVMs etc.

**Linear/Logistic regression.** For  $N$  samples of  $a^{(i)} \in \mathbf{R}^n$  and the associated label  $b^{(i)}$ , consider the regression problem

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

For our experiments we consider two popular loss functions: least square (linear) loss  $l(\theta; a, b) = \|\theta^T a - b\|_2^2$  and logistic loss  $l(\theta; a, b) = \log(1 + e^{-b\theta^T a})$ . As the non-differentiable regularizer we use nonnegative constraint or lasso penalty, i.e.,  $g(x) = \mathbf{1}_{\{z|z \geq 0\}}(x)$  or  $g(x) = \lambda \|x\|_1$  with  $\lambda \in \mathbf{R}_+$  respectively. For the experiments we fix  $\lambda = 10^{-4}$ .

**VM-PG algorithm parameters.** Throughout this paper we set  $\mu = 10^{-4}$ ,  $M = 1$  for metric selection (5), and  $M_{LS} = 10$ ,  $\beta = 2$  for line-search (6). We adopt modified stopping criterion [6] with  $\epsilon_{\text{tol}} = 10^{-6}$  for QP and LS problems,  $\epsilon_{\text{tol}} = 10^{-3}$  for logistic regression. Maximum iteration=500 is used. We compare the VM-PG (with Diagonal BB) to standard PG (with BB).

**Data generation.** For all the experiments we consider high-dimensional data with  $n = 1000$ . For QP, we generate quadratic objectives with two different condition numbers  $\kappa = 5$  and 500. For the regression problems, we use small ( $N = 200$ ) and large ( $N = 1000$ ) number of samples generated from  $a^{(i)} \sim \mathcal{N}(0, \Sigma)$ . For some  $x^* \in \mathbf{R}^n$ , we compute its associated true labels, adding some noise to generate noisy labels  $b^{(i)}$  (in  $\mathbf{R}$  or in  $\{-1, 1\}$ ) over  $N$  samples. For linear regression,  $b^{(i)} = (a^{(i)})^T x^* + v$  where  $v \sim \mathcal{N}(0, I)$ . For logistic regression, generate  $y = \sigma\left((a^{(i)})^T x^*\right) + 0.3 w$  where  $\sigma$  is sigmoid function  $\sigma(x) = \log(1 + e^{-x})$  and  $w \sim \text{Unif}[0, 1]$ , and then take  $b^{(i)} = 1$  if  $y \geq 0.5$  or  $b^{(i)} = -1$  otherwise. Finally, the data matrix  $A \in \mathbf{R}^{N \times n}$  is column-wise normalized to the unit  $\ell_2$  norm.

$f(x)$ $g(x)$ $(\kappa, n)$	QP nonneg. $(5, 10^3)$	QP nonneg. $(500, 10^3)$	$f(x)$ $g(x)$ $(N, n)$	LS nonneg. $(200, 10^3)$	log. reg. nonneg. $(200, 10^3)$	log. reg. lasso. $(200, 10^3)$
PG(BB)	9.8	16.1	PG (BB)	52.3	44.5	116.8
VM-PG (Diagonal BB)	8.2	12.2	VM-PG (Diagonal BB)	46.15	36.2	129.5

Table 1: Number of iterations required for the convergence of VM-PG (Diagonal BB) and PG (BB).

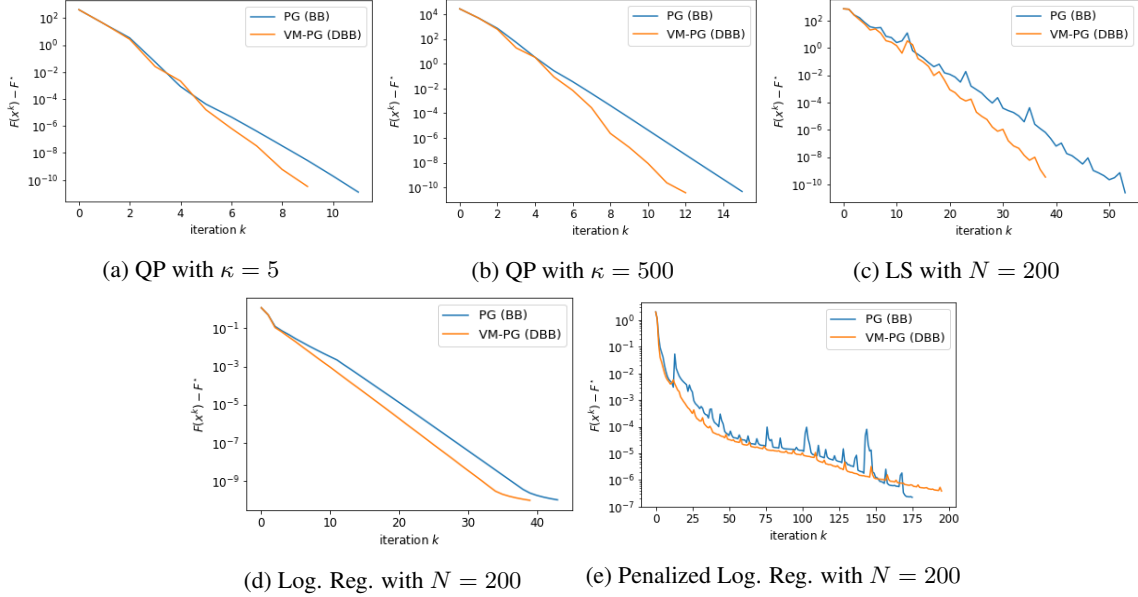


Figure 1: Convergence comparison. (a) - (d) are with nonnegative constraint, (e) is with lasso penalty with  $\lambda = 10^{-4}$ .

**Results.** Table 1 shows the total number of iterations required for convergence for VM-PG vs. PG (BB), averaged over 100 experiments. For well-conditioned  $Q$  with  $\kappa \sim 5$ , both the methods converge fast without any notable differences (Figure 1(a)). However the VM-PG with our diagonal BB selection outperforms standard PG (BB) for the ill-conditioned  $Q$  with  $\kappa \sim 500$  (Figure 1(b)).

For both the least square and logistic regression under nonnegative constraint, VM-PG exhibits faster or stable converge behavior compared to PG (with BB) (see Figure 1(c), (d)). However, for logistic regression under lasso penalty, there is no clear advantage. VM-PG is more stable in the low-precision regime, whereas PG (BB) outperforms in the high-precision regime (Figure 1(e)). The results with large sample size settings shows no notable difference between the two methods, and has been omitted. This shows that for ill-conditioned problems, having an Euclidean metric as the Hessian approximation may not be sufficient to capture the idiosyncrasies of the local geometry. Hence, allowing additional flexibility through a diagonal metric may be desirable. Additional experiments for other problem types like, QP with lasso penalty, phase retrieval, non-negative matrix factorization etc., shall be included in a longer version.

## 4 Discussion and Conclusion

This paper proposes a diagonal BB metric for the variable proximal gradient method. The proposed diagonal metric provides a better estimate of the local Hessian compared to the naive BB approach. Combined with a nonmonotonic line-search the overall algorithm is guaranteed to converge. Finally, empirical results shows improved convergence behavior for the proposed methodology under ill-conditioned settings.

## References

- [1] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA journal of numerical analysis*, 8(1):141–148, 1988.
- [2] S. Becker and J. Fadili. A quasi-newton proximal splitting method. In *Advances in Neural Information Processing Systems*, pages 2618–2626, 2012.
- [3] E. Chouzenoux, J.-C. Pesquet, and A. Repetti. Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function. *Journal of Optimization Theory and Applications*, 162(1):107–132, 2014.
- [4] P. L. Combettes, L. Condat, J.-C. Pesquet, and B. Vũ. A forward-backward view of some primal-dual optimization methods in image recovery. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 4141–4145. IEEE, 2014.
- [5] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.
- [6] T. Goldstein, C. Studer, and R. Baraniuk. A field guide to forward-backward splitting with a fast implementation. *arXiv preprint arXiv:1411.3406*, 2014.
- [7] L. Grippo, F. Lampariello, and S. Lucidi. A nonmonotone line search technique for newton’s method. *SIAM Journal on Numerical Analysis*, 23(4):707–716, 1986.
- [8] J.-B. Hiriart-Urruty and C. Lemaréchal. Convex analysis and minimization algorithms ii: Advanced theory and bundle methods, vol. 306 of *grundlehren der mathematischen wissenschaften*, 1993.
- [9] J. D. Lee, Y. Sun, and M. A. Saunders. Proximal newton-type methods for minimizing composite functions. *SIAM Journal on Optimization*, 24(3):1420–1443, 2014.
- [10] L. A. Parente, P. A. Lotito, and M. V. Solodov. A class of inexact variable metric proximal point algorithms. *SIAM Journal on Optimization*, 19(1):240–260, 2008.
- [11] N. Parikh, S. Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.
- [12] P. Tseng. A modified forward-backward splitting method for maximal monotone mappings. *SIAM Journal on Control and Optimization*, 38(2):431–446, 2000.
- [13] S. Villa, S. Salzo, L. Baldassarre, and A. Verri. Accelerated and inexact forward-backward algorithms. *SIAM Journal on Optimization*, 23(3):1607–1633, 2013.
- [14] H. Zhang and W. W. Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM journal on Optimization*, 14(4):1043–1056, 2004.
- [15] B. Zhou, L. Gao, and Y.-H. Dai. Gradient methods with adaptive step-sizes. *Computational Optimization and Applications*, 35(1):69–86, 2006.