A quasi-Newton proximal splitting method

Stephen Becker, Jalal M. Fadili

A quasi-Newton proximal splitting method

S. Becker[*] **M.J. Fadili**[†]

Abstract

A new result in convex analysis on the calculation of proximity operators in certain scaled norms is derived. We describe efficient implementations of the proximity calculation for a useful class of functions; the implementations exploit the piece-wise linear nature of the dual problem. The second part of the paper applies the previous result to acceleration of convex minimization problems, and leads to an elegant quasi-Newton method. The optimization method compares favorably against state-of-the-art alternatives. The algorithm has extensive applications including signal processing, sparse recovery and machine learning and classification.

1 Introduction

Convex optimization has proved to be extremely useful to all quantitative disciplines of science. A common trend in modern science is the increase in size of datasets, which drives the need for more efficient optimization schemes. For large-scale unconstrained smooth convex problems, two classes of methods have seen the most success: limited memory quasi-Newton methods and non-linear conjugate gradient (CG) methods. Both of these methods generally outperform simpler methods, such as gradient descent.

For problems with non-smooth terms and/or constraints, it is possible to generalize gradient descent with *proximal gradient descent* (which includes projected gradient descent as a sub-cases), which is just the application of the forward-backward algorithm [1].

Unlike gradient descent, it is not easy to adapt quasi-Newton and CG methods to problems involving constraints and non-smooth terms. Much work has been written on the topic, and approaches generally follow an active-set methodology. In the limit, as the active-set is correctly identified, the methods behave similar to their unconstrained counterparts. These methods have seen success, but are not as efficient or as elegant as the unconstrained versions. In particular, a sub-problem on the active-set must be solved, and the accuracy of this sub-iteration must be tuned with heuristics in order to obtain competitive results.

1.1 Problem statement

Let $\mathcal{H} = (\mathbb{R}^N, \langle\cdot,\cdot\rangle)$ equipped with the usual Euclidean scalar product $\langle x, y\rangle = \sum_{i=1}^N x_i y_i$ and associated norm $\|x\| = \sqrt{\langle x, x\rangle}$. For a matrix $V \in \mathbb{R}^{N\times N}$ in the symmetric positive-definite (SDP) cone $\mathbb{S}_{++}(N)$, we define $\mathcal{H}_V = (\mathbb{R}^N, \langle\cdot,\cdot\rangle_V)$ with the scalar product $\langle x, y\rangle_V = \langle x, Vy\rangle$ and norm $\|x\|_V$ corresponding to the metric induced by V. The dual space of $\mathcal{H}_V$, under $\langle\cdot,\cdot\rangle$, is $\mathcal{H}_{V^{-1}}$. We denote $\mathrm{I}_{\mathcal{H}}$ the identity operator on $\mathcal{H}$.

A real-valued function $f : \mathcal{H} \to \mathbb{R}\cup\{+\infty\}$ is (0)-*coercive* if $\lim_{\|x\|\to+\infty} f(x) = +\infty$. The *domain* of f is defined by $\mathrm{dom}\, f = \{x \in \mathcal{H} : f(x) < +\infty\}$ and f is *proper* if $\mathrm{dom}\, f \neq \emptyset$. We say that a real-valued function f is *lower semi-continuous* (lsc) if $\liminf_{x\to x_0} f(x) \geq f(x_0)$. The

[*]LJLL, CNRS-UPMC, Paris France (stephen.becker@upmc.fr) .

[†]GREYC, CNRS-ENSICAEN-Univ. of Caen, Caen France (Jalal.Fadili@greyc.ensicaen.fr) .

class of all proper lsc convex functions from $\mathcal{H}$ to $\mathbb{R} \cup \{+\infty\}$ is denoted by $\Gamma_0(\mathcal{H})$. The conjugate or Legendre-Fenchel transform of f on $\mathcal{H}$ is denoted f^* .

Our goal is the generic minimization of functions of the form

$$\min_{x \in \mathcal{H}} \{F(x) \triangleq f(x) + h(x)\} , \tag{P}$$

where $f, h \in \Gamma_0(\mathcal{H})$. We also assume the set of minimizers is nonempty (e.g. F is coercive) and that a standard domain qualification holds. We take $f \in C^1(\mathbb{R}^N)$ with L-Lipschitz continuous gradient, and we assume h is separable. Write $x^\star$ to denote an element of $\operatorname{Argmin} F(x)$.

The class we consider covers non-smooth convex optimization problems, including those with convex constraints. Here are some examples in regression, machine learning and classification.

Example 1 (LASSO)**.**

$$\min_{x \in \mathcal{H}} \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_1 . \tag{1}$$

Example 2 (Non-negative least-squares (NNLS))**.**

$$\min_{x \in \mathcal{H}} \frac{1}{2}\|Ax - b\|_2^2 \quad \textit{subject to} \quad x \geqslant 0 . \tag{2}$$

Example 3 (Sparse Support Vector Machines)**.** *One would like to find a linear decision function which minimizes the objective*

$$\min_{x \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} L(\langle x, z_i \rangle + b, y_i) + \lambda \|x\|_1 \tag{3}$$

where for $i = 1, \cdots, m$, $(z_i, y_i) \in \mathbb{R}^N \times \{\pm 1\}$ is the training set, and L is a smooth loss function with Lipschitz-continuous gradient such as the squared hinge loss $L(\hat{y}_i, y_i) = \max(0, 1 - \hat{y}_i y_i)^2$ or the logistic loss $L(\hat{y}_i, y_i) = \log(1 + e^{-\hat{y}_i y_i})$.

1.2 Contributions

This paper introduces a class of scaled norms for which we can compute a proximity operator; these results themselves are significant, for previous results only cover diagonal scaling (the diagonal scaling result is trivial). Then, motivated by the discrepancy between constrained and unconstrained performance, we define a class of limited-memory quasi-Newton methods to solve (P) and that extends naturally and elegantly from the unconstrained to the constrained case. Most well-known quasi-Newton methods for constrained problems, such as L-BFGS-B [2], are only applicable to box constraints $l \leq x \leq u$. The power of our approach is that it applies to a wide-variety of useful non-smooth functionals (see §3.1.4 for a list) and that it does not rely on an active-set strategy. The approach uses the zero-memory SR1 algorithm, and we provide evidence that the non-diagonal term provides significant improvements over diagonal Hessians.

2 Quasi-Newton forward-backward splitting

2.1 The algorithm

In the following, define the quadratic approximation

$$Q_k^B(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2}\|x - x_k\|_B^2, \tag{4}$$

where $B \in \mathbb{S}_{++}(N)$.

The standard (non relaxed) version of the forward-backward splitting algorithm (also known as proximal or projected gradient descent) to solve (P) updates to a new iterate x_{k+1} according to

$$x_{k+1} = \operatorname*{argmin}_x Q_k^{B_k}(x) + h(x) = \operatorname{prox}_{t_k h}(x_k - t_k \nabla f(x_k)) \tag{5}$$

with $B_k = t_k^{-1} \mathrm{I}_{\mathcal{H}}$, $t_k \in]0, 2/L[$ (typically $t_k = 1/L$ unless a line search is used).

Note that this specializes to the gradient descent when $h = 0$. Therefore, if f is a strictly convex quadratic function and one takes $B_k = \nabla^2 f(x_k)$, then we obtain the Newton method. Let's get back to $h \neq 0$. It is now well known that fixed $B = L\mathrm{I}_{\mathcal{H}}$ is usually a poor choice. Since f is smooth and can be approximated by a quadratic, and inspired by quasi-Newton methods, this suggest picking B_k as an approximation of the Hessian. Here we propose a diagonal+rank 1 approximation.

Our diagonal+rank 1 quasi-Newton forward-backward splitting algorithm is listed in Algorithm 1 (with details for the quasi-Newton update in Algorithm 2, see §4 for details). These algorithms are listed as simply as possible to emphasize their important components; the actual software used for numerical tests is open-source and available at `http://www.greyc.ensicaen.fr/~jfadili/software.html`.

Algorithm 1: Zero-memory Symmetric Rank 1 (0SR1) algorithm to solve $\min f + h$

Require: $x_0 \in \mathrm{dom}(f + h)$, Lipschitz constant estimate L of ∇f, stopping criterion ϵ

1: **for** $k = 1, 2, 3, \ldots$ **do**
2: $\quad s_k \leftarrow x_k - x_{k-1}$
3: $\quad y_k \leftarrow \nabla f(x_k) - \nabla f(x_{k-1})$
4: $\quad$ Compute H_k via Algorithm 2, and define $B_k = H_k^{-1}$.
5: $\quad$ Compute the rank-1 proximity operator (see §3)

$$\hat{x}_{k+1} \leftarrow \mathrm{prox}_h^{B_k}(x_k - H_k \nabla f(x_k)) \tag{6}$$

6: $\quad p_k \leftarrow \hat{x}_{k+1} - x_k$ and terminate if $\|p_k\| < \epsilon$
7: $\quad$ Line-search along the ray $x_k + t p_k$ to determine x_{k+1}, or choose $t = 1$.
8: **end for**

2.2 Relation to prior work

First-order methods The algorithm in (5) is variously known as proximal descent or iterated shrinkage/thresholding algorithm (IST or ISTA). It has a grounded convergence theory, and also admits over-relaxation factors $\alpha \in (0, 1)$ [3].

The spectral projected gradient (SPG) [4] method was designed as an extension of the Barzilai-Borwein spectral step-length method to constrained problems. In [5], it was extended to non-smooth problems by allowing general proximity operators; The Barzilai-Borwein method [6] uses a specific choice of step-length t_k motivated by quasi-Newton methods. Numerical evidence suggests the SPG/SpaRSA method is highly effective, although convergence results are not as strong as for ISTA.

FISTA [7] is a multi-step accelerated version of ISTA inspired by the work of Nesterov. The stepsize t is chosen in a similar way to ISTA; in our implementation, we tweak the original approach by using a Barzilai-Borwein step size, a standard line search, and restart[8], since this led to improved performance. Nesterov acceleration can be viewed as an over-relaxed version of ISTA with a specific, non-constant over-relaxation parameter α_k.

The above approaches assume B_k is a constant diagonal. The general diagonal case was considered in several papers in the 1980s as a simple quasi-Newton method, but never widely adapted. More recent attempts include a static choice $B_k \equiv B$ for a primal-dual method [9]. A convergence rate analysis of forward-backward splitting with static and variable B_k where one of the operators is maximal strongly monotone is given in [10].

Active set approaches Active set methods take a simple step, such as gradient projection, to identify active variables, and then uses a more advanced quadratic model to solve for the free variables. A well-known such method is L-BFGS-B [2, 11] which handles general box-constrained problems; we test an updated version [12]. A recent bound-constrained solver is ASA [13] which uses a conjugate gradient (CG) solver on the free variables, and shows good results compared to L-BFGS-B, SPG, GENCAN and TRON. We also compare to several active set approaches specialized for ℓ_1 penalties: "Orthant-wise Learning" (OWL) [14], "Projected Scaled Sub-gradient + Active Set" (PSSas) [15], "Fixed-point continuation + Active Set" (FPC_AS) [16], and "CG + IST" (CGIST) [17].

Other approaches By transforming the problem into a standard conic programming problem, the generic problem is amenable to interior-point methods (IPM). IPM requires solving a Newton-step equation, so first-order like "Hessian-free" variants of IPM solve the Newton-step approximately, either by approximately solving the equation or by subsampling the Hessian. The main issues are speed and robust stopping criteria for the approximations.

Yet another approach is to include the non-smooth h term in the quadratic approximation. Yu et al. [18] propose a non-smooth modification of BFGS and L-BFGS, and test on problems where h is typically a hinge-loss or related function.

The projected quasi-Newton (PQN) algorithm [19, 20] is perhaps the most elegant and logical extension of quasi-Newton methods, but it involves solving a sub-iteration. PQN proposes the SPG [4] algorithm for the subproblems, and finds that this is an efficient tradeoff whenever the cost function (which is not involved in the sub-iteration) is relatively much more expensive to evaluate than projecting onto the constraints. Again, the cost of the sub-problem solver (and a suitable stopping criteria for this inner solve) are issues. As discussed in [21], it is possible to generalize PQN to general non-smooth problems whenever the proximity operator is known (since, as mentioned above, it is possible to extend SPG to this case).

3 Proximity operators and proximal calculus

For space limitation reasons, we only recall essential definitions. More notions, results from convex analysis as well as proofs can be found in the supplementary material.

Definition 4 (Proximity operator [22])**.** *Let $h \in \Gamma_0(\mathcal{H})$. Then, for every $x \in \mathcal{H}$, the function $z \mapsto \frac{1}{2}\|x - z\|^2 + h(z)$ achieves its infimum at a unique point denoted by $\operatorname{prox}_h x$. The uniquely-valued operator $\operatorname{prox}_h : \mathcal{H} \to \mathcal{H}$ thus defined is the proximity operator or proximal mapping of h.*

3.1 Proximal calculus in $\mathcal{H}_V$

Throughout, we denote $\operatorname{prox}_h^V = (\mathrm{I}_{\mathcal{H}_V} + V^{-1}\partial h)^{-1}$, where ∂h is the subdifferential of h, the proximity operator of h w.r.t. the norm endowing $\mathcal{H}_V$ for some $V \in \mathbb{S}_{++}(N)$. Note that since $V \in \mathbb{S}_{++}(N)$, the proximity operator prox_h^V is well-defined.

Lemma 5 (Moreau identity in $\mathcal{H}_V$)**.** *Let $h \in \Gamma_0(\mathcal{H})$, then for any $x \in \mathcal{H}$*

$$\operatorname{prox}_{\rho h^*}^V(x) + \rho V^{-1} \circ \operatorname{prox}_{h/\rho}^{V^{-1}} \circ V(x/\rho) = x, \forall\, 0 < \rho < +\infty\,. \tag{7}$$

Corollary 6.

$$\operatorname{prox}_h^V(x) = x - V^{-1} \circ \operatorname{prox}_{h^*}^{V^{-1}} \circ V(x)\,. \tag{8}$$

3.1.1 Diagonal+rank-1: General case

Theorem 7 (Proximity operator in $\mathcal{H}_V$)**.** *Let $h \in \Gamma_0(\mathcal{H})$ and $V = D + uu^T$, where D is diagonal with (strictly) positive diagonal elements d_i, and $u \in \mathbb{R}^N$. Then,*

$$\operatorname{prox}_h^V(x) = D^{-1/2} \circ \operatorname{prox}_{h \circ D^{-1/2}}(D^{1/2}x - v)\,, \tag{9}$$

where $v = \alpha D^{-1/2}u$ and α is the unique root of

$$p(\alpha) = \left\langle u, x - D^{-1/2} \circ \operatorname{prox}_{h \circ D^{-1/2}} \circ D^{1/2}(x - \alpha D^{-1}u) \right\rangle + \alpha\,, \tag{10}$$

which is a Lipschitz continuous and strictly increasing function on $\mathbb{R}$ with Lipschitz constant $1 + \sum_i u_i^2/d_i$.

Remark 8.

- *Computing prox_h^V amounts to solving a scalar optimization problem that involves the computation of $\operatorname{prox}_{h \circ D^{-1/2}}$. The latter can be much simpler to compute as D is diagonal (beyond the obvious separable case that we will consider shortly). This is typically the case when h is the indicator of the ℓ_1-ball or the canonical simple. The corresponding projector can be obtained in expected complexity $O(N \log N)$ by simple sorting the absolute values*

- *It is of course straightforward to compute* $\mathrm{prox}_{h^*}^V$ *from* prox_h^V *either using Theorem 7, or using this theorem together with Corollary 6 and the Sherman-Morrison inversion lemma.*

3.1.2 Diagonal+rank-1: Separable case

The following corollary is key to our novel optimization algorithm.

Corollary 9. *Assume that* $h \in \Gamma_0(\mathcal{H})$ *is separable, i.e.* $h(x) = \sum_{i=1}^N h_i(x_i)$*, and* $V = D + uu^T$*, where* D *is diagonal with (strictly) positive diagonal elements* d_i*, and* $u \in \mathbb{R}^N$*. Then,*

$$\mathrm{prox}_h^V(x) = \left(\mathrm{prox}_{h_i/d_i}(x_i - v_i/d_i)\right)_i , \tag{11}$$

where $v = \alpha u$ *and* α *is the unique root of*

$$p(\alpha) = \left\langle u, x - \left(\mathrm{prox}_{h_i/d_i}(x_i - \alpha u_i/d_i)\right)_i \right\rangle + \alpha , \tag{12}$$

which is a Lipschitz continuous and strictly increasing function on $\mathbb{R}$*.*

Proof: As h is separable and $D \in \mathbb{S}_{++}(N)$ is diagonal, applying Theorem 7 yields the desired result. ■

Proposition 10. *Assume that for* $1 \leqslant i \leqslant N$*,* prox_{h_i} *is piecewise affine on* $\mathbb{R}$ *with* $k_i \geq 1$ *segments, i.e.*

$$\mathrm{prox}_{h_i}(x_i) = a_j x_i + b_j, \quad t_j \leqslant x_i \leqslant t_{j+1}, j \in \{1, \dots, k_i\} .$$

Let $k = \sum_{i=1}^N k_i$*. Then* $\mathrm{prox}_h^V(x)$ *can be obtained exactly by sorting at most the* k *real values* $\left(\frac{d_i}{u_i}(x_i - t_j)\right)_{(i,j)\in\{1,\dots,N\}\times\{1,\dots,k_i\}}$.

Proof: Recall that (10) has a unique solution. When prox_{h_i} is piecewise affine with k_i segments, it is easy to see that $p(\alpha)$ in (12) is also piecewise affine with slopes and intercepts changing at the k transition points $\left(\frac{d_i}{u_i}(x_i - t_j)\right)_{(i,j)\in\{1,\dots,N\}\times\{1,\dots,k_i\}}$. To get $\alpha^\star$, it is sufficient to isolate the unique segment that intersects the abscissa axis. This can be achieved by sorting the values of the transition points which can cost in average complexity $O(k \log k)$. ■

Remark 11.

- *The above computational cost can be reduced in many situations by exploiting e.g. symmetry of the* $h_i's$*, identical functions, etc. This turns out to be the case for many functions of interest, e.g.* ℓ_1*-norm, indicator of the* ℓ_∞*-ball or the positive orthant, and many others; see examples hereafter.*
- *Corollary 9 can be extended to the "block" separable (i.e. separable in subsets of coordinates) when* D *is piecewise constant along the same block indices.*

3.1.3 Semi-smooth Newton method

In many situations (see examples below), the root of $p(\alpha)$ can be found exactly in polynomial complexity. If no closed-form is available, one can appeal to some efficient iterative method to solve (10) (or (12)). As p is Lipschitz-continuous, hence so-called Newton (slantly) differentiable, semi-smooth Newton are good such solvers, with the proviso that one can design a simple slanting function which can be algorithmically exploited.

The semi-smooth Newton method for the solution of (10) can be stated as the iteration

$$\alpha_{t+1} = \alpha_t - g(\alpha_t)^{-1} p(\alpha_t) , \tag{13}$$

where g is a generalized derivative of p.

Proposition 12 (Generalized derivative of p)**.** *If* $\mathrm{prox}_{h\circ D^{-1/2}}$ *is Newton differentiable with generalized derivative* G*, then so is the mapping* p *with a generalized derivative*

$$g(\alpha) = 1 + \left\langle u, D^{-1/2} \circ G(D^{1/2}x - \alpha D^{-1/2}u) \circ D^{-1/2}u \right\rangle .$$

Furthermore, g *is nonsingular with a uniformly bounded inverse on* $\mathbb{R}$*.*

Function h	Algorithm
ℓ_1-norm	Separable: exact in $O(N \log N)$
Hinge	Separable: exact in $O(N \log N)$
ℓ_∞-ball	Separable: exact in $O(N \log N)$ from ℓ_1-norm by Moreau-identity
Box constraint	Separable: exact in $O(N \log N)$
Positivity constraint	Separable: exact in $O(N \log N)$
ℓ_1-ball	Nonseparable: semismooth Newton and $\mathrm{prox}_{h \circ D^{-1/2}}$ costs $O(N \log N)$
ℓ_∞-norm	Nonseparable: from projector on the ℓ_1-ball by Moreau-identity
Canonical simplex	Nonseparable: semismooth Newton and $\mathrm{prox}_{h \circ D^{-1/2}}$ costs $O(N \log N)$
$\max$ function	Nonseparable: from projector on the simplex by Moreau-identity

Table 1: Summary of functions which have efficiently computable rank-1 proximity operators

Proof: This follows from linearity and the chain rule [23, Lemma 3.5]. The second statement follows strict increasing monotonicity of p as established in Theorem 7. ■

Thus, as p is Newton differentiable with nonsingular generalized derivative whose inverse is also bounded, the general semi-smooth Newton convergence theorem implies that (13) converges superlinearly to the unique root of (10).

3.1.4 Examples

Many functions can be handled very efficiently using our results above. For instance, Table 1 summarizes a few of them where we can obtain either an exact answer by sorting when possible, or else by minimizing w.r.t. to a scalar variable (*i.e.* finding the unique root of (10)).

4 A primal rank 1 SR1 algorithm

Following the conventional quasi-Newton notation, we let B denote an approximation to the Hessian of f and H denote an approximation to the inverse Hessian. All quasi-Newton methods update an approximation to the (inverse) Hessian that satisfies the *secant condition*:

$$H_k y_k = s_k, \quad y_k = \nabla f(x_k) - \nabla f(x_{k-1}), \quad s_k = x_k - x_{k-1} \tag{14}$$

Algorithm 1 follows the SR1 method [24], which uses a rank-1 update to the inverse Hessian approximation at every step. The SR1 method is perhaps less well-known than BFGS, but it has the crucial property that updates are rank-1, rather than rank-2, and it is described "[SR1] has now taken its place alongside the BFGS method as the pre-eminent updating formula." [25].

We propose two important modifications to SR1. The first is to use limited-memory, as is commonly done with BFGS. In particular, we use zero-memory, which means that at every iteration, a new diagonal plus rank-one matrix is formed. The other modification is to extend the SR1 method to the general setting of minimizing $f + h$ where f is smooth but h need not be smooth; this further generalizes the case when h is an indicator function of a convex set. Every step of the algorithm replaces f with a quadratic approximation, and keeps h unchanged. Because h is left unchanged, the subgradient of h is used in an *implicit* manner, in comparison to methods such as [18] that use an approximation to h as well and therefore take an *explicit* subgradient step.

Choosing H_0 In our experience, the choice of H_0 is best if scaled with a Barzilai-Borwein spectral step length

$$\tau_{\text{BB2}} = \langle s_k, y_k \rangle \,/\, \langle y_k, y_k \rangle \tag{15}$$

(we call it τ_{BB2} to distinguish it from the other Barzilai-Borwein step size $\tau_{\text{BB1}} = \langle s_k, s_k \rangle \,/\, \langle s_k, y_k \rangle \geqslant \tau_{\text{BB2}}$).

In SR1 methods, the quantity $\langle s_k - H_0 y_k, y_k \rangle$ must be positive in order to have a well-defined update for u_k. The update is:

$$H_k = H_0 + u_k u_k^T, \quad u_k = (s_k - H_0 y_k) / \sqrt{\langle s_k - H_0 y_k, y_k \rangle}. \tag{16}$$

Algorithm 2: Sub-routine to compute the approximate inverse Hessian H_k

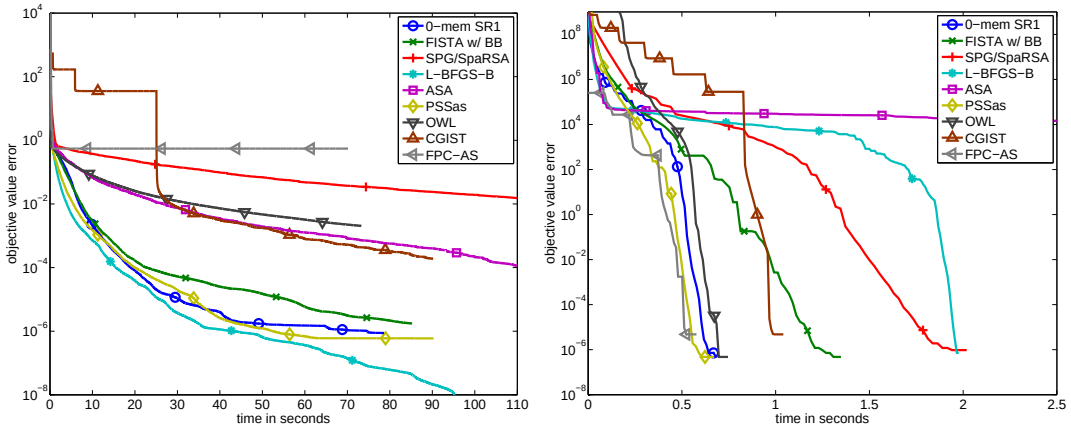
Require: $k, s_k, y_k,\ 0 < \gamma < 1,\ 0 < \tau_{\min} < \tau_{\max}$

if $k = 1$ **then**
  $H_0 \leftarrow \tau \mathrm{I}_{\mathcal{H}}$ where $\tau > 0$ is arbitrary
  $u_k \leftarrow 0$
else
  $\tau_{\mathrm{BB2}} \leftarrow \frac{\langle s_k, y_k \rangle}{\|y_k\|^2}$ {Barzilai-Borwein step length}
  Project τ_{BB2} onto $[\tau_{\min}, \tau_{\max}]$
  $H_0 \leftarrow \gamma \tau_{\mathrm{BB2}} \mathrm{I}_{\mathcal{H}}$
  if $\langle s_k - H_0 y_k, y_k \rangle \leq 10^{-8} \|y_k\|_2 \|s_k - H_0 y_k\|_2$ **then**
    $u_k \leftarrow 0$ {Skip the quasi-Newton update}
  else
    $u_k \leftarrow (s_k - H_0 y_k) / \sqrt{\langle s_k - H_0 y_k, y_k \rangle}$.
  end if
end if
return $H_k = H_0 + u_k u_k^T$ $\{B_k = H_k^{-1}$ can be computed via the Sherman-Morrison formula$\}$

For this reason, we choose $H_0 = \gamma \tau_{\mathrm{BB2}} \mathrm{I}_{\mathcal{H}}$ with $0 < \gamma < 1$, and thus $0 \leq \langle s_k - H_0 y_k, y_k \rangle = (1 - \gamma) \langle s_k, y_k \rangle$. If $\langle s_k, y_k \rangle = 0$, then there is no symmetric rank-one update that satisfies the secant condition. The inequality $\langle s_k, y_k \rangle > 0$ is the *curvature condition*, and it is guaranteed for all strictly convex objectives. Following the recommendation in [26], we skip updates whenever $\langle s_k, y_k \rangle$ cannot be guaranteed to be non-zero given standard floating-point precision.

A value of $\gamma = 0.8$ works well in most situations. We have tested picking γ adaptively, as well as trying H_0 to be non-constant on the diagonal, but found no consistent improvements.

5 Numerical experiments and comparisons



(a) (b)

Figure 1: (a) is first LASSO test, (b) is second LASSO test

Consider the unconstrained LASSO problem (1). Many codes, such as [27] and L-BFGS-B [2], handle only non-negativity or box-constraints. Using the standard change of variables by introducing the positive and negative parts of x, the LASSO can be recast as

$$\min_{x_+, x_- \geqslant 0} \frac{1}{2} \|Ax_+ - Ax_- - b\|^2 + \lambda \mathbf{1}^T (x_+ + x_-)$$

and then x is recovered via $x = x_+ - x_-$. With such a formulation solvers such as L-BFGS-B are applicable. However, this constrained problem has twice the number of variables, and the Hessian of

the quadratic part changes from A^TA to $\tilde{A} = \begin{pmatrix} A^TA & -A^TA \\ -A^TA & A^TA \end{pmatrix}$ which necessarily has (at least) n degenerate 0 eigenvalues and adversely affects solvers.

A similar situation occurs with the hinge-loss function. Consider the shifted and reversed hinge loss function $h(x) = \max(0, x)$. Then one can split $x = x_+ - x_-$, add constraints $x_+ \geqslant 0, x_- \geqslant 0$, and replace $h(x)$ with $\mathbf{1}^T(x_+)$. As before, the Hessian gains n degenerate eigenvalues.

We compared our proposed algorithm on the LASSO problem. The first example, in Fig. 1a, is a typical example from compressed sensing that takes $A \in \mathbb{R}^{m\times n}$ to have iid $\mathcal{N}(0, 1)$ entries with $m = 1500$ and $n = 3000$. We set $\lambda = 0.1$. L-BFGS-B does very well, followed closely by our proposed SR1 algorithm and PSSas. Note that L-BFGS-B and ASA are in Fortran and C, respectively (the other algorithms are in Matlab).

Our second example uses a square operator A with dimensions $n = 13^3 = 2197$ chosen as a 3D discrete differential operator. This example stems from a numerical analysis problem to solve a discretized PDE as suggested by [28]. For this example, we set $\lambda = 1$. For all the solvers, we use the same parameters as in the previous example. Unlike the previous example, Fig. 1b now shows that L-BFGS-B is very slow on this problem. The FPC-AS method, very slow on the earlier test, is now the fastest. However, just as before, our SR1 method is nearly as good as the best algorithm. This robustness is one benefit of our approach, since the method does not rely on active-set identifying parameters and inner iteration tolerances.

6 Conclusions

In this paper, we proposed a novel variable metric (quasi-Newton) forward-backward splitting algorithm, designed to efficiently solve non-smooth convex problems structured as the sum of a smooth term and a non-smooth one. We introduced a class of weighted norms induced by a diagonal+rank 1 symmetric positive definite matrices, and proposed a whole framework to compute a proximity operator in the weighted norm. The latter result is distinctly new and is of independent interest. We also provided clear evidence that the non-diagonal term provides significant acceleration over diagonal matrices.

The proposed method can be extended in several ways. Although we focused on forward-backward splitting, our approach can be easily extended to the new *generalized* forward-backward algorithm of [29]. However, if we switch to a primal-dual setting, which is desirable because it can handle more complicated objective functionals, updating B_k is non-obvious. Though one can think of non-diagonal pre-conditioning methods.

Another improvement would be to derive efficient calculation for rank-2 proximity terms, thus allowing a 0-memory BFGS method. We are able to extend (result not presented here) Theorem 7 to diagonal+rank r matrices. However, in general, one must solve an r-dimensional inner problem using the semismooth Newton method.

A final possible extension is to take B_k to be diagonal plus rank-1 on diagonal blocks, since if h is separable, this is still can be solved by our algorithm (see Remark 10). The challenge here is adapting this to a robust quasi-Newton update. For some matrices that are well-approximated by low-rank blocks, such as H-matrices [30], it may be possible to choose $B_k \equiv B$ to be a fixed preconditioner.

Acknowledgments

SB would like to acknowledge the Fondation Sciences Mathématiques de Paris for his fellowship.

References

[1] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces.* Springer-Verlag, New York, 2011.

[2] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Computing*, 16(5):1190–1208, 1995.

[3] P. L. Combettes and J. C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer-Verlag, New York, 2011.

[4] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optim.*, 10(4):1196–1211, 2000.

[5] S. Wright, R. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57, 2009. 2479–2493.

[6] J. Barzilai and J. Borwein. Two point step size gradient method. *IMA J. Numer. Anal.*, 8:141–148, 1988.

[7] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. on Imaging Sci.*, 2(1):183–202, 2009.

[8] B. O'Donoghue and E. Candès. Adaptive restart for accelerated gradient schemes. *Preprint: arXiv:1204.3982*, 2012.

[9] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *ICCV*, 2011.

[10] G. H.-G. Chen and R. T. Rockafellar. Convergence rates in forward–backward splitting. *SIAM Journal on Optimization*, 7(2):421–444, 1997.

[11] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Software*, 23(4):550–560, 1997.

[12] José Luis Morales and Jorge Nocedal. Remark on älgorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization. *ACM Trans. Math. Softw.*, 38(1):7:1–7:4, 2011.

[13] W. W. Hager and H. Zhang. A new active set algorithm for box constrained optimization. *SIAM J. Optim.*, 17:526–557, 2006.

[14] A. Andrew and J. Gao. Scalable training of l_1-regularized log-linear models. In *ICML*, 2007.

[15] M. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for l1 regularization: A comparative study and two new approaches. In *European Conference on Machine Learning*, 2007.

[16] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang. A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization and continuation. *SIAM J. Sci. Comput.*, 32(4):1832–1857, 2010.

[17] T. Goldstein and S. Setzer. High-order methods for basis pursuit. Technical report, CAM-UCLA, 2011.

[18] J. Yu, S.V.N. Vishwanathan, S. Guenter, and N. Schraudolph. A quasi-Newton approach to nonsmooth convex optimization problems in machine learning. *J. Machine Learning Research*, 11:1145–1200, 2010.

[19] M. Schmidt, E. van den Berg, M. Friedlander, and K. Murphy. Optimizing costly functions with simple constraints: A limited-memory projected quasi-Newton algorithm. In *AISTATS*, 2009.

[20] M. Schmidt, D. Kim, and S. Sra. Projected Newton-type methods in machine learning. In S. Sra, S. Nowozin, and S.Wright, editors, *Optimization for Machine Learning*. MIT Press, 2011.

[21] J. D. Lee, Y. Sun, and M. A. Saunders. Proximal Newton-type methods for minimizing convex objective functions in composite form. *Preprint: arXiv:1206.1623*, 2012.

[22] J.-J. Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *CRAS Sér. A Math.*, 255:2897–2899, 1962.

[23] R. Griesse and D. A. Lorenz. A semismooth Newton method for Tikhonov functionals with sparsity constraints. *Inverse Problems*, 24(3):035007, 2008.

[24] C. Broyden. Quasi-Newton methods and their application to function minimization. *Math. Comp.*, 21:577–593, 1967.

[25] N. Gould. Seminal papers in nonlinear optimization. In *An introduction to algorithms for continuous optimization*. Oxford University Computing Laboratory, 2006. `http://www.numerical.rl.ac.uk/nimg/course/lectures/paper/paper.pdf`.

[26] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[27] I. Dhillon, D. Kim, and S. Sra. Tackling box-constrained optimization via a new projected quasi-Newton approach. *SIAM J. Sci. Comput.*, 32(6):3548–3563, 2010.

[28] Roger Fletcher. On the Barzilai-Borwein method. In Liqun Qi, Koklay Teo, Xiaoqi Yang, Panos M. Pardalos, and Donald W. Hearn, editors, *Optimization and Control with Applications*, volume 96 of *Applied Optimization*, pages 235–256. Springer US, 2005.

[29] H. Raguet, J. Fadili, and G. Peyré. Generalized forward-backward splitting. Technical report, Preprint Hal-00613637, 2011.

[30] W. Hackbusch. A sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices. *Computing*, 62:89–108, 1999.

[31] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.