A quasi-Newton proximal splitting method

S. Becker∗ M.J. Fadili†

June 20, 2012

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 $H = (\mathbb{R}^N, \langle \cdot, \cdot \rangle)$ 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 $H_V = (\mathbb{R}^N, \langle \cdot, \cdot \rangle_V)$ with the scalar product $\langle x, y \rangle_V = \langle x, V y \rangle$ and norm $\|x\|_V$ corresponding to the metric induced by $V$. The dual space of $H_V$, under $\langle \cdot, \cdot \rangle$, is $H_{V^{-1}}$. We denote $I_H$ the identity operator on $H$.

A real-valued function $f : H \rightarrow \mathbb{R} \cup \{+\infty\}$ is $(0)$-coercive if $\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$. The domain of $f$ is defined by $\text{dom} f = \{x \in H : f(x) < +\infty\}$ and $f$ is proper if $\text{dom} f \neq \emptyset$. We say that a real-valued

∗LJLL, CNRS-UPMC, Paris France (stephen.becker@upmc.fr).
†GREYC, CNRS-ENSICAEN-Université de Caen, 14050 Caen France (Jalal.Fadili@greyc.ensicaen.fr).
function $f$ is lower semi-continuous (lsc) if $\liminf_{x \to x_0} f(x) \geq f(x_0)$. The 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
\[
\min_{x \in \mathcal{H}} \{ F(x) \triangleq f(x) + h(x) \},
\]
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 $x^*$ to denote an element of $\text{Argmin} F(x)$.

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

**Example 2** (Non-negative least-squares (NNLS)).
\[
\min_{x \in \mathcal{H}} \frac{1}{2} \| Ax - b \|_2^2 \text{ subject to } x \geq 0.
\]

**Example 3** (Sparse Support Vector Machines). One would like to find a linear decision function which minimizes the objective
\[
\min_{x \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^m L(\langle x, z_i \rangle + b, y_i) + \lambda \| x \|_1
\]
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, 4, 5] 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
\[
Q_k^B(x) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} \| x - x_k \|_B^2,
\]
where $B \in S_+(N)$.

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

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

with $B_k = t_{k-1} H$, $t_k \in ]0, 2/L[$ (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 $B = LH$ 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](http://www.greyc.ensicaen.fr/~jfadili/software.html).

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

<table>
<thead>
<tr>
<th>Require: $x_0 \in \text{dom}(f + h)$, Lipschitz constant estimate $L$ of $\nabla f$, stopping criterion $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: for $k = 1, 2, 3, \ldots$ do</td>
</tr>
<tr>
<td>2: $s_k \leftarrow x_k - x_{k-1}$</td>
</tr>
<tr>
<td>3: $y_k \leftarrow \nabla f(x_k) - \nabla f(x_{k-1})$</td>
</tr>
<tr>
<td>4: Compute $H_k$ via Algorithm 2 and define $B_k = H_k^{-1}$.</td>
</tr>
<tr>
<td>5: Compute the rank-1 proximity operator (see §3) $\hat{x}<em>{k+1} \leftarrow \operatorname{prox}</em>{t_k h}(x_k - H_k \nabla f(x_k))$</td>
</tr>
<tr>
<td>6: $p_k \leftarrow \hat{x}_{k+1} - x_k$ and terminate if $|p_k| &lt; \epsilon$</td>
</tr>
<tr>
<td>7: Line-search along the ray $x_k + tp_k$ to determine $x_{k+1}$, or choose $t = 1$.</td>
</tr>
<tr>
<td>8: end for</td>
</tr>
</tbody>
</table>

### 2.2 Relation to prior work

**First-order methods** The algorithm in [9] is variously known as proximal descent or iterated shrinkage/thresholding algorithm (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; we refer to this as SPG/SpaRSA (N.B. we do not use the SpaRSA implementation since we do not use warm-starts or restarts, in order to be fair to all algorithms). The Barzilai-Borwein method [6] use 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 $\alpha_k$.

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 \cite{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 \cite{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 \cite{2,11} which handles general box-constrained problems; we test an updated version \cite{12}. A recent bound-constrained solver is ASA \cite{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 $\ell_1$ penalties: “Orthant-wise Learning” (OWL) \cite{14}, “Projected Scaled Sub-gradient + Active Set” (PSSas) \cite{15}, “Fixed-point continuation + Active Set” (FPC\_AS) \cite{16}, and “CG + IST” (CGIST) \cite{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. \cite{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 \cite{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 \cite{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 \cite{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

We only recall essential definitions. More notions and results from convex analysis can be found in §A.

**Definition 4** (Proximity operator \cite{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 $\text{prox}_h x$. The uniquely-valued operator $\text{prox}_h : \mathcal{H} \to \mathcal{H}$ thus defined is the proximity operator or proximal mapping of $h$.

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

Throughout, we denote $\text{prox}^V_h = (I_{\mathcal{H}_V} + V^{-1} \partial h)^{-1}$, where $\partial h$ is the subdifferential of $h$, the proximity operator of $h$ w.r.t. the norm endowing $\mathcal{H}_V$ for some $V \in S_{++}(N)$. Note that since $V \in S_{++}(N)$, the proximity operator $\text{prox}^V_h$ is well-defined.

**Lemma 5** (Moreau identity in $\mathcal{H}_V$). Let $h \in \Gamma^0(\mathcal{H})$, then for any $x \in \mathcal{H}$

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

(7)

The proof is in \cite{B.1}

**Corollary 6.**

$$\text{prox}^V_h (x) = x - V^{-1} \circ \text{prox}^{V^{-1}}_h \circ V(x).$$  

(8)
3.1.1 Diagonal+rank-1: General case

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

$$\text{prox}_h^V(x) = D^{-1/2} \circ \text{prox}_{h_0 D^{-1/2}}(D^{1/2} x - v),$$

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

$$p(\alpha) = \left\langle u, x - D^{-1/2} \circ \text{prox}_{h_0 D^{-1/2}}(D^{1/2} x - \alpha D^{-1} u) \right\rangle + \alpha,$$

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

The proof is in §B.2.

**Remark 8.**

- Computing $\text{prox}_h^V$ amounts to solving a scalar optimization problem that involves the computation of $\text{prox}_{h_0 D^{-1/2}}$. 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 $\ell_1$-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 $\text{prox}_h^v$ from $\text{prox}_h^V$ 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 $h \in \Gamma_0(H)$ 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,

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

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

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

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

**Proof.** As $h$ is separable and $D \in S_+(N)$ is diagonal, applying Theorem 7 together with Lemma 24(ii)-(iii) the desired result follows.

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

$$\text{prox}_{h_i}(x_i) = a_j x_i + b_j, \quad t_j \leq x_i \leq t_{j+1}, j \in \{1, \ldots, k_i\}.$$

Let $k = \sum_{i=1}^N k_i$. Then $\text{prox}_h^V(x)$ can be obtained exactly by sorting at most the $k$ real values

$$\left( \frac{d_i}{w_i}(x_i - t_j) \right)_{(i,j) \in \{1,\ldots,N\} \times \{1,\ldots,k_i\}}.$$

5
Proof. Recall that (10) has a unique solution. When \( \text{prox}_{h_i} \) 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 \( \left( \frac{d_i}{m_i}(x_i - t_j) \right)_{(i,j) \in \{1,...,N\} \times \{1,...,k_i\}} \). To get \( \alpha^* \), 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. \( \ell_1 \)-norm, indicator of the \( \ell_\infty \)-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

\[
\alpha_{t+1} = \alpha_t - g(\alpha_t)^{-1} p(\alpha_t),
\]

(13)

where \( g \) is a generalized derivative of \( p \).

Proposition 12 (Generalized derivative of \( p \)). If \( \text{prox}_{h \circ D^{-1/2}} \) 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 \( \mathbb{R} \).

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)).

To put Proposition 12 on a more concrete footing, we briefly cover the positivity constraint explicitly. Let \( V = D + uu^T \) and \( h(x) = l_{[x:x \geq 0]} \). We will calculate

\[
\text{prox}_{h}^{V^{-1}}(x) = \arg\min_{y \geq 0} \frac{1}{2} \|y - x\|_{V^{-1}}^2.
\]

(14)
Function $h$ & Algorithm \\
$t_1$-norm & Separable: exact in $O(N \log N)$ \\
Hinge & Separable: exact in $O(N \log N)$ \\
$t_\infty$-ball & Separable: exact in $O(N \log N)$ from $t_1$-norm by Moreau-identity \\
Box constraint & Separable: exact in $O(N \log N)$ \\
Positivity constraint & Separable: exact in $O(N \log N)$ \\
$t_1$-ball & Nonseparable: semismooth Newton and prox $h \circ D_\frac{1}{2}$ costs $O(N \log N)$ \\
$t_\infty$-norm & Nonseparable: from projector on the $t_1$-ball by Moreau-identity \\
Canonical simplex & Nonseparable: semismooth Newton and prox $h \circ D_\frac{1}{2}$ 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

Since we work with $V^{-1}$ and not $V$, we will not use $p(\alpha)$ but rather $\hat{p}(\alpha)$ which will be used in a similar way to $p$.

If $(y, \lambda)$ is a primal-dual solution to (14), the KKT conditions must be satisfied:

$$y \geq 0, \lambda \geq 0, \quad y^T \lambda = 0, \quad y = x + (D + uu^T)\lambda$$

(15)

Define the scalar $\alpha = u^T \lambda$. The key observation is that if $\alpha$ is known, then the problem is solved since it becomes separable and the solution is

$$y_i = (x_i + \alpha u_i)_+, \quad \lambda_i = (- (x_i + \alpha u_i)/d_i)_+, \quad i = 1, \ldots, N$$

where $(x_i)_+ := \max(0, x_i)$. Let $\lambda^{(\alpha)}_i := (- (x_i + \alpha u_i)/d_i)_+$, so we search for a value of $\alpha$ such that $\alpha = u^T \lambda^{(\alpha)}$, or in other words, a root of $\hat{p}(\alpha) = \alpha - u^T \lambda^{(\alpha)}$.

Define $\hat{\alpha}_i$ to be the sorted values of $(-x_i/u_i)$, so we see that $\hat{p}$ is linear in the regions $[\hat{\alpha}_i, \hat{\alpha}_{i+1}]$ and so it is trivial to check if $\hat{p}$ has a root in this region. Thus the problem is reduced to finding the correct region $i$, which can be done efficiently by a binary search over $\log_2(n)$ values of $i$ since $\hat{p}$ is monotonic. To see that $\hat{p}$ is monotonic, we write it as

$$\hat{p}(\alpha) = \alpha + \sum_{i=1}^N (u_i x_i + \alpha u_i^2/d_i) \chi_i(\alpha)$$

where $\chi_i(\alpha)$ encodes the positivity constraint in the argument of $(\cdot)_+$ and is thus either 0 or 1, hence the slope is always positive.

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:

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

(16)

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.

**Algorithm 2:** Sub-routine to compute the approximate inverse Hessian \( H_k \)

<table>
<thead>
<tr>
<th>Require: ( k, s_k, y_k ), ( 0 &lt; \gamma &lt; 1 ), ( 0 &lt; \tau_{\text{min}} &lt; \tau_{\text{max}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: if ( k = 1 ) then</td>
</tr>
<tr>
<td>2: ( H_0 \leftarrow \tau I_H ) where ( \tau &gt; 0 ) is arbitrary</td>
</tr>
<tr>
<td>3: ( u_k \leftarrow 0 )</td>
</tr>
<tr>
<td>4: else</td>
</tr>
<tr>
<td>5: ( \tau_{\text{BB2}} \leftarrow \frac{\langle s_k, y_k \rangle}{| y_k |^2} ) {Barzilai-Borwein step length}</td>
</tr>
<tr>
<td>6: Project ( \tau_{\text{BB2}} ) onto ([\tau_{\text{min}}, \tau_{\text{max}}])</td>
</tr>
<tr>
<td>7: ( H_0 \leftarrow \gamma \tau_{\text{BB2}} I_H )</td>
</tr>
<tr>
<td>8: 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</td>
</tr>
<tr>
<td>9: ( u_k \leftarrow 0 ) {Skip the quasi-Newton update}</td>
</tr>
<tr>
<td>10: else</td>
</tr>
<tr>
<td>11: ( u_k \leftarrow (s_k - H_0 y_k) / \sqrt{\langle s_k - H_0 y_k, y_k \rangle} ).</td>
</tr>
<tr>
<td>12: end if</td>
</tr>
<tr>
<td>13: end if</td>
</tr>
<tr>
<td>14: return ( H_k = H_0 + u_k u_k^T ) {( B_k = H_k^{-1} ) can be computed via the Sherman-Morrison formula}</td>
</tr>
</tbody>
</table>

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

\[
\tau_{\text{BB2}} = \frac{\langle s_k, y_k \rangle}{\langle y_k, y_k \rangle} \tag{17}
\]

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

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:

\[
H_k = H_0 + u_k u_k^T, \quad u_k = (s_k - H_0 y_k) / \sqrt{\langle s_k - H_0 y_k, y_k \rangle}. \tag{18}
\]

For this reason, we choose \( H_0 = \gamma \tau_{\text{BB2}} I_H \) 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 [20], 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 \( \gamma \) adaptively, as well as trying \( H_0 \) to be non-constant on the diagonal, but found no consistent improvements.

### 5 Numerical experiments and comparisons

Consider the unconstrained LASSO problem [1]. Many codes, such as [27] and L-BFGS-B [4], 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

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

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 $A^T A$ to $\tilde{A} = \begin{pmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{pmatrix}$ 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 $x_+ \geq 0, x_- \geq 0$, and replace $h(x)$ with $\mathbb{1}^T(x_+)$. 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 op-
erator 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.
A Elements from convex analysis

We here collect some results from convex analysis that are key for our proof. Some lemmata are listed without proof and can be either easily proved or found in standard references such as [31, 1].

A.1 Background

Functions

Definition 13 (Indicator function). Let $C$ a nonempty subset of $H$. The indicator function $\mathbb{1}_C$ of $C$ is

$$\mathbb{1}_C(x) = \begin{cases} 0, & \text{if } x \in C, \\ +\infty, & \text{otherwise}. \end{cases}$$

$\text{dom}(\mathbb{1}_C) = C$.

Definition 14 (Infimal convolution). Let $h_1$ and $h_2$ two functions from $H$ to $\mathbb{R} \cup \{+\infty\}$. Their infimal convolution is the function from $H$ to $\mathbb{R} \cup \{\pm \infty\}$ defined by:

$$(h_1 \vee h_2)(x) = \inf \{h_1(x_1) + h_2(x_2) : x_1 + x_2 = x\} = \inf_{y \in H} h_1(y) + h_2(x - y).$$

Conjugacy

Definition 15 (Conjugate). Let $h : H \to \mathbb{R} \cup \{+\infty\}$ having a minorizing affine function. The conjugate or Legendre-Fenchel transform of $h$ on $H$ is the function $h^*$ defined by

$$h^*(v) = \sup_{x \in \text{dom}(h)} \langle v, x \rangle - h(x).$$

Lemma 16 (Calculus rules).

(i) $(h(x) + t)^*(v) = h^*(v) - t$.

(ii) $(th(x))^*(v) = t f^*(v/t), \ t > 0$.

(iii) $(h \circ A)^* = h^* \circ (A^{-1})^*$ if $A$ is a linear invertible operator.

(iv) $(h(x - x_0))^*(v) = h^*(v) + \langle v, x_0 \rangle$.

(v) Separability: $(\sum_{i=1}^n h_i(x_i))^*(v_1, \ldots, v_n) = \sum_{i=1}^n h_i^*(v_i)$, where $(x_1, \ldots, x_n) \in H_1 \times \cdots \times H_n$.

(vi) Conjugate of a sum: assume $h_1, h_2 \in \Gamma_0(H)$ and the relative interiors of their domains have a nonempty intersection. Then

$$(h_1 + h_2)^* = h_1^* \vee h_2^*.$$

(vii) Conjugate in $H_V$ for $V \in S_{++}(N)$: $h_V^*(u) = h^*(Vu)$.

Lemma 17 (Conjugate of a degenerate quadratic function). Let $Q$ be a symmetric positive semi-definite matrix. Let $Q^+$ be its Moore-Penrose pseudo-inverse. Then,

$$\left(\frac{1}{2} \|y - \cdot\|_Q^2\right)^*(v) = \begin{cases} \frac{1}{2} \|y - v\|_{Q^+}^2 & \text{if } v \in y + \text{Im}(Q), \\ +\infty, & \text{otherwise}. \end{cases} \quad (19)$$

Lemma 18 (Conjugate of a rank-1 quadratic function). Let $u \in H$. Then,

$$\left(\frac{1}{2} \langle u, \cdot \rangle^2\right)^*(v) = \begin{cases} \frac{\|v\|^2}{2\|u\|^2} & \text{if } v \in \text{R}u, \\ +\infty, & \text{otherwise}. \end{cases} \quad (20)$$
Subdifferential

**Definition 19** (Subdifferential). The subdifferential of a proper convex function \( h \in \Gamma_0(\mathcal{H}) \) at \( x \in \mathcal{H} \) is the set-valued map \( \partial h : \mathcal{H} \to 2^\mathcal{H} \)
\[
\partial h(x) = \{ v \in \mathcal{H} | \forall z \in \mathcal{H}, h(z) \geq h(x) + \langle v, z-x \rangle \}.
\]
An element \( v \) of \( \partial h \) is called a subgradient.

The subdifferential map \( \partial h \) is a maximal monotone operator from \( \mathcal{H} \to 2^\mathcal{H} \).

**Lemma 20.** If \( h \) is (Gâteaux) differentiable at \( x \), its only subgradient at \( x \) is its gradient \( \nabla h(x) \).

**Lemma 21.** Let \( V \in \mathbb{S}_+(\mathbb{N}) \). Then \( V \partial h \) is the subdifferential of \( h \) in \( \mathcal{H}_V \).

**Fenchel-Rockafellar duality** The duality formula to be stated shortly will be very useful throughout the rest of the paper.

**Lemma 22.** Let \( h \in \Gamma_0(\mathcal{H}) \) and \( g \in \Gamma_0(\mathcal{K}) \), and \( A \triangleq L \cdot -y : \mathcal{H} \to \mathcal{K} \) be a bounded affine operator, and \( \mathcal{K} = (\mathbb{R}^m, \langle \cdot, \cdot \rangle) \). Suppose that \( 0 \in \text{ri} (\text{dom} g - A(\text{dom} h)) \). Then
\[
\inf_{x \in \mathcal{H}} h(x) + g \circ A(x) = -\min_{u \in \mathcal{K}} h^*(L^* u) + g^*(u) + \langle u, y \rangle \tag{21}
\]
with the relationships between \( x^* \) and \( u^* \), respectively the solutions of the primal and dual problems
\[
h(x^*) + h^*(-L^* u^*) = \langle -L^* u^*, x^* \rangle, \tag{22}
\]
\[
g(A x^*) + g^*(u^*) = \langle u^*, A x^* \rangle \tag{23}
\]
or equivalently
\[
x^* \in \partial h^*(-L^* u^*) \text{ and } u^* \in \partial g(A x^*) \tag{24}
\]
\[
-L^* u^* \in \partial h(x^*) \text{ and } A x^* \in \partial g^*(u^*) . \tag{25}
\]

### A.2 Proximal calculus in \( \mathcal{H} \)

**Definition 23** (Moreau envelope \([22]\)). The function \( \rho h(x) = \inf_{z \in \mathcal{H}} \frac{1}{2\rho} \| x-z \|^2 + h(z) \) for \( 0 < \rho < +\infty \) is the Moreau envelope of index \( \rho \) of \( h \).

\( \rho h \) is also the infimal convolution of \( h \) with \( \frac{1}{2\rho} ||\cdot||^2 \).

**Lemma 24.**

(i) Translation: \( \text{prox}_{\rho h}(-y)(x) = y + \text{prox}_{h}(x-y) \).

(ii) Scaling: \( \forall \rho \in (-\infty, \infty), \text{prox}_{\rho h}(x) = \text{prox}_{\rho^2 f}(\rho x)/\rho \).

(iii) Separability: let \( (h_i)_{1 \leq i \leq n} \) a family of functions each in \( \Gamma_0(\mathbb{R}) \) and \( h(x) = \sum_{i=1}^{N} h_i(x_i) \). Then \( h \) is in \( \Gamma_0(\mathcal{H}) \) and \( \text{prox}_{h} = (\text{prox}_{h_i})_{1 \leq i \leq N} \).

**Lemma 25.** Let \( h \in \Gamma_0(\mathcal{H}) \). Then its Moreau envelope \( \rho h \) is convex and Fréchet-differentiable with \( 1/\rho \)-Lipschitz gradient
\[
\nabla \rho h = (I_H - \text{prox}_{\rho h})/\rho . \tag{26}
\]

**Lemma 26** (Moreau identity). Let \( h \in \Gamma_0(\mathcal{H}) \), then for any \( x \in \mathcal{H} \)
\[
\text{prox}_{\rho h}{^*}(x) + \rho \text{prox}_{h/\rho}(x/\rho) = x , \forall 0 < \rho < +\infty . \tag{27}
\]

From Lemma 26 we conclude that
\[
\text{prox}_{h^*} = I_H - \text{prox}_h , \text{ prox}_{h^*}(x) \in \partial h(x) .
\]
B Proofs

B.1 Proof of Lemma 5

Proof. We have
\[ p = \text{prox}_{\rho h^*}(x) = (I_{H_V} + V^{-1}\rho h^*)^{-1}(x) \quad \Leftrightarrow \quad V(x - p) \in \partial(\rho h^*)(p) \]
\[ p \in \partial h(V(x - p)/\rho) \]
\[ Vx/\rho - (Vx - Vp)/\rho \in V\partial(h/\rho)(Vx - p)/\rho \]
\[ V(x - p)/\rho = (I_{H_V} + V\partial(h/\rho))^{-1}(Vx) \]
\[ x = p + \rho^{-1} \circ (I_{H_V} + V\partial(h/\rho))^{-1}(Vx). \]

\[ \square \]

B.2 Proof of Theorem 7

Proof. Let \( p = \text{prox}_h^V(x) \). Then, we have to solve
\[ \min_{z} \frac{1}{2} \|x - z\|^2_{V} + h(z) \]
\[ \Leftrightarrow \min_{z} \left( \frac{1}{2} \|z\|^2_{D} - \langle x, z \rangle_{D} + h(z) \right) + \langle x - z, uu^T(x - z) \rangle \]
\[ y = D^{1/2}z, q = D^{-1/2}u \quad \Leftrightarrow \min_{v} \left( \frac{1}{2} \|y\|^2 - \langle D^{1/2}x, y \rangle + h \circ D^{-1/2} \right) \] \[ + \left( D^{1/2}x - y, qq^T(D^{1/2}x - y) \right) \] \[ \text{(Lemma 22 and Lemma 1[11][iv])} \quad \Leftrightarrow \min_{v} \left( \frac{1}{2} \|v\|^2 - \langle D^{1/2}x, \cdot \rangle + h \circ D^{-1/2} \right) \] \[ \text{(*)} \quad \Leftrightarrow \min_{v \in \mathbb{R}^q} \left( \left( \frac{1}{2} \|v\|^2 - \langle D^{1/2}x, \cdot \rangle \right) + \frac{\|v\|^2}{2 \|q\|^2} + \langle D^{1/2}x, v \rangle \right) \] \[ \text{(Definition 23)} \quad \Leftrightarrow \min_{v \in \mathbb{R}^q} \left( h \circ D^{1/2} \right)(D^{1/2}x - v) + \frac{\|v\|^2}{2 \|q\|^2} + \langle D^{1/2}x, v \rangle \] \[ \text{(28)} \]

By virtue of Lemma 25, \( (h \circ D^{1/2}) \) is continuously differentiable with 1-Lipschitz gradient. Together with Lemma 22, this yields
\[ p = D^{-1/2} \circ \nabla \left( h \circ D^{1/2} \right)(D^{1/2}x - v^*) = D^{-1/2} \circ (I_{H} - \text{prox}_{h \circ D^{1/2}})(D^{1/2}x - v^*) \]
\[ = D^{-1/2} \circ \text{prox}_{h \circ D^{-1/2}} \circ D^{1/2}(x - D^{-1/2}v^*), \]

where \( v^* \) is the unique solution to the above dual problem 25. This problem amounts to minimizing a proper convex smooth continuously differentiable objective with a Lipschitz gradient over a linear set. The latter can be parametrized by a real scalar \( \alpha \) such that \( v = \alpha q = \alpha D^{-1/2}u \), and is then equivalent to solving the scalar strongly convex smooth optimization problem
\[ \min_{\alpha \in \mathbb{R}} \left( h \circ D^{1/2} \right)(D^{1/2}x - \alpha D^{-1/2}u) + \frac{\alpha^2}{2} + \alpha \langle x, u \rangle, \] \[ \text{(29)} \]
whose solution $\alpha^*$ is unique. This is equivalent to saying that $\alpha^*$ is the unique root of

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

(30)

where we used again Lemma 25 and 26. Lipschitz continuity of $p(\alpha)$ follows from non-expansiveness of the proximal mapping, and the Lipschitz constant is straightforward from the triangle and Cauchy-Schwartz inequalities.

Let’s turn now to strict increasing monotonicity of $p$. Let $\beta > \alpha$. Denote the operator $P = \text{prox}_{h \circ D^{-1/2}}$. Then,

$$p(\beta) - p(\alpha) = (\beta - \alpha) - \left\langle D^{-1/2} u, P \circ D^{1/2}(x - \beta D^{-1/2} u) - P \circ D^{1/2}(x - \alpha D^{-1/2} u) \right \rangle$$

$$= (\beta - \alpha) + (\beta - \alpha)^{-1} \left\langle -(\beta - \alpha) D^{-1/2} u, P(D^{1/2} x - \beta D^{-1/2} u) - P(D^{1/2} x - \alpha D^{-1/2} u) \right \rangle$$

$$\geq (\beta - \alpha) + (\beta - \alpha)^{-1} \left\| P(D^{1/2} x - \beta D^{-1/2} u) - P(D^{1/2} x - \alpha D^{-1/2} u) \right\|^2$$

$$> 0 ,$$

where the first inequality is a consequence of the fact that the proximal mapping is firmly non-expansive.

References


