Proximal Minimization with Bregman Distances and the Goldstein-Osher Algorithm for Constrained Optimization

Charles L. Byrne*

March 17, 2015

Abstract

We consider here the problem of minimizing a convex function $h : \mathbb{R}^K \to \mathbb{R}$ over x with $T(x) = 0$, where $T : \mathbb{R}^K \to \mathbb{R}^M$ is (possibly) nonlinear. We examine first the split-Bregman iterative algorithm proposed by Goldstein and Osher for $L1$ regularized image reconstruction, and then turn to proximal minimization algorithms (PMA) with Bregman distances, sometimes called Bregman iteration. The PMA form a subclass of the SUMMA algorithms, from which we can deduce important properties of the PMA. We show that, while there is no PMA equivalent to the Goldstein-Osher (GO) algorithm in general, equivalence in the linear case provides useful suiggestions as to how the more general GO algorithm should be formulated. We also provide new results for the GO algorithm without using Bregman iteration.

Key Words: proximal minimization; Bregman distance; split-Bregman algorithm; proximity operator; constrained optimization; $L1$ regularization.
AMS subject classification: 94A12, 65K11, 65K05.

1 Background

In [11] the authors consider the $L1$-regularization problem of minimizing the function

$$\|\Phi(u)\|_1 + H(u), \tag{1.1}$$

where both $\|\Phi(u)\|_1$ and $H(u)$ are convex. Such problems occur frequently in image science and elsewhere. They reformulate the problem as a constrained minimization problem as follows: minimize

$$E(u, d) = \|d\|_1 + H(u), \tag{1.2}$$

*Charles_Byrne@uml.edu, Department of Mathematical Sciences, University of Massachusetts Lowell, Lowell, MA 01854

subject to $\Phi(u) - d = 0$. In this way they "decouple" the $L1$ term from the "energy" term.

They note that one penalty-function approach would be to minimize

$$\|d\|_1 + H(u) + \frac{\lambda_k}{2}\|\Phi(u) - d\|_2^2, \tag{1.3}$$

to get (u^k, d^k). As $\lambda_k \to +\infty$ the second term becomes increasingly important and the limit (u^*, d^*) of such a sequence would surely have $\Phi(u^*) - d^* = 0$. As k grows larger, however, the minimization problem becomes increasingly unstable. To avoid this, they propose an alternative algorithm, which we shall call here the Goldstein-Osher, or GO, algorithm.

1.1 The Goldstein-Osher Algorithm

The GO algorithm begins with an arbitrary choice of the vector b^0. Having found u^{k-1}, d^{k-1}, and b^{k-1}, the next iterate (u^k, d^k) minimizes

$$\|d\|_1 + H(u) + \frac{\lambda}{2}\|\Phi(u) - d - b^{k-1}\|_2^2. \tag{1.4}$$

The next b^k is

$$b^k = b^{k-1} - (\Phi(u^k) - d^k). \tag{1.5}$$

It is clear how the penalty-function approach in Equation (1.3) forces $\Phi(u^*) = d^*$, but it is not obvious how changing b^{k-1} to b^k would have the same effect. Essentially what appears to happen is that, as $k \to +\infty$, an increasingly large vector is added to $\Phi(u) - d$ prior to taking the square of the Euclidean norm. This has an effect similar to taking k increasingly large in the penalty-function approach. In the linear case of the GO problem $\Phi(u) = Ru$ for some M by N matrix R. The following theorem is essentially their Theorem 2.2.

Theorem 1.1 *If, for some k, we have $\Phi(u^k) = d^k$, then (u^k, d^k) minimizes $E(u, d) = \|d\|_1 + H(u)$, subject to $\Phi(u) - d = 0$.*

Proof: Let $(\hat{u}, \hat{d})$ satisfy $\Phi(\hat{u}) - \hat{d} = 0$, and minimize $E(u, d) = \|d\|_1 + H(u)$ over all (u, d) with $\Phi(u) - d = 0$. Then

$$E(u^k, d^k) + \frac{1}{2}\|\Phi(u^k) - d^k - b^{k-1}\|_2^2 \leq E(\hat{u}, \hat{d}) + \frac{1}{2}\|\Phi(\hat{u}) - \hat{d} - b^{k-1})\|_2^2,$$

so that $E(u^k, d^k) \leq E(\hat{u}, \hat{d})$. ∎

It is certainly restrictive to assume that $\Phi(u^k) - d^k = 0$ for finite k, although this can sometimes happen [12]. Clearly, a better result would assert that the sequence $\{(u^k, d^k)\}$ converges to some (u^*, d^*) with $\Phi(u^*) - d^* = 0$. Then (u^*, d^*) would solve their original problem.

1.2 The More General Problem

For the remainder of this paper we shall employ the following more general formulation of the basic problem. Let $K \geq M$, $T : \mathbb{R}^K \to \mathbb{R}^M$ be a (possibly nonlinear) operator, and $h : \mathbb{R}^K \to \mathbb{R}$ be a convex function. The problem is to minimize $h(x)$ over x in the nonempty set $S = \{x|T(x) = 0\}$. For the GO problem we have $K = N + M$, $x = (u, d)$, $h(x) = E(u, d) = \|d\|_1 + H(u)$, and $T(x) = \Phi(u) - d$. In the linear case of the GO problem we have

$$T(x) = \Phi(u) - d = Ru - d = \begin{bmatrix} R & -I \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix}, \tag{1.6}$$

so the operator T is linear.

Proximal minimization with Bregman functions plays a central role in the discussion in [11]. These PMA form a subclass of the SUMMA algorithms, from which important properties of PMA follow. In the next section we discuss the SUMMA class and PMA.

2 Proximal Minimization and the SUMMA Class

Let C be a nonempty subset of an arbitrary set X and $f : X \to \mathbb{R}$. Consider the problem of minimizing $f(x)$ over $x \in C$. For each $k = 1, 2, ...$ we minimize

$$G_k(x) = f(x) + g_k(x), \tag{2.1}$$

to get x^k. We say that the g_k are *auxiliary functions* (AF) if $g_k(x) \geq 0$ and $g_k(x^{k-1}) = 0$; then the algorithm is an AF method. For any AF method the sequence $\{f(x^k)\}$ is nonincreasing. We say that an AF method is in the SUMMA class [5] if, in addition, the SUMMA Inequality holds; that is,

$$G_k(x) - G_k(x^k) \geq g_{k+1}(x), \tag{2.2}$$

for all $x \in C$. We have the following theorem.

Theorem 2.1 *If the auxiliary functions g_k satisfy the SUMMA Inequality, then the sequence $\{f(x^k)\}$ converges to $\beta = \inf_{x \in C} f(x)$.*

Proof: If not, then there is $z \in C$ and β^* such that

$$f(x^k) \geq \beta^* > f(z) \geq \beta.$$

From

$$g_k(z) - g_{k+1}(z) \geq g_k(z) - G_k(z) + G_k(x^k)$$
$$= f(x^k) + g_k(x^k) - f(z) \geq \beta^* - f(z) > 0,$$

it follows that $\{g_k(z)\}$ is a decreasing sequence of nonnegative terms whose successive differences remain bounded away from zero, which is a contradiction. ∎

Let $h : \mathbb{R}^K \to \mathbb{R}$ be convex, but not necessarily differentiable. For each x and y and p in the subdifferential $\partial h(y)$, a Bregman distance associated with h is

$$D_h(x, y, p) = h(x) - h(y) - \langle p, x - y \rangle. \tag{2.3}$$

Clearly, $D_h(x, y, p) \geq 0$. Let $f : \mathbb{R}^K \to \mathbb{R}$ be convex. The iterative step of the PMA is then to minimize

$$G_k(x) = f(x) + D_h(x, x^{k-1}, p^{k-1}) \tag{2.4}$$

over all x to get x^k. An easy calculation shows that

$$G_k(x) - G_k(x^k) = D_f(x, x^k, v^k) + D_h(x, x^k, p^k) \geq D_h(x, x^k, p^k) = g_{k+1}(x),$$

for all x, where $v^k \in \partial f(x^k)$ and $p^k \in \partial h(x^k)$. Therefore, every PMA is in the SUMMA class and $\{f(x^k)\}$ is nonincreasing and converges to $\beta = \inf_x f(x)$.

Suppose that the sequence $\{x^k\}$ converges to some x^*. Then we know that $f(x^*) \leq f(x)$, for all x. Let $M = \{z | f(z) \leq f(x), \text{for all}\, x\}$. Does x^* minimize $h(z)$ over all z in M? Not necessarily. The PMA iteration involves D_h, not h itself, and D_h does not determine h uniquely; adding any affine linear function to h does not alter the corresponding Bregman distance. What has been shown for specific Bregman distances, such as the Euclidean and Kullback-Leibler distances, is that x^* minimizes $D_h(z, x^0)$ over all z in M. This is where the GO algorithm has the advantage. As we shall show later, with certain assumptions, if $\{x^k\}$ converges to some x^*, then $T(x^*) = 0$ and $z = x^*$ minimizes $h(z)$ over all z with $T(z) = 0$.

3 When $T(x)$ is Affine Linear

In this section we assume that $T(x) = Ax - b$ for some M by K matrix A and constant $b \in \mathbb{R}^M$. Our goal is to minimize $h(x) + \frac{\lambda}{2}\|Ax - b\|_2^2$. Consider first the PMA approach.

Begin with some x^0 and $p^0 \in \partial h(x^0)$. For each k we obtain x^k by minimizing

$$G_k(x) = \frac{\lambda}{2}\|Ax - b\|_2^2 + D_h(x, x^{k-1}, p^{k-1}). \tag{3.1}$$

Then there is $p^k \in \partial h(x^k)$ such that

$$0 = A^T(Ax^k - b) + p^k - p^{k-1}.$$

Now we look at the GO algorithm.

Select some b^0. For each k we obtain x^k by minimizing

$$\frac{\lambda}{2}\|Ax - b - b^{k-1}\|_2^2 + h(x). \tag{3.2}$$

This is equivalent to minimizing

$$\frac{\lambda}{2}\|Ax - b\|_2^2 + \langle A^T b - A^T b^{k-1}, x - x^{k-1}\rangle, \tag{3.3}$$

which suggests that, for equivalence, we want

$$p^{k-1} = A^T b^{k-1} - A^T b. \tag{3.4}$$

It follows that we should select b^k to be

$$b^k = b^{k-1} + b - Ax^k; \tag{3.5}$$

note that there is a typo in (2.10) of [11].

The PMA iteration begins with some x^0 and $p^0 \in \partial h(x^0)$. If we select x^0 to be a global minimizer of $h(x)$, then $0 \in \partial h(x^0)$, so we can select $p^0 = 0$. This tells us that the GO iteration should begin with $b^0 = b$.

As we shall discuss later, we cannot expect there to be an equivalent PMA for the GO iteration in the nonlinear case; it appears that equivalence holds only when T is affine linear. However, we can reason by analogy and define b^k as in [11] to be

$$b^k = b^{k-1} - T(x^k). \tag{3.6}$$

Although there is much discussion of Bregman iteration in [11], it would appear that it is used only to suggest Equation (3.6) for the nonlinear case.

In the sections to follow we shall obtain some new results concerning the GO algorithm, and investigate the question of whether or not each sequence generated by the GO algorithm is also a PMA sequence.

4 More on the Goldstein-Osher Algorithm

Once again, let $h : \mathbb{R}^K \to \mathbb{R}$ be a convex function, and $T : \mathbb{R}^K \to \mathbb{R}^M$ be a (possibly) nonlinear operator. Let $\hat{x} \in S$ minimize $h(x)$ over all x in S. For each $x \in \mathbb{R}^K$ let $T'(x) \in \mathbb{R}^{M\times K}$ be the Jacobian matrix for T at x and $\nabla T(x) = T'(x)^T$. We assume that the operators T and T' are continuous, that for each x there is an M by K matrix $B(x)$ such that $B(x)\nabla T(x) = I$, the identity matrix, and that the operator $B : \mathbb{R}^K \to \mathbb{R}^{M\times K}$ is continuous. For example, suppose that $T(x)$ is affine linear, so that there are matrix A and vector b with $T(x) = Ax - b$. Assuming that AA^T is invertible, we have $\nabla T(x) = A^T$ and $B(x) = (AA^T)^{-1}A$.

Now the GO algorithm takes the following form. For arbitrary vector b^0, and having found x^{k-1} and b^{k-1}, take x^k to be the minimizer of the function

$$h(x) + \frac{1}{2}\|T(x) - b^{k-1}\|_2^2. \tag{4.1}$$

Then there is $v^k \in \partial h(x^k)$ such that

$$v^k = \nabla T(x^k)(T(x^k) - b^{k-1}), \tag{4.2}$$

and so

$$b^{k-1} = T(x^k) - B(x^k)v^k. \tag{4.3}$$

If the sequences $\{x^k\}$ and $\{v^k\}$ converge, then so does the sequence $\{b^k\}$. Note that we have not yet said how the next b^k is to be defined.

If h is differentiable, then we have

$$\nabla h(x^k) = \nabla T(x^k)(T(x^k) - b^{k-1}), \tag{4.4}$$

so that

$$b^{k-1} = T(x^k) - B(x^k)\nabla h(x^k). \tag{4.5}$$

We have the following theorem.

Theorem 4.1 *If h is continuously differentiable, and the sequence $\{x^k\}$ converges to some x^*, then the sequence $\{b^k\}$ converges to some b^*. If $T(x^*) = 0$, then x^* minimizes $h(x)$ over x in S.*

Proof: The first assertion follows from Equation (4.5) and continuity. We have

$$h(x^k) + \frac{1}{2}\|T(x^k) - b^{k-1}\|_2^2 \leq h(\hat{x}) + \frac{1}{2}\|T(\hat{x}) - b^{k-1}\|_2^2,$$

so that, by taking limits, we have

$$h(x^*) + \frac{1}{2}\|b^*\|_2^2 \leq h(\hat{x}) + \frac{1}{2}\|b^*\|_2^2.$$

∎

This theorem is similar to Theorem 2.2 of [11]; the latter does not require that h be differentiable.

In the Goldstein-Osher algorithm we have

$$b^k = b^{k-1} - T(x^k). \tag{4.6}$$

Now we can strengthen Theorem 4.1.

Theorem 4.2 *Let h be continuously differentiable. Let b^k be defined as in Equation (4.6). If the sequence $\{x^k\}$ converges to some x^*, then $T(x^*) = 0$. Consequently, x^* minimizes $h(x)$ over x in S. If h is not differentiable, but the sequence $\{v^k\}$ converges, then the same result holds.*

Proof: We know that the sequence $\{b^k\}$ converges, according to Theorem 4.1 or the convergence of the sequence $\{v^k\}$. Therefore, by taking limits in Equation (4.6), we have $T(x^*) = 0$. ∎

5 Does the GO algorithm have an equivalent PMA?

First, consider the GO algorithm. For simplicity, we consider the case of differentiable h. Let b^0 be arbitrary. Let z^1 minimize the function

$$h(x) + \frac{1}{2}\|T(x) - b^0\|_2^2, \tag{5.1}$$

and

$$b^1 = b^0 - T(z^1). \tag{5.2}$$

Then

$$\nabla h(z^1) + \nabla T(z^1)(T(z^1) - b^0) = \nabla h(z^1) - \nabla T(z^1)b^1 = 0. \tag{5.3}$$

or

$$\nabla h(z^1) = \nabla T(z^1)b^1. \tag{5.4}$$

Similarly,

$$\nabla h(z^2) + \nabla T(z^2)(T(z^2) - b^1) = \nabla h(z^2) - \nabla T(z^2)b^2 = 0, \tag{5.5}$$

so

$$\nabla h(z^2) = \nabla T(z^2)b^2. \tag{5.6}$$

Now consider the PMA.

Let x^1 minimize

$$\frac{1}{2}\|T(x)\|_2^2 + D_h(x, x^0), \tag{5.7}$$

for some x^0. Then

$$\nabla h(x^0) = \nabla h(x^1) + \nabla T(x^1)T(x^1). \tag{5.8}$$

Similarly,

$$\nabla h(x^1) = \nabla h(x^2) + \nabla T(x^2)T(x^2). \tag{5.9}$$

If $x^1 = z^1$ and $x^2 = z^2$ then, from Equations (5.5) and (5.9) we have

$$\nabla h(x^1)\nabla h(x^2) + \nabla T(x^2)T(x^2) = \nabla T(x^2)b^1, \tag{5.10}$$

or

$$\nabla h(x^1) = \nabla T(x^2)b^1. \tag{5.11}$$

But we also have

$$\nabla h(x^1) = \nabla T(x^1)b^1. \tag{5.12}$$

This suggests that there will be an equivalent PMA only when $\nabla T(x)$ is constant, or $T(x) = Ax - b$; that is, only when $T(x)$ is affine linear.

6 Summary

xxxxx

References

[1] H. Bauschke and J. Borwein, *Legendre functions and the method of random Bregman projections,* J. Convex Analysis, 4 (1997), pp. 27–67.

[2] D. Butnariu, C. Byrne, and Y. Censor, *Redundant axioms in the definition of Bregman functions,* J. Convex Analysis, 10 (2003), pp. 245–254.

[3] D. Butnariu, Y. Censor, and S. Reich (eds.), *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications,* Studies in Computational Mathematics, 8, Elsevier, Amsterdam, 2001.

[4] C. Byrne, *Bregman-Legendre multi-distance projection algorithms for convex feasibility and optimization,* in [3], pp. 87–100.

[5] C. Byrne, *Sequential unconstrained minimization algorithms for constrained optimization,* Inverse Problems, 24(1) (2008), article no. 015013.

[6] C. Byrne, *Alternating minimization as sequential unconstrained minimization: a survey,* J. Opt. Th. Appl., electronic 154(3) (2012), DOI 10.1007/s1090134-2, and hardcopy 156(3) (2013), pp. 554–566.

[7] C. Byrne, *Iterative Optimization in Inverse Problems,* CRC Press, Boca Raton, FL (2014).

[8] Y. Censor and S. A. Zenios, *Proximal minimization algorithm with D-functions,* J. Opt. Th. Appl., 73(3) (1992), pp. 451–464.

[9] Y. Censor and S. A. Zenios, *Parallel Optimization: Theory, Algorithms and Applications,* Oxford University Press, New York (1997).

[10] A. Fiacco and G. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques,* SIAM Classics in Mathematics (reissue), Philadelphia, PA (1990).

[11] T. Goldstein and S. Osher, *The split Bregman algorithm for L1 regularized problems,* SIAM J. Imaging Sci., 2(2) (2009), pp. 323–343.

[12] S. Osher, Y. Mao, B. Dong, and W. Yin, *Fast linearized Bregman iterations forcompressed sensing and sparse denoising,* UCLA CAM Report 08-37 (2008).

[13] R. Rockafellar, (1970) *Convex Analysis,* Princeton University Press, Princeton, NJ (1970).