Written by Anders Skajaa, August 2012.

1 Problem and potential function

We are considering a general convex optimization problem of the form

$$\begin{array}{rl} \min_x & f(x) \\ \text{s.t.} & x \geq 0 \end{array} \tag{1}$$

under the following assumptions.

Assumptions I

(A1) f is a convex function in $x \in \mathbb{R}^n$

(A2) $f(x) > 0$ for all x

For the purpose of solving (1), we consider the potential function

$$\phi(x) = \rho \log f(x) - \sum_j \log x_j \tag{2}$$

which has been thoroughly studied in the literature, see e.g. [1, 2, 3, 6].

It is our intention to solve the problem (1) by reducing the potential function ϕ enough that an approximate minimizer of (1) has been found. This is the strategy utilized in potential reduction algorithms — see e.g. [5] for an overview of the properties of ϕ and for discussion of other potential functions.

2 First order potential reduction algorithm

2.1 Description of algorithm

The proximal gradient algorithm (PG-algorithm) works on an objective function which is the sum of two functions $g + h$. The algorithm consists of the following recursion:

$$x^+ = \text{prox}_{th}(x - t\nabla g(x)). \tag{3}$$

Here, x denotes the current iterate, x^+ the next iterate and $t \geq 0$ a steplength. The prox-operator occuring in (3) is defined as

$$\text{prox}_\varphi(x) = \text{argmin}_u \left(\varphi(u) + \frac{1}{2}\|u - x\|^2 \right). \tag{4}$$

We will use $\|\cdot\|$ for the 2-norm and denote any other norm by a subscript.

Under certain assumptions, the proximal gradient algorithm is known to bring the objective function $g + h$ within ϵ of its optimal value in $\mathcal{O}(1/\epsilon)$ iterations, see e.g. [4].

In order to apply the PG-algorithm to our potential function, we define the following splitting:

$$g(x) = \rho \log f(x)$$
$$h(x) = -\sum_j \log x_j$$

so that $\phi(x) = g(x) + h(x)$. We can explicitely compute

$$[\operatorname{prox}_{th}(x)]_i = \frac{1}{2}\left(x_i + \sqrt{x_i^2 + 4t}\right), \quad i = 1, \ldots, n. \tag{5}$$

The assumptions required for the convergence result of the PG-algorithm are not present in our situation. For example, g is required to be convex, which is not the case for the function $g(x) = \rho \log f(x)$.

Nevertheless, under weaker assumptions, the proximal gradient algorithm applied to ϕ still reduces f below ϵ in $\mathcal{O}(1/\epsilon)$-iterations which we show in the following section.

2.2 Convergence and complexity of the PG-algorithm

Let e be the vector of all ones and let us define the level set

$$\mathcal{L}_e = \{x : \ \phi(x) \leq \phi(e)\}.$$

We are going to make the following further assumptions about the potential function ϕ and the objective function f:

Assumptions II

(A3) $\rho \geq n + \sqrt{n}$.

(A4) $\exists\, M$ so that for all $x \in \mathcal{L}_e$: $\|x\|_\infty \leq M$ and $\|\nabla f(x)\|_\infty \leq M$.

(A5) $\exists\, \lambda \geq 1$ so that for all $x \in \mathcal{L}_e$:

$$f(x + d) - f(x) \leq \nabla f(x)^T d + \lambda \|d\|^2$$

Now let us define

$$G_t(x) = \frac{1}{t}\left(x - \operatorname{prox}_{th}(x - t\nabla g(x))\right) \tag{6}$$

so that we can write the recursion (3) as

$$x^+ = x - tG_t(x)$$

From the optimality condition of (4), it further follows that

$$\nabla h(x^+) = G_t(x) - \nabla g(x). \tag{7}$$

Lemma 1. *If*

$$0 \leq t \leq \frac{f(x)}{2\rho\lambda}, \tag{8}$$

then

$$g(x^+) \leq g(x) - t\nabla g(x)^T G_t(x) + \frac{t}{2}\|G_t(x)\|^2 \tag{9}$$

Proof. We have

$$\begin{aligned}
g(x^+) - g(x) &= \rho \log\left(1 + \frac{f(x^+) - f(x)}{f(x)}\right) \\
&\overset{(A5)}{\leq} \rho \log\left(1 + \frac{-t\nabla f(x)^T G_t(x) + \lambda t^2 \|G_t(x)\|^2}{f(x)}\right) \\
&\leq -\rho t \frac{\nabla f(x)^T G_t(x)}{f(x)} + \rho t^2 \lambda \frac{\|G_t(x)\|^2}{f(x)} \\
&\overset{(8)}{\leq} -t\nabla g(x)^T G_t(x) + \frac{t}{2}\|G_t(x)\|^2
\end{aligned}$$

□

Lemma 2. *If* (8) *holds, then*

$$\phi(x^+) = \phi(x) - \frac{t}{2}\|G_t(x)\|^2 \tag{10}$$

Proof. Using convexity of h and (7), we have

$$\begin{aligned}
h(x^+) &\leq h(x) - t\nabla h(x^+)^T G_t(x) \\
&\overset{(7)}{=} h(x) - t\left(G_t(x) - \nabla g(x)\right)^T G_t(x) \\
&= h(x) + t\nabla g(x)^T G_t(x) - t\|G_t(x)\|^2
\end{aligned} \tag{11}$$

Therefore,

$$\begin{aligned}
\phi(x^+) &= g(x^+) + h(x^+) \\
&\overset{(9)}{\leq} g(x) - t\nabla g(x)^T G_t(x) + \frac{t}{2}\|G_t(x)\|^2 + h(x^+) \\
&\overset{(11)}{\leq} g(x) + h(x) - \frac{t}{2}\|G_t(x)\|^2 \\
&= \phi(x) - \frac{t}{2}\|G_t(x)\|^2.
\end{aligned}$$

□

So Lemma 2 shows that the potential function will be reduced by at least $(t/2)\|G_t(x)\|^2$ in each iteration. This implies that the algorithm will generate a sequence of iterates on which ϕ decreases monotonically. Therefore, all iterates will stay in the initial level set $\mathcal{L}_e$.

Now it remains to establish a lower bound on $\|G_t(x)\|^2$. Before proving such a statement, we need the following lemma.

Lemma 3. *Let X be the diagonal matrix with diagonal elements $x_1, \ldots, x_n$. Then*

$$\|X\nabla\phi(x)\|^2 \geq 1$$

Proof. Since

$$X\phi(x) = \frac{\rho}{f(x)} X\nabla f(x) - e,$$

it is clear that if any one element of $\nabla f(x)$ is less than or equal to zero, then the statement holds. Thus we assume that every element of $\nabla f(x)$ is positive. Since f is convex, we have

$$\begin{aligned} f(x^*) - f(x) \leq \nabla f(x)(x^* - x) \qquad &\Rightarrow \\ \nabla f(x)^T x - f(x) \geq \nabla f(x) x^* \geq 0 \qquad &\Rightarrow \\ f(x) \leq \nabla f(x)^T x \end{aligned} \tag{12}$$

Now we compute

$$\begin{aligned} \|X\nabla\phi(x)\|^2 &= \frac{\rho^2}{f(x)^2}\|X\nabla f(x)\|^2 - 2\frac{\rho}{f(x)} x^T \nabla f(x) + n \\ &\geq \frac{\rho^2}{nf(x)^2}\|X\nabla f(x)\|_1^2 - 2\frac{\rho}{f(x)} x^T \nabla f(x) + n \\ &\geq \frac{\rho^2}{n}\left(\frac{x^T\nabla f(x)}{f(x)}\right)^2 - 2\rho\left(\frac{x^T\nabla f(x)}{f(x)}\right) + n \\ &= \frac{\rho^2}{n} z^2 - 2\rho z + n \end{aligned} \tag{13}$$

where we defined $z = x^T\nabla f(x)/f(x)$. From (12), we have $z \geq 1$. The quadratic in (13) has a single root in n/ρ, which is < 1 because of (A3). So the minimizer of (13) is achieved for $z = 1$ and therefore we can continue as

$$\begin{aligned} \|X\nabla\phi(x)\|^2 &= \frac{\rho^2}{n} z^2 - 2\rho z + n \\ &\geq \frac{\rho^2}{n} - 2\rho + n \\ &= \frac{(\rho - n)^2}{n} \geq 1 \end{aligned} \tag{14}$$

where the last inequality follows from (A1). $\square$

We are ready to prove a lower bound on the norm of $G_t(x)$:

Proposition 1. *If*

$$t \leq \frac{f(x)}{\rho}, \tag{15}$$

then

$$\|G_t(x)\|^2 \geq \frac{1}{4M^2}$$

Proof. We consider the i'th component of $G_t(x)$, which we denote by $[G_t(x)]_i$:

$$[G_t(x)]_i \overset{(5)}{=} \frac{1}{t}\left(\frac{x_i + t\,[\nabla g(x)]_i}{2} - \frac{1}{2}\sqrt{(x_i - t\,[\nabla g(x)]_i)^2 + 4t}\right)$$

or equivalently

$$\sqrt{(x_i - t\,[\nabla g(x)]_i)^2 + 4t} = x_i + t\,[\nabla g(x)]_i - 2t\,[G_t(x)]_i$$

which after squaring both sides gives

$$4t = 4tx_i\,[\nabla g(x)]_i + 4t^2\,[G_t(x)]_i^2 - 4t\,[G_t(x)]_i\,(x_i + t\,[\nabla g(x)]_i).$$

Dividing by $4t$ gives

$$\begin{aligned} x_i\,[\nabla g(x)]_i - 1 &= [G_t(x)]_i\,(x_i + t\,[\nabla g(x)]_i) - t\,[G_t(x)]_i^2 \\ &= [G_t(x)]_i\,\left(x_i^+ + t\,[\nabla g(x)]_i\right) \end{aligned}$$

Now squaring both sides and summing over i, we get

$$\sum_i (x_i\,[\nabla g(x)]_i - 1)^2 = \sum_i [G_t(x)]_i^2\,(x_i^+ + t\,[\nabla g(x)]_i)^2 \tag{16}$$

Finally, combining Lemma 3 and (16), we get

$$\begin{aligned} 1 &\overset{(14)}{\leq} \|X\nabla\phi(x)\|^2 \\ &= \sum_i (x_i\,[\nabla g(x)]_i - 1)^2 \\ &\overset{(16)}{=} \sum_i [G_t(x)]_i^2\,(x_i^+ + t\,[\nabla g(x)]_i)^2 \\ &\overset{(15)}{\leq} \sum_i [G_t(x)]_i^2\,(x_i^+ + [\nabla f(x)]_i)^2 \\ &\overset{(A4)}{\leq} 4M^2\|G_t(x)\|^2 \end{aligned} \tag{17}$$

which proves the statement. □

We can finally make a statement concerning the complexity of the algorithm applied to our potential function:

Theorem 1. *If we choose* $t = \frac{\epsilon}{2\rho\lambda}$*, the* PG*-algorithm produces an* x *with* $f(x) \leq \epsilon$ *in* $\mathcal{O}\Big(\frac{\lambda n^2 M^2}{\epsilon}\Big)$ *iterations.*

Proof. With $t = \epsilon/(2\rho\lambda)$, the premises of Lemma 1 and Proposition 1 are satisfied as long as $f(x) \geq \epsilon$. Therefore, each iteration will reduce ϕ according to

$$\begin{aligned}\phi(x^+) &\leq \phi(x) - \frac{t}{2}\|G_t(x)\|^2 \\ &\leq \phi(x) - \frac{\epsilon}{4\rho\lambda}\frac{1}{4M^2} \\ &= \phi(x) - \frac{\epsilon}{16\rho\lambda M^2}.\end{aligned}$$

Then, the k'th iterate x^k will satisfy

$$\begin{aligned}\phi(x^k) &\leq \phi(e) - k\frac{\epsilon}{16\rho\lambda M^2} \\ &= \rho \log f(e) - k\frac{\epsilon}{16\rho\lambda M^2}\end{aligned}$$

so that $\phi(x^k) \leq \rho \log \epsilon$ when

$$\begin{aligned}k &\geq \frac{16\rho^2\lambda M^2}{\epsilon}\log\frac{f(e)}{\epsilon} \\ &= \mathcal{O}\left(\frac{\lambda\rho^2 M^2}{\epsilon}\right)\end{aligned}$$

With $\rho = \mathcal{O}(n)$, this expression reduces to $\mathcal{O}\Big(\frac{n^2\lambda M^2}{\epsilon}\Big)$. □

So the previous theorem shows that the PG-algorithm converges in $\mathcal{O}(1/\epsilon)$ steps.

2.3 Convergence and complexity of the APG-algorithm

The *accelerated* PG-algorithm (APG-algorithm) is, at the k'th iteration defined by

$$y = x^{(k-1)} + \frac{k-2}{k+1}(x^{(k-1)} - x^{(k-2)}) \tag{18}$$

$$x^{(k)} = \mathrm{prox}_{t_k h}(y - t_k \nabla g(y)). \tag{19}$$

where t_k is the step length at iteration k.

A. Skajaa comment 1: This algorithm is known to converge in $\mathcal{O}(1/\sqrt{\epsilon})$ steps for certain classes of functions. As before, our objective function does *not* fall within this class. Nevertheless, our numerical experiments suggest that indeed the APG-algorithm converges in $\mathcal{O}(1/\sqrt{\epsilon})$ steps even when applied to our potential function.

Therefore, it would be extremely nice if we could establish that the APG-algorithm converges in $\mathcal{O}(1/\sqrt{\epsilon})$ steps when applied to our potential function. So far, we have *not* been able to prove this. I know that Zizhou has made attempts at this, but I'm not sure exactly to what extent. This would be the only remaining piece of work on the theoretical side.

3 Numerical experiments

A. Skajaa comment 2: I have carried out a number of numerical experiments. These include e.g. 1. large linear programs, 2. smaller linear programs from NETLIB and 3. image restoration problems.

This work/paper would be strengthened considerably if we can find a great "real world" application to which we can apply this algorithm. This is the view of Prof. Ye, and I also share this view. I.e., we should find a (very) large linear program from a real application. So large that a traditional interior-point methods can not handle the matrices. Possibilities in this direction n

References

[1] Gonzaga, C.C.: *Polynomial affine algorithms for linear programming.* Math. Program. 49, 7–21 (1990).

[2] Karmarkar, N.: *A new polynomial-time algorithm for linear programming.* Combinatorica 4, 373–395 (1984).

[3] Todd, M.J., Ye, Y.: *A centered projective algorithm for linear programming.* Math. Oper. Res. 15, 508–529 (1990).

[4] Tseng, P.: *On accelerated proximal gradient methods for convex-concave optimization.* Submitted to SIAM J. Optim. (2008)

[5] Wright, S.J.: *Primal-Dual Interior-Point Methods.* SIAM (1987)

[6] Ye, Y.: *An* $\mathcal{O}(n^3L)$ *potential reduction algorithm for linear programming.* Math. Program. 50, 239–258 (1991).