hw9 Optimization

Guillaume Jarry

November 2022

1 Problem 3.11

We will write x_I^k and u_I^k the sequences of PD3O and x_{II}^k, u_{II}^k the sequences of PAPC. First let us notice that:

$$x_I^{k+1} = x_I^k - \alpha A^T u_I^k - \alpha \nabla h(x_I^k)$$
$$u_I^{k+1} = Prox_{\beta g^*}\left(u_I^k + \beta A(x_I^{k+1} - \alpha A^T u_I^k - \alpha \nabla h(x^{k+1}))\right)$$

And:

$$u_{II}^{k+1} = Prox_{\beta g^*}\left(u_{II}^k + \beta A(x_{II}^k - \alpha A^T u_{II}^k - \alpha \nabla h(x^k)\right)$$

$$x_{II}^{k+1} = x_{II}^k - \alpha A^T u_{II}^{k+1} - \alpha \nabla h(x_{II}^k)$$

And once that simplification is done, then PD3O and PAPC are equivalent if we can manage to prove that PAPC is PD3O postponed by one iteration:

$$\forall k, \quad x_I^{k+1} = x_{II}^k$$

Let us try to prove it via induction. For the initilialization, let us assume $x_I^1 = x_{II}^0$. Then:

$$u_{II}^1 = Prox_{\beta g^*}\left(u_{II}^0 + \beta A(x_{II}^0 - \alpha A^T u_{II}^k - \alpha \nabla h(x^k)\right) = u_I^1$$

The first step is okay and allows us to get:

$$x_{II}^1 = \underset{=x_I^1}{x_{II}^0} - \alpha A^T \underset{u_I^1}{u_{II}^1} - \alpha \nabla h(\underset{=x_I^1}{x_{II}^k}) = x_I^2$$

The heredity of the proof stems from the two equality that we have written above (that's why I will not write it again).

Hence we have shown that the two methods aren't exactly equivalent but rather that PD3O is one iteration in advance of PAPC.

2 Problem 3.14

In the method of linearized multipliers, We will introduce the change in variables $v^0 = u^0 + \alpha(Ax^0 - b)$ and $v^{k+1} = v^k + \alpha(A(2x^{k+1} - x^k) - b)$. Then, by induction, we will prove that :

$$\forall k \geq 1, u^k = v^k - \alpha(Ax^k - b) \quad (1)$$

For $k = 1$, we have $u^0 = v^0 - \alpha(Ax^0 - b)$ and thus:

$$u^1 = v^0 - \alpha(Ax^0 - b) + \alpha(Ax^1 - b) = v^1 - \alpha(Ax^1 - b)$$

Which proves the initialiazation. For the heredity, let $k \in N$ satisfies the above property. Then;

$$u^{k+1} = u^k + \alpha(Ax^{k+1} - b) = v^k - \alpha(Ax^k - b) + \alpha(Ax^{k+1} - b) = v^{k+1} - \alpha(Ax^{k+1} - b)$$

Which finishes the proof our our induction and then we can prove that the algorithm below is equvalent to the linearized method of multiplayers.

$$x^{k+1} = Prox_{\beta f}(x^k + \beta A^T(v^k))$$
$$v^{k+1} = v^k + \alpha(A(2x^{k+1} - x^k) - b)$$

This is obvious when we inject the equation (1) inside:

$$x^{k+1} = Prox_{\beta f}(x^k + \beta A^T(u^k + \alpha(Ax^k - b)))$$
$$u^{k+1} = u^k + \alpha(Ax^{k+1} - b)$$

Then, let us remind ourselves that if $g = \delta\{0\}$, then $g^* = 0$ and thus $Prox_{\beta g^*}(v) = \underset{x}{argmin}\{\frac{1}{2}\|x - v\|^2\} = v$. Thus, the PDHG algorithm with $g = \delta\{0\}$ is exactly:

$$x^{k+1} = Prox_{\beta f}(x^k + \beta A^T(u^k))$$
$$u^{k+1} = u^k + \alpha(A(2x^{k+1} - x^k) - b)$$

And here we recognize the linearized method of multipliers !

3 Problem 3.15

First, let us establish what the variable metric proximal point method will be with $A = \partial L$. First:

$$\partial L \begin{bmatrix} x \\ z \\ u \end{bmatrix} = \begin{bmatrix} \nabla f(x) + A^T u \\ \nabla g(z) - u \\ Ax - z \end{bmatrix}$$

And thus:

$$(M+A)\begin{bmatrix}x\\z\\u\end{bmatrix} = \begin{bmatrix}\alpha x + \nabla f(x)\\ \nabla g(z) + \alpha z\\ \alpha u\end{bmatrix}$$

And thus:

$$(M+A)\begin{bmatrix}x^{k+1}\\z^{k+1}\\u^{k+1}\end{bmatrix} = M\begin{bmatrix}x\\z\\u\end{bmatrix} \iff \begin{bmatrix}\alpha x^{k+1} + \nabla f(x^{k+1})\\ \nabla g(z^{k+1}) + \alpha z^{k+1}\\ \alpha u^{k+1}\end{bmatrix} = \begin{bmatrix}\alpha x^k - A^T u^k\\ \alpha z^k + u^k\\ -Ax^k + z^k + \alpha u^k\end{bmatrix}$$

And thus we obtain the algorithm:

$$x^{k+1} = Prox_{\frac{1}{\alpha}f}(x^k + \frac{1}{\alpha}A^T u^k)$$

$$z^{k+1} = Prox_{\frac{1}{\alpha}g}(z^k + \frac{1}{\alpha}u^k)$$

$$u^{k+1} = u^k - \frac{1}{\alpha}(Ax^k - z^k)$$

And here, we get stuck if we don't recognize the fact that the α of the Tchen Teboulle algorithm and the α of the above method are actually inverse of one another. Once this is settled, we recognize that u^{k+1} of the above method and p^{k+1} of Tchen Teboulle actually have the exact same expression. Then considering the order of p^k and u^k in each algorithm, we realize that both algorithm are the same, except that Tchen Teboulle is postponed by one iteration, ie:

$$\forall k, \quad p^{k+1}_{Tchen-Teboulle} = u^k$$

4 Problem 3.16