ADMM Fused Lasso Signal Approximator

Vladimir Jojic

October 17, 2012

1 The FLSA problem and ADMM blueprint

We will first write down the optimization problem

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1 + \mu\|\mathbf{D}\mathbf{x}\|_1 \tag{1}$$

where $\mathbf{x},\mathbf{y} \in \mathbf{R}^n$,$\mathbf{D} \in \mathbf{R}^{d\times n}$.

Objective description We briefly give interpretation for the three different parts of the objective. The first term is a squared ℓ_2 loss that can also be seen as a linear regression problem $\frac{1}{2}\|\mathbf{y}-\mathbf{I}\mathbf{x}\|_2^2$ with feature matrix being an identity matrix of size $n \times n$. The second term promotes sparsity in $\mathbf{x}$ by virtue of being an ℓ_1 norm. The third term is again an ℓ_1 norm but of a linearly transformed vector $\mathbf{x}$. One common choice for matrix $\mathbf{D}$ is a matrix with each row containing only two non-zero entries, +1 and -1. Hence, multiplication by each row computes differences between different entries in $\mathbf{x}$. Sparsity in these values encourages entries in $\mathbf{x}$ corresponding to minuend and subtrahend to be the same. Taken together these parts of the objective encourage discovery of a sparse, block constant representation of vector y. Choice of matrix $\mathbf{D}$ determines what the blocks will look like.

Dual decomposition In order to avoid issues with multiple non-smooth penalties operating on the same set of variables, we will reformulate the optimization problem by introducing auxiliary variables

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{y}-\mathbf{z}_1\|_2^2 + \lambda\|\mathbf{z}_2\|_1 + \mu\|\mathbf{z}_3\|_1 \\ \text{subject to} \quad & \mathbf{z}_1 = \mathbf{x} \\ & \mathbf{z}_2 = \mathbf{x} \\ & \mathbf{z}_3 = \mathbf{D}\mathbf{x}. \end{aligned} \tag{2}$$

This problem is equivalent to (1).

Augmented Lagrangian For problem (2) we obtain following augmented Lagrangian

$$
\begin{aligned}
\mathrm{AL}(\mathbf{x},\mathbf{z}_1,\mathbf{z}_2,\mathbf{z}_3,\mathbf{u}_1,\mathbf{u}_2,\mathbf{u}_3) \quad = \quad & \frac{1}{2}\left\|\mathbf{y}-\mathbf{z}_1\right\|_2^2+\lambda\left\|\mathbf{z}_2\right\|_1+\mu\left\|\mathbf{z}_3\right\|_1+ \\
& \langle\mathbf{z}_1-\mathbf{x},\mathbf{u}_1\rangle+\langle\mathbf{z}_2-\mathbf{x},\mathbf{u}_2\rangle+\langle\mathbf{z}_3-\mathbf{D}\mathbf{x},\mathbf{u}_3\rangle+ \\
& \frac{\rho}{2}\left\|\mathbf{z}_1-\mathbf{x}\right\|_2^2+\frac{\rho}{2}\left\|\mathbf{z}_2-\mathbf{x}\right\|_2^2+\frac{\rho}{2}\left\|\mathbf{z}_3-\mathbf{D}\mathbf{x}\right\|_2^2.
\end{aligned}
$$

The ADMM algorithm iterates following updates

$$
\begin{aligned}
\mathbf{x}^{k} &= \operatorname*{argmin}_{\mathbf{x}} \mathrm{AL}(\mathbf{x},\mathbf{z}_1^{k-1},\mathbf{z}_2^{k-1},\mathbf{z}_3^{k-1},\mathbf{u}_1^{k-1},\mathbf{u}_2^{k-1},\mathbf{u}_3^{k-1}) \\
\mathbf{z}_1^{k} &= \operatorname*{argmin}_{\mathbf{x}} \mathrm{AL}(\mathbf{x}^{k},\mathbf{z}_1,\mathbf{z}_2^{k-1},\mathbf{z}_3^{k-1},\mathbf{u}_1^{k-1},\mathbf{u}_2^{k-1},\mathbf{u}_3^{k-1}) \\
\mathbf{z}_2^{k} &= \operatorname*{argmin}_{\mathbf{x}} \mathrm{AL}(\mathbf{x}^{k},\mathbf{z}_1^{k},\mathbf{z}_2,\mathbf{z}_3^{k-1},\mathbf{u}_1^{k-1},\mathbf{u}_2^{k-1},\mathbf{u}_3^{k-1}) \\
\mathbf{z}_3^{k} &= \operatorname*{argmin}_{\mathbf{x}} \mathrm{AL}(\mathbf{x}^{k},\mathbf{z}_1^{k},\mathbf{z}_2^{k},\mathbf{z}_3,\mathbf{u}_1^{k-1},\mathbf{u}_2^{k-1},\mathbf{u}_3^{k-1}) \\
\mathbf{u}_1^{k} &= \mathbf{u}_1^{k-1}+\rho(\mathbf{z}_1^{k}-\mathbf{x}^{k}) \\
\mathbf{u}_2^{k} &= \mathbf{u}_2^{k-1}+\rho(\mathbf{z}_2^{k}-\mathbf{x}^{k}) \\
\mathbf{u}_3^{k} &= \mathbf{u}_3^{k-1}+\rho(\mathbf{z}_3^{k}-\mathbf{D}\mathbf{x}^{k}),
\end{aligned}
$$

where $k-1$ and k denote previous and current iteration, respectively.

2 Three observation relevant to simplifying ADMM subproblems

Completing the square Suppose we have to solve a problem

$$
\operatorname*{argmin}_{\mathbf{x}} \underbrace{\frac{\rho}{2}\left\|\mathbf{x}-\mathbf{y}\right\|_2^2+\langle\mathbf{x},\mathbf{u}\rangle+f(\mathbf{x})}_{\mathrm{A}}.
$$

We claim that the optimal $\mathbf{x}$ for the above problem is equal to

$$
\operatorname*{argmin}_{\mathbf{x}} \underbrace{\frac{\rho}{2}\left\|\mathbf{x}-\mathbf{y}+\frac{1}{\rho}\mathbf{u}\right\|_2^2+f(\mathbf{x})}_{\mathrm{B}}
$$

because difference between the objectives A and B is $-\frac{1}{2\rho}\left\|\mathbf{u}\right\|^2+\langle\mathbf{u},\mathbf{y}\rangle$ a constant with respect to $\mathbf{x}$.

Stacking regression problems Suppose we have to solve a problem

$$
\operatorname*{argmin}_{\mathbf{x}} \frac{\rho}{2}\left\|\mathbf{B}\mathbf{x}-\mathbf{y}\right\|_2^2+\left\|\mathbf{C}\mathbf{x}-\mathbf{z}\right\|_2^2+f(\mathbf{x}).
$$

We claim that this problem is equivalent to solving

$$\underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2} \left\| \begin{bmatrix} \sqrt{\rho}\mathbf{B} \\ \mathbf{C} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \sqrt{\rho}\mathbf{y} \\ \mathbf{z} \end{bmatrix} \right\|_2^2 + f(\mathbf{x}).$$

Further if $f(\mathbf{x}) = \text{const}$ then this problem can be solved in a single line of Matlab code

```
[sqrt(rho)*B;C]\[sqrt(rho)*y;z]
```

Flipping signs under norms If you are looking at a normed expression, the whole expression under the norm can be negated without affecting the function value

$$\|x - y\| = \|y - x\|$$

this is a direct consequence of the definition of norms (positive homogeneity property).

3 Updates for FLSA ADMM algorithm

Deriving update for x Eliminating terms that are constants with respect to $\mathbf{x}$ we obtain

$$\begin{aligned}\mathbf{x}^{k} = \underset{\mathbf{x}}{\operatorname{argmin}} \quad & \left\langle \mathbf{z}_1^{k-1} - \mathbf{x}, \mathbf{u}_1^{k-1} \right\rangle + \left\langle \mathbf{z}_2^{k-1} - \mathbf{x}, \mathbf{u}_2^{k-1} \right\rangle + \left\langle \mathbf{z}_3^{k-1} - \mathbf{D}\mathbf{x}, \mathbf{u}_3^{k-1} \right\rangle + \\ & \frac{\rho}{2} \left\| \mathbf{z}_1^{k-1} - \mathbf{x} \right\|_2^2 + \frac{\rho}{2} \left\| \mathbf{z}_2^{k-1} - \mathbf{x} \right\|_2^2 + \frac{\rho}{2} \left\| \mathbf{z}_3^{k-1} - \mathbf{D}\mathbf{x} \right\|_2^2 .\end{aligned}$$

Now we complete the squares separately

$$\begin{aligned}\mathbf{x}^{k} &= \underset{\mathbf{x}}{\operatorname{argmin}} \frac{\rho}{2} \left\| \mathbf{z}_1^{k-1} - \mathbf{x} + \frac{1}{\rho}\mathbf{u}_1^{k-1} \right\|_2^2 + \frac{\rho}{2} \left\| \mathbf{z}_2^{k-1} - \mathbf{x} + \frac{1}{\rho}\mathbf{u}_2^{k-1} \right\|_2^2 + \frac{\rho}{2} \left\| \mathbf{z}_3^{k-1} - \mathbf{D}\mathbf{x} + \frac{1}{\rho}\mathbf{u}_3^{k-1} \right\|_2^2 \\ &= \underset{\mathbf{x}}{\operatorname{argmin}} \frac{\rho}{2} \left\| \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{D} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{z}_1^{k-1} + \frac{1}{\rho}\mathbf{u}_1^{k-1} \\ \mathbf{z}_2^{k-1} + \frac{1}{\rho}\mathbf{u}_2^{k-1} \\ \mathbf{z}_3^{k-1} + \frac{1}{\rho}\mathbf{u}_3^{k-1} \end{bmatrix} \right\|_2^2\end{aligned}$$

So we can use following Matlab code

```
x = [eye(n);eye(n);D]\[z1+1/rho*u1;z2+1/rho*u2;z2+1/rho*u3]
```

Deriving update for $\mathbf{z}_1$

$$\begin{aligned}
\mathbf{z}_1^{k} &= \underset{\mathbf{z}_1}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{y}-\mathbf{z}_1\|_2^2 + \left\langle \mathbf{z}_1 - \mathbf{x}^{k}, \mathbf{u}_1^{k-1}\right\rangle + \frac{\rho}{2}\left\|\mathbf{z}_1 - \mathbf{x}^{k}\right\|_2^2 \\
&= \underset{\mathbf{z}_1}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{z}_1-\mathbf{y}\|_2^2 + \frac{\rho}{2}\left\|\mathbf{z}_1 - \mathbf{x}^{k} + \frac{1}{\rho}\mathbf{u}_1^{k-1}\right\|_2^2 \\
&= \underset{\mathbf{z}_1}{\operatorname{argmin}} \frac{1}{2}\left\|\begin{bmatrix}\mathbf{I}\\ \sqrt{\rho}\mathbf{I}\end{bmatrix}\mathbf{z}_1 - \begin{bmatrix}\mathbf{y}\\ \sqrt{\rho}\mathbf{x}^{k} - \frac{1}{\sqrt{\rho}}\mathbf{u}_1^{k-1}\end{bmatrix}\right\|_2^2
\end{aligned}$$

So we can use following Matlab code

```
z1 = [eye(n);sqrt(rho)*eye(n)]\[y;sqrt(rho)*x - 1/sqrt(rho)*u1]
```

Deriving update for $\mathbf{z}_2$

$$\begin{aligned}
\mathbf{z}_2^{k} &= \underset{\mathbf{z}_2}{\operatorname{argmin}}\, \lambda\|\mathbf{z}_2\|_1 + \left\langle \mathbf{z}_2 - \mathbf{x}^{k}, \mathbf{u}_2^{k-1}\right\rangle + \frac{\rho}{2}\left\|\mathbf{z}_2 - \mathbf{x}^{k}\right\|_2^2 \\
&= \underset{\mathbf{z}_2}{\operatorname{argmin}} \frac{\lambda}{\rho}\|\mathbf{z}_2\|_1 + \frac{1}{2}\left\|\mathbf{z}_2 - \mathbf{x}^{k} + \frac{1}{\rho}\mathbf{u}_2^{k-1}\right\|_2^2
\end{aligned}$$

and we note that this is simply Lasso regression and further that it separates across entries $\mathbf{z}_2$ meaning that it can be solved in closed-form in a single update

$$\mathbf{z}_2 = S\left(\mathbf{x} - \frac{1}{\rho}\mathbf{u}_2, \frac{\lambda}{\rho}\right)$$

So we can use following Matlab code

```
z2 = shrinkThreshold(x - 1/rho*u2,lambda/rho)
```

assuming we have function

```
function x = shrinkThreshold(x,lambda)
x = sign(x).*max(abs(x) - lambda,0);
```

Deriving update for $\mathbf{z}_3$

$$\begin{aligned}
\mathbf{z}_3^{k} &= \underset{\mathbf{z}^3}{\operatorname{argmin}}\, \mu\|\mathbf{z}_3\|_1 + \left\langle \mathbf{z}_3 - \mathbf{D}\mathbf{x}^{k}, \mathbf{u}_3^{k-1}\right\rangle + \frac{\rho}{2}\left\|\mathbf{z}_3 - \mathbf{D}\mathbf{x}^{k}\right\|_2^2 \\
&= \underset{\mathbf{z}^3}{\operatorname{argmin}} \frac{\mu}{\rho}\|\mathbf{z}_3\|_1 + \frac{1}{2}\left\|\mathbf{z}_3 - \mathbf{D}\mathbf{x}^{k} + \frac{1}{\rho}\mathbf{u}_3^{k-1}\right\|_2^2
\end{aligned}$$

and hence

$$\mathbf{z}_3 = S\left(\mathbf{D}\mathbf{x} - \frac{1}{\rho}\mathbf{u}_3, \frac{\mu}{\rho}\right)$$

So we can use following Matlab code

```
z3 = shrinkThreshold(Dx - 1/rho*u3,mu/rho)
```

Updating dual variables $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ These updates are straightforward

$$
\begin{aligned}
\mathbf{u}_1^{k} &= \mathbf{u}_1^{k-1} + \rho(\mathbf{z}_1^{k} - \mathbf{x}^{k}) \\
\mathbf{u}_2^{k} &= \mathbf{u}_2^{k-1} + \rho(\mathbf{z}_2^{k} - \mathbf{x}^{k}) \\
\mathbf{u}_3^{k} &= \mathbf{u}_3^{k-1} + \rho(\mathbf{z}_3^{k} - \mathbf{D}\mathbf{x}^{k})
\end{aligned}
$$