

Alternating direction method of multipliers (ADMM)

Again, this material is mostly pulled from [BPC⁺10]. I have uploaded the paper to T-square so you can download it if you are interested.

ADMM extends the method of multipliers in such way that we get back some of the decomposability (i.e. ability to parallelize) of standard dual ascent algorithms. It also gives us a flexible framework for incorporating many types of convex constraints, though we will again focus on linear equality constraints to start.

ADMM **splits** the optimization variable into two parts, \mathbf{x} and \mathbf{z} , and solves programs of the form

$$\underset{\mathbf{x}, \mathbf{z}}{\text{minimize}} \quad f_0(\mathbf{x}) + h(\mathbf{z}) \quad \text{subject to} \quad \mathbf{Ax} + \mathbf{Bz} = \mathbf{c}.$$

The basic idea is to rotate through 3 steps:

1. Minimize the (augmented) Lagrangian over \mathbf{x} with \mathbf{z} and the Lagrange multipliers $\boldsymbol{\nu}$ fixed.
2. Minimize the (augmented) Lagrangian over \mathbf{z} with \mathbf{x} and $\boldsymbol{\nu}$ fixed.
3. Update the Lagrange multipliers using gradient ascent as before.

If the splitting is done in a careful manner, it can happen that each of the subproblems above can be easily computed and we can handle general convex constraints (more on this later).

To make the three steps above more explicit: the augmented La-

grangian is

$$L_\rho(\mathbf{x}, \mathbf{z}, \boldsymbol{\nu}) = f_0(\mathbf{x}) + h(\mathbf{z}) + \boldsymbol{\nu}^\top (\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|_2^2,$$

and the general ADMM iteration is

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \arg \min_{\mathbf{x}} L_\rho(\mathbf{x}, \mathbf{z}^{(k)}, \boldsymbol{\nu}^{(k)}) \\ \mathbf{z}^{(k+1)} &= \arg \min_{\mathbf{z}} L_\rho(\mathbf{x}^{(k+1)}, \mathbf{z}, \boldsymbol{\nu}^{(k)}) \\ \boldsymbol{\nu}^{(k+1)} &= \boldsymbol{\nu}^{(k)} + \rho(\mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{B}\mathbf{z}^{(k+1)} - \mathbf{c}).\end{aligned}$$

The only real different between ADMM and MoM is the we are splitting the primal minimization into two parts instead of optimizing over (\mathbf{x}, \mathbf{z}) jointly.

Scaled form.

We can write the ADMM iterations in a more convenient form by substituting

$$\boldsymbol{\mu} = \frac{1}{\rho} \boldsymbol{\nu}.$$

Then by “completing the square”, you can check at home that

$$\boldsymbol{\nu}^\top (\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|_2^2 = \frac{\rho}{2} \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c} + \boldsymbol{\mu}\|_2^2 - \frac{\rho}{2} \|\boldsymbol{\mu}\|_2^2,$$

and so we can write:

ADMM:

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \arg \min_{\mathbf{x}} \left(f_0(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^{(k)} - \mathbf{c} + \boldsymbol{\mu}^{(k)}\|_2^2 \right) \\ \mathbf{z}^{(k+1)} &= \arg \min_{\mathbf{z}} \left(h(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{B}\mathbf{z} - \mathbf{c} + \boldsymbol{\mu}^{(k)}\|_2^2 \right) \\ \boldsymbol{\mu}^{(k+1)} &= \boldsymbol{\mu}^{(k)} + \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{B}\mathbf{z}^{(k+1)} - \mathbf{c}\end{aligned}$$

Example: the LASSO

The following ℓ_1 regularized least-squares problem:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \tau \|\mathbf{x}\|_1$$

is called the LASSO; it is prevalent all across machine learning, model selection in statistics, and compressed sensing in signal processing. The $\tau > 0$ above is a user-defined “smoothing parameter”.

Taking

$$f_0(\mathbf{x}) = \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \quad \text{and} \quad h(\mathbf{z}) = \tau \|\mathbf{z}\|_1,$$

we can rewrite this in ADMM form as

$$\underset{\mathbf{x}, \mathbf{z}}{\text{minimize}} \quad f_0(\mathbf{x}) + h(\mathbf{z}) \quad \text{subject to} \quad \mathbf{x} - \mathbf{z} = \mathbf{0}.$$

The \mathbf{x} update is

$$\mathbf{x}^{(k+1)} = \arg \min_{\mathbf{x}} \left(\frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\rho}{2} \|\mathbf{x} - \mathbf{z}^{(k)} + \boldsymbol{\mu}^{(k)}\|_2^2 \right).$$

With both $\mathbf{z}^{(k)}$ and $\boldsymbol{\mu}^{(k)}$ fixed, this is equivalent to the least-squares problem:

$$\min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{A} \\ \sqrt{\rho} \mathbf{I} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} \\ \sqrt{\rho}(\mathbf{z}^{(k)} - \boldsymbol{\mu}^{(k)}) \end{bmatrix} \right\|_2^2.$$

This problem has a closed-form solution:

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \left(\mathbf{A}^T \mathbf{A} + \rho \mathbf{I} \right)^{-1} \begin{bmatrix} \mathbf{A}^T & \sqrt{\rho} \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ \sqrt{\rho}(\mathbf{z}^{(k)} - \boldsymbol{\mu}^{(k)}) \end{bmatrix} \\ &= \left(\mathbf{A}^T \mathbf{A} + \rho \mathbf{I} \right)^{-1} \left(\mathbf{A}^T \mathbf{b} + \rho(\mathbf{z}^{(k)} - \boldsymbol{\mu}^{(k)}) \right) \end{aligned}$$

The \mathbf{z} update update problem is:

$$\underset{\mathbf{z}}{\text{minimize}} \tau \|\mathbf{z}\|_1 + \frac{\rho}{2} \|\mathbf{z} - \mathbf{x}^{(k+1)} - \boldsymbol{\mu}^{(k)}\|_2^2.$$

Using the results from the Technical Details section below, we have a closed form for this as well:

$$\mathbf{z}^{(k+1)} = T_{\tau/\rho}(\mathbf{x}^{(k+1)} + \boldsymbol{\mu}^{(k)}),$$

where $T_\lambda(\cdot)$ is the term-by-term soft-thresholding operator,

$$(T_\lambda(\mathbf{v})) [n] = \begin{cases} v[n] - \lambda, & v[n] > \lambda, \\ 0, & |v[n]| \leq \lambda, \\ v[n] + \lambda, & v[n] < -\lambda. \end{cases}$$

To summarize:

ADMM iterations for the LASSO

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \left(\mathbf{A}^T \mathbf{A} + \rho \mathbf{I} \right)^{-1} \left(\mathbf{A}^T \mathbf{b} + \rho(\mathbf{z}^{(k)} - \boldsymbol{\mu}^{(k)}) \right), \\ \mathbf{z}^{(k+1)} &= T_{\tau/\rho}(\mathbf{x}^{(k+1)} + \boldsymbol{\mu}^{(k)}), \\ \boldsymbol{\mu}^{(k+1)} &= \boldsymbol{\mu}^{(k)} + \mathbf{x}^{(k+1)} - \mathbf{z}^{(k+1)}. \end{aligned}$$

Convergence properties

We will state one convergence result. If the following two conditions hold:

1. f_0 and h are closed, proper, and convex (i.e. their epigraphs are nonempty closed convex sets),
2. strong duality holds,

then

- $\mathbf{A}\mathbf{x}^{(k)} + \mathbf{B}\mathbf{z}^{(k)} - \mathbf{c} \rightarrow \mathbf{0}$ as $k \rightarrow \infty$. That is, the primal iterates are asymptotically feasible.
- $f_0(\mathbf{x}^{(k)}) + h(\mathbf{z}^{(k)}) \rightarrow p^*$ as $k \rightarrow \infty$. That is, the value of the objective function approaches the optimal value asymptotically.
- $\boldsymbol{\nu}^{(k)} \rightarrow \boldsymbol{\nu}^*$ as $k \rightarrow \infty$, where $\boldsymbol{\nu}^*$ is a dual optimal point.

Under additional assumptions, we can also have convergence to a primal optimal point, $(\mathbf{x}^{(k)}, \mathbf{z}^{(k)}) \rightarrow (\mathbf{x}^*, \mathbf{z}^*)$ as $k \rightarrow \infty$.

See [BPC⁺10, Section 3.2] for further discussion and references.

Convex constraints

We can write the general program

$$\underset{\mathbf{x} \in \mathcal{C}}{\text{minimize}} \quad f_0(\mathbf{x}),$$

where \mathcal{C} is a closed convex set, in ADMM form as

$$\underset{\mathbf{x} \in \mathbb{R}^N}{\text{minimize}} \quad f_0(\mathbf{x}) + h(\mathbf{z}) \quad \text{subject to} \quad \mathbf{x} - \mathbf{z} = \mathbf{0},$$

where $h(\mathbf{z})$ is the indicator function for \mathcal{C} :

$$h(\mathbf{z}) = \begin{cases} 0, & \mathbf{z} \in \mathcal{C}, \\ \infty, & \mathbf{z} \notin \mathcal{C}. \end{cases}$$

Note that in this case, the \mathbf{z} update is a closest-point-to-a-convex-set problem. For fixed $\mathbf{v} \in \mathbb{R}^N$,

$$\begin{aligned} \arg \min_{\mathbf{z}} h(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{z} - \mathbf{v}\|_2^2 &= \arg \min_{\mathbf{z} \in \mathcal{C}} \|\mathbf{z} - \mathbf{v}\|_2 \\ &= P_{\mathcal{C}}(\mathbf{v}) \quad (\text{closest point in } \mathcal{C} \text{ to } \mathbf{v}). \end{aligned}$$

ADMM iteration for general convex constraints:

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \arg \min_{\mathbf{x}} \left(f_0(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{x} - \mathbf{z}^{(k)} + \boldsymbol{\mu}^{(k)}\|_2^2 \right), \\ \mathbf{z}^{(k+1)} &= P_{\mathcal{C}} \left(\mathbf{x}^{(k+1)} + \boldsymbol{\mu}^{(k)} \right), \\ \boldsymbol{\mu}^{(k+1)} &= \boldsymbol{\mu}^{(k)} + \mathbf{x}^{(k+1)} - \mathbf{z}^{(k+1)}. \end{aligned}$$

Of course, this algorithm is most attractive when we have a fast method for computing $P_{\mathcal{C}}(\cdot)$.

Example: Basis Pursuit

A good proxy for finding the sparsest solution to an underdetermined system of equations $\mathbf{Ax} = \mathbf{b}$ is to solve

$$\underset{\mathbf{x}}{\text{minimize}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{Ax} = \mathbf{b}.$$

To put this in ADMM form, we are solving

$$\underset{\mathbf{x}, \mathbf{z}}{\text{minimize}} f_0(\mathbf{x}) + h(\mathbf{z}) \quad \text{subject to} \quad \mathbf{x} - \mathbf{z} = \mathbf{0},$$

with

$$f_0(\mathbf{x}) = \|\mathbf{x}\|_1, \quad \text{and} \quad h(\mathbf{z}) = \begin{cases} 0, & \mathbf{Az} = \mathbf{b}, \\ \infty, & \text{otherwise.} \end{cases}$$

The projection onto $\mathcal{C} = \{\mathbf{x} : \mathbf{Ax} = \mathbf{b}\}$ can be given in closed form using the pseudo-inverse \mathbf{A}^+ of \mathbf{A} as

$$\begin{aligned} P_{\mathcal{C}}(\mathbf{v}) &= \mathbf{A}^+(\mathbf{b} - \mathbf{Av}) + \mathbf{v} \\ &= (\mathbf{I} - \mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{A})\mathbf{v} + \mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{b}, \end{aligned}$$

where the last equality comes from $\mathbf{A}^+ = \mathbf{A}^T(\mathbf{AA}^T)^{-1}$ when \mathbf{A} has full row rank.

The updates in this case are

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \arg \min_{\mathbf{x}} \left(\|\mathbf{x}\|_1 + \frac{\rho}{2} \|\mathbf{x} - \mathbf{z}^{(k)} + \boldsymbol{\mu}^{(k)}\|_2^2 \right) \\ &= T_{1/\rho}(\mathbf{z}^{(k)} - \boldsymbol{\mu}^{(k)}) \\ \mathbf{z}^{(k+1)} &= (\mathbf{I} - \mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{A})(\mathbf{x}^{(k+1)} + \boldsymbol{\mu}^{(k)}) + \mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{b} \\ \boldsymbol{\mu}^{(k+1)} &= \boldsymbol{\mu}^{(k)} + \mathbf{x}^{(k+1)} - \mathbf{z}^{(k+1)}. \end{aligned}$$

Example: Linear programming

Consider the general linear program

$$\underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{c}^T \mathbf{x} \quad \text{subject to} \quad \mathbf{A} \mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0},$$

where \mathbf{A} is an $M \times N$ matrix with full row rank¹. We can put this in ADMM form by first eliminating the equality constraints, then introducing the indication function for the non-negativity constraint.

Let \mathbf{Q} be an $N \times (N - M)$ matrix whose columns span $\text{Null}(\mathbf{A})$, and let \mathbf{x}_0 be any point such that $\mathbf{A} \mathbf{x}_0 = \mathbf{b}$. Then we can re-write the LP as

$$\underset{\mathbf{w}}{\text{minimize}} \quad \mathbf{c}^T (\mathbf{x}_0 + \mathbf{Q} \mathbf{w}) \quad \text{subject to} \quad \mathbf{x}_0 + \mathbf{Q} \mathbf{w} \geq \mathbf{0},$$

which we can write in ADMM form as

$$\underset{\mathbf{w}}{\text{minimize}} \quad \mathbf{c}^T \mathbf{x}_0 + \mathbf{c}^T \mathbf{Q} \mathbf{w} + h(\mathbf{z}) \quad \text{subject to} \quad \mathbf{Q} \mathbf{w} - \mathbf{z} = -\mathbf{x}_0,$$

where

$$h(\mathbf{z}) = \begin{cases} 0, & \mathbf{z} \geq \mathbf{0}, \\ \infty, & \text{otherwise.} \end{cases}$$

(We can drop the $\mathbf{c}^T \mathbf{x}_0$ from the objective since it does not depend on either of the optimization variables.)

Notice that when \mathbf{Q} has full column rank, the program

$$\underset{\mathbf{w}}{\text{minimize}} \quad \mathbf{v}^T \mathbf{w} + \frac{1}{2} \|\mathbf{Q} \mathbf{w} - \mathbf{y}\|_2^2,$$

¹The full row rank assumption is not at all essential; I am just making it to keep things streamlined.

has the closed-form solution

$$\mathbf{w}^* = (\mathbf{Q}^T \mathbf{Q})^{-1} (\mathbf{Q}^T \mathbf{y} - \mathbf{v}).$$

Also, the projection onto the non-negative orthant $\mathcal{C} = \{\mathbf{x} : \mathbf{x} \geq \mathbf{0}\}$ is

$$P_{\mathcal{C}}(\mathbf{v}) = (\mathbf{v})_+, \quad \text{or} \quad (P_{\mathcal{C}}(\mathbf{v}))_n = \begin{cases} v[n], & v[n] \geq 0, \\ 0, & v[n] < 0. \end{cases}$$

For the general linear program, then, the ADMM iterations are

$$\begin{aligned} \mathbf{w}^{(k+1)} &= \arg \min_{\mathbf{w}} \left(\frac{1}{\rho} \mathbf{c}^T \mathbf{Q} \mathbf{w} + \frac{1}{2} \|\mathbf{Q} \mathbf{w} - \mathbf{z}^{(k)} + \mathbf{x}_0 + \boldsymbol{\mu}^{(k)}\|_2^2 \right) \\ &= (\mathbf{Q}^T \mathbf{Q})^{-1} \left[\mathbf{Q}^T (\mathbf{z}^{(k)} - \mathbf{x}_0 - \boldsymbol{\mu}^{(k)}) - \frac{1}{\rho} \mathbf{Q}^T \mathbf{c} \right], \\ \mathbf{z}^{(k+1)} &= P_{\mathcal{C}}(\mathbf{Q} \mathbf{w}^{(k+1)} + \mathbf{x}_0 + \boldsymbol{\mu}^{(k)}) \\ &= \left(\mathbf{Q} \mathbf{w}^{(k+1)} + \mathbf{x}_0 + \boldsymbol{\mu}^{(k)} \right)_+ \\ \boldsymbol{\mu}^{(k+1)} &= \boldsymbol{\mu}^{(k)} + \mathbf{Q} \mathbf{w}^{(k+1)} - \mathbf{z}^{(k+1)} + \mathbf{x}_0. \end{aligned}$$

Notice that especially when the columns of \mathbf{Q} are orthogonal, $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}$, all of these steps are very simple.

Distributed Recovery/Regression/Classification using ADMM

By being very crafty with how we do the splitting, we can use ADMM to solve certain kinds of optimization programs in a distributed manner.

We consider (this material comes from [BPC⁺10, Sec. 8]) the general problem of “fitting” a vector $\mathbf{x} \in \mathbb{R}^N$ to an observed vector $\mathbf{b} \in \mathbb{R}^M$ through an $M \times N$ matrix \mathbf{A} . We will encourage \mathbf{x} to have certain structure using a regularizer. This type of problem is ubiquitous in signal processing and machine learning — the math stays the same, only the words change from area to area.

At a high level, we are interested in solving

$$\underset{\mathbf{x}}{\text{minimize}} \text{Loss}(\mathbf{A}\mathbf{x} - \mathbf{b}) + \text{Regularizer}(\mathbf{x})$$

where the $M \times N$ matrix \mathbf{A} and the M -vector \mathbf{b} are given. Notice that

$$\text{Loss}(\cdot) : \mathbb{R}^M \rightarrow \mathbb{R}, \quad \text{and} \quad \text{Regularizer}(\cdot) : \mathbb{R}^N \rightarrow \mathbb{R}.$$

We will assume that one or both of these functions are separable, at least at the block level. This means we can write

$$\text{Loss}(\mathbf{A}\mathbf{x} - \mathbf{b}) = \sum_{i=1}^B \ell_i(\mathbf{A}_i\mathbf{x} - \mathbf{b}_i),$$

where \mathbf{A}_i are $M_i \times N$ matrices formed by partitioning the rows of \mathbf{A} , and $\mathbf{b}_i \in \mathbb{R}^{M_i}$ is the corresponding part of \mathbf{b} . For separable regularizers, we can write

$$\text{Regularizer}(\mathbf{x}) = \sum_{i=1}^C r_i(\mathbf{x}_i),$$

where the $\mathbf{x}_i \in \mathbb{R}^{N_i}$ partition the vector \mathbf{x} . These two types of separability will allow us to divide up the optimization in two different ways.

Example: Inverse Problems and Regression

Two popular methods for solving linear inverse problems and/or calculating regressors are solving

$$\underset{\mathbf{x}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \tau \|\mathbf{x}\|_2^2,$$

(*Tikhonov regularization* or *ridge regression*), and

$$\underset{\mathbf{x}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \tau \|\mathbf{x}\|_1,$$

(*basis pursuit denoising* or *the LASSO*).

These both clearly fit the separability criteria, as

$$\begin{aligned} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 &= \sum_{m=1}^M (\langle \mathbf{x}, \mathbf{a}_m \rangle - b[m])^2, \\ \|\mathbf{x}\|_2^2 &= \sum_{n=1}^N (x[n])^2 \\ \|\mathbf{x}\|_1 &= \sum_{n=1}^N |x[n]|. \end{aligned}$$

where \mathbf{a}_m^T is the m th row of \mathbf{A} .

Example: Support Vector Machines

Previously, we saw how if we are given a set of M training examples (\mathbf{x}_m, y_m) , where $\mathbf{x}_m \in \mathbb{R}^N$ and $y_m \in \{-1, 1\}$, we can find a maximum margin linear classifier by solving

$$\min_{\mathbf{w}, z} \|\mathbf{w}\|_2^2 \quad \text{subject to} \quad y_m(z - \langle \mathbf{x}_m, \mathbf{w} \rangle) + 1 \leq 0, \quad m = 1, \dots, M.$$

With the classifier trained (optimal solution \mathbf{w}^*, z^* computed), we can assign a label y' to a new point \mathbf{x}' using

$$y' = \text{sign}(\langle \mathbf{x}', \mathbf{w}^* \rangle + z^*).$$

Instead of enforcing the constraints above strictly, we can allow some errors by penalizing mis-classifications on the training data appropriately. One reasonable way to do this is make the loss zero if $y_m(z - \langle \mathbf{x}_m, \mathbf{w} \rangle) + 1 \leq 0$, and then have it increase linearly as this quantity exceeds zero. That is, we solve

$$\min_{\mathbf{w}, z} \sum_{m=1}^M \ell(y_m(z - \langle \mathbf{x}_m, \mathbf{w} \rangle) + 1) + \|\mathbf{w}\|_2^2,$$

where $\ell(\cdot)$ is the **hinge loss**

$$\ell(u) = (u)_+ = \begin{cases} 0, & u \leq 0, \\ u, & u > 0. \end{cases}$$

So “soft margin” SVM fits our model as what is inside the $\ell(\cdot)$ can be written as an affine function of the optimization variables:

$$y_m(z - \langle \mathbf{x}_m, \mathbf{w} \rangle) + 1 = \begin{bmatrix} -y_m \mathbf{x}_m & y_m \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ z \end{bmatrix} + 1.$$

Splitting across examples

This framework is useful when we have “many measurements of a small vector” or “large volumes of low-dimensional data”.

We partition the rows of \mathbf{A} and entries of \mathbf{b} :

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_B \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_B \end{bmatrix}.$$

If the loss function is separable over this partition, our optimization problem is

$$\underset{\mathbf{x}}{\text{minimize}} \quad \sum_{i=1}^B \ell_i(\mathbf{A}_i \mathbf{x} - \mathbf{b}_i) + r(\mathbf{x}),$$

where $r(\cdot)$ is the regularizer. We start by splitting the optimization variables in the loss function and those in the regularizer, arriving at the equivalent program

$$\underset{\mathbf{x}}{\text{minimize}} \quad \sum_{i=1}^B \ell_i(\mathbf{A}_i \mathbf{x} - \mathbf{b}_i) + r(\mathbf{z}) \quad \text{subject to} \quad \mathbf{x} - \mathbf{z} = \mathbf{0}.$$

This does not make the Lagrangian for the primal update separable, as the \mathbf{A}_i are still tying together all of the entries in \mathbf{x} . The trick is to introduce B different $\mathbf{x}_i \in \mathbb{R}^N$, one for each block, and then use the constraints to make them all agree. This is done with

$$\underset{\mathbf{x}_1, \dots, \mathbf{x}_B}{\text{minimize}} \quad \sum_{i=1}^B \ell_i(\mathbf{A}_i \mathbf{x}_i - \mathbf{b}_i) + r(\mathbf{z}) \quad \text{subject to} \quad \mathbf{x}_i - \mathbf{z} = \mathbf{0}, \quad i = 1, \dots, B.$$

The augmented Lagrangian for this last problem is

$$L_\rho(\mathbf{x}_1, \dots, \mathbf{x}_B, \mathbf{z}, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_B) = \sum_{i=1}^B \ell_i(\mathbf{A}_i \mathbf{x}_i - \mathbf{b}_i) + \frac{\rho}{2} \sum_{i=1}^B \|\mathbf{x}_i - \mathbf{z} + \boldsymbol{\mu}_i\|_2^2 + r(\mathbf{z}),$$

where $\boldsymbol{\mu}_i$ are the (rescaled) Lagrange multipliers for the constraint $\mathbf{x}_i - \mathbf{z} = \mathbf{0}$.

As the Lagrangian is separable over the B blocks, each of the primal updates for the \mathbf{x}_i can be performed independently. This makes the ADMM iteration

$$\mathbf{x}_i^{(k+1)} = \arg \min_{\mathbf{x}_i} \left(\ell_i(\mathbf{A}_i \mathbf{x}_i - \mathbf{b}_i) + \frac{\rho}{2} \|\mathbf{x}_i - \mathbf{z}^{(k)} + \boldsymbol{\mu}_i^{(k)}\|_2^2 \right)$$

$$i = 1, \dots, B$$

$$\mathbf{z}^{(k+1)} = \arg \min_{\mathbf{z}} \left(r(\mathbf{z}) + \frac{\rho}{2} \sum_{i=1}^B \|\mathbf{z} - \mathbf{x}_i^{(k+1)} - \boldsymbol{\mu}_i^{(k)}\|_2^2 \right)$$

$$\boldsymbol{\mu}_i^{(k+1)} = \boldsymbol{\mu}_i^{(k)} + \mathbf{x}_i^{(k+1)} - \mathbf{z}^{(k+1)}$$

$$i = 1, \dots, B.$$

The \mathbf{z} update can be written in terms of the average of the $\mathbf{x}_i^{(k+1)}$. To see this, first note that

$$\begin{aligned} \sum_{i=1}^B \|\mathbf{z} - \mathbf{v}_i\|_2^2 &= B\|\mathbf{z}\|_2^2 - 2 \left\langle \mathbf{z}, \sum_{i=1}^B \mathbf{v}_i \right\rangle + \sum_{i=1}^B \|\mathbf{v}_i\|_2^2 \\ &= B\|\mathbf{z}\|_2^2 - 2B \langle \mathbf{z}, \bar{\mathbf{v}} \rangle + B\|\bar{\mathbf{v}}\|_2^2 + \left(-B\|\bar{\mathbf{v}}\|_2^2 + \sum_{i=1}^B \|\mathbf{v}_i\|_2^2 \right) \\ &= B\|\mathbf{z} - \bar{\mathbf{v}}\|_2^2 + \left(-B\|\bar{\mathbf{v}}\|_2^2 + \sum_{i=1}^B \|\mathbf{v}_i\|_2^2 \right). \end{aligned}$$

where $\bar{\mathbf{v}} = \frac{1}{B} \sum_{i=1}^B \mathbf{v}_i$. Thus

$$\begin{aligned} & \arg \min_{\mathbf{z}} \left(r(\mathbf{z}) + \frac{\rho}{2} \sum_{i=1}^B \|\mathbf{z} - \mathbf{x}_i^{(k+1)} - \boldsymbol{\mu}_i^{(k)}\|_2^2 \right) \\ &= \arg \min_{\mathbf{z}} \left(r(\mathbf{z}) + \frac{B\rho}{2} \|\mathbf{z} - \bar{\mathbf{x}}^{(k+1)} - \bar{\boldsymbol{\mu}}^{(k)}\|_2^2 \right) \end{aligned}$$

Distributed ADMM (dividing rows of \mathbf{A})

$$\mathbf{x}_i^{(k+1)} = \arg \min_{\mathbf{x}_i} \left(\ell_i(\mathbf{A}_i \mathbf{x}_i - \mathbf{b}_i) + \frac{\rho}{2} \|\mathbf{x}_i - \mathbf{z}^{(k)} + \boldsymbol{\mu}_i^{(k)}\|_2^2 \right)$$

$$i = 1, \dots, B$$

$$\mathbf{z}^{(k+1)} = \arg \min_{\mathbf{z}} \left(r(\mathbf{z}) + \frac{B\rho}{2} \|\mathbf{z} - \bar{\mathbf{x}}^{(k+1)} - \bar{\boldsymbol{\mu}}^{(k)}\|_2^2 \right)$$

$$\boldsymbol{\mu}_i^{(k+1)} = \boldsymbol{\mu}_i^{(k)} + \mathbf{x}_i^{(k+1)} - \mathbf{z}^{(k+1)}$$

$$i = 1, \dots, B.$$

where

$$\bar{\mathbf{x}}^{(k+1)} = \frac{1}{B} \sum_{i=1}^B \mathbf{x}_i^{(k+1)}, \quad \bar{\boldsymbol{\mu}}^{(k)} = \frac{1}{B} \sum_{i=1}^B \boldsymbol{\mu}_i^{(k)}.$$

The high-level architecture is that B separate units solve independent optimization programs for the B \mathbf{x}_i updates. These are collected and averaged, and a single optimization program is solved to get the \mathbf{x} update. The new \mathbf{z} is then communicated back to each of the B units. The Lagrange multiplier update can easily be com-

puted both centrally and at the B units, so these do not have to be communicated.

Example: the LASSO

With $\ell_i(\mathbf{A}_i \mathbf{x}_i - \mathbf{b}_i) = \|\mathbf{A}_i \mathbf{x}_i - \mathbf{b}_i\|_2^2$ and $r(\mathbf{x}) = \tau \|\mathbf{x}\|_1$, the ADMM iteration becomes

$$\mathbf{x}_i^{(k+1)} = \arg \min_{\mathbf{x}_i} \left(\|\mathbf{A}_i \mathbf{x}_i - \mathbf{b}_i\|_2^2 + \frac{\rho}{2} \|\mathbf{x}_i - \mathbf{z}^{(k)} + \boldsymbol{\mu}_i^{(k)}\|_2^2 \right)$$

$$i = 1, \dots, B$$

$$\mathbf{z}^{(k+1)} = T_{\tau/(B\rho)} \left(\bar{\mathbf{x}}^{(k+1)} + \bar{\boldsymbol{\mu}}^{(k)} \right)$$

$$\boldsymbol{\mu}_i^{(k+1)} = \boldsymbol{\mu}_i^{(k)} + \mathbf{x}_i^{(k+1)} - \mathbf{z}^{(k+1)}$$

$$i = 1, \dots, B.$$

The \mathbf{x}_i updates are all small unconstrained least-squares problems whose solutions can be computed independently; the \mathbf{z} update is a simple soft thresholding, and the $\boldsymbol{\mu}_i$ updates are computed simply by adding vectors.

Example: SVM

For the SVM, we collect the weights and the offset into a single optimization vector

$$\mathbf{v} = \begin{bmatrix} \mathbf{w} \\ z \end{bmatrix} \in \mathbb{R}^{N+1}$$

and use

$$\mathbf{A}_i = \begin{bmatrix} -y_1 \mathbf{x}_1 & y_1 \\ \vdots & \vdots \\ -y_{N_1} \mathbf{x}_{N_1} & y_{N_1} \end{bmatrix}$$

Note that the regularization does not include the last term in \mathbf{v} :

$$r(\mathbf{v}) = \sum_{n=1}^N |v[n]|^2.$$

This makes the ADMM iteration

$$\mathbf{v}_i^{(k+1)} = \arg \min_{\mathbf{v}_i} \left(\mathbf{1}^T (\mathbf{A}_i \mathbf{v}_i + \mathbf{1})_+ + \frac{\rho}{2} \|\mathbf{v}_i - \mathbf{z}^{(k)} + \boldsymbol{\mu}_i^{(k)}\|_2^2 \right)$$

$$\mathbf{z}_{1:N}^{(k+1)} = \frac{\rho}{1 + N\rho} \left(\bar{\mathbf{v}}_{1:N}^{(k+1)} + \bar{\boldsymbol{\mu}}_{1:N}^{(k)} \right)$$

$$z^{(k+1)}[N + 1] = \bar{\mathbf{v}}^{(k+1)}[N + 1] + \bar{\boldsymbol{\mu}}^{(k)}[N + 1]$$

$$\boldsymbol{\mu}_i^{(k+1)} = \boldsymbol{\mu}_i^{(k)} + \mathbf{v}_i^{(k+1)} - \mathbf{z}^{(k+1)}.$$

where $\mathbf{x}_{1:N}$ is the first N entries of the vector \mathbf{x} , and $x[N + 1]$ is the last entry.

Splitting across features

Similarly, we can divide up the *columns* of \mathbf{A} . This is described in [\[BPC⁺10, Section 8.3\]](#).

Technical Details: Decoupled ℓ_1 minimization

Consider the optimization problem

$$\underset{\mathbf{z}}{\text{minimize}} \quad \lambda \|\mathbf{z}\|_1 + \frac{1}{2} \|\mathbf{z} - \mathbf{v}\|_2^2, \quad (1)$$

where \mathbf{v} is a fixed vector. This program is separable:

$$\begin{aligned} & \underset{\mathbf{z}}{\text{minimize}} \quad \lambda \sum_{n=1}^N |z[n]| + \frac{1}{2} \sum_{n=1}^N (z[n] - v[n])^2 \\ & = \underset{\mathbf{z}}{\text{minimize}} \quad \sum_{n=1}^N \left(\lambda |z[n]| + \frac{1}{2} (z[n] - v[n])^2 \right) \end{aligned}$$

and so we can solve each 1 dimensional problem individually.

For fixed $v \in \mathbb{R}$, we can compute the minimizer of

$$\underset{z \in \mathbb{R}}{\text{minimize}} \quad \lambda |z| + \frac{1}{2} (z - v)^2$$

explicitly. This function is convex, and is differentiable everywhere except at $z = 0$. Away from zero, the derivative is

$$\frac{df}{dz} = \begin{cases} \lambda + z - v, & z > 0 \\ -\lambda + z - v, & z < 0. \end{cases}$$

For the optimal value z^* to be positive, we need $\lambda + z^* - v = 0$; this can only hold for $z^* > 0$ if $v > \lambda$. Similarly, for z^* to be negative, we need $-\lambda + z^* - v = 0$; this can only hold for $z^* < 0$ if $v < -\lambda$. If neither of these conditions hold, we must have $z^* = 0$. Thus

$$z^* = \begin{cases} v - \lambda, & v > \lambda \\ 0, & |v| \leq \lambda \\ v + \lambda, & v < -\lambda. \end{cases}$$

We use $T_\lambda(\cdot)$ to denote the nonlinear mapping above, and so

$$z^* = T_\lambda(v).$$

T_λ is called a *soft thresholding* or *shrinkage* operator.

The solution to (1) just applies the shrinkage operator term by term:

$$\mathbf{z}^* = T_\lambda(\mathbf{v}), \quad \text{or} \quad z^*[n] = T_\lambda(v[n]).$$

References

- [BPC⁺10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

1 Alternating Direction Method of Multipliers

1.1 Algorithm

Consider the following optimization problem

$$\begin{aligned} \min & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{aligned} \quad (1)$$

where $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, and $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$.

Let the primal optimal value be denoted by p^* .

For $\rho > 0$, the augmented Lagrangian is given by

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2$$

The ADMM consists of iterations

$$x^{k+1} := \arg \min_x L_\rho(x, z^k, y^k) \quad (2)$$

$$z^{k+1} := \arg \min_z L_\rho(x^{k+1}, z, y^k) \quad (3)$$

$$y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \quad (4)$$

1.1.1 Scaled Form

The ADMM is usually written in another form. First consider the residual $r = Ax + Bz - c$. We have

$$\begin{aligned} y^T r + \frac{\rho}{2}\|r\|_2^2 &= \frac{\rho}{2}\|r + \frac{1}{\rho}y\|_2^2 - \frac{1}{2\rho}\|y\|_2^2 \\ &= \frac{\rho}{2}\|r + u\|_2^2 + \frac{\rho}{2}\|u\|_2^2 \end{aligned}$$

where $u := \frac{1}{\rho}y$ is called the scaled dual variable (y is the dual variable). Then ADMM can be expressed as

$$x^{k+1} := \arg \min_x \left(f(x) + \frac{\rho}{2}\|Ax + Bz^k - c + u^k\|_2^2 \right) \quad (5)$$

$$z^{k+1} := \arg \min_z \left(g(z) + \frac{\rho}{2}\|Ax^{k+1} + Bz - c + u^k\|_2^2 \right) \quad (6)$$

$$u^{k+1} := u^k + Ax^{k+1} + Bz^{k+1} - c \quad (7)$$

Note that if $r^k = Ax^k + Bz^k - c$, then

$$u^k = u^0 + \sum_{j=1}^k r^j$$

1.2 Convergence

In order to prove the convergence of ADMM, we need some assumptions.

Assumption 1 The functions $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ and $g : \mathbb{R}^m \rightarrow \overline{\mathbb{R}}$ are closed, proper and convex.

Assumption 2 The Lagrangian $L(= L_0)$ has a saddle point. That is there exist (x^*, z^*, y^*) such that

$$L(x^*, z^*, y) \leq L(x^*, z^*, y^*) \leq L(x, z, y^*)$$

for all x, z, y .

Assumption 3 The Slater's condition is satisfied. That is, there exists $(x, z) \in \text{ri}(\text{dom } f \times \text{dom } g)$ such that $Ax + Bz = c$.

Assumption 4 For each iteration, the subproblems (2), (3) are solvable.

The KKT condition of (1) is

$$-A^T y \in \partial f(x), \quad -B^T y \in \partial g(z), \quad Ax + Bz = c$$

Under the Slater's condition, (x^*, z^*) is a solution to (1) if and only if there exists y^* such that $(x^*, z^*), y^*$ satisfy the KKT condition.

Under some conditions, the subproblems can be proved to be solvable. Please look at [1] for reference.

Under the above assumptions, the ADMM iterates satisfy the following:

Residual convergence : $r^k \rightarrow 0$

Objective convergence : $f(x^k) + g(z^k) \rightarrow p^*$

Dual variable convergence : $y^k \rightarrow y^*$

Proof. Let (x^*, z^*, y^*) be a saddle point of L . Define

$$p^k := f(x^k) + g(z^k)$$

$$V^k := \frac{1}{\rho} \|y^k - y^*\|_2^2 + \rho \|B(z^k - z^*)\|_2^2$$

We first need to prove 3 inequalities

$$p^* - p^{k+1} \leq (y^*)^T r^{k+1} \quad (8)$$

$$p^{k+1} - p^* \leq -(y^{k+1})^T r^{k+1} - \rho(B(z^{k+1} - z^k))^T (-r^{k+1} + B(z^{k+1} - z^*)) \quad (9)$$

$$V^{k+1} \leq V^k - \rho \|r^{k+1}\|_2^2 - \rho \|B(z^{k+1} - z^k)\|_2^2 \quad (10)$$

Proof of (8) Since (x^*, z^*, y^*) is a saddle point of L , then

$$L(x^*, z^*, y^*) \leq L(x^{k+1}, z^{k+1}, y^*)$$

Since $Ax^* + Bz^* = c$, $L(x^*, z^*, y^*) = p^*$.

Also, since $p^{k+1} = f(x^{k+1}) + g(z^{k+1})$, $L(x^{k+1}, z^{k+1}, y^*) = p^{k+1} + (y^*)^T r^{k+1}$.
Therefore, $p^* - p^{k+1} \leq (y^*)^T r^{k+1}$.

Proof of (9) Since $x^{k+1} = \arg \min_x L_\rho(x, z^k, y^k)$, by the optimal condition we have

$$0 \in \partial f(x^{k+1}) + A^T y^k + \rho A^T (Ax^{k+1} + Bz^k - c).$$

Note that $y^{k+1} = y^k + \rho r^{k+1}$, so $y^k = y^{k+1} - \rho r^{k+1}$. Then

$$0 \in \partial f(x^{k+1}) + A^T (y^{k+1} - \rho B(z^{k+1} - z^k)).$$

This implies that x^{k+1} minimizes

$$f(x) + (y^{k+1} - \rho B(z^{k+1} - z^k))^T Ax.$$

Similarly, since $z^{k+1} = \arg \min_z L_\rho(x^{k+1}, z, y^k)$, we have

$$\begin{aligned} 0 &\in \partial g(z^{k+1}) + B^T y^k + \rho B^T (Ax^{k+1} + Bz^{k+1} - c) \\ &= \partial g(z^{k+1}) + B^T y^k + \rho B^T (r^{k+1}) \\ &= \partial g(z^{k+1}) + B^T (y^k + \rho r^{k+1}) \\ &= \partial g(z^{k+1}) + B^T y^{k+1} \end{aligned}$$

This is equivalent to z^{k+1} minimizes

$$g(z) + (y^{k+1})^T Bz.$$

Hence,

$$\begin{aligned} &f(x^{k+1}) + (y^{k+1} - \rho B(z^{k+1} - z^k))^T Ax^{k+1} \\ &\leq f(x^*) + (y^{k+1} - \rho B(z^{k+1} - z^k))^T Ax^* \end{aligned}$$

Similarly,

$$g(z^{k+1}) + (y^{k+1})^T B z^{k+1} \leq g(z^*) + (y^{k+1})^T B z^*$$

Adding the above two inequalities and using the fact that $Ax^* + Bz^* = c$, we have

$$\begin{aligned} p^{k+1} - p^* &\leq (y^{k+1} - \rho B(z^{k+1} - z^k))^T A(x^* - x^{k+1}) + (y^{k+1})^T B(z^* - z^{k+1}) \\ &= (y^{k+1})^T (Ax^* + Bz^* - Ax^{k+1} - Bz^{k+1}) - \rho(B(z^{k+1} - z^k))^T A(x^* - x^{k+1}) \\ &= -(y^{k+1})^T r^{k+1} - \rho(B(z^{k+1} - z^k))^T A(x^* - x^{k+1}) \end{aligned}$$

Note that $A(x^* - x^{k+1}) = -r^{k+1} + B(z^{k+1} - z^*)$, then

$$p^{k+1} - p^* \leq -(y^{k+1})^T r^{k+1} - \rho(B(z^{k+1} - z^k))^T (-r^{k+1} + B(z^{k+1} - z^*))$$

Proof of (10) Adding (8),(9) (and multiply by 2) gives

$$2(y^{k+1} - y^*)^T r^{k+1} - 2\rho(B(z^{k+1} - z^k))^T r^{k+1} + 2\rho(B(z^{k+1} - z^k))^T (B(z^{k+1} - z^*)) \leq 0 \quad (11)$$

Since $y^{k+1} = y^k + \rho r^{k+1}$, $r^{k+1} = (y^{k+1} - y^k)/\rho$,

$$\begin{aligned} 2(y^{k+1} - y^*)^T r^{k+1} &= 2(y^k - y^*)^T r^{k+1} + \rho \|r^{k+1}\|_2^2 + \rho \|r^{k+1}\|_2^2 \\ &= \frac{2}{\rho} (y^k - y^*)^T (y^{k+1} - y^k) + \frac{1}{\rho} \|y^{k+1} - y^k\|_2^2 + \rho \|r^{k+1}\|_2^2 \\ &= \frac{1}{\rho} (\|y^{k+1} - y^*\|_2^2 - \|y^k - y^*\|_2^2) + \rho \|r^{k+1}\|_2^2 \end{aligned}$$

Since $z^{k+1} - z^* = (z^{k+1} - z^k) + (z^k - z^*)$,

$$\begin{aligned} \rho \|r^{k+1}\|_2^2 - 2\rho(B(z^{k+1} - z^k))^T r^{k+1} + 2\rho(B(z^{k+1} - z^k))^T (B(z^{k+1} - z^*)) \\ = \rho \|r^{k+1} - B(z^{k+1} - z^k)\|_2^2 + \rho \|B(z^{k+1} - z^k)\|_2^2 + 2\rho(B(z^{k+1} - z^k))^T (B(z^k - z^*)) \end{aligned}$$

Since $z^{k+1} - z^k = (z^{k+1} - z^*) - (z^k - z^*)$,

$$\begin{aligned} \rho \|r^{k+1}\|_2^2 - 2\rho(B(z^{k+1} - z^k))^T r^{k+1} + 2\rho(B(z^{k+1} - z^k))^T (B(z^{k+1} - z^*)) \\ = \rho \|r^{k+1} - B(z^{k+1} - z^k)\|_2^2 + \rho \|B(z^{k+1} - z^k)\|_2^2 + 2\rho(B(z^{k+1} - z^k))^T (B(z^k - z^*)) \\ = \rho \|r^{k+1} - B(z^{k+1} - z^k)\|_2^2 + \rho (\|B(z^{k+1} - z^*)\|_2^2 - \|B(z^k - z^*)\|_2^2) \end{aligned}$$

Combining the above, (11) is equivalent to

$$V^k - V^{k+1} \geq \rho \|r^{k+1} - B(z^{k+1} - z^k)\|_2^2$$

Recall that z^{k+1} minimizes $g(z) + (y^{k+1})^T Bz$ and z^k minimizes $g(z) + (y^k)^T Bz$. Then

$$g(z^{k+1}) + (y^{k+1})^T Bz^{k+1} \leq g(z^k) + (y^{k+1})^T Bz^k$$

and

$$g(z^k) + (y^k)^T Bz^k \leq g(z^{k+1}) + (y^k)^T Bz^{k+1}$$

Hence

$$(y^{k+1} - y^k)^T (B(z^{k+1} - z^k)) \leq 0$$

This shows that (10) holds.

Therefore, V^k is a bounded decreasing sequence. Hence, y^k, Bz^k are also bounded.

By iterating (10), we have

$$\rho \sum (\|r^{k+1}\|_2^2 + \|B(z^{k+1} - z^k)\|_2^2) \leq V^0.$$

This shows that $r^k \rightarrow 0$, $B(z^{k+1} - z^k) \rightarrow 0$.

Since $B(z^{k+1} - z^k)$ is bounded, the right hand side of (9) tends to 0.

Moreover, the right hand side of (8) also tends to 0.

Combining the above, we have $p^k \rightarrow p^*$.

The proof that $y^k \rightarrow y^*$ can be found in [3].

For more details of the proof, please have a look at [1], [2], [3].

□

Reference

1. Chen L., Sun D.F., Toh K-C: *A note on the convergence of ADMM for linearly constrained convex optimization problems*
2. Fazel M., Pong T.K., Sun D.F., Tseng P: *Hankel matrix rank minimization with applications to system identification and realization.*
3. Boyd S., Parikh N., Chu E., Peleato B., Eckstein J.: *Distributed optimization and statistical learning via the alternating direction method of multipliers*

A

Convergence Proof

The basic convergence result given in §3.2 can be found in several references, such as [81, 63]. Many of these give more sophisticated results, with more general penalties or inexact minimization. For completeness, we give a proof here.

We will show that if f and g are closed, proper, and convex, and the Lagrangian L_0 has a saddle point, then we have primal residual convergence, meaning that $r^k \rightarrow 0$, and objective convergence, meaning that $p^k \rightarrow p^*$, where $p^k = f(x^k) + g(z^k)$. We will also see that the dual residual $s^k = \rho A^T B(z^k - z^{k-1})$ converges to zero.

Let (x^*, z^*, y^*) be a saddle point for L_0 , and define

$$V^k = (1/\rho)\|y^k - y^*\|_2^2 + \rho\|B(z^k - z^*)\|_2^2,$$

We will see that V^k is a *Lyapunov function* for the algorithm, *i.e.*, a nonnegative quantity that decreases in each iteration. (Note that V^k is unknown while the algorithm runs, since it depends on the unknown values z^* and y^* .)

We first outline the main idea. The proof relies on three key inequalities, which we will prove below using basic results from convex analysis

along with simple algebra. The first inequality is

$$V^{k+1} \leq V^k - \rho \|r^{k+1}\|_2^2 - \rho \|B(z^{k+1} - z^k)\|_2^2. \quad (\text{A.1})$$

This states that V^k decreases in each iteration by an amount that depends on the norm of the residual and on the change in z over one iteration. Because $V^k \leq V^0$, it follows that y^k and Bz^k are bounded. Iterating the inequality above gives that

$$\rho \sum_{k=0}^{\infty} \left(\|r^{k+1}\|_2^2 + \|B(z^{k+1} - z^k)\|_2^2 \right) \leq V^0,$$

which implies that $r^k \rightarrow 0$ and $B(z^{k+1} - z^k) \rightarrow 0$ as $k \rightarrow \infty$. Multiplying the second expression by ρA^T shows that the dual residual $s^k = \rho A^T B(z^{k+1} - z^k)$ converges to zero. (This shows that the stopping criterion (3.12), which requires the primal and dual residuals to be small, will eventually hold.)

The second key inequality is

$$\begin{aligned} p^{k+1} - p^* & \leq -(y^{k+1})^T r^{k+1} - \rho (B(z^{k+1} - z^k))^T (-r^{k+1} + B(z^{k+1} - z^*)), \end{aligned} \quad (\text{A.2})$$

and the third inequality is

$$p^* - p^{k+1} \leq y^{*T} r^{k+1}. \quad (\text{A.3})$$

The righthand side in (A.2) goes to zero as $k \rightarrow \infty$, because $B(z^{k+1} - z^*)$ is bounded and both r^{k+1} and $B(z^{k+1} - z^k)$ go to zero. The righthand side in (A.3) goes to zero as $k \rightarrow \infty$, since r^k goes to zero. Thus we have $\lim_{k \rightarrow \infty} p^k = p^*$, *i.e.*, objective convergence.

Before giving the proofs of the three key inequalities, we derive the inequality (3.11) mentioned in our discussion of stopping criterion from the inequality (A.2). We simply observe that $-r^{k+1} + B(z^{k+1} - z^k) = -A(x^{k+1} - x^*)$; substituting this into (A.2) yields (3.11),

$$p^{k+1} - p^* \leq -(y^{k+1})^T r^{k+1} + (x^{k+1} - x^*)^T s^{k+1}.$$

Proof of inequality (A.3)

Since (x^*, z^*, y^*) is a saddle point for L_0 , we have

$$L_0(x^*, z^*, y^*) \leq L_0(x^{k+1}, z^{k+1}, y^*).$$

Using $Ax^* + Bz^* = c$, the lefthand side is p^* . With $p^{k+1} = f(x^{k+1}) + g(z^{k+1})$, this can be written as

$$p^* \leq p^{k+1} + y^{*T} r^{k+1},$$

which gives (A.3).

Proof of inequality (A.2)

By definition, x^{k+1} minimizes $L_\rho(x, z^k, y^k)$. Since f is closed, proper, and convex it is subdifferentiable, and so is L_ρ . The (necessary and sufficient) optimality condition is

$$0 \in \partial L_\rho(x^{k+1}, z^k, y^k) = \partial f(x^{k+1}) + A^T y^k + \rho A^T (Ax^{k+1} + Bz^k - c).$$

(Here we use the basic fact that the subdifferential of the sum of a subdifferentiable function and a differentiable function with domain \mathbf{R}^n is the sum of the subdifferential and the gradient; see, *e.g.*, [140, §23].)

Since $y^{k+1} = y^k + \rho r^{k+1}$, we can plug in $y^k = y^{k+1} - \rho r^{k+1}$ and rearrange to obtain

$$0 \in \partial f(x^{k+1}) + A^T (y^{k+1} - \rho B(z^{k+1} - z^k)).$$

This implies that x^{k+1} minimizes

$$f(x) + (y^{k+1} - \rho B(z^{k+1} - z^k))^T Ax.$$

A similar argument shows that z^{k+1} minimizes $g(z) + y^{(k+1)T} Bz$. It follows that

$$\begin{aligned} f(x^{k+1}) + (y^{k+1} - \rho B(z^{k+1} - z^k))^T Ax^{k+1} \\ \leq f(x^*) + (y^{k+1} - \rho B(z^{k+1} - z^k))^T Ax^* \end{aligned}$$

and that

$$g(z^{k+1}) + y^{(k+1)T} Bz^{k+1} \leq g(z^*) + y^{(k+1)T} Bz^*.$$

Adding the two inequalities above, using $Ax^* + Bz^* = c$, and rearranging, we obtain (A.2).

Proof of inequality (A.1)

Adding (A.2) and (A.3), regrouping terms, and multiplying through by 2 gives

$$\begin{aligned} & 2(y^{k+1} - y^*)^T r^{k+1} - 2\rho(B(z^{k+1} - z^k))^T r^{k+1} \\ & + 2\rho(B(z^{k+1} - z^k))^T (B(z^{k+1} - z^*)) \leq 0. \end{aligned} \quad (\text{A.4})$$

The result (A.1) will follow from this inequality after some manipulation and rewriting.

We begin by rewriting the first term. Substituting $y^{k+1} = y^k + \rho r^{k+1}$ gives

$$2(y^k - y^*)^T r^{k+1} + \rho \|r^{k+1}\|_2^2 + \rho \|r^{k+1}\|_2^2,$$

and substituting $r^{k+1} = (1/\rho)(y^{k+1} - y^k)$ in the first two terms gives

$$(2/\rho)(y^k - y^*)^T (y^{k+1} - y^k) + (1/\rho) \|y^{k+1} - y^k\|_2^2 + \rho \|r^{k+1}\|_2^2.$$

Since $y^{k+1} - y^k = (y^{k+1} - y^*) - (y^k - y^*)$, this can be written as

$$(1/\rho) \left(\|y^{k+1} - y^*\|_2^2 - \|y^k - y^*\|_2^2 \right) + \rho \|r^{k+1}\|_2^2. \quad (\text{A.5})$$

We now rewrite the remaining terms, *i.e.*,

$$\rho \|r^{k+1}\|_2^2 - 2\rho(B(z^{k+1} - z^k))^T r^{k+1} + 2\rho(B(z^{k+1} - z^k))^T (B(z^{k+1} - z^*)),$$

where $\rho \|r^{k+1}\|_2^2$ is taken from (A.5). Substituting

$$z^{k+1} - z^* = (z^{k+1} - z^k) + (z^k - z^*)$$

in the last term gives

$$\begin{aligned} & \rho \|r^{k+1} - B(z^{k+1} - z^k)\|_2^2 + \rho \|B(z^{k+1} - z^k)\|_2^2 \\ & + 2\rho(B(z^{k+1} - z^k))^T (B(z^k - z^*)), \end{aligned}$$

and substituting

$$z^{k+1} - z^k = (z^{k+1} - z^*) - (z^k - z^*)$$

in the last two terms, we get

$$\rho \|r^{k+1} - B(z^{k+1} - z^k)\|_2^2 + \rho \left(\|B(z^{k+1} - z^*)\|_2^2 - \|B(z^k - z^*)\|_2^2 \right).$$

With the previous step, this implies that (A.4) can be written as

$$V^k - V^{k+1} \geq \rho \|r^{k+1} - B(z^{k+1} - z^k)\|_2^2. \quad (\text{A.6})$$

To show (A.1), it now suffices to show that the middle term $-2\rho r^{(k+1)T}(B(z^{k+1} - z^k))$ of the expanded right hand side of (A.6) is positive. To see this, recall that z^{k+1} minimizes $g(z) + y^{(k+1)T}Bz$ and z^k minimizes $g(z) + y^{kT}Bz$, so we can add

$$g(z^{k+1}) + y^{(k+1)T}Bz^{k+1} \leq g(z^k) + y^{(k+1)T}Bz^k$$

and

$$g(z^k) + y^{kT}Bz^k \leq g(z^{k+1}) + y^{kT}Bz^{k+1}$$

to get that

$$(y^{k+1} - y^k)^T(B(z^{k+1} - z^k)) \leq 0.$$

Substituting $y^{k+1} - y^k = \rho r^{k+1}$ gives the result, since $\rho > 0$.