Statistical Optimization: Lecture 13

Alternating Direction Method of Multipliers

Zijian Guo

Zhejiang University
Center for Data Science

May 2, 2026

Outline

ADMM: Alternating Direction Method of Multipliers

Optimality Conditions and Stopping Criterion

Case study: LASSO

Case study: Distributed optimization

Additive structure

In Lecture 12, we already introduced the equality-constrained problem

$$\min_{\theta} \; f_0(\theta) \qquad \text{s.t.} \quad h(\theta) = 0,$$

and discussed the ordinary Lagrangian, dual ascent, and the augmented Lagrangian.

In Lecture 13, the point is to study an important structured special case:

$$\begin{aligned} \min_{\theta_1, \theta_2} \quad & f_1(\theta_1) + f_2(\theta_2) \\ \text{s.t.} \quad & A_1\theta_1 + A_2\theta_2 = b. \end{aligned}$$

This is exactly Lecture 12 with

$$f_0(\theta) = f_1(\theta_1) + f_2(\theta_2), \qquad h(\theta_1, \theta_2) = A_1\theta_1 + A_2\theta_2 - b.$$

How the augmented Lagrangian arises

For the structured problem

$$\begin{aligned} \min_{\theta_1,\theta_2} \quad & f_1(\theta_1) + f_2(\theta_2) \\ \text{s.t.} \quad & A_1\theta_1 + A_2\theta_2 = b, \end{aligned}$$

a natural idea is to first add a quadratic penalty to the objective:

$$\begin{aligned} \min_{\theta_1,\theta_2} \quad & f_1(\theta_1) + f_2(\theta_2) + \frac{\mu}{2}\|A_1\theta_1 + A_2\theta_2 - b\|_2^2 \\ \text{s.t.} \quad & A_1\theta_1 + A_2\theta_2 = b, \qquad \mu > 0. \end{aligned}$$

Now take the ordinary Lagrangian of this penalized constrained problem:

$$\begin{aligned} L_A(\theta_1, \theta_2, \nu; \mu) := f_1(\theta_1) + f_2(\theta_2) + \nu^\top (A_1\theta_1 + A_2\theta_2 - b) \\ + \frac{\mu}{2}\|A_1\theta_1 + A_2\theta_2 - b\|_2^2. \end{aligned}$$

This is exactly the augmented Lagrangian.

ALM primal step is coupled

Applying ALM to problem gives

$$(\theta_1^{k+1}, \theta_2^{k+1}) \approx \arg\min_{\theta_1,\theta_2} L_A(\theta_1, \theta_2, \nu^k; \mu),$$

followed by the multiplier update

$$\nu^{k+1} = \nu^k + \mu\big(A_1\theta_1^{k+1} + A_2\theta_2^{k+1} - b\big).$$

This is completely natural. The difficulty is that the quadratic term

$$\frac{\mu}{2}\|A_1\theta_1 + A_2\theta_2 - b\|_2^2$$

still couples the two blocks, so the primal step is usually not separable.

Why decomposition matters

In many applications, the variable split has real computational meaning: θ_1 and θ_2 may belong to different models, different data blocks, or different machines.

If we must solve one large coupled ALM subproblem each iteration, then we lose several advantages:

- simple blockwise minimizations or proximal steps,
- sparsity and problem structure inside each block,
- distributed or parallel implementation across blocks.

So decomposition is important because it often makes each iteration cheaper, easier to implement, and more scalable.

ADMM: Alternating Direction Method of Multipliers

ADMM keeps the same augmented Lagrangian as ALM, but replaces the joint minimization by alternating block updates:

$$\theta_1^{k+1} := \arg\min_{\theta_1} L_A(\theta_1, \theta_2^k, \nu^k; \mu),$$

$$\theta_2^{k+1} := \arg\min_{\theta_2} L_A(\theta_1^{k+1}, \theta_2, \nu^k; \mu),$$

$$\nu^{k+1} = \nu^k + \mu\big(A_1\theta_1^{k+1} + A_2\theta_2^{k+1} - b\big).$$

So ADMM can be viewed as ALM + decomposition.

Scaled ADMM form

For the coupling constraint, define the residual

$$r(\theta_1, \theta_2) := A_1\theta_1 + A_2\theta_2 - b.$$

Then the augmented Lagrangian can be written as

$$L_A(\theta_1, \theta_2, \nu) = f_1(\theta_1) + f_2(\theta_2) + \nu^\top r(\theta_1, \theta_2) + \frac{\mu}{2}\|r(\theta_1, \theta_2)\|_2^2.$$

Now define the scaled dual variable

$$u := \frac{1}{\mu}\nu.$$

Scaled ADMM form

Using $\nu = \mu u$ and $r = r(\theta_1, \theta_2)$, we complete the square:

$$\nu^\top r + \frac{\mu}{2}\|r\|_2^2 = \frac{\mu}{2}\|r + u\|_2^2 - \frac{\mu}{2}\|u\|_2^2.$$

Therefore,

$$L_A(\theta_1, \theta_2, u) = f_1(\theta_1) + f_2(\theta_2) + \frac{\mu}{2}\|r + u\|_2^2 - \frac{\mu}{2}\|u\|_2^2.$$

For the θ_1- and θ_2-subproblems, the last term is constant, so we may drop it.

Scaled ADMM form

The scaled ADMM iteration is

$$\theta_1^{k+1} := \arg\min_{\theta_1} \left\{ f_1(\theta_1) + \frac{\mu}{2} \|A_1\theta_1 + A_2\theta_2^k - b + u^k\|_2^2 \right\},$$

$$\theta_2^{k+1} := \arg\min_{\theta_2} \left\{ f_2(\theta_2) + \frac{\mu}{2} \|A_1\theta_1^{k+1} + A_2\theta_2 - b + u^k\|_2^2 \right\},$$

$$u^{k+1} = u^k + r^{k+1}, \qquad r^{k+1} := A_1\theta_1^{k+1} + A_2\theta_2^{k+1} - b.$$

$$(\text{due to } \nu^{k+1} = \nu^k + \mu\big(A_1\theta_1^{k+1} + A_2\theta_2^{k+1} - b\big))$$

Hence

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

so u^k is the running sum of the primal residuals.

Theorem (Convergence of ADMM)

Assume that $f_1 : \mathbb{R}^{n_1} \to \mathbb{R}$ and $f_2 : \mathbb{R}^{n_2} \to \mathbb{R}$ are closed, convex, and differentiable, and there exists $(\theta_1^\star, \theta_2^\star, \nu^\star)$ such that for all θ_1, θ_2, ν,

$$L_0(\theta_1^\star, \theta_2^\star, \nu) \leq L_0(\theta_1^\star, \theta_2^\star, \nu^\star) \leq L_0(\theta_1, \theta_2, \nu^\star).$$

Let $p^\star := f_1(\theta_1^\star) + f_2(\theta_2^\star)$ denote the optimal objective value. Then the ADMM iterates satisfy

- ***residual convergence:***
$$r^k \to 0;$$
- ***objective convergence:***
$$f_1(\theta_1^k) + f_2(\theta_2^k) \to p^\star;$$
- ***dual variable convergence:***
$$\nu^k \to \nu^\star.$$

Remarks on the theorem

- **Practical implication.** ADMM is often very effective when moderate accuracy is enough. If very high accuracy is required, the later-stage convergence can be slower.
- **Intuition for the saddle point assumption.** The primal variables minimize L_0 at fixed $\nu^\star$, while the multiplier $\nu^\star$ maximizes L_0 at fixed $(\theta_1^\star, \theta_2^\star)$. So the Lagrangian looks like a valley in the primal directions and a peak in the dual direction.
- **What is not guaranteed.** The theorem shows residual convergence, objective convergence, and dual variable convergence. But θ_1^k and θ_2^k need not themselves converge to optimal values, unless additional assumptions are imposed.

Outline

ADMM: Alternating Direction Method of Multipliers

Optimality Conditions and Stopping Criterion

Case study: LASSO

Case study: Distributed optimization

Optimality conditions of the original problem

The unaugmented Lagrangian is

$$L_0(\theta_1, \theta_2, \nu) = f_1(\theta_1) + f_2(\theta_2) + \nu^\top (A_1\theta_1 + A_2\theta_2 - b).$$

The necessary and sufficient optimality conditions are:

Primal feasibility

$$A_1\theta_1^\star + A_2\theta_2^\star - b = 0,$$

Stationary condition

$$\nabla f_1(\theta_1^\star) + A_1^\top \nu^\star = 0,$$
$$\nabla f_2(\theta_2^\star) + A_2^\top \nu^\star = 0.$$

The θ_2-update

By definition, θ_2^{k+1} minimizes

$$L_A(\theta_1^{k+1}, \theta_2, \nu^k) = f_1(\theta_1^{k+1}) + f_2(\theta_2) + (\nu^k)^\top \big(A_1\theta_1^{k+1} + A_2\theta_2 - b\big) + \frac{\mu}{2}\big\|A_1\theta_1^{k+1} + A_2\theta_2 - b\big\|_2^2.$$

with respect to θ_2. Therefore,

$$\nabla f_2(\theta_2^{k+1}) + A_2^\top \nu^k + \mu A_2^\top \big(A_1\theta_1^{k+1} + A_2\theta_2^{k+1} - b\big) = 0.$$

Using the multiplier update

$$\nu^{k+1} = \nu^k + \mu\big(A_1\theta_1^{k+1} + A_2\theta_2^{k+1} - b\big),$$

we obtain

$$\nabla f_2(\theta_2^{k+1}) + A_2^\top \nu^{k+1} = 0.$$

So the second stationary condition is satisfied exactly at every iteration.

The θ_1-update

By definition, θ_1^{k+1} minimizes

$$L_A(\theta_1, \theta_2^k, \nu^k) = f_1(\theta_1) + f_2(\theta_2^k) + (\nu^k)^\top \left(A_1\theta_1 + A_2\theta_2^k - b\right) + \frac{\mu}{2}\left\|A_1\theta_1 + A_2\theta_2^k - b\right\|_2^2$$

with respect to θ_1. Therefore,

$$\nabla f_1(\theta_1^{k+1}) + A_1^\top \nu^k + \mu A_1^\top \left(A_1\theta_1^{k+1} + A_2\theta_2^k - b\right) = 0.$$

Add and subtract $A_2\theta_2^{k+1}$ inside the last term:

$$0 = \nabla f_1(\theta_1^{k+1}) + A_1^\top \nu^k + \mu A_1^\top \left(A_1\theta_1^{k+1} + A_2\theta_2^{k+1} - b\right) + \mu A_1^\top A_2(\theta_2^k - \theta_2^{k+1}).$$

The θ_1-update

Using

$$\nu^{k+1} = \nu^k + \mu\big(A_1\theta_1^{k+1} + A_2\theta_2^{k+1} - b\big),$$

this identity becomes

$$\mu A_1^\top A_2(\theta_2^{k+1} - \theta_2^k) = \nabla f_1(\theta_1^{k+1}) + A_1^\top \nu^{k+1}.$$

This motivates the definition of the **dual residual**

$$s^{k+1} := \mu A_1^\top A_2(\theta_2^{k+1} - \theta_2^k).$$

If s^{k+1} is small, then the first stationary condition is nearly satisfied.

Residuals and stopping criterion

The **primal residual** is the constraint violation

$$r^{k+1} := A_1\theta_1^{k+1} + A_2\theta_2^{k+1} - b.$$

The **dual residual** is

$$s^{k+1} := \mu A_1^\top A_2(\theta_2^{k+1} - \theta_2^k).$$

So the optimality conditions are monitored by two quantities:

- r^{k+1}: how well the primal constraint is satisfied;
- s^{k+1}: how well the first stationary condition is satisfied.

So a standard practical stopping test is

$$\|r^k\|_2 \le \varepsilon_{\text{pri}}, \qquad \|s^k\|_2 \le \varepsilon_{\text{dual}}.$$

Recall: scaled ADMM form

We start from the two-block problem

$$\begin{aligned} \min_{\theta_1,\theta_2} \quad & f_1(\theta_1) + f_2(\theta_2) \\ \text{s.t.} \quad & A_1\theta_1 + A_2\theta_2 = b. \end{aligned}$$

Using the scaled dual variable $u = \nu/\mu$, the scaled ADMM iteration is

$$\theta_1^{k+1} := \arg\min_{\theta_1} \left\{ f_1(\theta_1) + \frac{\mu}{2}\|A_1\theta_1 + A_2\theta_2^k - b + u^k\|_2^2 \right\},$$

$$\theta_2^{k+1} := \arg\min_{\theta_2} \left\{ f_2(\theta_2) + \frac{\mu}{2}\|A_1\theta_1^{k+1} + A_2\theta_2 - b + u^k\|_2^2 \right\},$$

$$u^{k+1} = u^k + A_1\theta_1^{k+1} + A_2\theta_2^{k+1} - b.$$

A generic θ_1-update

From the first scaled ADMM step, define

$$v^k := b - A_2\theta_2^k - u^k.$$

Then

$$\theta_1^{k+1} := \arg\min_{\theta_1} \left\{ f_1(\theta_1) + \frac{\mu}{2}\|A_1\theta_1 - v^k\|_2^2 \right\}.$$

This is the basic subproblem behind many practical ADMM updates. We can similarly define the subproblem of θ_2^{k+1}

Outline

ADMM: Alternating Direction Method of Multipliers

Optimality Conditions and Stopping Criterion

Case study: LASSO

Case study: Distributed optimization

LASSO via ADMM

We consider the lasso problem

$$\min_{\theta} \frac{1}{2}\|X\theta - y\|_2^2 + \lambda\|\theta\|_1, \qquad \lambda > 0.$$

The squared loss is smooth, while the ℓ_1 term is nonsmooth at zero.

To fit the ADMM form, introduce a copy variable z and rewrite the problem as

$$\min_{\theta,z} \frac{1}{2}\|X\theta - y\|_2^2 + \lambda\|z\|_1 \qquad \text{s.t.} \quad \theta - z = 0.$$

So here we take

$$\theta_1 = \theta, \qquad \theta_2 = z, \qquad A_1 = I, \qquad A_2 = -I, \qquad b = 0.$$

Scaled augmented Lagrangian

Recall the scaled augmented Lagrangian for

$$\min_{\theta_1,\theta_2} f_1(\theta_1) + f_2(\theta_2) \qquad \text{s.t.} \quad A_1\theta_1 + A_2\theta_2 - b = 0$$

is

$$L_A(\theta_1, \theta_2, u; \mu) = f_1(\theta_1) + f_2(\theta_2) + \frac{\mu}{2}\|A_1\theta_1 + A_2\theta_2 - b + u\|_2^2 - \frac{\mu}{2}\|u\|_2^2.$$

For lasso, we have

$$f_1(\theta) = \frac{1}{2}\|X\theta - y\|_2^2, \qquad f_2(z) = \lambda\|z\|_1, \qquad A_1 = I,\ A_2 = -I,\ b = 0.$$

Hence

$$L_A(\theta, z, u) = \frac{1}{2}\|X\theta - y\|_2^2 + \lambda\|z\|_1 + \frac{\mu}{2}\|\theta - z + u\|_2^2 - \frac{\mu}{2}\|u\|_2^2.$$

Since the last term is independent of (θ, z), it does not affect the minimization steps.

ADMM iteration

Therefore the ADMM iteration is

$$\theta^{k+1} := \arg\min_{\theta} \left\{ \frac{1}{2}\|X\theta - y\|_2^2 + \frac{\mu}{2}\|\theta - z^k + u^k\|_2^2 \right\},$$

$$z^{k+1} := \arg\min_{z} \left\{ \lambda\|z\|_1 + \frac{\mu}{2}\|\theta^{k+1} - z + u^k\|_2^2 \right\},$$

$$u^{k+1} = u^k + \theta^{k+1} - z^{k+1}.$$

θ-update: a ridge-type linear system

The θ-subproblem is quadratic, so we set its gradient to zero:

$$X^\top (X\theta - y) + \mu\big(\theta - (z^k - u^k)\big) = 0.$$

Hence

$$\big(X^\top X + \mu I\big)\theta = X^\top y + \mu\big(z^k - u^k\big).$$

So the update is

$$\theta^{k+1} = \big(X^\top X + \mu I\big)^{-1}\Big(X^\top y + \mu\big(z^k - u^k\big)\Big).$$

If μ is fixed, then $X^\top X + \mu I$ is constant across iterations, so its factorization can be cached and reused.

z-update: where soft-thresholding appears

Now consider the z-subproblem:

$$z^{k+1} := \arg\min_z \left\{ \lambda \|z\|_1 + \frac{\mu}{2} \|\theta^{k+1} - z + u^k\|_2^2 \right\}.$$

Let

$$w := \theta^{k+1} + u^k.$$

Then this becomes

$$z^{k+1} = \arg\min_z \left\{ \lambda \|z\|_1 + \frac{\mu}{2} \|w - z\|_2^2 \right\} \tag{1}$$

So in lasso-ADMM, the nonsmooth block reduces exactly to an ℓ_1+ quadratic minimization problem.

This is the key reason why ADMM works well here: the smooth least-squares part and the nonsmooth sparsity term are handled separately.

z-update: soft-thresholding

The objective in (1) is separable across coordinates:

$$\lambda\|z\|_1 + \frac{\mu}{2}\|w - z\|_2^2 = \sum_{i=1}^{d}\left(\lambda|z_i| + \frac{\mu}{2}(w_i - z_i)^2\right).$$

Therefore the update can be computed componentwise:

$$(z^{k+1})_i = \mathcal{S}_{\lambda/\mu}(w_i), \qquad i = 1, \ldots, d,$$

where

$$\mathcal{S}_\kappa(a) = \operatorname{sign}(a)\max\{|a| - \kappa, 0\}.$$

Thus the z-block is just a soft-thresholding step, which shrinks small coordinates to zero exactly.

LASSO via ADMM: Algorithm

Given z^0, u^0 and $\mu > 0$, repeat:

$$\theta^{k+1} = \left(X^\top X + \mu I\right)^{-1}\Big(X^\top y + \mu(z^k - u^k)\Big),$$

$$z^{k+1} = \mathcal{S}_{\lambda/\mu}\big(\theta^{k+1} + u^k\big),$$

$$u^{k+1} = u^k + \theta^{k+1} - z^{k+1}.$$

A common stopping rule is based on the primal and dual residuals:

$$r^k = \theta^k - z^k, \qquad s^k = \mu(z^k - z^{k-1}).$$

Outline

ADMM: Alternating Direction Method of Multipliers

Optimality Conditions and Stopping Criterion

Case study: LASSO

Case study: Distributed optimization

Recall: scaled ADMM form

We start from the two-block problem

$$\begin{aligned} \min_{\theta_1,\theta_2} \quad & f_1(\theta_1) + f_2(\theta_2) \\ \text{s.t.} \quad & A_1\theta_1 + A_2\theta_2 = b. \end{aligned}$$

Using the scaled dual variable $u = \nu/\mu$, the scaled ADMM iteration is

$$\theta_1^{k+1} := \arg\min_{\theta_1} \left\{ f_1(\theta_1) + \frac{\mu}{2}\|A_1\theta_1 + A_2\theta_2^k - b + u^k\|_2^2 \right\},$$

$$\theta_2^{k+1} := \arg\min_{\theta_2} \left\{ f_2(\theta_2) + \frac{\mu}{2}\|A_1\theta_1^{k+1} + A_2\theta_2 - b + u^k\|_2^2 \right\},$$

$$u^{k+1} = u^k + A_1\theta_1^{k+1} + A_2\theta_2^{k+1} - b.$$

Why distributed optimization is difficult

Many large-scale problems have the form

$$\min_{\theta} \sum_{i=1}^{N} f_i(\theta),$$

where machine i stores only the local loss f_i.

The issue is that θ is still shared by all machines. So the problem is not truly separable.

- data are distributed,
- the model variable is global,
- local updates must still agree.

Core idea: consensus form

Introduce local copies $\theta_1, \ldots, \theta_N$ and one global variable z:

$$\min_{\theta_1,\ldots,\theta_N,z} \quad \sum_{i=1}^{N} f_i(\theta_i)$$
$$\text{s.t.} \quad \theta_i - z = 0, \qquad i = 1, \ldots, N.$$

This is the consensus form.

- machine i updates its own local copy θ_i,
- the constraint $\theta_i = z$ enforces global agreement.

Why ADMM instead of pure dual decomposition?

Dual decomposition enforces agreement only through multipliers. In practice, this can be sensitive to the dual step size.

ADMM adds the quadratic term

$$\frac{\mu}{2}\|\theta_i - z\|_2^2,$$

which pulls the local variables toward consensus.

So ADMM keeps

- **decomposition**: local blocks are still separate,
- **stability**: the quadratic term improves coordination.

What one ADMM iteration does

For the consensus problem, one ADMM iteration has three steps.

Step 1: local update. Each processor solves

$$\theta_i^{k+1} = \arg\min_{\theta_i} \left\{ f_i(\theta_i) + \frac{\mu}{2} \|\theta_i - z^k + u_i^k\|_2^2 \right\}.$$

Step 2: global coordination. Combine the local results and update the global variable z^{k+1}.

Step 3: dual update.

$$u_i^{k+1} = u_i^k + \theta_i^{k+1} - z^{k+1}.$$

So each iteration has the pattern

$$\text{local solve} \longrightarrow \text{global agree} \longrightarrow \text{disagreement correction.}$$

Why ADMM is well suited for distributed optimization

ADMM is effective for distributed optimization because:

1. Parallel local updates. Once the current global variable is fixed, each machine solves its own local subproblem independently.

2. Small communication cost. Machines exchange only variables or summaries, rather than the entire dataset.

3. Simple global coordination. After the local solves, one global step enforces consistency.

So ADMM separates a large coupled problem into many local solves + one coordination step.