

# Lasso Regression

---

Nipun Batra

IIT Gandhinagar

September 11, 2025

# Outline

1. Introduction and Motivation
2. Mathematical Formulation
3. Why Lasso Gives Sparsity
  - 3.1 Geometric Interpretation
  - 3.2 Gradient Descent Interpretation
4. Geometric Interpretation
5. Regularization Effects
6. Feature Selection Properties
7. Subgradient Methods
8. Coordinate Descent Algorithm
9. Worked Example
10. Visual Coordinate Descent
11. When Coordinate Descent Fails
12. Mathematical Derivation
13. Lasso vs Ridge Comparison
14. Summary and Applications

# Introduction and Motivation

# What is Lasso Regression?

## Definition: LASSO

Least **A**bsolute **S**hrinkage and **S**election **O**perator

# What is Lasso Regression?

## Definition: LASSO

Least **A**bsolute **S**hrinkage and **S**election **O**perator

## Key Points: Key Properties

- Uses L1 penalty (absolute values) instead of L2 penalty
- Leads to **sparse solutions** (many coefficients become exactly zero)
- Performs automatic feature selection
- Popular for high-dimensional problems

# Mathematical Formulation

## Problem: Why Not Just Use Ridge?

### Important: Limitation of Ridge Regression

Ridge regression shrinks coefficients but **never makes them exactly zero**

# Problem: Why Not Just Use Ridge?

## Important: Limitation of Ridge Regression

Ridge regression shrinks coefficients but **never makes them exactly zero**

## Example: High-Dimensional Problem

- 1000 features, only 50 are truly relevant
- Ridge gives tiny but non-zero coefficients for irrelevant features
- Model is not interpretable
- Need automatic feature selection!



# Lasso Objective Function: Constrained Form

## Definition: Constrained Optimization

Find  $\theta_{\text{opt}}$  such that:

$$\theta_{\text{opt}} = \arg \min_{\theta} \|(\mathbf{y} - \mathbf{X}\theta)\|_2^2 \text{ subject to } \|\theta\|_1 \leq s$$

# Lasso Objective Function: Constrained Form

## Definition: Constrained Optimization

Find  $\theta_{\text{opt}}$  such that:

$$\theta_{\text{opt}} = \arg \min_{\theta} \|(\mathbf{y} - \mathbf{X}\theta)\|_2^2 \text{ subject to } \|\theta\|_1 \leq s$$

L1 Norm (Manhattan Distance)

$$\|\theta\|_1 = |\theta_1| + |\theta_2| + \cdots + |\theta_d| = \sum_{j=1}^d |\theta_j|$$

# Lasso Objective Function: Penalized Form

## Theorem: Using Lagrangian Duality (KKT Conditions)

Constrained form is equivalent to:

$$\boldsymbol{\theta}_{\text{opt}} = \arg \min_{\boldsymbol{\theta}} \underbrace{\|(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1}_{\text{Lasso Objective}}$$

# Lasso Objective Function: Penalized Form

## Theorem: Using Lagrangian Duality (KKT Conditions)

Constrained form is equivalent to:

$$\theta_{\text{opt}} = \arg \min_{\theta} \underbrace{\|(\mathbf{y} - \mathbf{X}\theta)\|_2^2 + \lambda \|\theta\|_1}_{\text{Lasso Objective}}$$

## Key Points: Key Components

- $\|(\mathbf{y} - \mathbf{X}\theta)\|_2^2$ : **Data fitting term** (minimize prediction error)
- $\lambda \|\theta\|_1$ : **L1 penalty** (encourage sparsity)
- $\lambda \geq 0$ : **Regularization parameter** (controls sparsity)

# The Challenge: Non-Differentiability

## Important: Problem

The L1 norm  $\|\boldsymbol{\theta}\|_1 = \sum_j |\theta_j|$  is **not differentiable** at  $\theta_j = 0$

# The Challenge: Non-Differentiability

## Important: Problem

The L1 norm  $\|\boldsymbol{\theta}\|_1 = \sum_j |\theta_j|$  is **not differentiable** at  $\theta_j = 0$

Cannot Use Standard Calculus

$$\frac{\partial}{\partial \boldsymbol{\theta}} [\|(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1] = 0$$

This fails because  $\frac{\partial |\theta_j|}{\partial \theta_j}$  is undefined at  $\theta_j = 0$

# Solution Approaches

## Key Points: Three Main Approaches

- **Coordinate Descent:** Optimize one coefficient at a time
- **Subgradient Methods:** Generalize derivatives to non-smooth functions
- **Proximal Methods:** Use soft-thresholding operators

## Example: Focus

We'll concentrate on coordinate descent - most popular for Lasso

# Why Lasso Gives Sparsity



# Sparsity: The Key Question

## **Important: Central Question**

Why does Lasso produce sparse solutions while Ridge doesn't?

# Sparsity: The Key Question

## Important: Central Question

Why does Lasso produce sparse solutions while Ridge doesn't?

## Key Points: Two Perspectives

- **Geometric:** Shape of constraint regions
- **Algorithmic:** Behavior of optimization algorithms

# Sparsity: The Key Question

## Important: Central Question

Why does Lasso produce sparse solutions while Ridge doesn't?

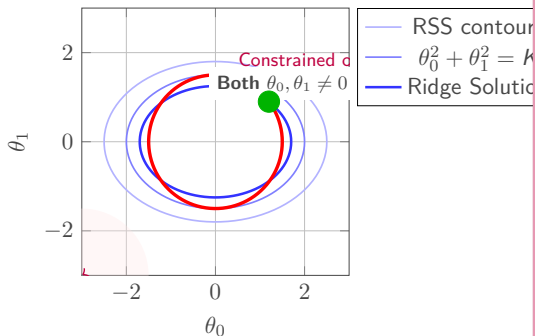
## Key Points: Two Perspectives

- **Geometric:** Shape of constraint regions
- **Algorithmic:** Behavior of optimization algorithms

## Example: Preview

We'll see why  $L_p$  norms with  $p < 2$  promote sparsity

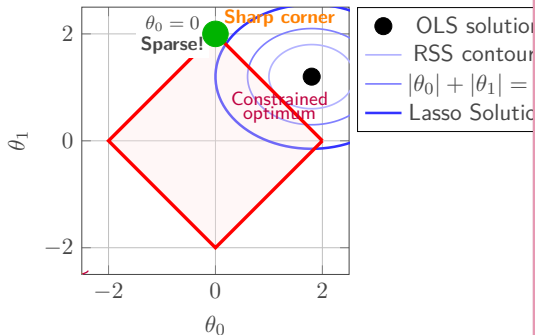
# L2 Norm: Ridge Regression



## Key Points: L2 Constraint Properties

- **Shape:** Perfect circle
- **Boundary:** Smooth everywhere
- **Intersection:** Rarely on axes
- **Result:** No sparsity
- **Effect:** Shrinks coefficients proportionally

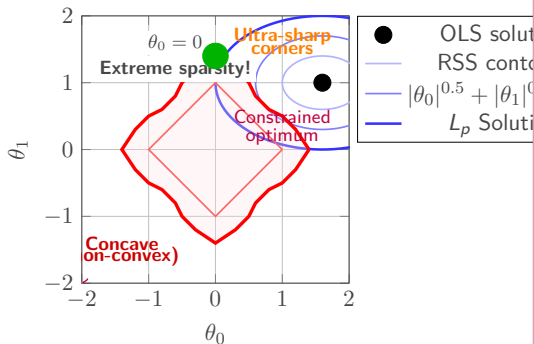
# L1 Norm: Lasso Regression



## Key Points: L1 Constraint Properties

- **Shape:** Diamond/rhombus
- **Corners:** Sharp at axes
- **Intersection:** High probability on axes
- **Result:** Automatic sparsity!
- **Effect:** Sets coefficients to

## $L_p$ Norm: $0 < p < 1$ (Example: $p = 0.5$ )



### Key Points: $L_p$ Properties ( $p < 1$ )

- **Shape:** Highly concave
- **Corners:** Ultra-sharp at axes
- **Sparsity:** Extremely high probability
- **Optimization:** Non-convex, much harder
- **Trade-off:**

## Sparsity Trend: $L_2 \rightarrow L_1 \rightarrow L_p$

### Theorem: Key Insight

As  $p$  decreases from 2 to 1 to  $p < 1$ :

- Constraint regions become more **pointed** at axes
- Probability of intersection at axes **increases**
- Sparsity **increases**
- Optimization difficulty **increases**

## Sparsity Trend: $L_2 \rightarrow L_1 \rightarrow L_p$

### Theorem: Key Insight

As  $p$  decreases from 2 to 1 to  $p < 1$ :

- Constraint regions become more **pointed** at axes
- Probability of intersection at axes **increases**
- Sparsity **increases**
- Optimization difficulty **increases**

### Important: Why $p = 1$ is Special

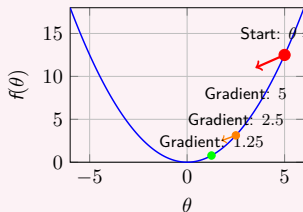
- Still promotes sparsity (sharp corners)
- Remains convex (unlike  $p < 1$ )
- Computationally tractable



## L2 vs L1: Gradient Descent Behavior

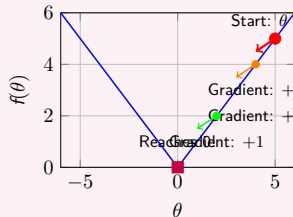
### Key Points: L2 Penalty:

$$f(\theta) = \frac{1}{2}\theta^2$$



### Key Points: L1 Penalty:

$$f(\theta) = |\theta|$$



### Example: Key Difference

L2: Gradient  $\propto \theta$  (decreases). L1: Constant gradient  $= \pm 1$

# Gradient Descent: Step 1

## L2 Update

- $\frac{df}{d\theta} = \theta = 5$
- $\theta_{new} = 5 - 0.5 \times 5 = 2.5$
- $f(2.5) = 3.125$

## L1 Update

- $\frac{df}{d\theta} = \text{sign}(\theta) = +1$
- $\theta_{new} = 5 - 0.5 \times 1 = 4.5$
- $f(4.5) = 4.5$

## Important: Key Difference

L2 gradient depends on  $\theta$  value, L1 gradient is constant  $\pm 1$

# Gradient Descent: Multiple Steps

## Key Points: L2 Sequence

- $\theta_0 = 5.0$
- $\theta_1 = 2.5$
- $\theta_2 = 1.25$
- $\theta_3 = 0.625$
- $\theta_4 = 0.3125$
- $\vdots$  (never exactly 0)

## Key Points: L1 Sequence

- $\theta_0 = 5.0$
- $\theta_1 = 4.5$
- $\theta_2 = 4.0$
- $\theta_3 = 3.5$
- $\vdots$
- $\theta_{10} = 0.0$  (exactly!)

## Theorem: Sparsity Mechanism

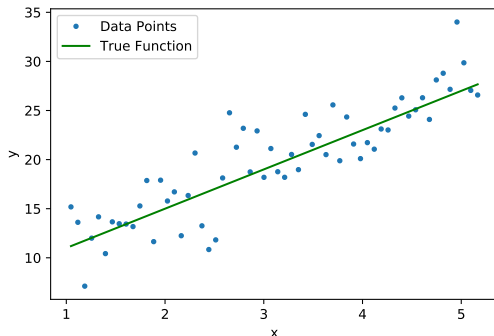
L1 penalty creates **constant gradient** that drives parameters to exactly zero in finite steps!

# Geometric Interpretation

# Sample Dataset for Demonstration

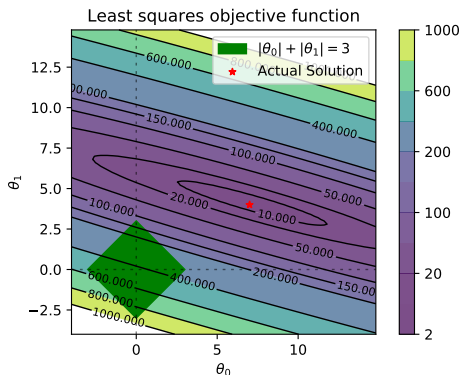
## Example: True Function

We'll demonstrate Lasso on a simple linear relationship:  $y = 4x + 7$



Sample data from  $y = 4x + 7$  with noise

# Geometric Interpretation: L1 vs L2 Constraints



L1 vs L2 constraint regions

## Key Points: Key Insight

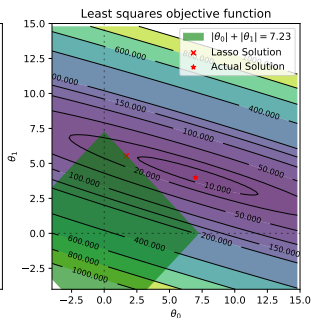
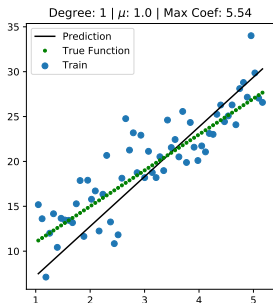
Diamond corners  $\Rightarrow$  exact zeros! Circle  $\Rightarrow$  no sparsity.

# Regularization Effects

# Effect of $\lambda$ on Solution Path

## Important: Regularization Parameter

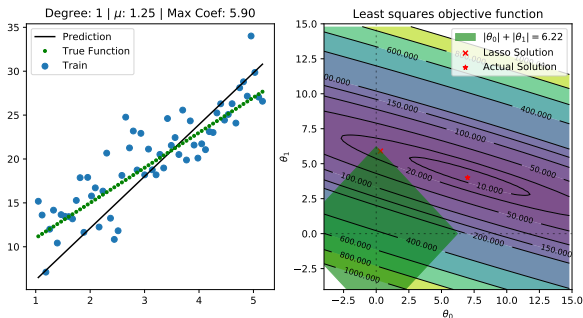
$\lambda$  controls fit vs sparsity trade-off



$\lambda = 1.0$  - Moderate regularization



# Increasing Regularization: $\lambda = 1.25$

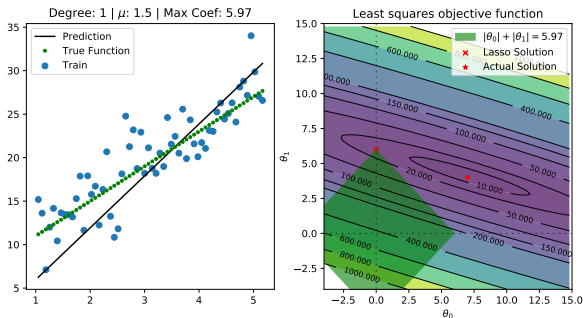


$\lambda = 1.25$  - Higher regularization

## Key Points: Observation

As  $\lambda$  increases  $\rightarrow$  solution becomes sparser

## Further Regularization: $\lambda = 1.5$

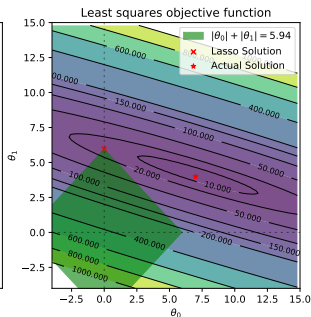
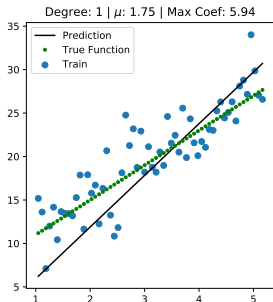


$\lambda = 1.5$  - Even higher regularization

**Important: Sparsity Effect**

More coefficients  $\rightarrow$  exactly zero

# High Regularization: $\lambda = 1.75$

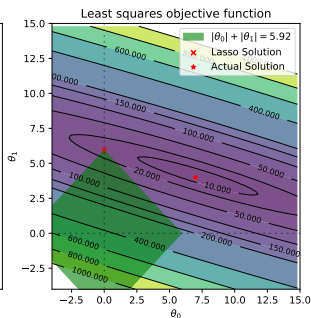
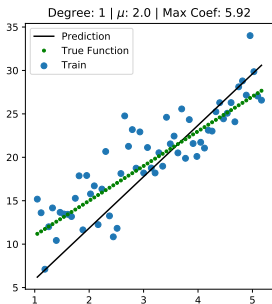


$\lambda = 1.75$  - Strong regularization

Feature Selection

Automatic selection of most important features

# Maximum Regularization: $\lambda = 2.0$

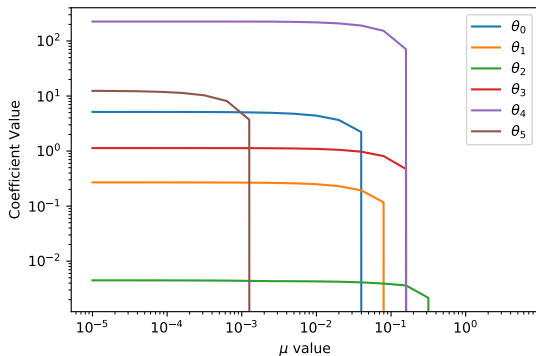


$\lambda = 2.0$  - Very strong regularization

**Important: Extreme Sparsity**

Only most crucial features remain non-zero

# Lasso Regularization Path



Coefficient values vs  $\lambda$

## Key Points: Key Observations

- Coefficients shrink to zero as  $\lambda$  increases
- Different coefficients zero at different  $\lambda$

# Feature Selection Properties

# Lasso for Automatic Feature Selection

## Definition: Automatic Feature Selection

Lasso performs regression and feature selection simultaneously by setting irrelevant coefficients to exactly zero

## Key Points: Key Advantages

- **Sparsity:** Many coefficients  $\rightarrow$  exactly zero
- **Interpretability:** Understand which features matter
- **Efficiency:** Fewer parameters, faster prediction

# Real-World Feature Selection

## Example: Genomics Example

Start with 20,000 genes → Lasso selects 50 relevant ones

## Key Points: Other Applications

- Text mining: 100k+ words → select key terms
- Finance: 1000+ indicators → find predictive signals
- Image processing: millions of pixels → identify features



# Subgradient Methods

# What is a Subgradient?

## **Definition: Subgradient**

A subgradient generalizes the concept of gradient to convex but non-differentiable functions

# What is a Subgradient?

## Definition: Subgradient

A subgradient generalizes the concept of gradient to convex but non-differentiable functions

## Example: Classic Example

For  $f(x) = |x|$ :

- $f'(x) = 1$  when  $x > 0$
- $f'(x) = -1$  when  $x < 0$
- $f'(0)$  is undefined, but subgradient  $\in [-1, 1]$

# What is a Subgradient?

## Definition: Subgradient

A subgradient generalizes the concept of gradient to convex but non-differentiable functions

## Example: Classic Example

For  $f(x) = |x|$ :

- $f'(x) = 1$  when  $x > 0$
- $f'(x) = -1$  when  $x < 0$
- $f'(0)$  is undefined, but subgradient  $\in [-1, 1]$

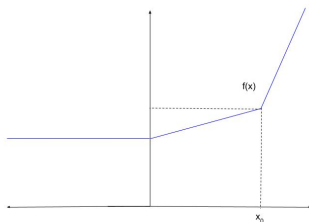
## Important: Why Important for Lasso?

The  $L_1$  penalty  $|\theta|$  is non-differentiable at  $\theta = 0$

# Subgradient: Visual Intuition

## Important: Task

Find the "derivative" of  $f(x)$  at the non-differentiable point  $x = x_0$

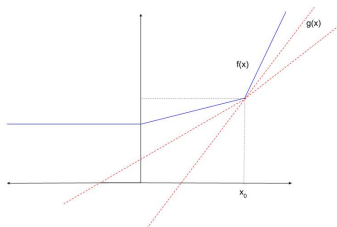


# Subgradient Construction

## Construction Method

Find a differentiable function  $g(x)$  such that:

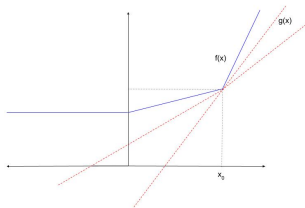
- $g(x_0) = f(x_0)$  (intersects at the point)
- $g(x) \leq f(x)$  for all  $x$  (lies below or on  $f$ )



# Computing the Subgradient

## Theorem: Subgradient Definition

Slope of  $g(x)$  at  $x = x_0$  gives subgradient of  $f$  at  $x_0$



Slope  $\rightarrow$  subgradient

# Subgradient Sets

## Key Points: Key Insight

Multiple supporting lines  $\Rightarrow$  set of valid subgradients

**Example:** For  $f(x) = |x|$

At  $x = 0$ : subgradient set is  $[-1, +1]$

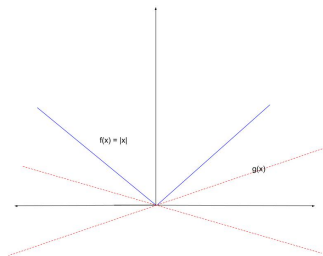


## Example: Subgradient of $f(x) = |x|$

### Subgradient Set

For  $f(x) = |x|$  at  $x = 0$ :

$$\partial f(0) = [-1, 1]$$



Lines with slope in  $[-1, 1]$  support  $|x|$  at  $x = 0$

# Connection to Lasso

## Important: Lasso Connection

This subgradient concept is exactly what we need for the L1 penalty term!

## Key Points: Next Step

We'll use subgradients to derive coordinate descent for Lasso

# Coordinate Descent Algorithm

# Introduction to Coordinate Descent

## **Definition: Coordinate Descent**

Optimization method: minimize one coordinate at a time

# Introduction to Coordinate Descent

## Definition: Coordinate Descent

Optimization method: minimize one coordinate at a time

## Key Points: Key Idea

- Hard: optimize all coordinates together
- Easy: optimize one coordinate at a time
- Perfect for non-differentiable Lasso!

# Coordinate Descent Algorithm

## Algorithm Overview

$$\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \text{ becomes } \min_{\theta_j} f(\theta_1, \dots, \theta_{j-1}, \theta_j, \theta_{j+1}, \dots, \theta_d)$$

## Important: Process

Cycle through coordinates, optimizing one at a time

# Coordinate Descent Properties

## Key Points: Advantages

- **No step-size:** Exact 1D minimization
- **Convergence:** Guaranteed for convex Lasso
- **Efficient:** Closed-form updates

# Coordinate Descent Properties

## Key Points: Advantages

- **No step-size:** Exact 1D minimization
- **Convergence:** Guaranteed for convex Lasso
- **Efficient:** Closed-form updates

## Selection Strategies

Cyclic, Random, or Greedy coordinate selection



# Worked Example

# Coordinate Descent Example Setup

## Example: Problem

Learn  $y = \theta_0 + \theta_1 x$  using coordinate descent on the dataset below

x	y
1	1
2	2
3	3

## Initial Conditions

- Initial parameters:  $(\theta_0, \theta_1) = (2, 3)$
- We'll run for 2 iterations
- Using standard least squares (no regularization for simplicity)

# Coordinate Descent : Example

Our predictor,  $\hat{y} = \theta_0 + \theta_1 x$

Error for  $i^{th}$  datapoint,  $\epsilon_i = y_i - \hat{y}_i$

$$\epsilon_1 = 1 - \theta_0 - \theta_1$$

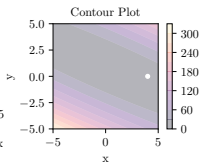
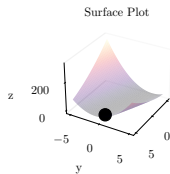
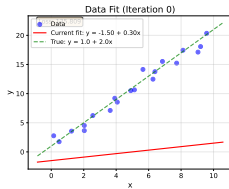
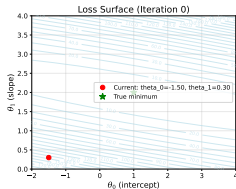
$$\epsilon_2 = 2 - \theta_0 - 2\theta_1$$

$$\epsilon_3 = 3 - \theta_0 - 3\theta_1$$

$$\text{MSE} = \frac{\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2}{3} = \frac{14 + 3\theta_0^2 + 14\theta_1^2 - 12\theta_0 - 28\theta_1 + 12\theta_0\theta_1}{3}$$

# Iteration 0

$$\text{MSE} = \frac{1}{3}(14 + 3\theta_0^2 + 14\theta_1^2 - 12\theta_0 - 28\theta_1 + 12\theta_0\theta_1)$$



# Coordinate Descent : Example

## Iteration 1

INIT:  $\theta_0 = 2$  and  $\theta_1 = 3$

$\theta_1 = 3$  optimize for  $\theta_0$

# Coordinate Descent : Example

## Iteration 1

INIT:  $\theta_0 = 2$  and  $\theta_1 = 3$

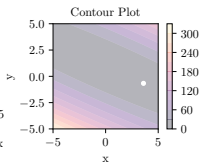
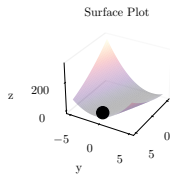
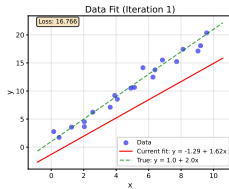
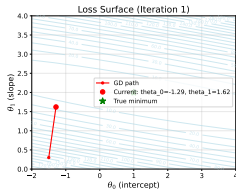
$\theta_1 = 3$  optimize for  $\theta_0$

$$\frac{\partial \text{MSE}}{\partial \theta_0} = 6\theta_0 + 24 = 0$$

$$\theta_0 = -4$$

# Iteration 1

$$\text{MSE} = \frac{1}{3}(14 + 3\theta_0^2 + 14\theta_1^2 - 12\theta_0 - 28\theta_1 + 12\theta_0\theta_1)$$



# Coordinate Descent : Example

## Iteration 2

INIT:  $\theta_0 = -4$  and  $\theta_1 = 3$

$\theta_0 = -4$  optimize for  $\theta_1$



# Coordinate Descent : Example

## Iteration 2

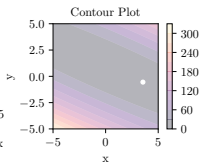
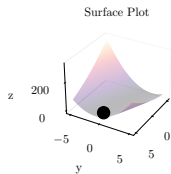
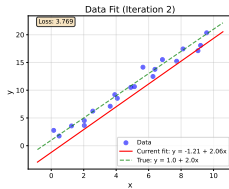
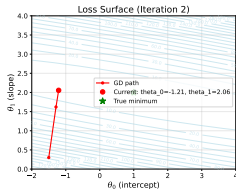
INIT:  $\theta_0 = -4$  and  $\theta_1 = 3$

$\theta_0 = -4$  optimize for  $\theta_1$

$\theta_1 = 2.7$

## Iteration 2

$$\text{MSE} = \frac{1}{3}(14 + 3\theta_0^2 + 14\theta_1^2 - 12\theta_0 - 28\theta_1 + 12\theta_0\theta_1)$$



# Coordinate Descent : Example

## Iteration 3

INIT:  $\theta_0 = -4$  and  $\theta_1 = 2.7$

$\theta_1 = 2.7$  optimize for  $\theta_0$

# Coordinate Descent : Example

## Iteration 3

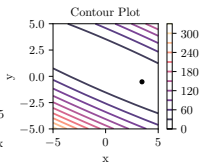
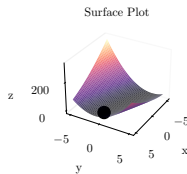
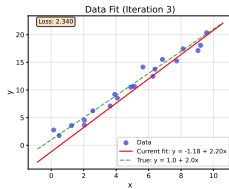
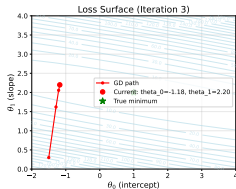
INIT:  $\theta_0 = -4$  and  $\theta_1 = 2.7$

$\theta_1 = 2.7$  optimize for  $\theta_0$

$\theta_0 = -3.4$

# Iteration 3

$$\text{MSE} = \frac{1}{3}(14 + 3\theta_0^2 + 14\theta_1^2 - 12\theta_0 - 28\theta_1 + 12\theta_0\theta_1)$$

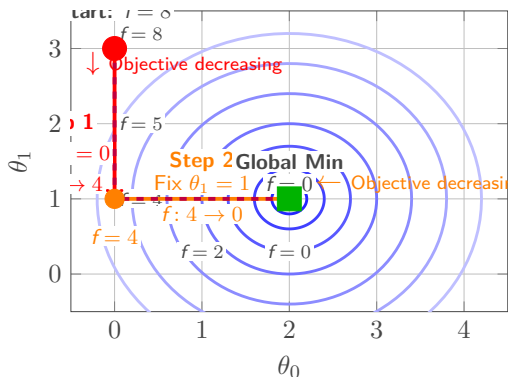


# Visual Coordinate Descent

# Coordinate Descent: Visual Algorithm

## Example: Setup

Minimize  $f(\theta_0, \theta_1) = (\theta_0 - 2)^2 + (\theta_1 - 1)^2$  starting from  $(0, 3)$



# Coordinate Descent: Step-by-Step

**Step 1: Fix  $\theta_0 = 0$**

Minimize:  $f(0, \theta_1) = 4 + (\theta_1 - 1)^2$

$$\frac{\partial f}{\partial \theta_1} = 2(\theta_1 - 1) = 0$$

$$\theta_1^* = 1$$

**Step 2: Fix  $\theta_1 = 1$**

Minimize:  $f(\theta_0, 1) = (\theta_0 - 2)^2$

$$\frac{\partial f}{\partial \theta_0} = 2(\theta_0 - 2) = 0$$

$$\theta_0^* = 2$$

## Key Points: Key Observations

- Each step moves parallel to coordinate axes
- Each step finds exact 1D minimum
- Converges to global minimum in 2 steps (for this quadratic)

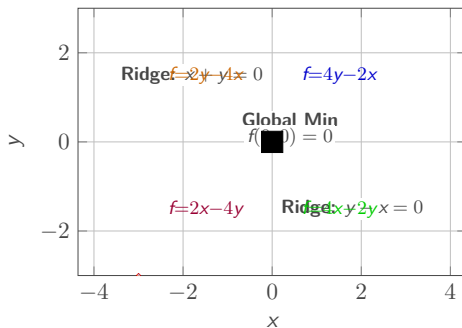


# When Coordinate Descent Fails

Function:  $f(x, y) = |x + y| + 3|y - x|$

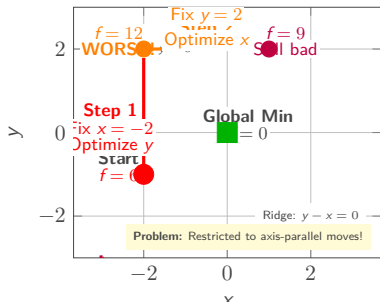
### Example: Problematic Function

Let's visualize:  $f(x, y) = |x + y| + 3|y - x|$  - a non-smooth function with ridges



**Important: Non-Smooth Surface with Ridges**

# Coordinate Descent Failure: Step by Step



## Function Values

- **Start**  $(-2, -1)$ :  
 $f = 6$
- **After Step 1**  
 $(-2, 2)$ :  $f = 12 \uparrow$   
**WORSE**
- **After Step 2**  $(1, 2)$ :  
 $f = 9$  Still bad
- **Global Min**  $(0, 0)$ :  
 $f = 0 \checkmark$  Optimal

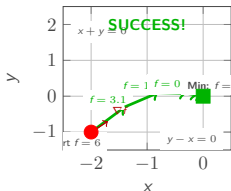
**Important: Root Cause**

**1D Optimization Prob-**

# Gradient Descent vs Coordinate Descent

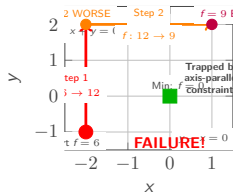
## Key Points: Gradient Descent

**Strategy:** Move in direction of steepest descent - can move diagonally



## Important: Coordinate Descent

**Constraint:** Only axis-parallel moves - gets trapped!



# Why Coordinate Descent Fails Here

## Important: Problem

Function  $f(x, y) = |x + y| + 3|y - x|$  is non-separable

## Analysis

- Start at  $(-2, -1)$ :  $f(-2, -1) = |-3| + 3|-1| = 6$
- Fix  $x = -2$ , optimize  $y$ : optimal  $y = 2$
- New point  $(-2, 2)$ :  $f(-2, 2) = 12$  (worse!)

# When Coordinate Descent Fails

## Theorem: Failure Conditions

- Non-separable functions
- Strong coupling between variables
- Need simultaneous movement in multiple directions

## Key Points: Fortunately

Lasso objective IS separable, so coordinate descent works well!

# Coordinate Descent for Unregularised Regression

- Express error as a difference of  $y_i$  and  $\hat{y}_i$

$$\hat{y}_i = \sum_{j=0}^d \theta_j x_i^j = \theta_0 x_i^0 + \theta_1 x_i^1 + \theta_2 x_i^2 + \dots + \theta_d x_i^d \quad (1)$$

$$\epsilon_i = y_i - \hat{y}_i = y_i - \theta_0 x_i^0 - \theta_1 x_i^1 - \dots - \theta_d x_i^d = y_i - \sum_{j=0}^d \theta_j x_i^j \quad (2)$$

## Coordinate Descent for Unregularised regression

$$\sum_{i=1}^n \epsilon^2 = \text{RSS} = \sum_{i=1}^n \left( y_i - \left( \theta_0 x_i^0 + \dots + \theta_j x_i^j + \theta_d x_i^d \right) \right)^2$$



## Coordinate Descent for Unregularised regression

$$\sum_{i=1}^n \epsilon^2 = \text{RSS} = \sum_{i=1}^n \left( y_i - \left( \theta_0 x_i^0 + \dots + \theta_j x_i^j + \theta_d x_i^d \right) \right)^2$$

$$\frac{\partial \text{RSS}(\theta_j)}{\partial \theta_j} = 2 \sum_{i=1}^n \left( y_i - \left( \theta_0 x_i^0 + \dots + \theta_j x_i^j + \dots \right) \right) \left( -x_i^j \right)$$

## Coordinate Descent for Unregularised regression

$$\sum_{i=1}^n \epsilon^2 = \text{RSS} = \sum_{i=1}^n \left( y_i - \left( \theta_0 x_i^0 + \dots + \theta_j x_i^j + \theta_d x_i^d \right) \right)^2$$

$$\frac{\partial \text{RSS}(\theta_j)}{\partial \theta_j} = 2 \sum_{i=1}^n \left( y_i - \left( \theta_0 x_i^0 + \dots + \theta_j x_i^j + \dots \right) \right) \left( -x_i^j \right)$$

$$= 2 \sum_{i=1}^n \left( y_i - \left( \theta_0 x_i^0 + \dots + \theta_d x_i^d \right) \right) \left( -x_i^j \right) + 2 \sum_{i=1}^n \theta_j (x_i^j)^2$$

## Coordinate Descent for Unregularised regression

$$\sum_{i=1}^n \epsilon^2 = \text{RSS} = \sum_{i=1}^n \left( y_i - \left( \theta_0 x_i^0 + \dots + \theta_j x_i^j + \theta_d x_i^d \right) \right)^2$$

$$\begin{aligned} \frac{\partial \text{RSS}(\theta_j)}{\partial \theta_j} &= 2 \sum_{i=1}^n \left( y_i - \left( \theta_0 x_i^0 + \dots + \theta_j x_i^j + \dots \right) \right) \left( -x_i^j \right) \\ &= 2 \sum_{i=1}^n \left( y_i - \left( \theta_0 x_i^0 + \dots + \theta_d x_i^d \right) \right) \left( -x_i^j \right) + 2 \sum_{i=1}^n \theta_j (x_i^j)^2 \end{aligned}$$

where:

$$\hat{y}_i^{(-j)} = \theta_0 x_i^0 + \dots + \theta_d x_i^d$$

is  $\hat{y}_i$  without  $\theta_j$

## Coordinate Descent for Unregularised regression

$$\text{Set } \frac{\partial \text{RSS}(\theta_j)}{\partial \theta_j} = 0$$

$$\theta_j = \sum_{i=1}^n \frac{(y_i - (\theta_0 x_i^0 + \dots + \theta_d x_i^d)) (x_i^j)}{(x_i^j)^2} = \frac{\rho_j}{z_j}$$

$$\rho_j = \sum_{i=1}^n x_i^j (y_i - \hat{y}_i^{(-j)}) \quad \text{and} \quad z_j = \sum_{i=1}^n (x_i^j)^2$$

$z_j$  is the squared of  $\ell_2$  norm of the  $j^{\text{th}}$  feature

# Mathematical Derivation

# Lasso Coordinate Descent: Setup

## Lasso Objective

$$\text{Minimize } \underbrace{\sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^d |\theta_j|}_{\text{Lasso Objective}}$$

# Lasso Coordinate Descent: Setup

## Lasso Objective

$$\text{Minimize } \underbrace{\sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=0}^d |\theta_j|}_{\text{Lasso Objective}}$$

## Key Points: Key Definitions

- $\rho_j = \sum_{i=1}^n x_{ij}(y_i - \hat{y}_i^{(-j)})$  (partial residual correlation)
- $z_j = \sum_{i=1}^n x_{ij}^2$  (feature norm squared)
- $\hat{y}_i^{(-j)}$  = prediction without  $j$ -th feature

## Coordinate Descent: Subgradient Analysis

Subgradient of Lasso Objective w.r.t.  $\theta_j$

$$\frac{\partial}{\partial \theta_j}(\text{Lasso Objective}) = -2\rho_j + 2\theta_j z_j + \lambda \frac{\partial}{\partial \theta_j} |\theta_j|$$



## Coordinate Descent: Subgradient Analysis

Subgradient of Lasso Objective w.r.t.  $\theta_j$

$$\frac{\partial}{\partial \theta_j}(\text{Lasso Objective}) = -2\rho_j + 2\theta_j z_j + \lambda \frac{\partial}{\partial \theta_j} |\theta_j|$$

**Theorem: Subgradient of  $|\theta_j|$**

$$\frac{\partial}{\partial \theta_j} |\theta_j| = \begin{cases} +1 & \text{if } \theta_j > 0 \\ [-1, +1] & \text{if } \theta_j = 0 \\ -1 & \text{if } \theta_j < 0 \end{cases}$$

## Case Analysis: $\theta_j > 0$

Case 1:  $\theta_j > 0$

Subgradient is +1, so optimality condition:

$$-2\rho_j + 2\theta_j z_j + \lambda = 0$$

## Case Analysis: $\theta_j > 0$

Case 1:  $\theta_j > 0$

Subgradient is +1, so optimality condition:

$$-2\rho_j + 2\theta_j z_j + \lambda = 0$$

**Theorem: Solution**

$$\theta_j = \frac{\rho_j - \lambda/2}{z_j}$$

This is valid when  $\rho_j > \lambda/2$  (ensures  $\theta_j > 0$ )

## Case Analysis: $\theta_j > 0$

**Case 1:  $\theta_j > 0$**

Subgradient is +1, so optimality condition:

$$-2\rho_j + 2\theta_j z_j + \lambda = 0$$

**Theorem: Solution**

$$\theta_j = \frac{\rho_j - \lambda/2}{z_j}$$

This is valid when  $\rho_j > \lambda/2$  (ensures  $\theta_j > 0$ )

**Important: Soft Thresholding**

Notice the  $-\lambda/2$  term: this "shrinks" the coefficient!

## Case Analysis: $\theta_j < 0$

Case 2:  $\theta_j < 0$

Subgradient is  $-1$ , so optimality condition:

$$-2\rho_j + 2\theta_j z_j - \lambda = 0$$

## Case Analysis: $\theta_j < 0$

Case 2:  $\theta_j < 0$

Subgradient is  $-1$ , so optimality condition:

$$-2\rho_j + 2\theta_j z_j - \lambda = 0$$

**Theorem: Solution**

$$\theta_j = \frac{\rho_j + \lambda/2}{z_j}$$

This is valid when  $\rho_j < -\lambda/2$  (ensures  $\theta_j < 0$ )

## Case Analysis: $\theta_j < 0$

**Case 2:  $\theta_j < 0$**

Subgradient is  $-1$ , so optimality condition:

$$-2\rho_j + 2\theta_j z_j - \lambda = 0$$

**Theorem: Solution**

$$\theta_j = \frac{\rho_j + \lambda/2}{z_j}$$

This is valid when  $\rho_j < -\lambda/2$  (ensures  $\theta_j < 0$ )

**Important: Symmetric Shrinkage**

Same shrinkage effect, but in the opposite direction!

## Case Analysis: $\theta_j = 0$

Case 3:  $\theta_j = 0$

Subgradient  $\in [-1, +1]$ , so optimality requires:

$$0 \in [-2\rho_j - \lambda, -2\rho_j + \lambda]$$



## Case Analysis: $\theta_j = 0$

**Case 3:**  $\theta_j = 0$

Subgradient  $\in [-1, +1]$ , so optimality requires:

$$0 \in [-2\rho_j - \lambda, -2\rho_j + \lambda]$$

### Theorem: Zero Condition

This happens when:

$$-2\rho_j - \lambda \leq 0 \text{ and } -2\rho_j + \lambda \geq 0$$

$$\Rightarrow -\frac{\lambda}{2} \leq \rho_j \leq \frac{\lambda}{2}$$

## Case Analysis: $\theta_j = 0$

**Case 3:**  $\theta_j = 0$

Subgradient  $\in [-1, +1]$ , so optimality requires:

$$0 \in [-2\rho_j - \lambda, -2\rho_j + \lambda]$$

### Theorem: Zero Condition

This happens when:

$$-2\rho_j - \lambda \leq 0 \text{ and } -2\rho_j + \lambda \geq 0$$

$$\Rightarrow -\frac{\lambda}{2} \leq \rho_j \leq \frac{\lambda}{2}$$

### Important: Sparsity Mechanism

If  $\rho_j > \frac{\lambda}{2}$ , then  $\theta_j = -\frac{\lambda}{2\rho_j} < -1$ . If  $\rho_j < -\frac{\lambda}{2}$ , then  $\theta_j = \frac{\lambda}{2\rho_j} > 1$ .

# Soft-Thresholding Operator

## Definition: Complete Lasso Update Rule

$$\theta_j = \begin{cases} \frac{\rho_j + \lambda/2}{z_j} & \text{if } \rho_j < -\lambda/2 \\ 0 & \text{if } |\rho_j| \leq \lambda/2 \\ \frac{\rho_j - \lambda/2}{z_j} & \text{if } \rho_j > \lambda/2 \end{cases}$$

# Soft-Thresholding Properties

## Key Points: Key Properties

- **Shrinkage:** Coefficients pulled toward zero
- **Selection:** Small coefficients  $\rightarrow$  exactly zero
- **Soft-thresholding:** Smooth shrinkage + selection

## Example: Intuition

Weak correlation  $|\rho_j| \leq \lambda/2 \Rightarrow$  eliminate feature!

# Lasso vs Ridge Comparison

## Lasso vs Ridge: Key Differences

Property	Ridge (L2)	Lasso (L1)
Penalty	$\sum \theta_j^2$	$\sum  \theta_j $
Sparsity	Never exactly zero	Can be exactly zero
Feature Selection	No	Yes
Differentiable	Yes	No (at $\theta_j = 0$ )
Solution Method	Closed form	Coordinate descent
Constraint Shape	Circle	Diamond
Best for	Multicollinearity	Feature selection

# When to Use Lasso vs Ridge

## Key Points: Use Lasso When:

- High-dimensional data ( $p \gg n$ )
- Need interpretable model
- Expect only few features are truly relevant
- Want automatic feature selection

# When to Use Lasso vs Ridge

## Key Points: Use Lasso When:

- High-dimensional data ( $p \gg n$ )
- Need interpretable model
- Expect only few features are truly relevant
- Want automatic feature selection

## Key Points: Use Ridge When:

- All features might be somewhat relevant
- Multicollinearity is the main problem
- Want to keep all features with reduced impact
- Need stable solution with correlated features



# Summary and Applications

# Lasso Regression: Summary

## Definition: Lasso in a Nutshell

Lasso = Linear regression + L1 penalty for automatic feature selection

## Key Points: Key Advantages

- Regression + feature selection simultaneously
- Sparse, interpretable models
- Handles high-dimensional data well

# Lasso: Limitations and Applications

## Key Points: Limitations

- Arbitrary selection among correlated features
- May underperform when all features are relevant

## Example: Applications

Genomics, text mining, signal processing, finance, marketing analytics