Iterative Soft Thresholding for LASSO (ISTA)

Oscar Ma, Max Maejima, Alan Zhong

What is LASSO?

- LASSO stands for Least Absolute Shrinkage and Selection Operator
- Modified version of the Least Squares Regression Model
  - LASSO is used for regression problems similar to ordinary least squares but uses additional parameters
  - Lambda is the regularization parameter
  - L1 Norm of Beta is the sum of the absolute value of the Beta coefficients

Why do we use LASSO?

- LASSO is a modified version of the Least Squares Regression Model
- Linear Regression with $\{X_1,\ldots X_n\}$ where n is large often overfits the data
- In real world examples, some X_1's are often statistically irrelevant
- LASSO's optimization problem emphasizes the most important independent variables in our analysis, attempting to avoid large coefficients when possible

LASSO Optimization Formula

$$\hat{\beta} = \arg\min_{\beta} \left\{ ||y - X\beta||_2^2 + \lambda ||\beta||_1 \right\}$$

- **Similar to Multiple Linear Regression Optimization,**

  Our Beta Vector is the coefficients $\{\beta_0, \beta_1, \ldots, \beta_i\}$ for our variables $\{X_1, \ldots, X_i\}$

  β_0 is the intercept (It is the predicted value when all $X_i = 0$ for all i)

- Our y-vector contains the true y-values for the variable we are aiming to predict
- X is the matrix that contains the values of the X_i's that coincide with our actual Y_i values as rows
- **Unique to LASSO**, Lambda is a non-negative real number that controls the strength of LASSO's influence on our optimization problem.
  - When **Lambda = 0**, this is an ordinary Least Squares Regression problem
  - When **Lambda is large**, many coefficients in our Beta Vector are forced to 0, extracting the strongest of X_i predictor variables for regression

The Beta Vector

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j \, x_{ij}$$

- Elements of beta are coefficients of a linear polynomial to predict elements in vector y
- Each element of the Beta Vector β_i corresponds to predictor variable X_i

- One way to initialize β is the Zero Vector
  - Treating all coefficients equally
  - LASSO's goal is to drive some coefficients to 0 for variable selection; it makes sense to start iterating with the zero vector

Importance of the L1-Norm

$$||\beta||_1 = \sum_{j=1}^{p} |\beta_j|$$

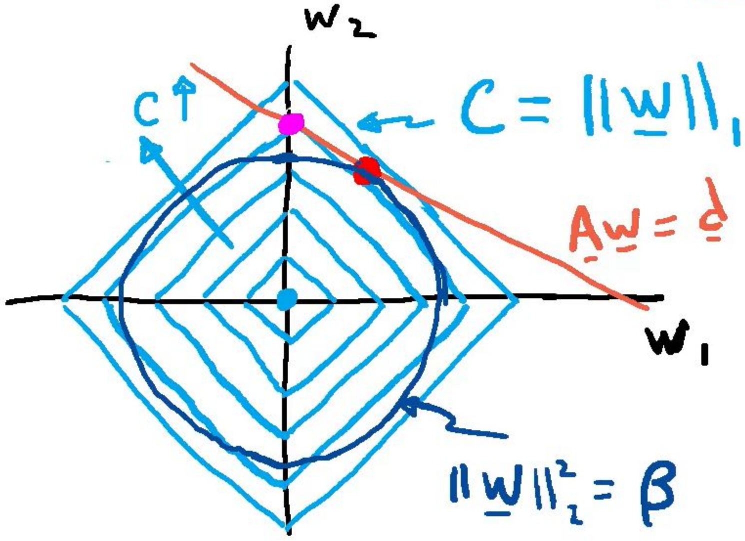
- L1-Norm of Beta - the **sum** of all the absolute coefficients of the Xi's
- This optimization problem requires **minimization**
    - Adding a factor of the L1-Norm to the objective function can drastically change the optimized Beta Vector
    - Compared to a Least Squares Minimization problem, LASSO prefers pushing β_j to 0 for less relevant variables X_j
    - This is known as the L1 Penalty

- But we have a PROBLEM!
    - The L1 Norm is not differentiable at 0 due to Absolute Values

Convex relaxation gives tractable problem

$\min_{\underline{w}} \|\underline{w}\|_1$ s.t. $\|\underline{A}\underline{w} - \underline{d}\|_2^2 < \varepsilon$ LASSO: Least Absolute Selection & Shrinkage Operator

convex



w_2
w_1
$c\uparrow$
$\underline{A}\underline{w} = \underline{d}$
$\|\underline{w}\|_2^2 = \beta$

$C = \|\underline{w}\|_1 = \sum_{i=1}^{M} |w_i|$: $|w_1| + |w_2| = c$

1st quad $w_1 + w_2 = c$

$\min \|\underline{w}\|_1$ s.t. $\underline{A}\underline{w} = \underline{d}$

"corners" on $\|\underline{w}\|_1 \Rightarrow$ sparse sol'ns

$\min \|\underline{w}\|_2^2$ s.t. $\underline{A}\underline{w} = \underline{d}$ circular $\|w\|_2^2 \Rightarrow$ non sparse solutions

Iterative Soft Thresholding for LASSO

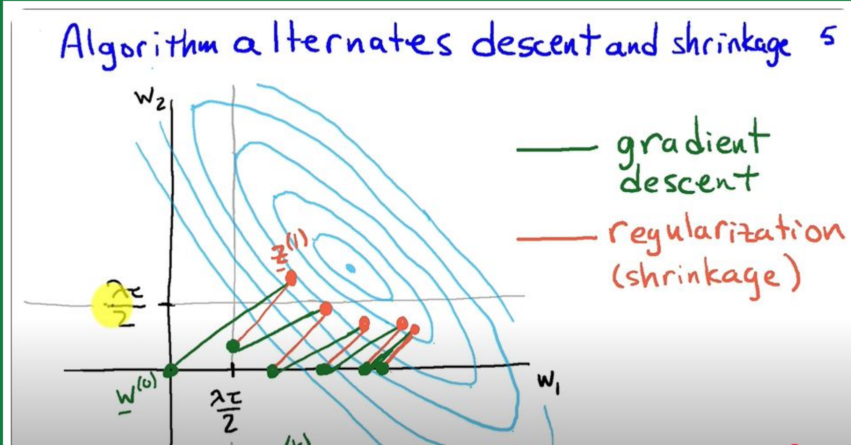
- To solve this L1-Norm differentiation issue, we use **Iterative Soft Thresholding**

Step 1: Gradient Descent

- Take the Gradient of the smooth Least Squares part of our objective function

Step 2: Soft Thresholding

- Shrinks smaller coefficient values toward 0
- Handle Non-Differentiability of the L1 Penalty



Gradient Descent

- First, split the objective function into two parts
  - A Smooth part, denoted f(β), which is our OLS objective function
  - A Non-Smooth part, denoted g(β)

$$f(\beta) = \|y - X\beta\|_2^2.$$

$$g(\beta) = \lambda\|\beta\|_1$$

- Then, find the gradient of f
  - Expanded, $f(\beta) = (y-X\beta)^T(y-X\beta) = y^Ty - 2\beta^TX^Ty + \beta^TX^TX\beta$
  - Taking the derivative with respect to β:
    - $d/d\beta\ (y^Ty) = 0$
    - $d/d\beta\ (2\beta^TX^Ty) = -2X^Ty$
    - $d/d\beta\ (\beta^TX^TX\beta) = 2X^TX^T\beta$

$$\nabla f(\beta) = 0 - 2X^\top y + 2X^\top X\beta = 2X^\top(X\beta - y).$$

Intermediate Step for Proximal Operator

$$\tilde{\beta}^{(k)} = \beta^{(k)} - \alpha \nabla f(\beta^{(k)}).$$

- $\beta^k \longrightarrow \beta$-tilde$^k \longrightarrow \beta^{k+1}$
  - Big Idea: Take a step in the direction of the gradient
  - High number of iterations $\longrightarrow$ use a small fixed alpha (we used 10^{-3})

- This is our **input vector** for the proximal operator

Proximal Operator Function

$$\beta^{(k+1)} = \mathrm{prox}_{\alpha_k g}\Big(\beta^{(k)} - \alpha_k \nabla f(\beta^{(k)})\Big).$$

- For LASSO, we choose Soft Thresholding as our Proximal Operator, hence **Iterative Soft Thresholding Algorithm (ISTA)**

- Soft Thresholding Algorithm

$$\beta^{(k+1)} = \mathrm{soft}\big(\beta^{(k)} - \alpha \nabla f(\beta^{(k)}),\ \alpha\lambda\big),$$

Where each element of

$$\beta_j^{(k+1)} = \mathrm{sign}\big(\tilde{\beta}_j^{(k)}\big) \max\Big(|\tilde{\beta}_j^{(k)}| - \alpha\lambda,\ 0\Big).$$

Soft Thresholding

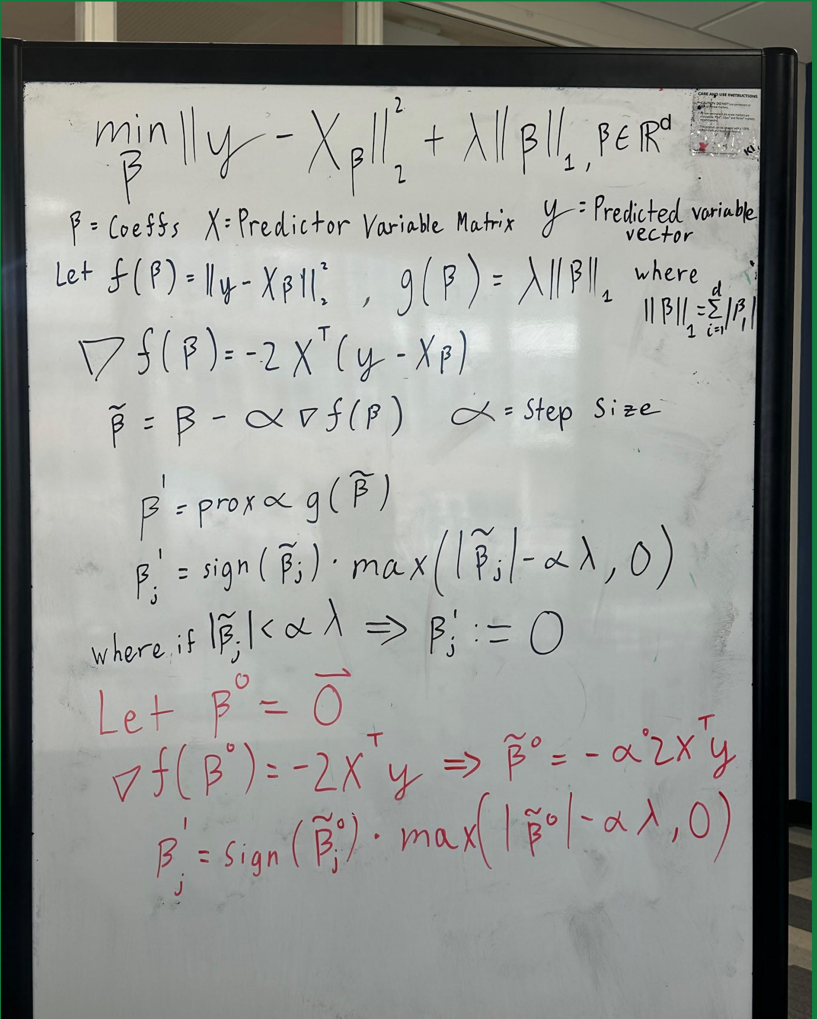
$$\beta_j^{(k+1)} = \operatorname{sign}(\tilde{\beta}_j^{(k)}) \max\left(|\tilde{\beta}_j^{(k)}| - \alpha\lambda, 0\right).$$

- The ISTA Algorithm:
  - Input: jth element of intermediate step β^k–tilde
  - Output: jth element of β^{k+1}

β^k_j–tilde $< -\alpha\lambda$ ---> $\beta_j^{k+1} := \beta^k_j$–tilde $+ \alpha\lambda$

$-\alpha\lambda < \beta^k_j$–tilde $< \alpha\lambda$ ---> $\beta^{k+1}_j := 0$

β^k_j–tilde $> \alpha\lambda \longrightarrow \beta^{k+1}_j := \beta^k_j$–tilde $- \alpha\lambda$



$$\min_{\beta} \|y - X\beta\|_2^2 + \lambda\|\beta\|_1, \ \beta \in \mathbb{R}^d$$

β = Coeffs X: Predictor Variable Matrix y: Predicted variable vector

Let $f(\beta) = \|y - X\beta\|_2^2$, $g(\beta) = \lambda\|\beta\|_1$ where $\|\beta\|_1 = \sum_{i=1}^{d}|\beta_i|$

$$\nabla f(\beta) = -2X^T(y - X\beta)$$

$\tilde{\beta} = \beta - \alpha\nabla f(\beta)$ α = Step Size

$$\beta' = \text{prox}_{\alpha g}(\tilde{\beta})$$

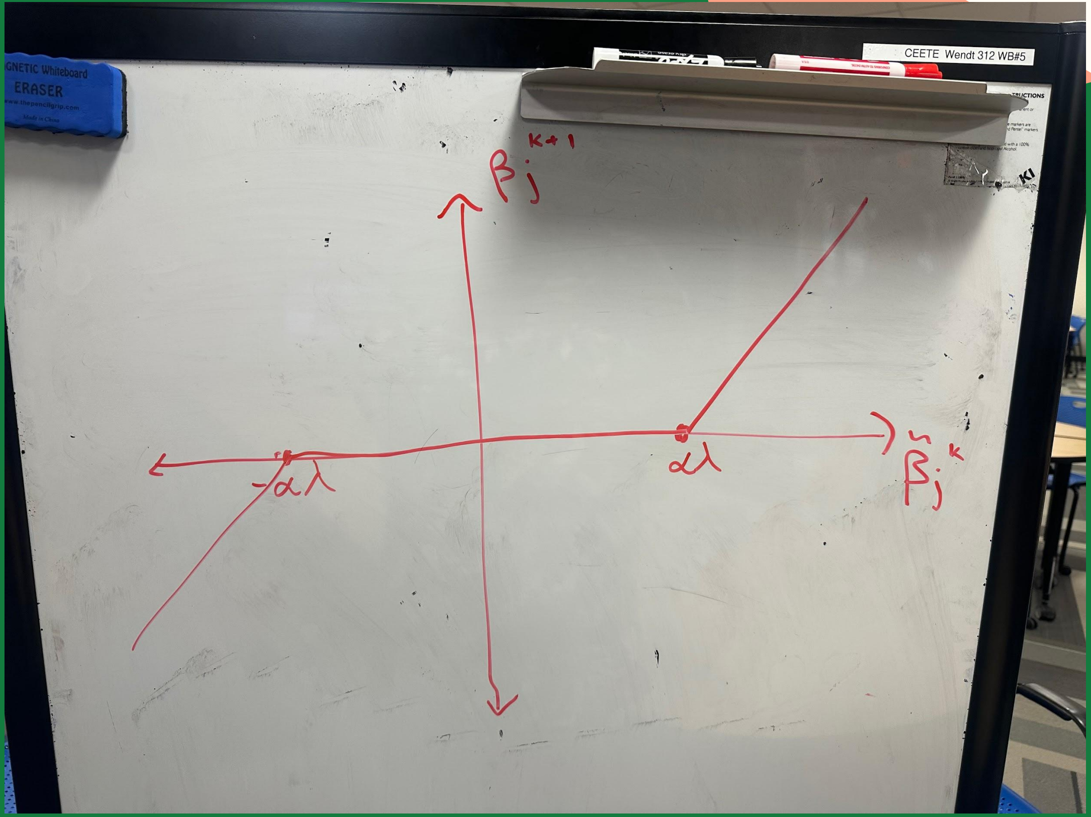
$$\beta_j' = \text{sign}(\tilde{\beta}_j)\cdot\max(|\tilde{\beta}_j| - \alpha\lambda, 0)$$

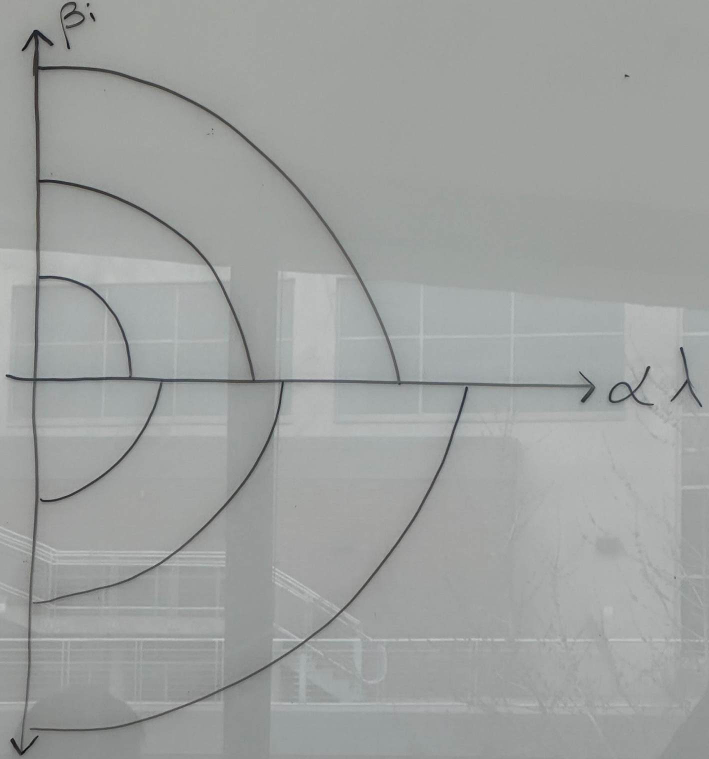
where if $|\tilde{\beta}_j| < \alpha\lambda \Rightarrow \beta_j' := 0$

Let $\beta^0 = \vec{0}$

$$\nabla f(\beta^0) = -2X^Ty \Rightarrow \tilde{\beta}^0 = -\alpha\cdot 2X^Ty$$

$$\beta_j^1 = \text{sign}(\tilde{\beta}_j^0)\cdot\max(|\tilde{\beta}^0| - \alpha\lambda, 0)$$




βi
$\alpha \lambda$

Implementation

We applied Lasso Regression using the Iterative Soft Thresholding Algorithm (ISTA) to identify key factors influencing career outcomes based on a dataset of student profiles.

```
def soft_thresholding(x, alpha,lambda_):
    """Applies soft-thresholding to shrink values toward
zero."""
    return np.sign(x) * np.maximum(np.abs(x) -
alpha*lambda_, 0)
```

Implementation Continued

```
def ista(X, y, lambda_, alpha=1e-3, max_iter=1000, tol=1e-6):
    m, n = X.shape
    beta = np.zeros(n)   # Initialize beta coefficients to zero

    for _ in range(max_iter):
        gradient = X.T @ (X @ beta - y) / m  # Compute the gradient of loss
        beta_new = soft_thresholding(beta - alpha * gradient, alpha, lambda_)

        if np.linalg.norm(beta_new - beta, ord= 2) < tol:
            break # Check if the update is small enough to consider convergence

        beta = beta_new

    return beta
```

Issue with Cross Validation for Choosing λ

A common way to choose parameters in machine learning is cross validation, a method of iterating over possible parameter values and selecting one that minimizes the squared error.

With LASSO we are balancing two priorities; a Beta with sparse weights and reducing squared error. We found using cross validation results in bias towards smallest λ possible. Thus in LASSO it is important to recognize minimizing error alone doesn't guarantee a sparse model and chosen λ may lead to overfitting or dense models, defeating the purpose of feature selection.

$$\hat{\beta} = \arg\min_{\beta} \left\{ ||y - X\beta||_2^2 + \underline{\lambda||\beta||_1} \right\}$$

Data

Our dataset includes academic, personal, and career information for recent graduates:

Results

Feature Importance (ISTA Lasso) for Job_Offers (lambda = 0.01):

Projects_Completed	0.012913
Age	0.009617
Internships_Completed	0.008134
SAT_Score	0.004453
University_GPA	0.003893
Gender_Other	0.003813
University_Ranking	0.003028
High_School_GPA	0.002633
Soft_Skills_Score	0.002628
Networking_Score	0.001614
Field_of_Study_Engineering	0.001511
Field_of_Study_Business	0.001395
Certifications	0.000381
Gender_Male	**0.000000**
Field_of_Study_Computer Science	**0.000000**
Field_of_Study_Law	**0.000000**
Field_of_Study_Mathematics	**0.000000**
Field_of_Study_Medicine	**0.000000**

Results

Feature Importance (ISTA Lasso) for Starting_Salary (lambda = 0.05):

Internships_Completed	0.013612
Field_of_Study_Law	0.009995
University_Ranking	0.009057
Certifications	0.008917
Field_of_Study_Medicine	0.008152
Field_of_Study_Business	0.003515
Soft_Skills_Score	0.002962
Gender_Male	0.002493
High_School_GPA	0.001999
University_GPA	0.001986
Projects_Completed	0.000467
Field_of_Study_Computer Science	0.000261
Age	**0.000000**
SAT_Score	**0.000000**
Networking_Score	**0.000000**
Gender_Other	**0.000000**
Field_of_Study_Engineering	**0.000000**
Field_of_Study_Mathematics	**0.000000**

References

1. Barry Van Veen. (n.d.). Solving l1 Regularized Least Squares via Proximal Gradient Descent. YouTube. https://www.youtube.com/watch?v=CBMDwMESuWs&list=PLGI7M8vwfrFMo6i-kmYROMN-GUdKNHGLv&index=4
2. Murphy, K. P. (2012). *Machine learning : a probabilistic perspective*. Cambridge, MA: MIT. ISBN: 9780262018029 0262018020
3. Professor Brian Powers. (4/10/2024). UW Madison Department of Statistics. brpowers2@wisc.edu
4. ChatGPT