LARGE-SCALE COORDINATE DESCENT: FROM CONVEX TO NON-CONVEX

Chuxuan He, Haipeng Zhao, Haoyi Wang, Kevin Ji

1. INTRODUCTION

Sparse regression has become highly valuable across a wide range of applications due to its enhanced model interpretability and ability to improve efficiency by selecting only the most relevant features. Among the optimization techniques used in sparse regression, coordinate descent and block coordinate descent is particularly effective in methods like LASSO and SCAD, which promote sparsity. This technique is highly regarded for its computational efficiency, scalability, reliable convergence, and ability to handle non-differentiable functions.

In this report, we explore the application of coordinate descent and block coordinate descent in the contexts of LASSO and SCAD regression, two widely-used approaches for achieving sparse solutions. We implement all methods and analyzing their performance and outcomes through experiments.

2. BACKGROUND

2.1. Sparse Regression

Sparse regression is a statistical approach that focuses on producing models with only a subset of the most relevant features, resulting in interpretable and efficient solutions. By encouraging sparsity—where many feature coefficients are zero or close to zero—sparse regression reduces complexity. The main idea of enhancing sparsity is based on L_0 regularization, which aims to control the number of non-zero elements in the model parameters to achieve sparsity. However, directly solving the L_0 regularization problem is NP-hard. Therefore, the problem is usually transformed to the L_1 regularization[1] problem like LASSO.

2.2. Convex Approach: LASSO

Lasso is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model [2]. Lasso adds a penalty term, called the L1 regularization term equal to the absolute value of the magnitude of coefficients to the loss function, which helps to avoid overfitting by shrinking the coefficients of less important features toward zero:

$$\mathcal{L}(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(y_i - \sum_{j=0}^{n}\theta_j x_{ij}\right)^2 + \lambda\sum_{j=0}^{n}|\theta_j|$$

where λ is the regularization strength parameter, which controls the sparsity of the model.

Figure.1 presents the objective function of LASSO regression. We can see that from Figure.1, the LASSO function has a very obvious L1 regularization characteristic: The contour lines take on a ”rhombus” shape and the encouragement coefficient is close to zero.

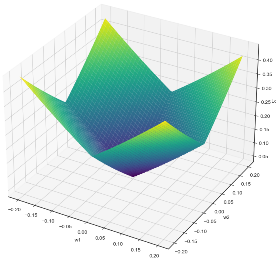

Fig. 1: LASSO Loss Surface

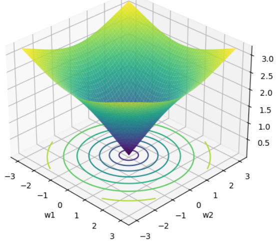

Fig. 2: SCAD Loss Surface

2.3. Non-convex Approach: SCAD

SCAD (Smoothly Clipped Absolute Deviation) is a regression analysis method that performs variable selection and regularization while addressing the bias limitation of LASSO [3]. SCAD introduces a novel penalty term that smoothly transitions from high to low penalization as coefficient magnitudes increase, which helps produce nearly unbiased estimates for large coefficients while maintaining sparsity for small ones. The SCAD-penalized loss function is defined as:

$$\mathcal{L}(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(y_i - \sum_{j=0}^{n}\theta_j x_{ij}\right)^2 + \sum_{j=0}^{n}p_\phi(|\theta_j|)$$

where the SCAD penalty $p_\phi(\theta)$ is defined by its derivative:

$$p'_\phi(\theta) = \phi\left\{I(\theta \leq \phi) + \frac{(a\phi - \theta)_+}{(a-1)\phi}I(\theta > \phi)\right\}, \quad \theta > 0$$

So $p_\phi(\theta)$ is given by:

$$p_\phi(\theta) = \begin{cases} \phi|\theta| & \text{when } |\theta| \leq \phi \\ -\frac{\theta^2 - 2a\phi|\theta| + \phi^2}{2(a-1)} & \text{when } \phi < |\theta| \leq a\phi \\ \frac{(a+1)\phi^2}{2} & \text{when } |\theta| > a\phi \end{cases}$$

where ϕ is the regularization parameter controlling the sparsity of the model, and $a > 2$ is a shape parameter which achieves three key properties: unbiased for large coefficients, sparsity for zero coefficients, and continuity of the estimation procedure.

Figure.2 presents the objective function of SCAD penalty. It can be obviously seen that from Figure.2, the regression with SCAD penalty is a non-convex function.

3. METHODOLOGY

3.1. Coordinate Descent

Coordinate descent is an optimization algorithm used to minimize multivariate functions by iteratively solving simpler univariate problems. At each iteration, the algorithm selects a single coordinate direction and minimizes the objective function along that direction, while keeping all other coordinates fixed. This process repeats cyclically for each coordinate until convergence. Given an initial point $\theta^{(0)} = (\theta_1^0, \theta_2^0, \ldots, \theta_n^0)$, the algorithm updates each coordinate by solving:

$$\theta_i^{(k+1)} = \arg\min_{\theta_i} f(\theta_1^{(k+1)}, \ldots, \theta_{i-1}^{(k+1)}, \theta_i, \theta_{i+1}^{(k)}, \ldots, \theta_n^{(k)})$$

This method is particularly efficient for high-dimensional problems, as each iteration focuses on a simpler, one-dimensional subproblem.

3.2. Coordinate Descent For LASSO

We first defines the **soft thresholding function**, $S(\alpha, \epsilon)$, which is widely used in Lasso regression for variable selection:

$$S(\alpha, \epsilon) = \begin{cases} \alpha - \epsilon & \text{if } \alpha > \epsilon \\ 0 & \text{if } |\alpha| \leq \epsilon \\ \alpha + \epsilon & \text{if } \alpha < -\epsilon \end{cases}$$

For easier computation, we transform the soft shresholding function into the following form:

$$S(\alpha, \epsilon) = \text{sign}(\alpha) \max(|\alpha| - \epsilon, 0)$$

The soft thresholding operator shrinks the coefficient α by ϵ, setting it to zero if it falls below the threshold.

Then, we define the computing rule for ρ_j,

$$\rho_j = \sum_{i=1}^{m} x_j^i \left(y_i - \sum_{k \neq j}^{n} \theta_k x_k^i \right),$$

is the **partial residual sum** used in coordinate descent for Lasso regression. Here, ρ_j is computed for each feature x_j, accounting for the current values of the other coefficients θ_k (for $k \neq j$).

Finally, we rewrite the residual sum by isolating the contribution of x_j to the prediction:

$$\rho_j = \sum_{i=1}^{m} x_j^i \left(y_i - \hat{y}_{\text{pred}}^i + \theta_j x_j^i \right),$$

where $\hat{y}_{\text{pred}}^i$ is the predicted value excluding the contribution from the current feature x_j. This reformulation highlights the effect of updating θ_j on the residual. Algorithm 1 shows the pseudocode for lasso coordinate descent.

For our regression problem, we used X as the input matrix and y as the output vector. Additionally, θ represent the coefficient matrix and ϵ represents the input variable of soft thresholding function.

Figure.3 illustrates the contour plot of this objective function alongside the variable changes throughout the optimization.

Algorithm 1 Coordinate Descent For LASSO

```
Input: $\theta$, $X$, $y$, $\epsilon$
Initialize $\theta$
repeat
    For each feature j in X
        $y_{\text{pred}} \leftarrow X \cdot \theta$
        $\rho_j \leftarrow \mathbf{X}_{:,j}^\top (\mathbf{y} - \hat{\mathbf{y}}_{\text{pred}} + \theta_j \mathbf{X}_{:,j})$
        $\theta_j \leftarrow S(\rho_j, \epsilon)$
    return updated $\theta$
until Stopping criterion is satisfied
```

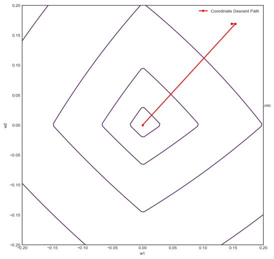

Fig. 3: Coordinate Descent for LASSO

3.3. Coordinate Descent For SCAD

The first equation defines the thresholding function for SCAD regularization. Given the correlation ρ, regularization parameter ϕ, and shape parameter a, the thresholding function can be expressed as:

$$S(\rho, \phi, a) = \begin{cases} \text{sign}(\rho) \cdot (|\rho| - \phi) & \text{if } |\rho| \leq \phi \\ \text{sign}(\rho) \cdot \frac{(a-1) \cdot |\rho| - a\phi}{a-1} & \text{if } \phi < |\rho| \leq a\phi \\ \rho & \text{if } |\rho| > a\phi \end{cases}$$

This thresholding function reflects SCAD's three-phase regularization strategy: strong penalization (linear) for small coefficients (like LASSO), gradually reduced (quadratic) penalization for moderate coefficients, and no penalization for large coefficients.

The second equation computes the correlation ρ between feature j and the current residuals:

$$\rho = \mathbf{X}_{:,j}^\top (\mathbf{y} - \mathbf{X} \cdot \theta + \mathbf{X}_{:,j} \cdot \theta_j)$$

where $\mathbf{X}_{:,j}$ represents the j-th feature column, θ is the current weight vector, and θ_j is the j-th coefficient. This correlation term helps determine the appropriate update for each coefficient.

Finally, the coefficient update is normalized by the diagonal element of the Gram matrix:

$$\theta_j \leftarrow \frac{S(\rho, \phi, a)}{X_{:,j}^\top X_{:,j}}$$

This normalization accounts for the scale of the features and ensures stable updates. Algorithm 2 shows the pseudocode for SCAD coordinate descent.

For our regression problem, we used $\mathbf{X}$ as the input matrix and $\mathbf{y}$ as the output vector. Additionally, θ represents the weight vector, ϕ is the regularization parameter, and $a > 2$ is the SCAD shape parameter that controls the transition between different penalization phases.

Algorithm 2 Coordinate Descent with SCAD Regularization

```
Input: $X, y, \theta, \phi, a$
Initialize $\theta$
repeat
    For each feature j in X
        $\mathbf{r} \leftarrow \mathbf{y} - \mathbf{X} \cdot \theta + \mathbf{X}_{:,j} \cdot \theta_j$
        $\rho \leftarrow \mathbf{X}_{:,j}^\top \mathbf{r}$
        $\theta_j \leftarrow \frac{S(\rho,\phi,a)}{X_{:,j}^\top \cdot X_{:,j}}$
    return updated $\theta$
until Stopping criterion is satisfied
```

3.4. Block Coordinate Descent Method

Block coordinate descent [4] is an optimization paradigm that iteratively updates one block of variables at a time, making it quite amenable to big data applications due to its scalability and performance.

Given an initial point $\theta^0 = (\theta_1^0, \ldots, \theta_N^0)$. We randomly divide these coefficients into s blocks (θ_1,...,θ_s) and the Block Coordinate Descent method updates one block θ_i in each iteration. solving:

$$\theta_i^k = \arg\min_{\theta_i} \|y - X \cdot \theta^k\|_2^2, \; k \geq 0$$

In the Block Coordinate Descent (BCD) method, blocks are randomly chosen to improve convergence speed and algorithmic efficiency. When blocks are updated sequentially, the algorithm may get stuck in inefficient update patterns under the non-convex problem, particularly if the blocks are highly correlated. Randomly selecting blocks helps the algorithm avoid poor local optima by introducing diversity in the update directions. This randomness also increases the likelihood of reaching the global minimum in non-convex problems.

Algorithm 3 Block Coordinate Descent Algorithm

```
Input: $X, y$, B
Initialization: B
repeat for $k = 1, 2, \ldots,$ N
    for $i = 1, 2, \ldots, s$ do
        Update block $\theta_i^k$ while keeping all other blocks fixed:
            $$\theta_i^k \leftarrow \arg\min_{\theta_i} \|y - X \cdot \theta^k\|_2^2$$
    end for
until Stopping criterion is satisfied
Output: Optimized Block $(\theta_1^k, \theta_2^k, \ldots, \theta_s^k)$
```

For our regression problem, we used X as the input matrix and y as the output vector. Additionally, B is a matrix containing all the block vectors θ_i. Each block vector θ_i consists of m coefficients, where m is the length of the block.

4. EXPERIMENT

With the coordinate descent methods discussed in Part 3, we implemented two versions to test our methods. The first version applies coordinate descent on the Diabetes Dataset to optimize a parameter vector θ, aiming to evaluate the fundamental effectiveness of the coordinate descent algorithm. The second version utilizes block coordinate descent (BCD) on the Boston Housing Dataset, where parameter matrices are updated by vectors instead of elements. This version tests both the efficacy of BCD and the importance of introducing randomness in block selection during optimization.

4.1. Coordinate Descent

We firstly test our coordinate method in Part 3.1 with the Diabetes Dataset[5], a classic dataset used for regression tasks, which contains data on 442 patients. This dataset includes 10 input features, such as age, sex, body mass index, average blood pressure, and six blood serum measurements (e.g., cholesterol and glucose levels). These features are used to predict a single dimensional output: a quantitative measure of diabetes progression, indicating the severity of the condition.

In this experiment, we set both of the penalty hyperparameter λ from LASSO and ϕ from SCAD to 0.9, and the hyperparameter α for SCAD is set to 6. Figure.4 and Figure.5 present the change of objective function value of LASSO and SCAD regression with coordinate descent method. The LASSO loss function demonstrates rapid convergence when optimized using coordinate descent. By iteratively updating each parameter, the algorithm achieves a sharp decrease in loss within the first few iterations. Around iteration 20, the loss stabilizes at approximately 0.338. For SCAD, the convergence pattern is less regular compared to LASSO. The use of a non-convex penalty introduces additional complexity in the optimization, resulting in fluctuations in the early iterations. However, by iteration 10, the loss stabilizes at approximately 0.340. The coordinate descent approach handles SCAD's piecewise penalty effectively. These results indicate that coordinate descent method is well-suited for both convex sparse regression LASSO and non-convex regression SCAD.

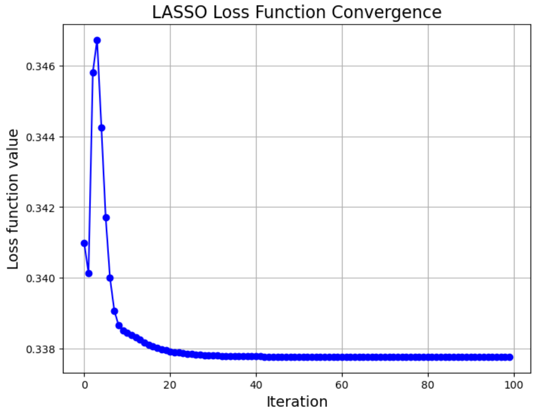

Fig. 4: Coordinate Descent for LASSO

4.2. Block Coordinate Descent

We tested our block coordinate method in Part 3.4 on both LASSO and SCAD penalty on the Boston Housing Dataset[6], a dataset that comprises 506 samples of housing data from Boston Metropolitan Area census tracts, featuring 13 predictor variables including crime rates, structural characteristics, and socio-economic indicators. We tested BCD with this dataset because BCD is usually used for large scale optimization problems, this dataset is larger than Diabetes Dataset with more samples and larger feature dimensionality.

We set all the hyperparameters in LASSO and SCAD as same as the experiments in coordinate descent. And map the 1 dimensional output to a 5 dimensional output matrix, which allows us to update a

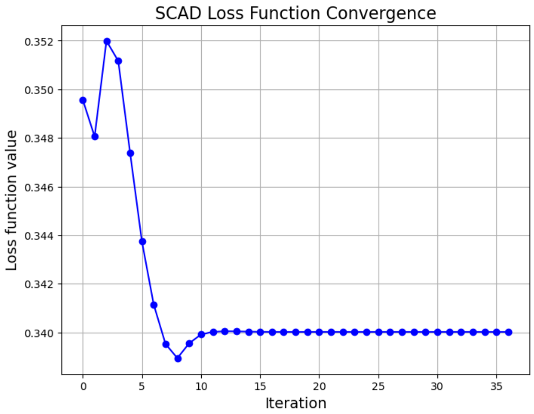

Fig. 5: Coordinate Descent for LASSO

real vector in each iteration. Figure.6 shows the loss function value of LASSO during the BCD process. The red line showed the results of sequentially update blocks. Each of the rest of the lines (blue, orange, green, and yellow curves) showed the results that updating blocks with a random order. It can observed that for LASSO, regardless of the update order, all methods eventually converge to the same minimum value. This is because LASSO is convex, ensuring that any update sequence will ultimately lead to the global minimum.

Figure.7 shows the loss function value of SCAD during the BCD process. The red line still showed the results of sequentially update blocks. The rest of the lines showed the results with different random orders of updating blocks. Unlike LASSO, in SCAD optimization, different update orders lead to convergence at different minimum values. It can be observed that for some randomly updated blocks, the function value converged to the same optimal value as the sequential update. However, in some cases, it converged to a better value than the sequential update. This aligns with our description in the BCD section: since SCAD is a non-convex function, using sequential updates in coordinate descent can cause the optimization to get stuck in local minima. Introducing randomness helps mitigate this issue, allowing the function to potentially reach the global optimum.

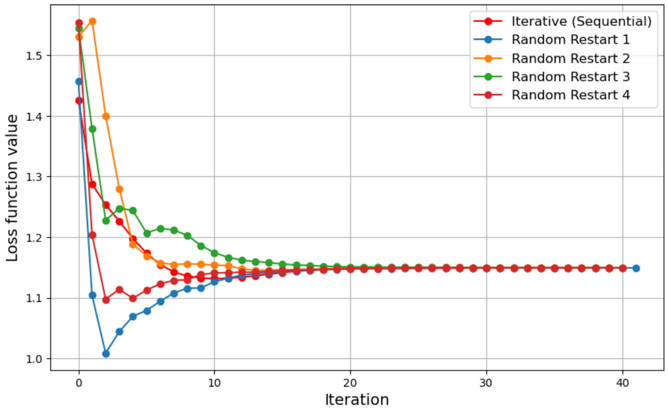

Fig. 6: Block Coordinate Descent for LASSO

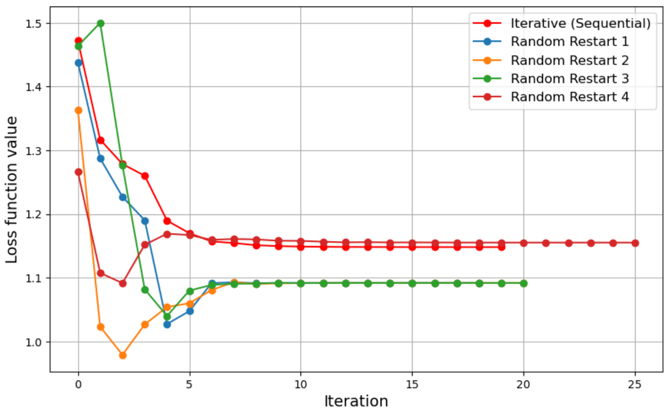

Fig. 7: Block Coordinate Descent for SCAD

5. FUTURE WORKS

In the future, we aim to explore sparse regression models with gradient properties by using non-$\mathcal{L}_1$ regularization techniques such as $\mathcal{L}_2$ regularization. The limitation of $\mathcal{L}_1$ regularization is within its non-differentiability, which makes it difficult to apply gradient-based optimization methods when we are updating blocks in BCD. This often results in suboptimal solutions in some cases. By focusing on alternative regularization methods, such as non-convex approaches like MCP (Minimax Concave Penalty), we plan to overcome these challenges. These techniques offer smooth properties that allow for gradient computation, which enables more efficient optimization and improving the overall performance of sparse regression models, especially in high-dimensional data scenarios.

6. REFERENCES

[1] Tom Jacobs and Rebekka Burkholz, “Mask in the mirror: Implicit sparsification,” *arXiv preprint arXiv:2408.09966*, 2024.

[2] Robert Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 58, no. 1, pp. 267–288, 1996.

[3] Jianqing Fan and Runze Li, “Variable selection via nonconcave penalized likelihood and its oracle properties,” *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[4] Liangzu Peng and René Vidal, “Block coordinate descent on smooth manifolds: Convergence theory and twenty-one examples,” *arXiv preprint arXiv:2305.14744*, 2023.

[5] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani, “Least angle regression,” 2004.

[6] David Harrison Jr and Daniel L Rubinfeld, “Hedonic housing prices and the demand for clean air,” *Journal of environmental economics and management*, vol. 5, no. 1, pp. 81–102, 1978.