Advanced Machine Learning — Project 1

Cyprien Fourcroy, Paulina Kulczyk, Michał Puścian

March 2025

Contents

1 Introduction

Logistic regression is a widely used statistical method for binary classification, particularly in high-dimensional data scenarios. It is a fundamental technique applied across various domains, including bioinformatics, economics, and social sciences. However, when the number of features significantly exceeds the number of observations, logistic regression is prone to overfitting, which diminishes the model's generalizability and interpretability. To address this issue, regularization techniques such as l_1 (Lasso) penalty and l_2 (Ridge) penalty are commonly employed. These techniques shrink the model coefficients, helping to prevent overfitting and improve predictive accuracy [1].

Elastic Net regularization, a hybrid approach combining Lasso and Ridge penalties, has gained popularity due to its ability to handle correlated features and promote both sparsity and group selection [2]. Elastic Net is particularly useful in high-dimensional datasets, where the relationships among features are often complex, and some features may exhibit multicollinearity. While Lasso tends to select individual features, Elastic Net is more suited for identifying and grouping correlated features, thus improving model performance in many real-world applications. In practice, however, choosing the optimal regularization method and corresponding parameters can be challenging, particularly when the number of features is large or when data is limited.

In response to these challenges, we introduce **LogRegCCD** developed by Jerome H. Friedman, Trevor Hastie and Rob Tibshirani [3]. It is an enhanced logistic regression algorithm designed to efficiently operate in high-dimensional spaces while incorporating Elastic Net regularization. LogRegCCD leverages Cyclic Coordinate Descent (CCD), a computationally efficient optimization strategy that iteratively updates model coefficients to converge to the optimal solution. By combining the advantages of Elastic Net with the efficiency of CCD, LogRegCCD not only balances sparsity and feature group selection but also provides an effective solution even in cases of high-dimensional and imbalanced datasets.

To assess the effectiveness of LogRegCCD, we evaluate its performance across several synthetic and well-known benchmark datasets, examining its strengths compared to standard logistic regression. We investigate the influence of different parameters, including sample size, class imbalance, feature dimensionality, and feature correlation, on the algorithm's ability to generalize and make accurate predictions.

2 Methodology

The LogRegCCD algorithm is designed to perform logistic regression with elastic net regularization, effectively balancing l (lasso) and l (ridge) penalties to promote both sparsity and group selection among features (we can choose only lasso or ridge). The implementation leverages the cyclic coordinate descent method, known for its computational efficiency in high-dimensional settings [4]. This section details the methodological approach underlying the algorithm's implementation, including data preprocessing, regularization path computation and optimization strategy. Here, we also describe the datasets used in our experiments to evaluate the algorithm's performance.

2.1 Algorithm implementation

2.1.1 Data Preprocessing

Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ denote the feature matrix and $\mathbf{y} \in \{0, 1\}^n$ the binary response variable. Before optimization, feature standardization is applied by centering each column of $\mathbf{X}$ to have zero mean. This ensures numerical stability and facilitates regularization.

2.1.2 Regularization and Optimization Strategy

To prevent overfitting and enhance model generalization, we fit the model by minimizing the negative log-likelihood with Elastic Net regularization, which incorporates both l_1 (Lasso) and l_2 (Ridge) penalties:

$$\min_{\beta \in \mathbb{R}^p} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \left[y_i \cdot (x_i^\top \beta) - \log(1 + e^{x_i^\top \beta}) \right] + \lambda \left[\alpha \|\beta\|_1 + \frac{1-\alpha}{2} \|\beta\|_2^2 \right] \right\},$$

where λ controls the overall penalty strength, and α adjusts the balance between l_1 and l_2 regularization.

This objective is approximated at each iteration using the IRLS approach, where we solve the weighted least-squares problem:

$$\ell_Q(\beta) = \frac{1}{2n} \sum_{i=1}^{n} w_i (z_i - x_i^\top \beta)^2$$

with

$$w_i = p_i(1 - p_i), \tag{1}$$

$$z_i = x_i^\top \beta + \frac{y_i - p_i}{w_i}, \tag{2}$$

$$p_i = \frac{1}{1 + e^{-x_i^\top \beta}} \tag{3}$$

The optimization is performed using cyclic coordinate descent (CCD), which iteratively updates each coefficient β_j while holding others fixed.

$$\beta_j \leftarrow \frac{S\left(\sum_{i=1}^{n} w_i x_{ij} r_i^{(j)},\ \lambda\alpha\right)}{\sum_{i=1}^{n} w_i x_{ij}^2 + \lambda(1 - \alpha)}$$

where $r_i^{(j)} = z_i - \sum_{k \neq j} x_{ik}\beta_k$, and $S(z, \gamma) = \text{sign}(z) \cdot \max(|z| - \gamma, 0)$ is the soft-thresholding operator. We define a sequence of regularization parameter λ over the itertions:

$$\lambda_{\max} = \frac{1}{n\alpha} \max_j \left| x_j^\top (y - \bar{y}) \right|, \tag{4}$$

$$\lambda_{\min} = \lambda_{\max} \cdot \text{lambda\_min\_ratio} \tag{5}$$

This ensures that for $\lambda = \lambda_{\max}$, all coefficients are zero. The algorithm computes the regularization path by warm-starting the solution at each λ_k from the previous λ_{k-1}. Model selection is performed by validating on a held-out set to determine the λ that minimizes a chosen metric such as accuracy or log-loss.

2.2 Datasets generation & selection

For the purpose of our work we generate variety of datasets to compare the algorithm behavior of them. Generated datasets have binary target variables $\mathbf{y}$ sampled from Bernoulli(p). The feature vectors $\mathbf{X}$ are drawn from a d-dimensional multivariate normal distribution. We assume:

- when $\mathbf{y} = 0, X$ follows $N(0, S)$ where $S[i, j] = g^|i - j|$;
- when $\mathbf{y} = 1, X$ follows $N((1, \frac{1}{2}, \frac{1}{3}, ..., \frac{1}{d}), S)$ where $S[i, j] = g^|i - j|$.

To make datasets diverse we use different parameters: p (float) - probability of $Y = 1$ in Bernoulli distribution, n (int) - number of observations, d (int) - dimensionality of the feature space, g (float) - parameter controlling the covariance matrix structure. We take this argument into consideration cause it allow us to evaluate our algorithm in case of different inputs. Table 1 shows the key consideration about parameters.

During our study we also use real datasets to evaluate performance of our algorithm on realistic event. We choose:

- **Spambase** ($n = 4,601, d = 57$) [5]: Moderately imbalanced (39% spam), with high feature correlation, testing LogRegCCD's ability to handle multicollinearity;
- **Wine Quality** ($n = 6,497, d = 11$) [6]: Imbalanced across quality levels, converted into a binary classification problem. Tests generalization with low-to-moderate feature correlation;
- **Breast Cancer** ($n = 569, d = 30$) [7]: Nearly balanced dataset, useful for evaluating small sample size effects and feature dependencies in medical data;
- **Ionosphere** ($n = 351, d = 34$) [8]: Moderately imbalanced (64% "good" class) with strong feature correlations, assessing model robustness under multicollinearity

datasets that are available on Kaggle. By testing LogRegCCD across these datasets, we can compare its performance with standard logistic regression under different data conditions and gain insights into its strengths and weaknesses. Moreover, they are re well-known in machine learning and enable easy comparison with other studies evaluating similar algorithms.

Parameter	Values	Purpose
Sample size (n)	150, 1500, 15000	To evaluate the impact of dataset size on model performance, convergence speed, and generalization. Small n may lead to overfitting, but also show algorithm performance in limited data scenarios (as in rare diseases). Medium n reflects realistic settings, while large n allows better parameter estimation and also tests scalability.
Class imbalance (p)	0.1, 0.3, 0.5, 0.7, 0.9	To analyze how class imbalance affects logistic regression and LogRegCCD performance. Balanced ($p = 0.5$) vs. imbalanced cases ($p \neq 0.5$) help understand robustness to skewed distributions.
Feature dimensionality (d)	5, 25, 125	To study the effect of increasing feature space on logistic regression. Larger d may increase overfitting risk and computational cost.
Feature correlation (g)	0.05, 0.1, 0.4, 0.5, 0.6, 0.9, 0.95	Controls the correlation structure of features. Low g means nearly independent features, while high g introduces multicollinearity, testing the ability of methods to handle correlated variables.

Table 1: Synthetic datasets parameters and their purpose

3 Discussion about correctness of the LogRegCDD algorithm

We analyze the correctness and performance of the `LogRegCDD` algorithm by evaluating it across three different regularization settings: $\alpha = 1.0$ (L1), $\alpha = 0.5$ (Elastic Net), and $\alpha = 0.0$ (L2). For each configuration, we compare:

- the validation accuracy over the regularization path (i.e., across different λ),
- the coefficient trajectories learned across λ (coefficient path),
- the selected λ value and final test accuracy versus scikit-learn's implementation.

We have chosen breast cancer dataset for the comparaison.

Case 1: L1 Regularization ($\alpha = 1.0$) 3

Validation accuracy: The performance steadily improves with decreasing λ, peaking at 96.5% before slightly dropping.

Coefficient behavior: The learned coefficients are sparse and stabilize as λ decreases.

Comparison:

- **Best λ (LogRegCDD):** 0.00111
- **Best λ (Scikit-learn):** 0.829
- **Test accuracy (both):** 96.5%

Case 2: Elastic Net ($\alpha = 0.5$) 6

Validation accuracy: Accuracy peaks early at 95.6% and remains stable over a wide range of λ values.

Coefficient behavior: Coefficients show smoother paths compared to $\alpha = 1.0$, with more features staying non-zero.

Comparison:

- **Best λ (LogRegCDD):** 0.00016
- **Best λ (Scikit-learn):** 0.391
- **Test accuracy (both):** 96.5%

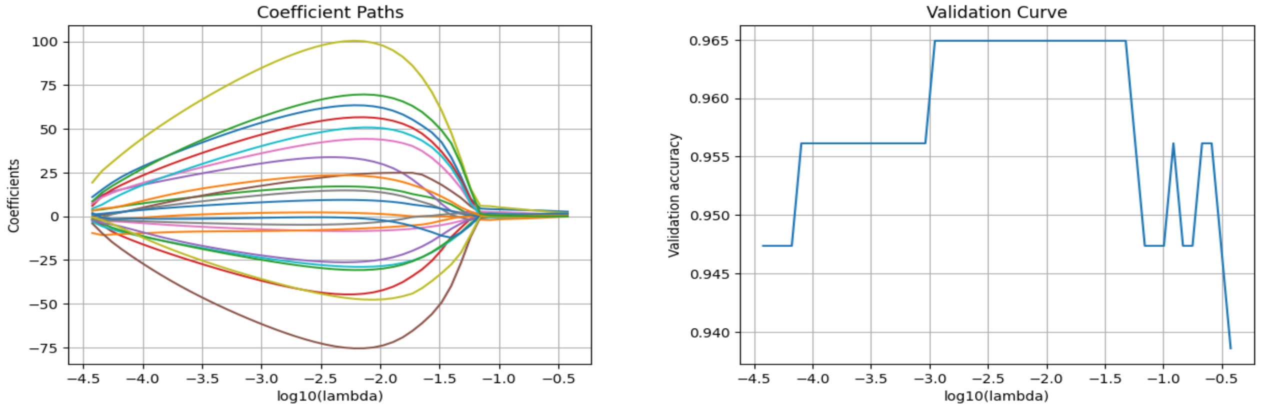

Figure 1: Validation accuracy across λ for LogRegCDD with $\alpha = 1.0$

Figure 2: Coefficient paths for LogRegCDD with $\alpha = 1.0$

Figure 3: $= 1$

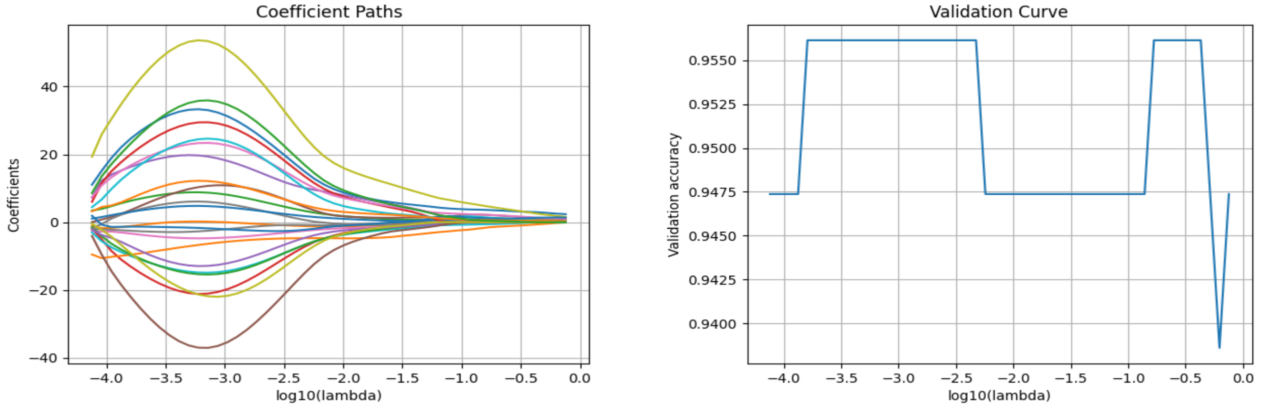

Figure 4: Validation accuracy across λ for LogRegCDD with $\alpha = 0.5$

Figure 5: Coefficient paths for LogRegCDD with $\alpha = 0.5$

Figure 6: $= 0.5$

Case 3: L2 Regularization ($\alpha = 0.0$) 9

Validation accuracy: Accuracy increases with lower λ values and reaches a maximum of 96.5%, similar to the previous settings.

Coefficient behavior: As expected, no coefficients are shrunk to zero. The regularization is smooth and all weights evolve gradually.

Comparison:

- **Best λ (LogRegCDD):** 0.478
- **Best λ (Scikit-learn):** 0.569
- **Test accuracy (both):** 96.5%

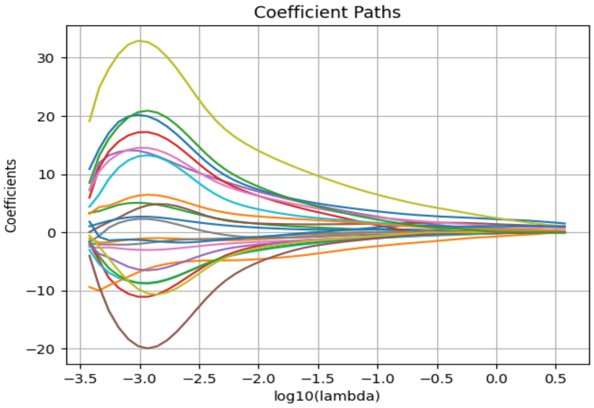

Figure 7: Validation accuracy across λ for LogRegCDD with $\alpha = 0.0$

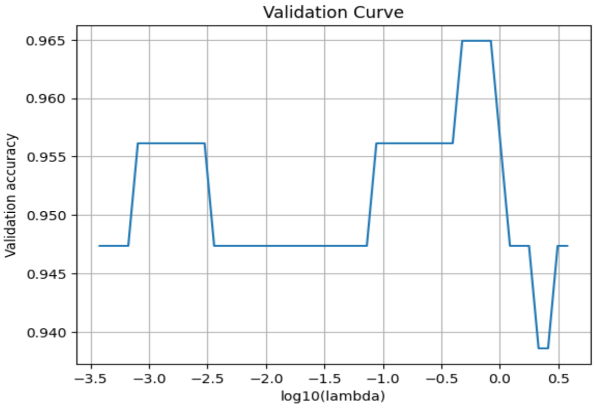

Figure 8: Coefficient paths for LogRegCDD with $\alpha = 0.0$

Figure 9: $= 0$

3.1 General discussion:

Across all values of α, the LogRegCDD algorithm exhibits consistent performance and agrees well with scikit-learn's model selection. The algorithm successfully adapts to both sparse and dense regimes. In the $\alpha = 1.0$ case, the solution is highly sparse, while for $\alpha = 0.0$, all coefficients remain non-zero. In every configuration, the best λ found by LogRegCDD differs slightly from scikit-learn, but leads to the same final test accuracy (96.5%), validating the correctness of the coordinate descent implementation.

As λ increases, the regularization effect becomes stronger, leading to smaller coefficient magnitudes; many coefficients shrink toward zero, especially in the case of L1 regularization.

Synthetic datasets have also been produced on a large scale for further experiments. They will be discussed during the presention.

References

[1] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[2] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320, 2005.

[3] Jerome H Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33:1–22, 2010.

[4] Stephen J Wright. Coordinate descent algorithms. *Mathematical programming*, 151(1):3–34, 2015.

[5] Unknown https://www.kaggle.com/datasets/colormap/spambase.

[6] Unknown https://www.kaggle.com/datasets/yasserh/wine-quality dataset.

[7] Unknown https://www.kaggle.com/datasets/jamieleech/ionosphere.

[8] Johns Hopkins University https://www.kaggle.com/datasets/jamieleech/ionosphere.