Ridge Regression for S&P 500 Analysis

Raul Unnithan

May 17, 2025

Contents

Chapter 1 Introduction

1.1 Background

High-dimensional data refers to data in which the number of variables, p, is much larger than the number of observations, n.[1] It is increasingly common in various fields, such as finance, genomics, and social sciences. Analysing high-dimensional data poses issues, mainly due to over-fitting, multicollinearity and identifying the most relevant subset of variables.[1] Multicollinearity is an issue because it makes it difficult to tell which predictor is impacting the response variable. Ridge regression is a penalised regression technique which solves these problems by incorporating a penalty term that reduces the impact of less important variables, thus shrinking their coefficients towards zero without removing them from the model.[2]

1.2 Research Objectives and Aims

This report applies Ridge Regression to an S&P 500 dataset to identify the key stocks influencing average market returns. It simplifies the model and shrinks irrelevant features through the L_2 penalty: $\|\boldsymbol{\beta}\|_2^2 = \sum_{j=1}^{p} \beta_j^2$. The larger coefficients in the Ridge model represent the more influential stocks, which significantly impact overall returns. Cross-validation optimises the regularisation parameter (λ), thus balancing model simplicity and predictive accuracy by minimising Mean Squared Error (MSE). The report also uses a coefficient path to understand how the estimated coefficients of the Ridge regression model change as the regularisation parameter changes.

1.3 Literature Review

The curse of dimensionality is prevalent in financial modelling, where datasets can involve hundreds of predictors. Fan and Li (2001) highlighted the limitations of Ordinary Least Squares (OLS)-based variable selection methods in such cases because these selection procedures ignore random errors inherited in each variable selection stage.[3]

Tikhonov (1943) introduced Ridge regression to solve this by incorporating a penalty term, shrinking coefficients toward zero while keeping all predictors in the model.[2] Hoerl and Kennard (1970) established the theory for Ridge regression, demonstrating its superiority over OLS in multi-collinear data by introducing a bias term, reducing the variance of the estimators.[4]

Ridge regression is used widely in financial research. Giannone (2016) highlighted Ridge regression's Bayesian foundation by demonstrating its predictive accuracy and extending it beyond simple forecasts, demonstrating its effectiveness in high-dimensional financial datasets.[5] Ridge regression, despite its versatility in financial contexts like bond pricing, has a limited application to time series datasets like the S&P 500 in current literature.

Existing studies, such as Sant'Anna (2020), have used Lasso regression to analyse index funds. Lasso handles variable selection better because it can set coefficients of irrelevant features completely to zero, unlike Ridge. However, its use of the L_1 parameter: $\|\boldsymbol{\beta}\|_1 = \sum_{i=1}^{n} |\beta_i|$ makes it less suitable for multi-collinear datasets like S&P 500, as it excludes some predictors. The S&P 500 is multi-collinear because all these stocks can be affected by substantial global events such as recessions. This report addresses gaps in Ridge regression analysis by using its improved ability to handle multicollinearity and identify influential stocks that have the most impact on average market returns.

1.4 Ridge Regression

1.4.1 Theory

Ridge regression is a linear regression method that includes a penalty term to constrain coefficient size.[2] Its objective function general form is:

$$\hat{\boldsymbol{\beta}}_{\text{Ridge}} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 \right\}$$

where $\hat{\boldsymbol{\beta}}_{\text{Ridge}}$ is the vector of coefficient estimates, $\boldsymbol{\beta}$ is the vector of regression coefficients, $\boldsymbol{Y}$ is the response variable vector, $\boldsymbol{X}$ is the matrix of predictor (feature) variables, with rows as observations and columns as predictors, the penalty term $\|\boldsymbol{\beta}\|_2^2 = \sum_{j=1}^{p} \beta_j^2$ is the L_2 norm of $\boldsymbol{\beta}$, and λ controls the trade-off between fit and coefficient magnitude, with larger values increasing the influence of the penalty. As λ increases, the Ridge Regression model shrinks all coefficients toward zero, as shown in the coefficient path plot (Figure 3.1).

The assumption that $\hat{\boldsymbol{\beta}}_{\text{Ridge}}$ minimises the general function can be shown by solving the above equation using differentiation with respect to $\boldsymbol{\beta}$, giving the closed-form estimator: $\hat{\boldsymbol{\beta}}_{\text{Ridge}} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p\right)^{-1}\mathbf{X}^T\mathbf{Y}$.[1] This is the cost function Ridge Regression aims to minimise:

$$\|\boldsymbol{Y} - \boldsymbol{X\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 = RSS + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}$$

The second equality is expressing the residual sum of squares (RSS) as a matrix, which will be expanded upon and differentiated.

$$\begin{aligned} RSS + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta} &= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta} \\ &= \mathbf{y}^T\mathbf{y} - 2\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{y} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta} \end{aligned}$$

Take the gradient with respect to $\boldsymbol{\beta}$ and set it to zero for the minimum:

$$\begin{aligned} -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} + 2\lambda\boldsymbol{\beta} &= 0, \\ -\mathbf{X}^T\mathbf{y} + \mathbf{X}^T\mathbf{X}\boldsymbol{\beta} + \lambda\boldsymbol{\beta} &= 0, \\ \mathbf{X}^T\mathbf{X}\boldsymbol{\beta} + \lambda\boldsymbol{\beta} &= \mathbf{X}^T\mathbf{y}, \\ \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)\boldsymbol{\beta} &= \mathbf{X}^T\mathbf{y}, \\ \boldsymbol{\beta} &= \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y}. \quad \blacksquare \end{aligned}$$

The L_2 penalty shrinks all coefficients toward zero without making them precisely zero, leading to simpler models that reduce multicollinearity and mitigate over-fitting.[2] This also leads to more stable models, especially when the number of variables is much larger than the number of observations.[1] Estimates from the ridge regression cost function, $\|\boldsymbol{Y} - \boldsymbol{X\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2$, solve the LS problem subject to the constraint $\sum_{j=1}^{p} \beta_j^2 \leq c$.[2] (Fletcher 2008) uses Karush-Kuhn Tucker (KKT) conditions to verify equivalence. It uses the Lagrangian along with this constraint to start the proof:
$\mathcal{L}(\beta, \nu) = \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \nu\left(\|\beta\|_2^2 - c\right)$, where $\nu \geq 0$ is the Lagrange multiplier.

The first KKT condition needs the Lagrangian gradient with respect to β to equal zero at the optimal solution:

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\beta} &= -2\mathbf{X}^\top(\mathbf{Y} - \mathbf{X}\beta) + 2\nu\beta = 0, \\ (\mathbf{X}^\top\mathbf{X} + \nu\mathbf{I})\beta &= \mathbf{X}^\top\mathbf{Y}. \end{aligned}$$

The second KKT condition is the complementarity property: $\nu(\|\beta\|_2^2 - c) = 0$, which has two consequences. If $\nu > 0$, the constraint $\|\beta\|_2^2 = c$ is active, meaning the solution lies on the constraint boundary, and if the constraint is inactive, i.e. $(\|\beta\|_2^2 < c)$, then $\nu = 0$, so the Lagrange multiplier is zero.[6]

If $\nu = \lambda$ and $c = \|\hat{\beta}(\lambda)\|_2^2$, then the ridge estimator $\hat{\beta}(\lambda)$ satisfies both KKT conditions, and both forms produce the same solution here.[6] ■

The constraint defines a smooth L_2 ball in the coefficient space. This smooth constraint means that Ridge does not completely zero coefficients.[2]

1.5 Application

1.5.1 Dataset Processing and Analysis

The high-dimensional dataset for analysis comes from Mendeley Data.[7] This is the S&P 500 dataset, which includes the daily closing prices of multiple stocks from the S&P 500 index. Ridge regression can be applied to this dataset to select influential stocks and spot their trends. This work is helpful for portfolio managers because they can improve their equity asset allocation.

The first analysis stage was to replace `NA`s in the `returns_data` with the column mean. Mean imputation minimises bias and preserves the underlying distributions. The features in the matrix `X` were standardised with the `scale()` function, so each feature has a mean of 0 and a standard deviation of 1 and helps λ identify relevant features. The dataset is split so the Ridge regression model can be trained. The `glmnet` package is used in R to carry out the ridge regression implementation. The key steps were computing daily returns from stock prices, imputing missing values with column means, and standardising predictors. The ridge model and cross-validation determined the optimal λ. The model's performance was evaluated on the test via the MSE, and a plot showed the highest contributing predictors.

1.5.2 Results

The MSE on the test set was `2.02e-07`, so the ridge model worked well. The top contributing predictors from Figure 3.2 were STZ (Constellation Brands Inc.), GNW (Genworth Financial), and DVN (Devon Energy Corp). The results are all positive, so these companies did well in this period. DVN had a significant increase in oil production, explaining its higher returns. STZ acquired one of its competitors, Grupo Modelo, thus, why its stock prices were higher.[8] Lasso could apply better to this dataset because it can remove features by shrinking their coefficients to zero, but Ridge deals with multicollinearity better. The condition number gives the multicollinearity. It is the square root of the largest eigenvalue over the smallest eigenvalue.[9] A condition number over 30 means high multicollinearity, and the condition number here was `72.49`. The `best_lambda` was `0.0666`, so, despite the high multicollinearity, ridge deals with it well. In the coefficient path plot, Figure 3.1, the paths shrink to zero, which should not occur for Ridge. It occurs due to the `glmnet` package treating small coefficient values as zero.

Chapter 2 Conclusion

2.1 Summary of Findings

Key findings from this application of Ridge Regression to a high-dimensional time series dataset included:

- The Ridge regression achieving a low MSE of 2.02×10^{-7}, so the ridge model has a high prediction accuracy despite high dimensionality.
- STZ (Constellation Brands Inc.), GNW (Genworth Financial), and DVN (Devon Energy Corp.) being identified as the top contributors, with STZ having the most significant coefficient contribution.
- A high condition number confirming severe multicollinearity and appropriate use of Ridge Regression rather than Lasso, but Ridge regression addresses this well with a low λ value.

2.2 Implications for Future Research & Ridge Comparison

Future research into this area could look into different ways of integrating Ridge regression with other regression techniques to improve prediction accuracy. For instance, Gasparin et al. (2020) use an adaptive elastic-net regularisation approach, which is a combination of Lasso, adaptive Lasso, and elastic-net techniques, showing how integrating sparsity and regularisation can effectively improve index tracking.[10] This was done using a tracking error, which indicates the portfolio tracks the benchmark closely.

Testing other shrinkage methods proved to be interesting. Lasso achieved an MSE of 3.82×10^{-7}, which, while a good accuracy, was not as good as Ridge. However, what was most noteworthy was testing the elastic net regularisation technique. Elastic net combines ridge and lasso so it considers both. This produced an MSE of 4.71×10^{-7}, which is higher than the ridge MSE, indicating that ridge is the best-performing simple shrinkage method on this dataset.

Bibliography

1. Reza Drikvandi. *High Dimensional Statistics IV (MATH4287) Lecture Notes.* Durham University, Durham, 2024. Available for the year 2024-25.

2. Hailiang Du. *Machine Learning and Neural Networks.* Durham University, Durham, 2023. Lecture notes for the module MATH3431.

3. Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

4. Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

5. Domenico Giannone. Comment. *Journal of Business & Economic Statistics*, 34(3):342–344, 2016.

6. Roger Fletcher. *Practical optimization methods.* John Wiley, 2000.

7. Chi Seng Pun. Low and high-dimensional stock price data, 2017. Dataset contains financial research data including low-dimensional and high-dimensional daily stock prices of S&P 500 and Russell 2000 indices from Jan 1, 2013 to Dec 31, 2014. Licensed under CC BY 4.0.

8. Constellation Brands, Inc. Constellation brands completes acquisition of Grupo Modelo's U.S. Beer Business, June 2013. Accessed: 2024-12-01.

9. Jong Hae Kim. Multicollinearity and misleading statistical results. *Korean Journal of anesthesiology*, 72(6):558–569, 2019.

10. Lianjie Shu, Fangquan Shi, and Guoliang Tian. High-dimensional index tracking based on the adaptive elastic net. *Quantitative Finance*, 20(9):1513–1530, 2020.

Chapter 3 Appendix

3.1 R Code

```
install.packages("glmnet")
library(glmnet)

sp500_data <- read.csv("C:\\Users\\raula\\OneDrive\\Documents\\Durham\\4th␣
   Year\\High-Dimensional␣Statistics\\Mini␣Project␣Report\\Code\\sp500data.csv
   ")

# Remove first row and first column - dates and stock tickers
sp500_data <- sp500_data[-1, -1]

# Compute daily returns: (current - previous price)/previous price
returns_data <- diff(as.matrix(sp500_data)) / as.matrix(sp500_data[-nrow(sp500
   _data), ])

# Replace NA values in returns_data with column mean
returns_data[is.na(returns_data)] <- sapply(returns_data, function(x) mean(x,
   na.rm = TRUE))

# Define target variable ‘y’ as average return of stocks per time
y <- rowMeans(returns_data, na.rm = TRUE)

# Convert X to matrix form and standardise predictors
X <- as.matrix(returns_data)
X <- scale(X)

# Split data into training and test sets
train_index <- sample(1:nrow(X), 0.7 * nrow(X)) # 70% for training

X_train <- X[train_index, ]
y_train <- y[train_index]
```

```
X_test <- X[-train_index, ]
y_test <- y[-train_index]

ridge_model <- cv.glmnet(X_train, y_train, alpha = 0)
best <- ridge_model$lambda.min

# Fit the final Ridge model using best lambda on the training data
final_model <- glmnet(X_train, y_train, alpha = 0, lambda = best)

# Evaluate the model on the test set
predictions <- predict(final_model, s = best, newx = X_test)
mse <- mean((y_test - predictions)^2)

# Ridge Minimiser Proof Verification: Generate synthetic data
n <- 100
p <- 10

X <- matrix(rnorm(n * p), nrow = n, ncol = p) # Random predictor matrix
beta_true <- rnorm(p) # True coefficients
y <- X %*% beta_true + rnorm(n, sd = 0.5) # response with noise

# Ridge regression closed-form estimator
lambda <- 1
I <- diag(p)

ridge_beta <- solve(t(X) %*% X + lambda * I) %*% t(X) %*% y

# Verify by comparing with the glmnet package
ridge_model <- glmnet(X, y, alpha = 0, lambda = lambda / n) # dividing by n
    scales lambda
ridge_beta_glmnet <- as.vector(coef(ridge_model, s=lambda/n))[-1]

cat("Closed-form␣Ridge␣Coefficients:\n", ridge_beta, "\n")
cat("GLMNet␣Ridge␣Coefficients:\n", ridge_beta_glmnet, "\n")

# Verifying constraint proof satisfies KKT: simulate data
n <- 50 # Number of observations
p <- 100 # Number of predictors

X <- matrix(rnorm(n * p), nrow = n, ncol = p) # Design matrix
beta_true <- rnorm(p) # True coefficients
```

```
Y <- X %*% beta_true + rnorm(n) # Response vector
lambda <- 10

# Ridge regression using the normal equation
ridge_solution <- function(X, Y, lambda) {
  p <- ncol(X)
  I <- diag(p) # Identity matrix
  beta_hat <- solve(t(X) %*% X + lambda * I) %*% t(X) %*% Y
  return(beta_hat)}

beta_ridge <- ridge_solution(X, Y, lambda)

verify_kkt <- function(X, Y, beta, lambda, tolerance = 1e-6) {
  # 1. Gradient condition (stationarity)
  gradient <- -2 * t(X) %*% (Y - X %*% beta) + 2 * lambda * beta
  gradient_condition <- all(abs(gradient) < tolerance)

  # 2. Complementarity condition
  beta_norm_squared <- sum(beta^2)
  complementarity_condition <- abs(lambda * (beta_norm_squared - beta_norm_
      squared)) < tolerance

  # Both conditions must be satisfied
  all_conditions <- gradient_condition & complementarity_condition
  return(all_conditions)}

kkt_satisfied <- verify_kkt(X, Y, beta_ridge, lambda)
print(kkt_satisfied) # returns True
```

3.2 Additional Data Visualisations

```
# Plot Ridge Coefficient Paths, highlighting the best lambda

plot(ridge_model$glmnet.fit, xvar = "lambda", label = FALSE)
abline(v = log(best), col = "red", lty = 2)
title(main = "Ridge␣Regression␣Coefficient␣Paths", line = 2.5)
```

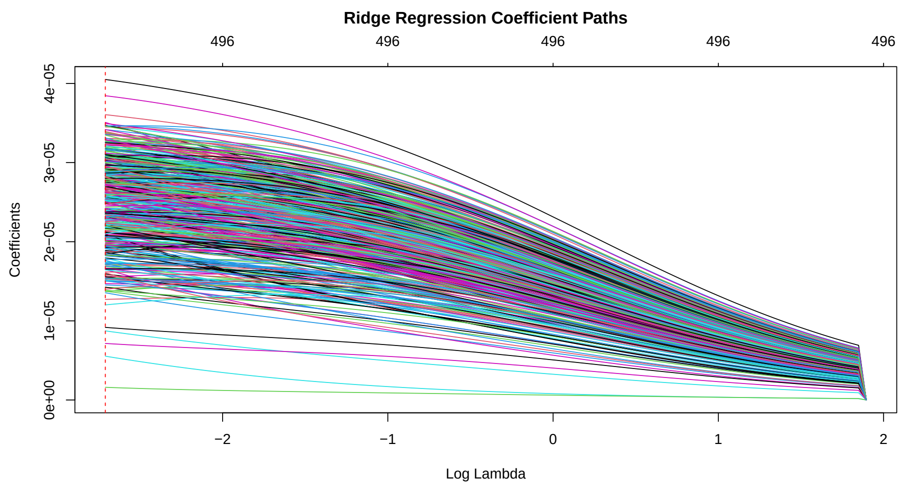

Figure 3.1: Ridge Coefficient Paths

```
# Convert sparse matrix to a numeric vector and remove intercept
coefficients_vector <- as.numeric(coefficients)[-1]
# Assign correct stock names
names(coefficients_vector) <- rownames(coefficients)[-1]

coefficients_vector <- coefficients_vector[!is.na(coefficients_vector) & is.
    finite(coefficients_vector)] # Remove invalid
top_10_coefs<-sort(abs(coefficients_vector),decreasing=TRUE)[1:10]
# Plot the top 10 coefficients
barplot(top_10_coefs, main = "Top␣10␣Stocks␣by␣Coefficient␣Magnitude␣(
    Excluding␣Intercept)", xlab = "Stocks", las = 2, col = "steelblue", cex.
    names = 0.8)
```

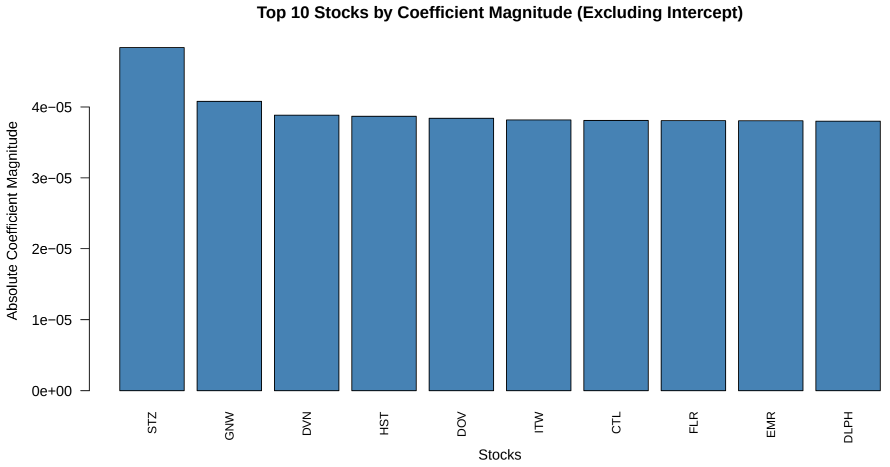

Figure 3.2: Top 10 Contributing Stocks by Coefficient Magnitude