Some Perspectives to Penalized Regression Model (Part of 2024 SURF)

Yu Lu (XJTLU Bsc Applied Statistics (Biomedical statistics))

OLS / MLE Solution to LR with issues

We have learned the equivalence of Ordinary Least Squares (OLS) and Maximum Likelihood Estimation (MLE) in the context of linear regression assuming the error term follows a Gaussian distribution but with limitations as below:

Issues

Even if the OLS or MLE solution is optimal in terms of minimizing the error, it may lead to overfitting, especially when the number of features is large relative to the number of observations. This can result in high variance and poor generalization to new data.

How to handle the issues?

Definition

$$\text{LASSO:}\quad Argmin_{\beta}\sum_{i=1}^{N}\left(y_i-\sum_{j}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j}^{p}|\beta_j|,\quad \text{where } \lambda>0$$

$$\text{Ridge:}\quad Argmin_{\beta}\sum_{i=1}^{N}\left(y_i-\sum_{j}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j}^{p}\beta_j^2,\quad \text{where } \lambda>0$$

- OLS perspective thinking: This shrinkage method not only achieves the objective of making the error to be smallest like OLS, but also prevents overfitting by penalizing large coefficients.
- MLE perspective thinking: Compared with MLE, this method could be regarded as a Bayesian estimation that can avoid overfitting by introducing the prior distribution $P(\beta)$, especially when the data volume is small, the prior information can play a regularization role.

Why does the terms of the regularization look like this?

Just like what we have learned about the relationship between the OLS and MLE and get the idea of why the error term is the square shape, now we are going to use the perspective of **Bayesian** to understand the penalized terms

Brief Intro to Bayesian Method

Bayesian Method

Bayesian statistics is a statistical paradigm that interprets probability as a measure of belief or certainty rather than a frequency. It combines prior beliefs with evidence to update the probability of a hypothesis.

- **Prior Distribution**: Represents our beliefs about the parameters before observing the data.
- **Likelihood**: The probability of the observed data given the parameters.
- **Posterior Distribution**: The updated beliefs about the parameters after observing the data, calculated using Bayes' theorem.
- **Bayes' Theorem**:

$$P(\beta|D) = \frac{P(D|\beta)P(\beta)}{P(D)}$$

Choose Prior Distribution $P(\beta)$

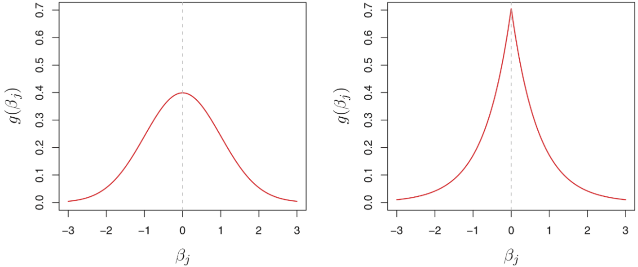
Idea:

We insist that β should be more likely to be small also with small variances (very near to 0). This means we should pick a distribution that has a peak around zero and decays quickly as we move away from zero.

- **Prior Distribution**: In the context of penalized regression, we choose a prior distribution for the coefficients β that reflects our beliefs about their values. Common choices include:
  - **Laplace Prior**: Assumes coefficients follow a Laplace distribution, leading to L^1 regularization (Lasso Regression).
  - **Gaussian Prior**: Assumes coefficients are normally distributed around zero, which leads to L^2 regularization (Ridge Regression).

This also means it would take extreame evidence with the data that we see in order to accept very large and very high variances beta because of the prior.

Choose Prior Distribution $P(\beta)$



Maximize the posterior probability to get the point estimation of β – Ridge

$$P(\beta|y) \propto P(y|\beta) \cdot P(\beta)$$

$$P(y|\beta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(y-\beta^T x)^2}{2\sigma^2}\right\}$$

And for the prior part, we assume β follows a Gaussian distribution, $\beta \sim \mathcal{N}(0, \sigma'^2)$,

$$P(\beta) = \frac{1}{\sqrt{2\pi}\sigma'} \exp\left\{-\frac{\beta^2}{2\sigma'^2}\right\}$$

Thus,

$$\begin{aligned}\hat{\beta} &= \arg\max_{\beta} \log P(\beta|Y) \\ &= \arg\max_{\beta} \log P(Y|\beta)P(\beta) \\ &= \arg\max_{\beta} \log \prod_{i=1}^{N} P(y_i|\beta)P(\beta) \\ &= \arg\max_{\beta} \sum_{i=1}^{N} \log[P(y_i|\beta)P(\beta)]\end{aligned}$$

Maximize the posterior probability to get the point estimation of β – Ridge (continue)

$$= \arg\max_{\beta} \sum_{i=1}^{N} \left[\log\left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{(y - \beta^T x)^2}{2\sigma^2} \right\} \right) + \log(\frac{1}{\sqrt{2\pi}\sigma'}) - \frac{\beta^2}{2\sigma'^2} \right]$$

$$= \arg\max_{\beta} \sum_{i=1}^{N} \left[\log\left(\frac{1}{2\pi\sigma'\sigma} \right) - \frac{(y_i - \beta^T x_i)^2}{2\sigma^2} - \frac{\beta^2}{2\sigma'^2} \right]$$

$$= \arg\min_{\beta} \sum_{i=1}^{N} \left[\frac{(y_i - \beta^T x_i)^2}{2\sigma^2} + \frac{\beta^2}{2\sigma'^2} \right]$$

$$= \arg\min_{\beta} (Y - X\beta)^T (\frac{1}{2\sigma^2} I_N)(Y - X\beta) + \frac{1}{\sigma'^2} \beta^2$$

((Weighted least squares with the weight $\frac{1}{2\sigma^2} I_N$ and the penalty term $\frac{1}{\sigma'^2} \beta^2$))

If we let $\lambda = \frac{1}{\sigma'^2}$, then the loss function is $L(\beta) = \sum_{i=1}^{N} (\beta^T x_i - y_i)^2 + \lambda \beta^T \beta$, which is equivalent to the minimization of the regularized least squares in L^2.

Remark of the Ridge Regression

Ridge regression solution is also the posterior mean, this is because the likelihood of the data given the parameters is Gaussian, and the prior is also Gaussian, which results in a posterior that is also Gaussian by the property of the conjugate prior of the Gaussian distribution, i.e.

Ridge Regression Posterior Mean

NEED to ADD!

Maximize the posterior probability to get the point estimation of β – Lasso

Now the prior part is assumed to be a Laplace distribution, $\beta \sim \text{Laplace}(0, b), i.e., P(\beta) = \frac{1}{2b} \exp\left(-\frac{|\beta|}{b}\right)$.

$$\arg\max_{\beta} \sum_{i=1}^{N} \left[\log\left(\frac{1}{\sqrt{2\pi}\sigma_{\beta}} \exp\left\{ -\frac{(y-\beta^T x)^2}{2\sigma^2} \right\} \right) - \log 2b - \frac{|\beta|}{b} \right]$$

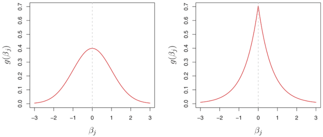
$$= \arg\max_{\beta} \sum_{i=1}^{N} \left[\log\left(\frac{1}{\sqrt{2\pi}\sigma_{\beta} 2b} \right) - \frac{(y_i-\beta^T x_i)^2}{2\sigma^2} - \frac{|\beta|}{b} \right]$$

$$= \arg\min_{\beta} \sum_{i=1}^{N} \left[\frac{(y_i-\beta^T x_i)^2}{2\sigma^2} + \frac{|\beta|}{b} \right]$$

$$= \arg\min_{\beta} (Y - X\beta)^T (\frac{1}{2\sigma^2} I_N)(Y - X\beta) + \frac{1}{b}|\beta|$$

If we let $\lambda = \frac{1}{b}$, then the loss function is $L(\beta) = \sum_{i=1}^{N} (\beta^T x_i - y_i)^2 + \lambda|\beta|$, which is equivalent to the minimization of the regularized least squares in L^1.

Further check λ and corresponding term in the posterior likelihood–How the penalized terms are derived from the Bayesian perspective



Gaussian:

$\lambda = \frac{1}{\sigma'^2}$ i.e. Larger λ with less σ'^2 means more regularization corresponding to smaller variance of the prior distribution (normal)

Laplace:

$\lambda = \frac{1}{b}$ i.e. Larger λ with less b means more regularization corresponding to smaller variance of the prior distribution (laplace)

How to choose the λ?

- **Cross-Validation**: A common method to choose the regularization parameter λ is through cross-validation. This involves splitting the data into training and validation sets, fitting the model with different values of λ, and selecting the one that minimizes the prediction error on the validation set.

Lagrange Multipliers Perspective of Penalized Models

Lagrange Multipliers Perspective

Lagrange multipliers provide a way to incorporate constraints into optimization problems. In the context of penalized regression, we can view the regularization term as a constraint on the size of the coefficients.

The formula could be expressed as:

$$\min_{\beta} \left\{ \|y - X\beta\|_2^2 + \lambda \|\beta\|_p^p \right\}$$

where $\|\beta\|_p^p$ is the L^p norm of the coefficients, which serves as a penalty term.

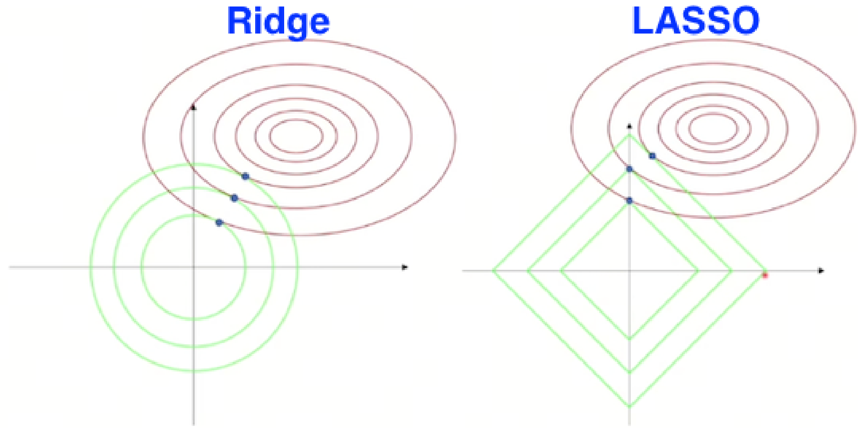
The gradient of the objective function of Ridge with respect to β is given by:

$$\nabla J(\beta) = -2X^T(y - X\beta) + \lambda \nabla \|\beta\|_p^p$$

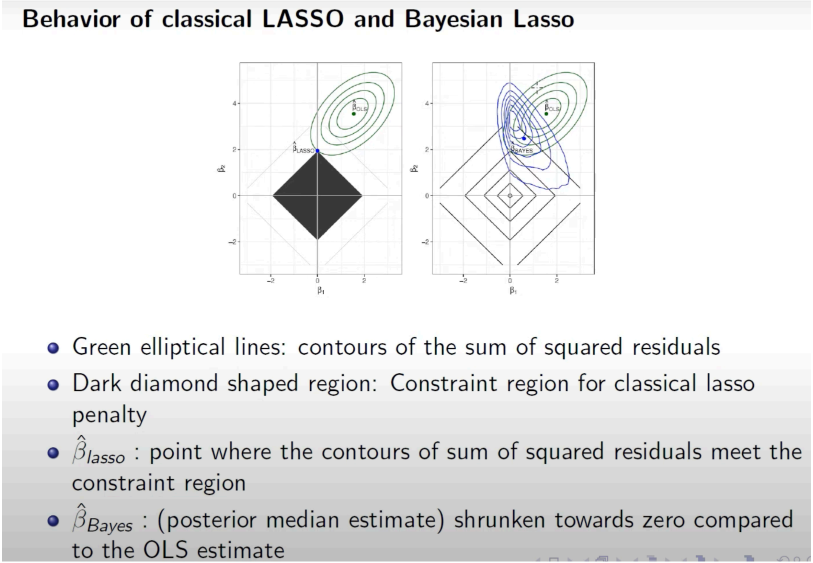
(see matrix form (Go to Matrix Form) below for more details)

This should be set to zero to find the optimal β, which means the gradient of the loss function (the first term) is equal to the gradient of the penalty term (the second term) scaled by λ.

Geometric Interpretation of Penalized Models in Lagrange Multipliers Perspective



Geometric Interpretation of LASSO in Lagrange Multipliers Perspective and Bayesian Perspective



Behavior of classical LASSO and Bayesian Lasso

- Green elliptical lines: contours of the sum of squared residuals
- Dark diamond shaped region: Constraint region for classical lasso penalty
- $\hat{\beta}_{lasso}$: point where the contours of sum of squared residuals meet the constraint region
- $\hat{\beta}_{Bayes}$: (posterior median estimate) shrunken towards zero compared to the OLS estimate

Understanding of λ

Lagrange Multipliers help explain how λ balances fitting the data well:

Bigger λ makes the ellipse (representing the constraint region in the parameter space in WLS, if it satisfies the homoscedasticity, it would be a circle) smaller. In this case, the gradient of β becomes steeper with a larger λ, which means that the optimization algorithm will grow the penalty of β to stay within the shrinking feasible region. This helps reduce variance and prevent overfitting, but overly large λ can cause underfitting.

Matrix Form of Penalized Regression Models

Ridge Regression

Objective Function:

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

Matrix Form Derivation:

1. Expand the loss term:

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

2. Add the L^2 penalty:

$$J(\beta) = y^T y - 2y^T X\beta + \beta^T X^T X\beta + \lambda\beta^T \beta.$$

3. Compute the gradient:

$$\frac{\partial J}{\partial \beta} = \nabla J(\beta) = -2X^T (y - X\beta) + \lambda \nabla \|\beta\|_p^p = -2X^T y + 2X^T X\beta + 2\lambda\beta.$$

4. Set gradient to zero:

$$-X^T y + X^T X\beta + \lambda\beta = 0 \implies (X^T X + \lambda I)\beta = X^T y.$$

5. Solve for β:

$$\hat{\beta}_{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y$$

Why Regularization Helps: The term λI ensures $X^T X + \lambda I$ is always invertible (since $\lambda > 0$ adds positive values to the diagonal, guaranteeing full rank). This avoids singularity issues in $X^T X$ when features are collinear.

Matrix Form of Penalized Regression Models

Lasso Regression

Formula:

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

Objective Function:

$$\min_{\beta} \left\{ \|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

Matrix Form Derivation: Lasso lacks a closed-form solution due to the non-differentiable L1 norm. Instead, we use the subgradient optimality condition:

1. Subgradient equation:

$$-2X^T(y - X\beta) + \lambda \cdot \text{sign}(\beta) = 0,$$

where $\text{sign}(\beta)$ is defined component-wise:

Key Insight:

- For $\beta_j \neq 0$, the solution balances data fit and shrinkage.
- For $\beta_j = 0$, the condition requires:

$$\left|2\mathbf{x}_j^T(y - X\beta)\right| \leq \lambda.$$

This induces sparsity (exact zeros in β), which Ridge cannot achieve.

Differences between Lasso and Ridge

- Lasso could realize variable selection by shrinking some coefficients to exactly zero, while Ridge regression shrinks all coefficients but does not set any to zero.

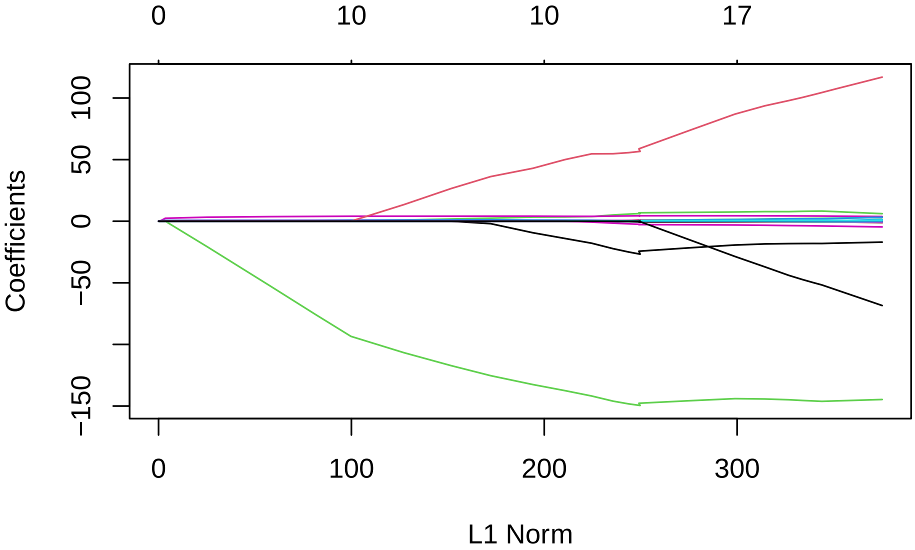
R code Example of Penalized Linear Regression Models

```r
# Install and load necessary packages
# install.packages("glmnet")
# install.packages("ISLR2")
library(glmnet)
library(ISLR2)

# Load the Hitters dataset
data(Hitters)

# Remove missing values
Hitters <- na.omit(Hitters)

# Define the predictor matrix and response variable
# exclude the intercept term
x <- model.matrix(Salary ~ ., Hitters)[, -1]
y <- Hitters$Salary
```

R code Example of Penalized Linear Regression Models (continue)

```
# Split the data into training and test sets
set.seed(1)
train <- sample(1:nrow(x), nrow(x) / 2)
test <- (-train)
y.test <- y[test]

# Fit the Lasso model on the training data
grid <- 10^seq(10, -2, length = 100)
# LASSO-LR
lasso.mod <- glmnet(x[train, ], y[train], alpha = 1,
                    lambda = grid, family = 'gaussian')
# Plot the Lasso model to visualize coefficient paths
plot(lasso.mod)
```

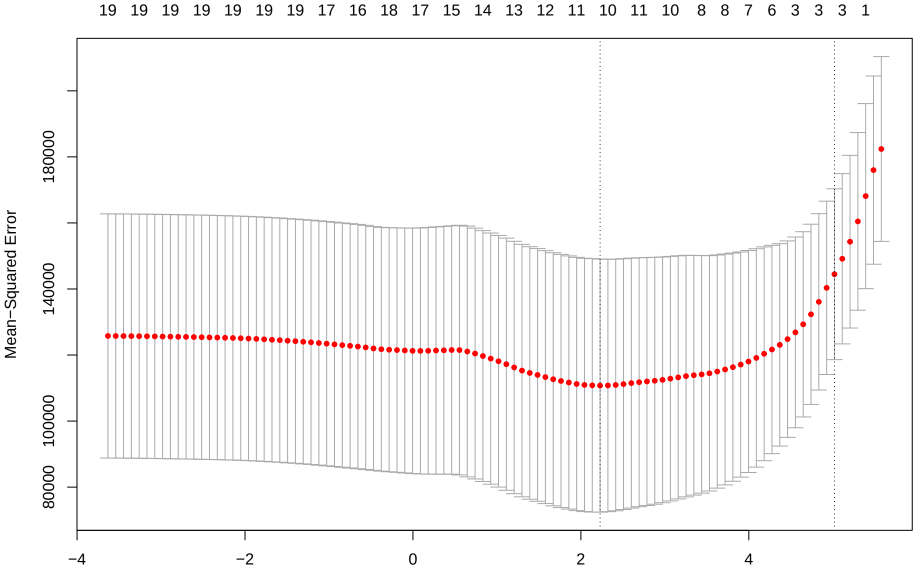
R code Example of Penalized Linear Regression Models (continue)



R code Example of Penalized Linear Regression Models (continue)

```r
# Perform cross-validation to find the optimal lambda
set.seed(1)
cv.out <- cv.glmnet(x[train, ], y[train], alpha = 1)
plot(cv.out)
```

R code Example of Penalized Linear Regression Models (continue)



R code Example of Penalized Linear Regression Models (continue)

```r
bestlam <- cv.out$lambda.min
# Make predictions on the test set using the optimal lambda
lasso.pred <- predict(lasso.mod, s = bestlam, newx = x[test, ])
test.mse <- mean((lasso.pred - y.test)^2)
print(paste("Test MSE with Lasso and optimal lambda:", test.mse))
# Fit the Lasso model on the full dataset using the optimal lambda
lasso.full <- glmnet(x, y, alpha = 1,lambda = bestlam)
lasso.coef <- coef(lasso.full)
print("Lasso coefficients on the full dataset:")
print(lasso.coef)
lasso.coef[lasso.coef != 0]
# print the name of the variables that are not zero
print("Variables with non-zero coefficients:")
names(lasso.coef)[lasso.coef != 0]
```

R code Example of Penalized Linear Regression Models (continue)

```
[1] "Test MSE with Lasso and optimal lambda: 143673.618543046"

[1] "Lasso coefficients on the full dataset:"

20 x 1 sparse Matrix of class "dgCMatrix"
                       s0
(Intercept)   -3.42073206
AtBat          .
Hits           2.02965136
HmRun          .
Runs           .
RBI            .
Walks          2.24850782
Years          .
CAtBat         .
CHits          .
CHmRun         0.04994886
```

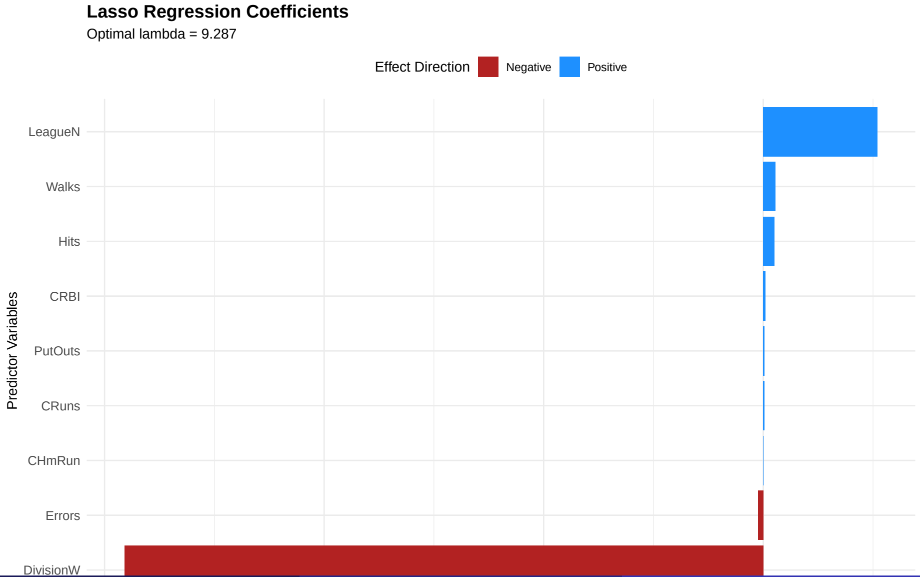
R code example of Penalized Linear Regression Models (continue)

```
            [,1]         [,2]      [,3]
[1,] -3.42073206    2.0296514 2.2485078
[2,]  0.04994886    0.2221244 0.4018303
[3,] 20.83775664 -116.3901920 0.2376831
[4,] -0.93567863   -3.4207321 2.0296514
```

R code example of Penalized Linear Regression Models (continue)

```r
# plot
non_zero_coef <- lasso.coef[lasso.coef[,1] != 0, ][-1]  #
coef_df <- data.frame(
  Variable = names(non_zero_coef),
  Coefficient = as.numeric(non_zero_coef)
)

# Sort by the absolute value of the coefficient
coef_df <- coef_df[order(abs(coef_df$Coefficient), decreasing = TRUE), ]

library(ggplot2)

ggplot(coef_df, aes(x = reorder(Variable, Coefficient),
                   y = Coefficient,
                   fill = ifelse(Coefficient > 0, "Positive", "Negative"))) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("Positive" = "dodgerblue", "Negative" = "firebrick")) +
  labs(title = "Lasso Regression Coefficients",
       subtitle = paste("Optimal lambda =", round(bestlam, 4)),
       x = "Predictor Variables",
       y = "Coefficient Value",
       fill = "Effect Direction") +
  coord_flip() +
  theme_minimal() +
  theme(legend.position = "top",
        plot.title = element_text(face = "bold", size = 14),
        axis.text.y = element_text(size = 10))
```

R code example of Penalized Linear Regression Models (continue)



Remark

Actually the penalized model could be applied in many regression models not only for linear regression, but also for logistic regression, Poisson regression, Cox regression, etc.

The penalized terms could be added to the loss function of these models in a similar way.

R Code of Penalized Cox Regression - Lasso using `glmnet` package

Example

Want to minimize

$$-\log\left(\prod_{i:\delta_i=1}\frac{\exp\left(\sum_{j=1}^{p}x_{ij}\beta_j\right)}{\sum_{i':y_{i'}\geq y_i}\exp\left(\sum_{j=1}^{p}x_{i'j}\beta_j\right)}\right)+\lambda P(\beta), \tag{1}$$

where δ_i is the indicator function for censoring and $P(\beta)=\sum_{j=1}^{p}\beta_j^2$ corresponds to a ridge penalty, or $P(\beta)=\sum_{j=1}^{p}|\beta_j|$ corresponds to a lasso penalty.

R Code example for LASSO in Cox Regression applied on selecting important features of pesticide poisoning

Click here to see LASSO code provided by SURF2024 instructed by Dr.He

Summary

We have seen many perspectives to understand the penalized regression models, including:

- Bayes
- Lagrange multipliers
- matrix form

Further exploration

Gribonval, R. (2011). Should penalized least squares regression be interpreted as maximum a posteriori estimation?. IEEE Transactions on Signal Processing, 59(5), 2405-2410.

In this paper, Gribonval challenges the conventional interpretation of penalized least squares (PLS) regression as merely MAP estimation with a prior proportional to $\exp(-\phi(x))$. The author demonstrates that for any prior distribution, the MMSE estimator can also be expressed as a PLS problem with a specific penalty function ϕ_{MMSE}, which generally differs from $-\log p_X(x)$. Conversely, a single PLS estimator can simultaneously represent MAP estimation under one prior and MMSE estimation under another. This non-uniqueness implies that penalty functions should be chosen based on empirical performance rather than Bayesian beliefs about underlying signal distributions.

Thanks for listening

The greatest truths are the simplest.

Penalized regression models balance the trade-off between fitting the data well and keeping the model simple.

Everything is connected to each other. Good theory interpretes this connection interestingly.

Linear Regression: **OLS** and **MLE** (Last time)
Penalized regression models: **Lagrange multiplier** (restriction in the geometric meaning and also algebra interpretation) and **Bayesian perspective** (prior: history knowledge restriction)