Metodi di Shrinkage

Tikhonov–Phillips regularization

Tikhonov, Andrey Nikolayevich (**1943**). "Об устойчивости обратных задач" [On the stability of inverse problems]. Doklady Akademii Nauk SSSR. 39 (5): 195–198.

Tikhonov, A. N. (**1963**). "О решении некорректно поставленных задач и методе регуляризации". Doklady Akademii Nauk SSSR. 151: 501–504.. Translated in "Solution of incorrectly formulated problems and the regularization method". Soviet Mathematics. 4: 1035–1038.

Ridge regression

Hoerl, A. E.; R. W. Kennard (1970). "Ridge regression: Biased estimation for nonorthogonal problems". Technometrics. 12 (1): 55–67.

Shrinkage Methods: Ridge regression
LASSO regression
Elastic Net

Warnings:

- ✓ they can also produce models that make no sense.
- ✓ they ignore nonsignificant variables that may, nevertheless, be interesting or important.
- ✓ they don't follow any hierarchy principle.

1. RIDGE REGRESSION

Consider the standard model for multiple linear regression:

$$\underbrace{Y}_{nx1} = \underbrace{X}_{nxp} \underbrace{\beta}_{px1} + \varepsilon$$

With: $X = rank\ p$

$\beta = unknown$

$E[\varepsilon] = 0\ E[\varepsilon\varepsilon^T] = \sigma^2 I_{nxn}$

The usual approach to solve this problem is to use the Gauss-Markov linear estimator:

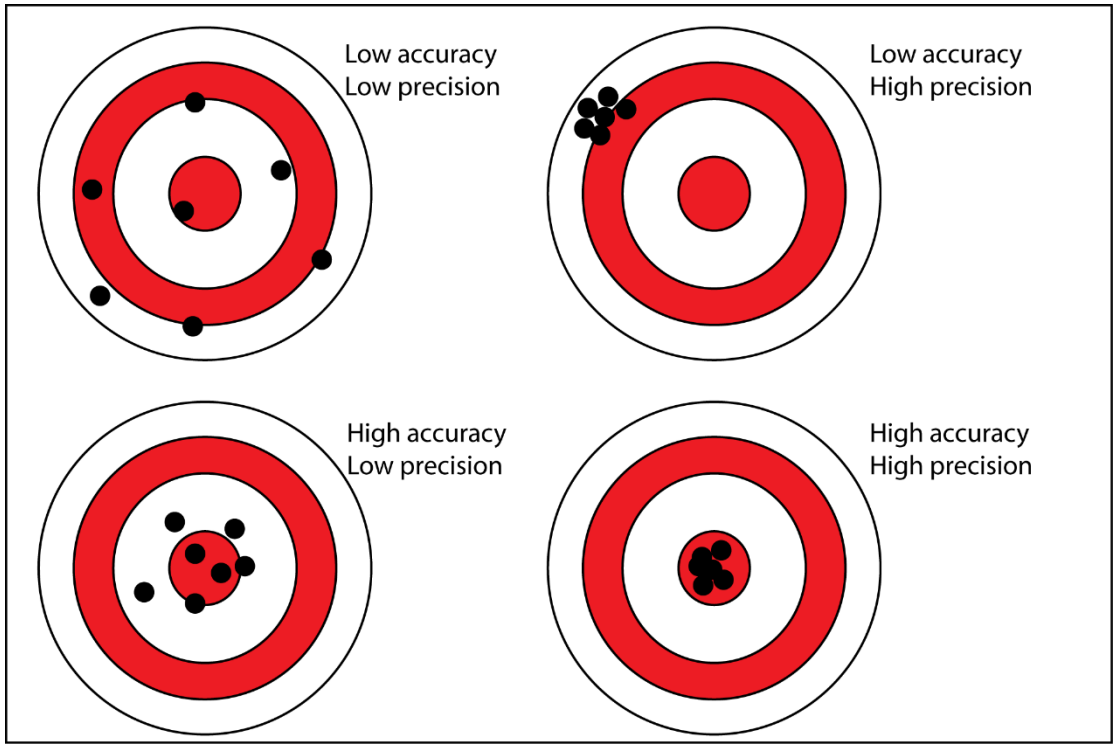
$$\beta^{LS} = (X^T X)^{-1} X^T Y$$

That solve the minimization problem:

$$\beta^{LS} = (X^T X)^{-1} X^T Y = \underset{\beta \in \Re^p}{\arg\min} (Y - X\beta)^T (Y - X\beta) = \underset{\beta \in \Re^p}{\arg\min} \|Y - X\beta\|_2^2$$

Accuracy (bias) and Precision (variance):

Accuracy refers to the closeness of a measured value to a standard or known value.
Precision refers to the closeness of two or more measurements to each other.



1. Predictive ability: Linear regression has low bias (zero bias) but suffers from high variance. So it may be worth sacrificing some bias to achieve a lower variance

2. Interpretative ability: with a large number of predictors, it can be helpful to identify a smaller subset of important variables.

Linear regression doesn't do this

Also: linear regression is not defined when $p > n$

Ridge regression is like least squares but shrinks the estimated coefficients towards zero:

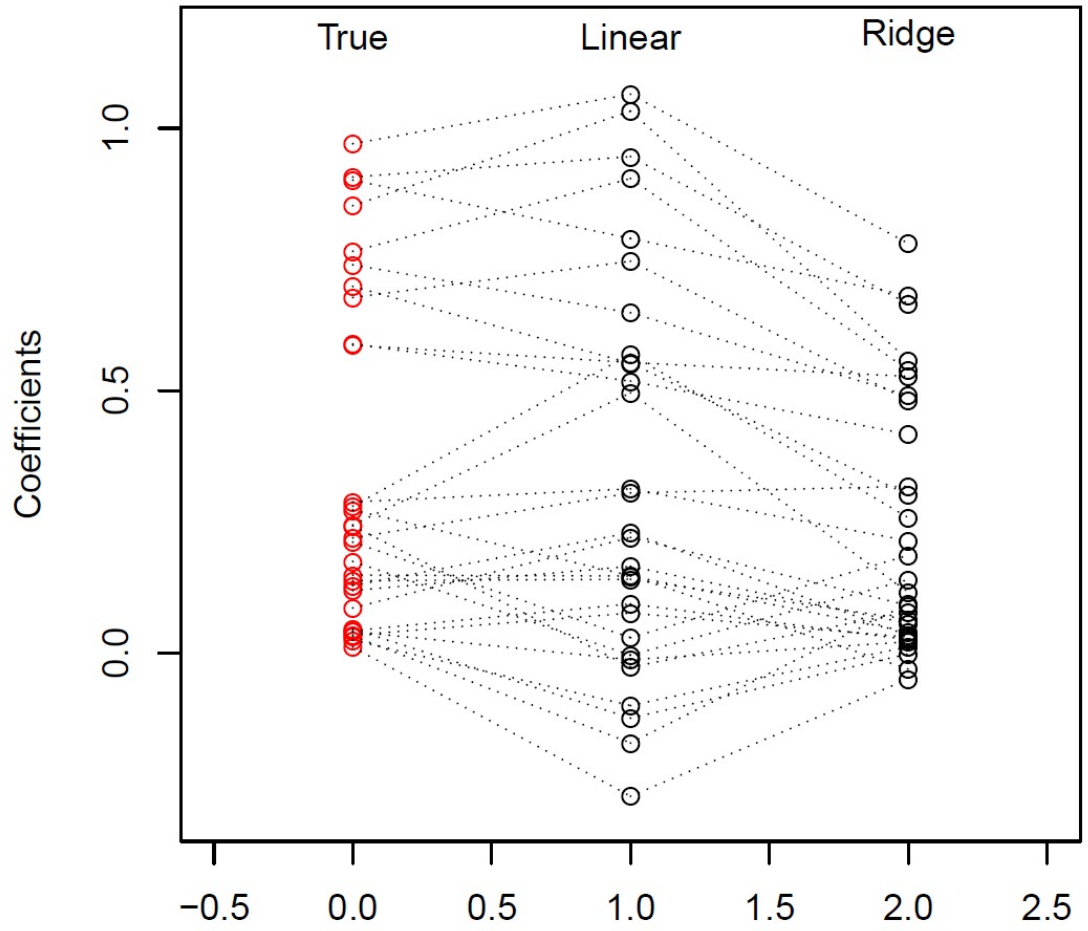
$$\beta^{Ridge} = \arg\min_{\beta \in \Re^p} (Y - X\beta)^T (Y - X\beta) + \boldsymbol{\lambda}\beta^T\beta$$
$$= \min_{\beta \in \Re^p} \|Y - X\beta\|_2^2 + \boldsymbol{\lambda} \underbrace{\|\beta\|_2^2}_{\boldsymbol{penalty}}$$

$\lambda \geq 0$ is the regularization coefficient, i.e. a tuning parameter, which controls the strength of the penalty term. Note that:

when $\lambda = 0$, we get the linear regression estimate
when $\lambda \to \infty$, we get $\beta^{Ridge} = 0$

$$\beta^{Ridge} = \left(X^T X + \lambda I_{pxp}\right)^{-1} X^T Y = \left(I_{pxp} + \lambda (X^T X)^{-1}\right)\beta^{LS}$$

Ridge regression is biased: ($n = 50$, $p = 30$, and $\sigma^2 = 1$; 10 large true coefficients, 20 small). Here is a visual representation of the ridge regression coefficients for $\lambda = 25$:



IMPORTANT DETAILS

When including an intercept term in the regression, we usually leave this coefficient unpenalized.

Hence ridge regression with intercept solves

$$\beta_0, \beta^{Ridge} = \underset{\beta_0 \in \Re, \beta \in \Re^p}{\arg\min} \|Y - \beta_0 - X\beta\|_2^2 + \boldsymbol{\lambda} \|\beta\|_2^2$$

penalty

Used in the code

If we center the columns of X, then the intercept estimate ends up just being $\beta_0 = \bar{Y}$, so it is usually assumed that Y, X have been centered and we don't include an intercept

Also, the penalty term $\|\beta\|_2^2 = \sum_{j=1}^{p} \beta_j^2$ is unfair is the predictor variables are not on the same scale. Therefore, if we know that the variables are not measured in the same units, we typically scale the columns of X (to have sample variance 1), and then we perform ridge regression.

- λ is the shrinkage parameter
  - λ controls the size of the coefficients
  - λ controls amount of **regularization**
  - As $\lambda \downarrow 0$, we obtain the least squares solutions
  - As $\lambda \uparrow \infty$, we have $\hat{\boldsymbol{\beta}}^{\text{ridge}}_{\lambda=\infty} = 0$ (intercept-only model)

Methods for λ selection:

1. ridge traces (in their original paper, Hoerl and Kennard)
2. Discrepancy Principle (DP)
3. Generalized cross validation (GCV)
4. The L-curve criterion
5. The NCP method
6. Leave-one-out Cross Validation

CROSS VALIDATION

E' una tecnica statistica usata per valutare la bontà di performance di un modello (e non solo). Il concetto fondamentale è quelo di suddividere il data set in due parti:

training set
validation set

Il fit del modello viene eseguito nel training set (Y_1), i parametri fissati alle stime ottenute, e il modello ri-utilizzato non per la stima ma solo per il passo di predizione su Y_2:

$$1)\ Y_1 = X\beta + \varepsilon \quad \Longrightarrow \quad \hat{\beta}^{LS}$$

$$2)\ calcolo\ i\ redidui\ \ Y_2 - X\hat{\beta}^{LS}$$

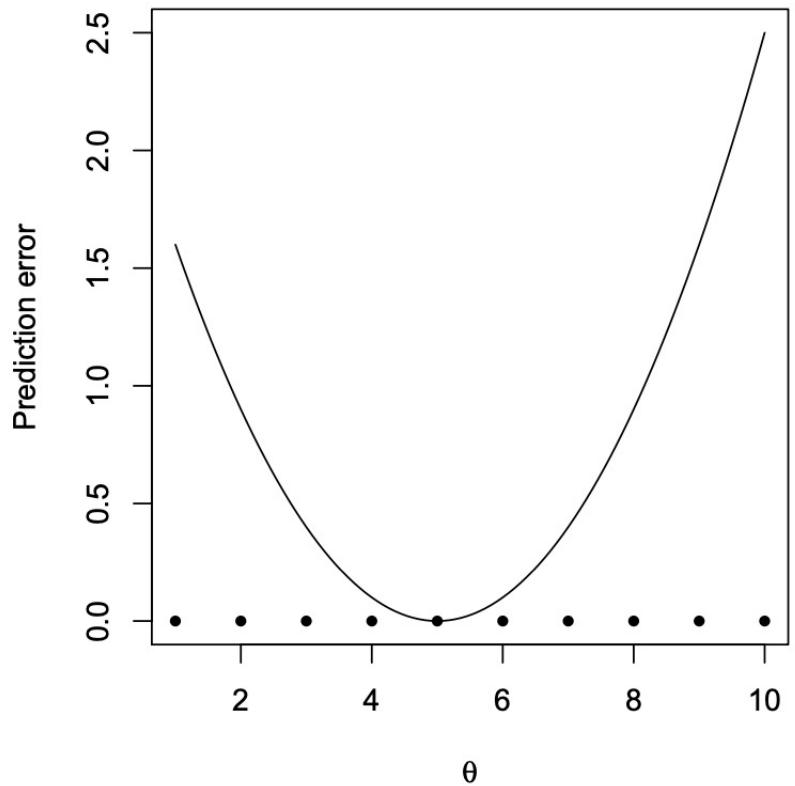
$$3)\ calcolo\ il\ MSE = \left[Y_2 - X\hat{\beta}^{LS}\right]^T \left[Y_2 - X\hat{\beta}^{LS}\right]$$

Cross-validation

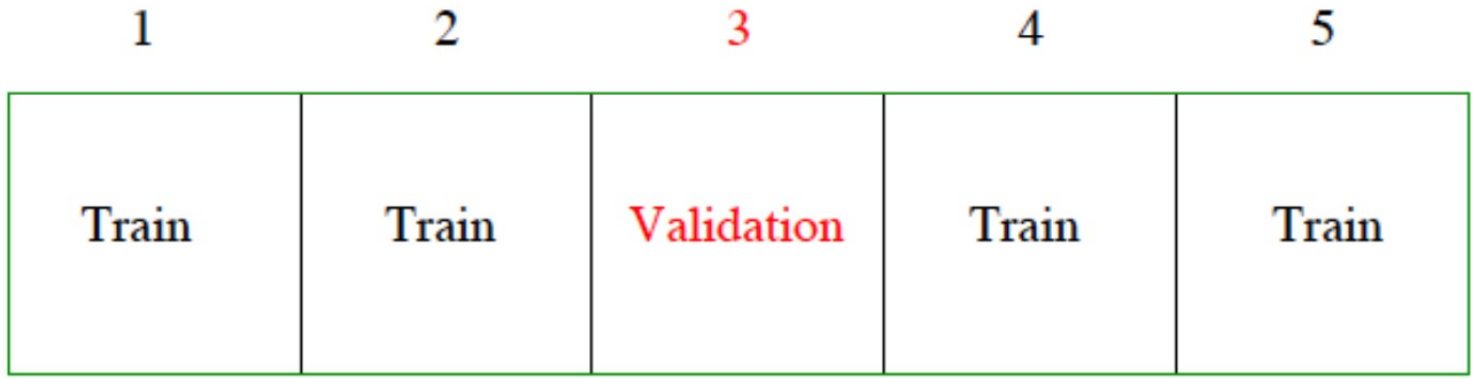
Cross-validation is a simple, intuitive way to estimate prediction error

Given training data (x_i, y_i), $i = 1, \ldots n$ and an estimator $\hat{f}_\theta$, depending on a tuning parameter θ

Even if θ is a continuous parameter, it's usually not practically feasible to consider all possible values of θ, so we discretize the range and consider choosing θ over some discrete set $\{\theta_1, \ldots \theta_m\}$



For a number K, we split the training pairs into K parts or "folds" (commonly $K = 5$ or $K = 10$)



K-fold cross validation considers training on all but the kth part, and then validating on the kth part, iterating over $k = 1, \ldots K$

(When $K = n$, we call this leave-one-out cross-validation, because we leave out one data point at a time)

K-fold cross validation procedure:

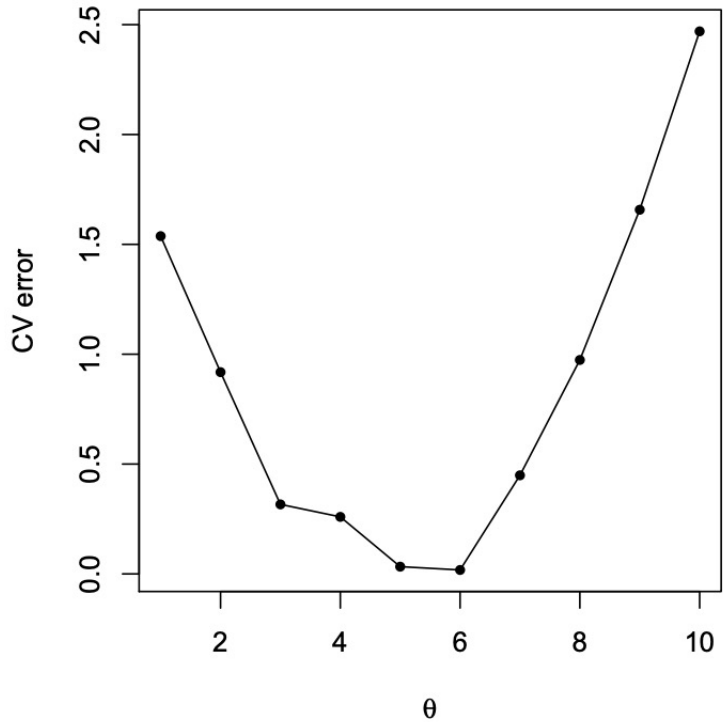
- Divide the set $\{1, \ldots n\}$ into K subsets (i.e., folds) of roughly equal size, $F_1, \ldots F_K$
- For $k = 1, \ldots K$:
  - Consider training on (x_i, y_i), $i \notin F_k$, and validating on (x_i, y_i), $i \in F_k$
  - For each value of the tuning parameter $\theta \in \{\theta_1, \ldots \theta_m\}$, compute the estimate $\hat{f}_\theta^{-k}$ on the training set, and record the total error on the validation set:

$$e_k(\theta) = \sum_{i \in F_k} \left(y_i - \hat{f}_\theta^{-k}(x_i)\right)^2$$

- For each tuning parameter value θ, compute the average error over all folds,

$$\mathrm{CV}(\theta) = \frac{1}{n} \sum_{k=1}^{K} e_k(\theta) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in F_k} \left(y_i - \hat{f}_\theta^{-k}(x_i)\right)^2$$

Having done this, we get a cross-validation error curve $\mathrm{CV}(\theta)$ (this curve is a function of θ), e.g.,



and we choose the value of tuning parameter that minimizes this curve,

$$\hat{\theta} = \underset{\theta \in \{\theta_1, \ldots \theta_m\}}{\operatorname{argmin}} \ \mathrm{CV}(\theta)$$

LASSO REGRESSION

Tibshirani (Journal of the Royal Statistical Society 1996) introduced the LASSO: least absolute shrinkage and selection operator

LASSO coefficients are the solutions to the ℓ_1 optimization problem:

$$\text{minimize } (\mathbf{y} - \mathbf{Z}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}) \text{ s.t. } \sum_{j=1}^{p} |\beta_j| \leq t$$

Again, we have a tuning parameter λ that controls the amount of regularization

One-to-one correspondence with the threshhold t:
recall the constraint:

$$\sum_{j=1}^{p} |\beta_j| \leq t$$

The only difference is instead of taking the square of the coefficients, magnitudes are taken into account.

This type of regularization (L1) can lead to zero coefficients i.e. some of the features are completely neglected for the evaluation of output.

So Lasso regression not only helps in reducing over-fitting but it can help us in feature selection.

LASSO regression is like least squares but shrinks the estimated coefficients towards zero:

$$\beta^{Lasso} = \min_{\beta \in \Re^p} \|Y - X\beta\|_2^2 + \boldsymbol{\lambda} \underbrace{|\beta|}_{\boldsymbol{penalty}}$$

$\lambda \geq 0$ is the regularization coefficient, i.e. a tuning parameter, which controls the strength of the penalty term. Note that:

when $\lambda = 0$, we get the linear regression estimate
when $\lambda \to \infty$, we get $\beta^{Lasso} = 0$

- Unlike ridge regression, LASSO has no closed form
- Original implementation involves quadratic program techniques from convex optimization

Often, we believe that many of the β_j's should be 0

Hence, we seek a set of **sparse solutions**

Large enough λ (or small enough t) will set some coefficients exactly equal to 0!

- So the LASSO will perform model selection for us!

ELASTIC NET REGRESSION

Elastic-Net penalty is given by a combination of L_1 and L_2 penalties, and that simultaneously does automatic variable selection, shrinks the coefficients and can select groups of correlated variables, while LASSO usually tends to select only one variable from these groups; hence it seems that Elastic-Net performs better than LASSO in terms of prediction accuracy.

The optimization problem to solve in this types of regression is

$$\hat{\beta} = argmin_{\beta} \frac{1}{n} \sum_{i=1}^{n} (Y_i - X\beta_i)^2 + \lambda[\alpha \sum_{j=1}^{p} |\beta_j| + (1-\alpha) \sum_{j=1}^{p} \beta_j^2]$$

where α controls the influence of L_1-penalty and L_2-penalty and λ is the tuning parameter. Notice that even in Elastic-Net regression we are working with a convex objective function.

Moreover, we can write the optimization problem

$$\hat{\beta} = argmin_{\beta} \frac{1}{n} \sum_{i=1}^{n} (Y_i - X\beta_i)^2 \qquad \text{subject to} \quad \alpha \sum_{j=1}^{p} |\beta_j| + (1-\alpha) \sum_{j=1}^{p} \beta_j^2 \leq t$$

Hence if $\alpha = 1$ we perform the LASSO regression, if $\alpha = 0$ the RIDGE regression and if $\alpha \in (0, 1)$ the Elastic-Net regression.