Business Statistics

Tommaso Proietti

DEF - Università di Roma 'Tor Vergata'

Model Evaluation and Selection

Predictive Ability of a Model: Definition and Estimation

- We aim at achieving a balance between parsimony (model complexity) and goodness of fit.
- When we increase model complexity (e.g. by including further regressors) we improve the fit within the training sample, but the improvement is not necessarily generalisable outside the sample (for predictive purposes).
- The least squares (LS) estimates have high variability when the number of inputs is large. This results in inaccurate predictions.
- A parsimonious model (adhering to Occam's Razor: *Entia non sunt multiplicanda praeter necessitatem*) is estimated precisely and is easily interpretable, but is potentially biased.

Hypothesis testing is one approach to model selection. The problem is to control the size of the test procedure when a sequence of correlated tests is carried out.
A more fruitful approach is to estimate the predictive ability of a model and select the one that maximises it.

Learning objectives of this unit:

- Define and understand the relation between model complexity and predictive ability.
- Validation of a model in terms of predictive ability.
- Define operational criteria for model selection.
- Illustrate alternative model selection strategies (subset selection, regularization).

Training sample

Assume that the true data generating process is $Y = f(X) + \varepsilon$.
The training sample of size N, $\mathbf{y}$, is generated as above, that is

$$\mathbf{y} = \mathbf{f} + \boldsymbol{\varepsilon}, \mathsf{Var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}.$$

The true regression function $f(X)$ is unknown. We estimate it as a linear function of a set of measurable characteristics in $\mathbf{X}$.
From the training sample we estimate $\mathbf{f}$ as

$$\hat{\mathbf{f}} = \mathbf{X}\hat{\boldsymbol{\beta}}, \qquad \hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

The fitted values are $\hat{\mathbf{y}} = \hat{\mathbf{f}} = \mathbf{H}\mathbf{y}$, where $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ and the residuals are $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{M}\mathbf{y}$, $\mathbf{M} = \mathbf{I} - \mathbf{H}$.

The estimation of the regression function $\mathbf{f}$ using $\hat{\mathbf{f}}$ faces a fundamental *bias-variance trade-off*.

- The statistical accuracy of $\hat{f}_i$ as an estimator of f_i, for $i = 1, \ldots, N$, is measured by $\mathsf{MSE}(\hat{f}_i) = \mathsf{E}[(\hat{f}_i - f_i)^2]$.
- The MSE has two components: $\mathsf{MSE}(\hat{f}_i) = \mathsf{Bias}^2(\hat{f}_i) + \mathsf{Var}(\hat{f}_i)$
- We can reduce the bias by increasing the complexity of the model, which in turn determines an increase of the variance.
- It can be shown that $\mathsf{Var}(f_i) = \sigma^2 h_i$ (it is more difficult to predict accurately observations that are remote in the input space).
- The bias term depends on $f(X)$ being nonlinear and on the omission of relevant inputs.
- We denote the bias $b_i = \mathsf{E}(\hat{f}_i) - f_i$.

Figure : The true regression function is $f(x) = 4 + 1.5x + 3.2x^2 - 4x^3$. The fitted model contains the intercept and x (misspecification). Plot of Bias($\hat{f}_i$) and Var($\hat{f}_i$) $= \sigma^2 h_i$

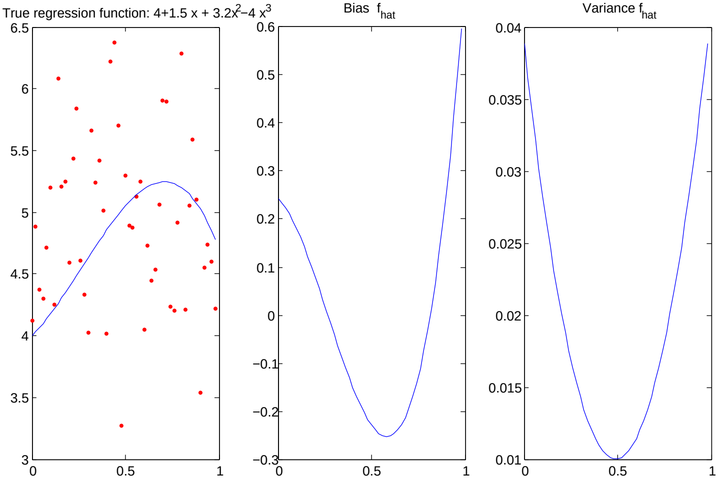
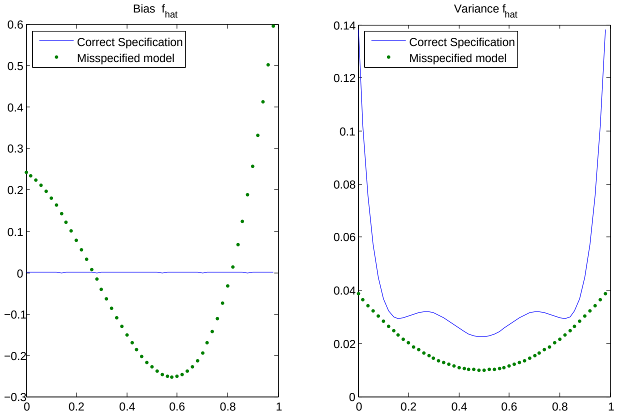


Figure : The true regression function is $f(x) = 4 + 1.5x + 3.2x^2 - 4x^3$. The fitted model contains the intercept, x, x^2, and x^3 (Correctly specified model). Plot of Bias($\hat{f}_i$) and Var($\hat{f}_i$) $= \sigma^2 h_i$



Training sample and training error

The LS residuals $\mathbf{e}$ are used to assess the goodness of the training sample fit. Recall their definition:

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{H}\mathbf{y} = (\mathbf{I} - \mathbf{H})\mathbf{y} = \mathbf{M}\mathbf{y}$$

We define the **training error**

$$\overline{err} = \frac{1}{N}\mathbf{e}'\mathbf{e} = \frac{1}{N}\sum_i e_i^2 = \frac{1}{N}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \frac{RSS}{N}$$

The model that maximises the fit for the training sample is the one for which the training error is a minimum. However, what is best for in-sample fit is not best for out-of-sample prediction (it does not generalize to a different test sample), as we shall see shortly.

The **expected training error**

$$\begin{aligned} \mathsf{E}(\overline{err}|X) &= \mathsf{E}\left(\frac{1}{N}\sum_i e_i^2\right) = \frac{1}{N}\sum_i \mathsf{E}(e_i^2) \\ &= \frac{1}{N}\sum_i (b_i^2 + \mathsf{Var}(e_i)) \\ &= \frac{1}{N}\sum_i b_i^2 + \sigma^2 - \sigma^2\frac{p+1}{N} \end{aligned}$$

is a downward biased (i.e. an optimistic) estimate of the expected test error. It appears as if only accuracy gains accrue from increasing the complexity of the model.

Test sample and test error

Consider drawing, for every training sample $(\mathbf{y}, \mathbf{X})$, a test sample $\mathbf{y}^*$ of size N from the same population, independently of $\mathbf{y}$ and matching the same X's (again this is quite unrealistic, but it simplifies the analysis considerably), so that the systematic part is $\mathbf{f}$, and $\mathbf{y}^* = \mathbf{f} + \boldsymbol{\varepsilon}^*$.

Using $\mathbf{X}$ we aim at predicting $\mathbf{f}$ and $\mathbf{y}^*$ for the new units. The optimal predictor based on the linear model is $\hat{\mathbf{y}}^* = \mathbf{X}\hat{\boldsymbol{\beta}}$ ($\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}\mathbf{y}$ has already been estimated from the training sample, whereas $\mathbf{y}^*$ is not used for fitting).
The test sample prediction error is

$$\mathbf{e}^* = \mathbf{y}^* - \hat{\mathbf{y}}^*$$

The **test error** (average prediction error over the test sample)

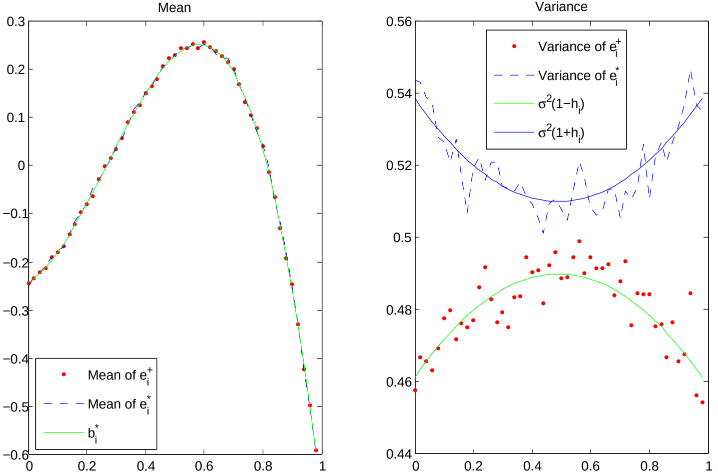
$$\mathsf{Err}_{in} = \frac{1}{N}\mathbf{e}^{*'}\mathbf{e}^{*} = \frac{1}{N}\sum_{i} e_i^{2*},$$

has expected value

$$\mathsf{E}(\mathsf{Err}_{in}|X) \quad = \quad \tfrac{1}{N}\textstyle\sum_i b_i^2 + \sigma^2 + \sigma^2\tfrac{p+1}{N} = \mathsf{E}(\overline{err}|X) + 2\tfrac{p+1}{N}\sigma^2$$

In fact, there are two independent sources of error: the error that accrues from estimating the true $f(X)$ using the training sample, and the intrinsic variation induced by ϵ^*.
We wish to select the model which yields the minimum expected test error $\mathsf{E}(\mathsf{Err}_{in}|X)$. The main issue deals with the estimation of $\mathsf{E}(\mathsf{Err}_{in}|X)$. We wish to use the training sample for this purpose.

Figure : The true regression function is $f(x) = 4 + 1.5x + 3.2x^2 - 4x^3$. The fitted model contains the intercept and x (misspecification). Mean and variance of the training sample residuals e_i and of the test sample prediction errors e_i^*.



- We have shown that the expected training error

$$\mathsf{E}(\overline{err}|X) \quad = \quad \mathsf{E}(\mathsf{Err}_{in}|X) - 2\sigma^2\tfrac{p+1}{N}$$

  is a downward biased (i.e. an optimistic) estimate of the expected test error.
- We define the "Optimism" as: $\mathsf{op} = \mathsf{Err}_{in} - \overline{err}$
- The expected optimism is $\mathsf{E}(\mathsf{op}) = 2\frac{p+1}{N}\sigma^2$.
- Notice that $\frac{p+1}{N} = \frac{1}{N}\mathsf{trace}(\mathbf{H})$.
- The trace is the sum of the values on the diagonal of a matrix.
- Number of **degrees of freedom** used for fitting (model complexity): $\mathsf{trace}(\mathbf{H})$.

- It also can be shown that

  $$\mathsf{E(op)} = \frac{2}{N}\mathsf{trace}\left(\mathsf{Cov}(\mathbf{y}, \hat{\mathbf{y}}|\mathbf{X})\right)$$

  In fact, $\mathsf{Cov}(\mathbf{y}, \hat{\mathbf{y}}|\mathbf{X}) = \sigma^2\mathbf{H}$
  Interesting interpretation: $\mathsf{Cov}(\mathbf{y}, \hat{\mathbf{y}}|\mathbf{X})$ measures overfitting.
  The larger the covariance between the fitted and the observed values, the more the model is likely to overfit.
- Notice that Err_{in} measures the bias-variance trade-off: increasing the complexity reduces the bias component, but it inflates the variance via the term depending on p.

Criteria for Model Selection

A model selection procedure should focus on the expected prediction error.

In practice we have to estimate σ^2. We can use $\tilde{\sigma}^2 = RSS_{p_{max}}/(N - p_{max} - 1)$ where p_{max} is the largest p considered

Popular criteria are:

- Mallows' $C_p = \overline{err} + 2\frac{p+1}{N}\tilde{\sigma}^2 = \frac{RSS}{N} + 2\frac{p+1}{N}\tilde{\sigma}^2$ (this is obtained from E($\text{Err}_{in}|X$) by replacing $\tilde{\sigma}^2$ for σ^2).
- Akaike information criterion: same as Mallow's C_p statistic,

$$AIC = \ln \hat{\sigma}^2 + 2\frac{p+1}{N}$$

  (Taking logs of C_p and by 1st order Taylor approximation).
- Bayesian information criterion:

$$BIC = \ln \hat{\sigma}^2 + \frac{p+1}{N} \ln N$$

  The factor 2 is replaced by $\ln N$, so that for $N > 8$, BIC penalises complex models more heavily.
- Cross validation

These criteria penalize model complexity.

Cross-validation

Method that estimates the average generalization prediction error.
Useful when we are unable to evaluate the expected optimism.
Let us consider leave-one-out CV.
The CV criterion is

$$CV = \sum_{i=1}^{N} \left(y_i - \hat{y}_{(i)} \right)^2$$

where $\hat{y}_{(i)}$ is the prediction of y_i obtained using all the remaining observations. Indeed, we saw that the problem with RSS is that the same y is used for fitting and assessing the gof.

In a linear regression framework,

$$CV = \sum_i \left(y_i - \hat{y}_{(i)}\right)^2 = \sum_i \frac{e_i^2}{(1-h_i)^2}$$

Recall that

$$\frac{1}{N} \leq h_i \leq 1, \quad \frac{1}{N}\sum_i h_i = \frac{p+1}{N}$$

Replacing h_i by their average, we get the generalized CV criterion:

$$GCV = \frac{RSS}{\left(1 - N^{-1}\text{trace}(\mathbf{H})\right)^2}$$

Considering the first order Taylor approximation,

$$GCV \approx RSS\left(1 + 2\frac{p+1}{N}\right)$$

and thus GCV/N is approximately equal to C_p.

In a more general setting CV works as follows:

- The sample is divided into K segments of equal size
- For $k = 1, \ldots, K$, we fit the model using the other $K - 1$ segments, construct the predictor $\hat{\mathbf{y}}_{(k)}$, and calculate the prediction error

$$\mathbf{e}_{(k)} = \mathbf{y}_k - \hat{\mathbf{y}}_{(k)}.$$

- The CV score is computed as

$$CV = \frac{1}{N} \sum_k \mathbf{e}'_{(k)} \mathbf{e}_{(k)}$$

Model selection

Model selection refers to choice of $\mathbf{y}$ (transformation of the response variable), $\mathbf{X}$ (selection of the inputs and tranformation of the inputs), choice of an estimation method and of a prediction rule $\hat{f}(X)$.

- Subset selection methods deal with the choice of the X's.
- Shrinkage methods deal with the choice of the prediction rule.
- Methods using derived input directions deal with deriving linear or nonlinear combinations of the inputs that summarise their information.

I. Subset selection

In principle we could fit all the possible models and select the one that has minimum C_p or AIC.

However, if there are p regressors, there are 2^p competitor models (all of them including the intercept). If $p = 10$, $2^p = 1024$. With 50 explanatory variables, there are 1125899907 million models.

We focus on a situation in which the investigator has available a number of potential inputs that is too large, either because $p \geq N$ or 2^p is unfeasible.

I1. Best subset selection

- For each $k \in \{0, 1, \ldots, p\}$, BSS selects the subset of size k among the possible candidates that minimise the RSS, exploring all the possibilities.
- Select k that minimises the expected prediction error (C_p, AIC, etc)

I2. Stepwise selection

1. Forward stepwise Selection

    1.1 Start with a model containing only the intercept

    1.2 Add the input which improves the fit (max t-statistic, or F-stat from addition, or R^2).
    This is also the variable with largest squared correlation with the residuals of the previous step regression.

    1.3 select the model with minimum AIC or C_p

2. Backward stepwise selection.

    2.1 Start with the full model (requires $p < N$).

    2.2 Drop the variable with smallest t-statistic.

    2.3 Select model with smallest C_p, AIC, etc.

3. An hybrid strategy can be implemented (add or delete according to AIC - step function in R).

II. Shrinkage Methods

- We do not select a subset of the inputs.
- These methods yield an estimate of $f(X)$ depending on a regularization parameter which regulates the amount of shrinkage of the regression coefficients to zero.
- These methods trade-off some increased bias for a reduction in the variance.

II.1. Ridge Regression

The ridge regression estimator is the minimizer of the penalised LS criterion

$$(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} \beta_j^2$$

$\lambda \geq 0$ is a penalty parameter which regulates the tradeoff.
$\lambda = 0$ yields the usual LS criterion, as a particular case.
The second addend penalises the departure from zero of the regression parameter and shrinks them toward zero.
Now, zero is a sensible target if the the variables have the same scale.
Thus we assume that both the response and the inputs are standardized (the regression model does not include the intercept and $\mathbf{X}'\mathbf{X}$ is N times the correlation matrix of the inputs).

Redefining $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$,

$$RSS_\lambda(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}'\boldsymbol{\beta},$$

the minimiser of $RSS_\lambda(\boldsymbol{\beta})$ wrt $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}}_r = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$$

Assume that $\mathbf{f} = \mathbf{X}\boldsymbol{\beta}$ (the model is correctly specified: the systematic part is linear in the available X's).

- $\hat{\boldsymbol{\beta}}_r$ is biased: $\mathsf{E}(\boldsymbol{\beta}_r) = [\mathbf{I} + \lambda(\mathbf{X}'\mathbf{X})]^{-1}\boldsymbol{\beta}$
- The variance is smaller than that of the LS estimator

Choice of λ

λ regulates model complexity. $\lambda = 0$ corresponds to the greatest complexity (bias is a minimum, but variance is high). As λ increases we increase the bias at the advantage of precision.

We choose λ so as to minimise the (estimated) expected test error:

$$\frac{RSS}{N} + 2\sigma^2 \frac{\text{trace}(\mathbf{H}_\lambda)}{N}.$$

The fitted values are

$$\hat{y}_r = \mathbf{X}\hat{\boldsymbol{\beta}}_r = \mathbf{X}(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{H}_\lambda\mathbf{y}.$$

Denoting by d_i^2 the eigenvalues of $\mathbf{X}'\mathbf{X}$, we define the effective degrees of freedom as

$$df(\lambda) = \text{trace}(\mathbf{H}_\lambda) = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}$$

The eigenvalues of $\mathbf{X}'\mathbf{X}$ are positive p scalars that are relevant for characterizing a matrix. Usually, they are ordered s.t.
$d_1^2 \geq d_2^2 \geq \cdots \geq d_p^2$.
They have several important properties.

- The number of non zero eigenvalues is equal to the rank.
- The trace of a matrix is equal to the sum of the eigenvalues: $\text{trace}(\mathbf{X}'\mathbf{X}) = \sum_j d_j^2$.
- For each eigenvalue, there exists a vector $\mathbf{v}_j$, called eigenvector, s.t. $(\mathbf{X}'\mathbf{X})\mathbf{v}_j = d_j^2\mathbf{v}_j$.
- They are orthogonal and with unit norm: $\mathbf{v}_j'\mathbf{v}_j = 1, \mathbf{v}_j'\mathbf{v}_h = 0, h \neq j$.
- If we collect the eigenvectors $\mathbf{v}_j$ in the matrix $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_p]$, $\mathbf{V}'\mathbf{V} = \mathbf{I}$ and define $\mathbf{D} = \text{diag}(d_1^2, d_2^2, \ldots, d_p^2)$, then we can decompose $\mathbf{X}'\mathbf{X} = \mathbf{VDV}'$ (spectral decomposition).

Figure : Housing dataset: $p = 20$. Coefficient profiles as a function of λ.

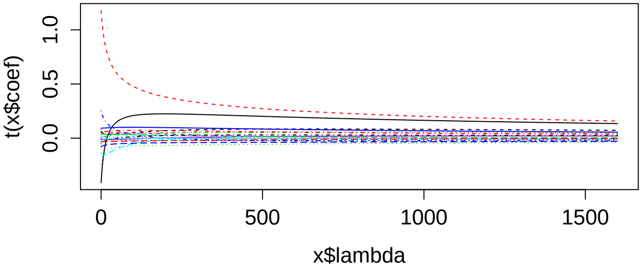
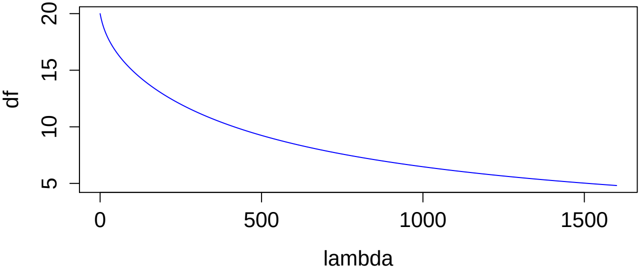


Figure : Housing dataset: $p = 20$. Degrees of freedom $df(\lambda) = \text{trace}(\mathbf{H}_\lambda)$ as a function of λ.



II.2 Lasso

The lasso (least absolute shrinkage and selection operator) estimator is the minimizer of the penalised LS criterion

$$(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j|,$$

where λ is a penalty parameter, or, equivalently, it is obtained as the solution to the constrained minimization problem:

$$\text{argmin}_{\beta} \left\{ (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right\}, s.t. \sum_{j=1}^{p} |\beta_j| < t,$$

where t is a tuning parameter.

- No closed form solution for $\hat{\boldsymbol{\beta}}_l$ (solve a quadratic programming problem).
- Lasso performs variable selection and shrinkage. Coefficients are forced to zero as t decreases (effectively a subset selection).
- We assume that both the response and the inputs are standardized (the regression model does not include the intercept).

The single predictor case: soft-thresholding

Let us consider a training sample $\{x_i, y_i\}$ on two standardized variables ($\bar{x} = \bar{y} = 0$ and $\sum x_i^2/N = \sum_i y_i^2/N = 1$).
We aim at estimating the model $y_i = \beta x_i + \epsilon_i$, subject to the constraint $|\beta| < t$. This is equivalent to

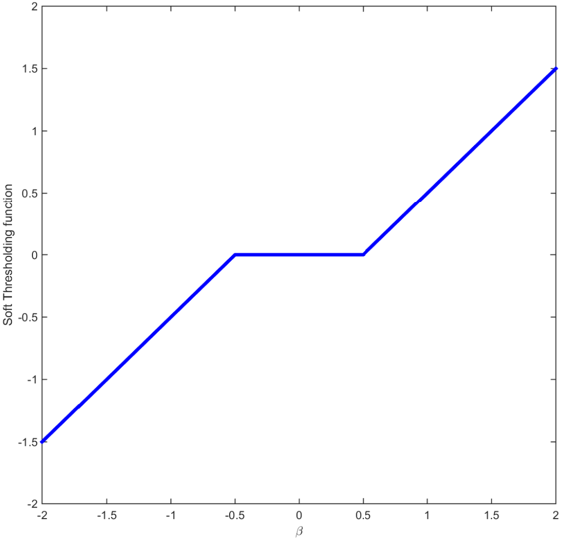
$$\min_{\beta} \left\{ \sum_{i=1}^{N} (y_i - \beta x_i)^2 + \lambda |\beta| \right\} .$$

If $\hat{\beta} = \sum_i x_i y_i / N$ denotes the least square estimate (i.e. the value that is obtained if $\lambda = 0$), then the lasso estimate is

$$\hat{\beta}_L = \begin{cases} \hat{\beta} - \lambda, & \text{if } \hat{\beta} > \lambda \\ 0, & \text{if } -\lambda \leq \hat{\beta} \leq \lambda \\ \hat{\beta} + \lambda, & \text{if } \hat{\beta} < -\lambda \end{cases}$$

which can compactly be written $\hat{\beta}_L = \text{sign}(\hat{\beta}) \max\{|\hat{\beta}| - \lambda, 0\}$.

Figure : Soft-Thresholding operator for $\lambda = 0.5$



In the multiple predictor case, the lasso solution can be computed using the *Cyclic Coordinate Descent* algorithm.

This repeatedly cycles through the predictors in some fixed (but arbitrary) order (say $j = 1, 2, \ldots, p$). At the j-th step, the coefficient β_j is updated by minimising with respect to β_j the objective function

$$\left\{ \sum_{i=1}^{N} (y_i - \sum_{k \neq j} \beta_k x_{ik} - \beta_j x_{ij})^2 + \lambda \sum_{k \neq j} |\beta_k| + \lambda |\beta_j| \right\},$$

holding fixed all other coefficients at their current values.

Letting $\hat{\delta}_j = N^{-1} \sum_i x_{ij}(y_i - \sum_{k \neq j} \hat{\beta}_{kL} x_{ik})$, the generic coefficient is updated as $\hat{\beta}_{j,L} = \text{sign}(\hat{\delta}_j) \max\{|\hat{\delta}_j| - \lambda, 0\}$.

II.3 Incremental Forward-Stagewise Regression

This stands half way from subset selection and shrinkage.

1. All the variables are standardized.
2. Set the residuals $\mathbf{r} = \mathbf{y}$ and $\hat{\beta}_1 = \ldots = \hat{\beta}_p = 0$
3. Find the input $\mathbf{x}_j$ for which $\mathbf{x}_j'\mathbf{r}$ is maximum (max correlation)
4. Update the j-th regression coefficient $\hat{\beta}_j \leftarrow \hat{\beta}_j + \delta_j, \delta_j = \epsilon\,\text{sign}(\mathbf{x}_j'\mathbf{r})$. Update the residuals: $\mathbf{r} \leftarrow \mathbf{r} - \delta_j \mathbf{x}_j$.
5. Repeat until $\max(\mathbf{x}_j'\mathbf{r}) = 0$

The last step provides the OLS estimate of $\boldsymbol{\beta}$. If $\delta_j = \mathbf{x}_j'\mathbf{r}$ we get the Forward-Stagewise Regression (FSR). FSR converges to the LS solution after a number of steps larger than p.

II.4. Least Angle Regression (LAR)

Special case of Stagewise. Rather than using a predictor all at once, it gradually blends predictors.

1. All the variables are standardized.
2. Set the residuals $\mathbf{r} = \mathbf{y}$ and $\hat{\beta}_1 = \ldots = \hat{\beta}_p = 0$
3. Find the input $\mathbf{x}_j$ for which $\mathbf{x}_j'\mathbf{r}$ is maximum (max correlation)
4. Move $\hat{\beta}_j$ from zero towards $\mathbf{x}_j'\mathbf{r}$, until another variable $\mathbf{x}_k$ has the same correlation with the residuals as does $\mathbf{x}_j$. Update $\mathbf{r}$.
5. Move $\hat{\beta}_j$ and $\hat{\beta}_k$ in the direction defined by the LS coefficients obtained by regressing $\mathbf{r}$ on $[\mathbf{x}_j, \mathbf{x}_k]$, until some other variable $\mathbf{x}_l$ has as much correlation with the current residual.
6. Repeat until all p predictors have been entered. The LS solution is obtained after $\min(N-1, p)$ steps.

Interestingly, a modified LAR gives the entire path of lasso solutions: if a non-zero coefficient hits zero, the corresponding predictor is removed from the active set of predictors and the joint direction of the remaining coefficients is recomputed.

III. Methods Using Derived Input Directions

Principal components Regression

Suppose that $\mathbf{X}$ denotes a matrix of p standardized variables.
Principal components regression is based on the regression of $\mathbf{y}$ on new variables, called principal components, obtained from the linear combination of the original ones:

$$\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_p] = \mathbf{X}\mathbf{A}, \quad \mathbf{z}_k = \mathbf{X}\mathbf{a}_k$$

The loadings matrix $\mathbf{A}$ is obtained from the spectral decomposition of the matrix $\mathbf{S} = N^{-1}\mathbf{X}'\mathbf{X}$ (correlation matrix), $\mathbf{S} = \mathbf{A}\boldsymbol{\Lambda}\mathbf{A}'$, where $\mathbf{A}$ is the eigenvector matrix, $\mathbf{A}'\mathbf{A} = \mathbf{I}$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_p)$ is the diagonal matrix collecting the eigenvalues of the covariance matrix $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$.

- The p.c.'s are orthogonal (uncorrelated) and have variance equal to λ_k: $\frac{1}{N}\mathbf{Z}'\mathbf{Z} = \mathbf{\Lambda}$.
- The first component is designed to capture as much of the variability in the data as possible, and the succeeding components in turn extract as much of residual variability as possible.
- If we consider only the first $M \leq p$ variables, then PCR is similar to ridge regression.

Partial Least Squares

1. All the variables are standardized. Set $\mathbf{x}_k^{(0)} = \mathbf{x}_k$, the k-th input, and $\hat{\mathbf{y}}^{(0)} = \mathbf{0}$.
2. For $k = 1, 2, \ldots, p$, and $m = 1, 2, \ldots,$
   2.1 Compute $\hat{\varphi}_{mk} = \mathbf{x}_k^{(m-1)'}\mathbf{y}$ and form the derived input variable

   $$\mathbf{z}_m = \sum_{k=1}^{p} \hat{\varphi}_{mk}\mathbf{x}_k^{(m-1)}$$

   2.2 Regress $\mathbf{y}$ on $\mathbf{z}_m$ to obtain:

   $$\hat{\theta}_m = \frac{\mathbf{z}'_m\mathbf{y}}{\mathbf{z}'_m\mathbf{z}_m}$$

   2.3 Update the fitted values

   $$\hat{\mathbf{y}}^{(m)} = \hat{\mathbf{y}}^{(m-1)} + \hat{\theta}_m\mathbf{z}_m$$

   2.4 Obtain a new set of $\mathbf{x}_k^{(m)}$ by orthogonalizing each of the predictors $\mathbf{x}_k^{(m-1)}$ with respect to $\mathbf{z}_m$:

   $$\mathbf{x}_k^{(m)} = [\mathbf{I} - \mathbf{z}_m(\mathbf{z}'_m\mathbf{z}_m)^{-1}\mathbf{z}'_m]\mathbf{x}_k^{(m-1)}$$

3. Output the sequence of fitted vectors $\hat{\mathbf{y}}^{(m)}$. Since $\mathbf{z}_m$ is linear in the inputs, we can express $\hat{\mathbf{y}}^{(m)} = \mathbf{X}\boldsymbol{\beta}_{pls}(m)$. The coefficients $\boldsymbol{\beta}_{pls}(m)$ can be recovered from the sequence of PLS transformations.

Example: eigenvalues and eigenvectors of a correlation matrix

```
X = scale(data.frame(sqft,sqft^2,Age,Age^2,sqft*Age,Baths,Bedrooms));

S = cor(X) # correlation matrix
            sqft sqft.2    Age  Age.2 sqft...Age  Baths Bedrooms
sqft        1.00  0.952 -0.138 -0.100      0.332  0.716     0.68
sqft.2      0.95  1.000 -0.091 -0.068      0.338  0.668     0.60
Age        -0.14 -0.091  1.000  0.927      0.813 -0.293    -0.17
Age.2      -0.10 -0.068  0.927  1.000      0.755 -0.228    -0.12
sqft...Age  0.33  0.338  0.813  0.755      1.000  0.071     0.16
Baths       0.72  0.668 -0.293 -0.228      0.071  1.000     0.58
Bedrooms    0.68  0.603 -0.168 -0.120      0.159  0.579     1.00
```

```
eigen(S)

$values
[1] 3.258 2.687 0.459 0.364 0.145 0.055 0.031

$vectors
      [,1]   [,2]   [,3]   [,4]    [,5]   [,6]    [,7]
[1,] -0.52 -0.122  0.213 -0.268 -0.0858  0.377  0.6670
[2,] -0.50 -0.140  0.362 -0.346 -0.2912 -0.404 -0.4814
[3,]  0.18 -0.566 -0.042  0.069 -0.0054 -0.660  0.4521
[4,]  0.16 -0.556 -0.100  0.197 -0.6349  0.422 -0.1886
[5,] -0.08 -0.577  0.086 -0.071  0.7076  0.261 -0.2812
[6,] -0.47  0.025  0.142  0.866  0.0617 -0.075 -0.0093
[7,] -0.44 -0.050 -0.886 -0.099  0.0108 -0.077 -0.0650
```