Fundamentals of machine learning

Course 3: linear regression

Mathurin Massias & Titouan Vayer
email: mathurin.massias@inria.fr, titouan.vayer@inria.fr

February 3, 2025

ENS DE LYON

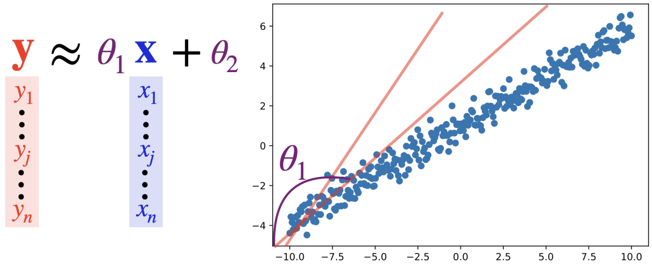
Linear regression

A linear regression model assumes that the regression function f is linear in the inputs $\mathbf{x} = (x_1, \dots, x_d)^T \in \mathbb{R}^d$, or that a linear function is a good approximation to it:

$$y = f(\mathbf{x}) \sim \theta^\top \mathbf{x} + \theta_0, \quad \theta \in \mathbb{R}^d, \theta_0 \in \mathbb{R}$$

1D toy example

$\{x_j\} \in \mathbb{R}, \{y_j\} \in \mathbb{R}$. Goal: find the linear function $f : \mathbb{R} \to \mathbb{R}$ that "best predicts" y_j from x_j for all j.



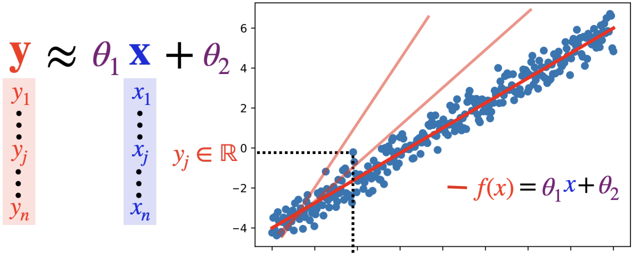
Linear regression

A linear regression model assumes that the regression function f is linear in the inputs $\mathbf{x} = (x_1, \ldots, x_d)^T \in \mathbb{R}^d$, or that a linear function is a good approximation to it:

$$y = f(\mathbf{x}) \sim \theta^\top \mathbf{x} + \theta_0, \quad \theta \in \mathbb{R}^d, \theta_0 \in \mathbb{R}$$

1D toy example

$\{x_j\} \in \mathbb{R}, \{y_j\} \in \mathbb{R}$. Goal: find the linear function $f : \mathbb{R} \to \mathbb{R}$ that "best predicts" y_j from x_j for all j.



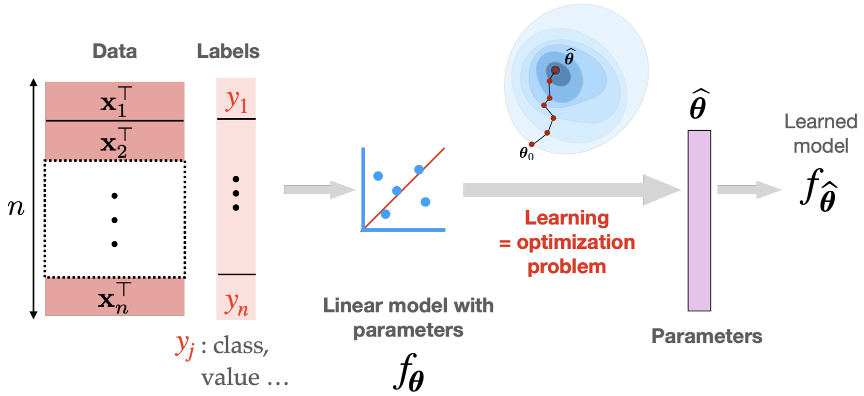
Why linear regression?

Advantages ?

- simple
- interpretable (easy to understand how the input affects the output)
- can outperform nonlinear models in case of: small training sets, sparsity (few important features)
- linear models can be applied to transformations of the inputs

How does linear regression work?

Given training data $\mathbf{x}_i \in \mathbb{R}^d,\ y_i \in \mathbb{R}\ \forall i = 1, \dots, n$



Learning step: find the parameters $\widehat{\theta}$ (hence the function) that **minimizes a measure of error on the data**

$$\min_{\theta} \sum_{i=1}^{n} \ell(f_\theta(\mathbf{x}_i) - \mathbf{y}_i)$$

The linear model

For one sample $\mathbf{x}$:

- Model:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \theta_0 + \sum_{j=1}^{d} x_j \theta_j \in \mathbb{R}$$

- Parameters (unknowns):

$$\boldsymbol{\theta} = (\theta_0, \dots, \theta_d)^T \in \mathbb{R}^{d+1}$$

- Training data:

$$\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\},$$
$$\mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}$$

Notation

$\mathbf{x}_i$: vector

x_i: scalar

For the entire training set:

Denote

$$X = \begin{pmatrix} 1 & \mathbf{x}_1^T \\ 1 & \mathbf{x}_2^T \\ \vdots & \\ 1 & \mathbf{x}_n^T \end{pmatrix} \in \mathbb{R}^{n \times (d+1)}$$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n$$

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = X\boldsymbol{\theta} \sim \mathbf{y}$$

Which measure of error?

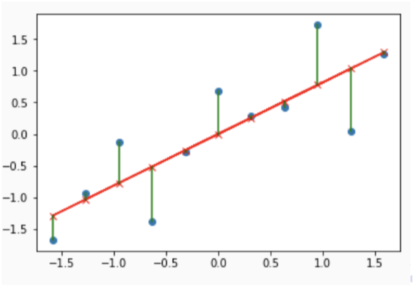
Least squares estimation method

Square loss $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$:

$$\ell(y_i, f_{\boldsymbol{\theta}}(\mathbf{x}_i)) = (y_i - f_{\boldsymbol{\theta}}(\mathbf{x}_i))^2.$$

Error: RSS= residual sum of squares

$$RSS(\boldsymbol{\theta}) = \sum_{i=1}^{n} (y_i - f_{\boldsymbol{\theta}}(\mathbf{x}_i))^2 = \sum_{i=1}^{n} (y_i - \theta_0 - \sum_{j=1}^{d} X_{ij}\theta_j)^2$$



How to minimize the least-squares?

We can write the residual sum of squares as:

$$RSS(\boldsymbol{\theta}) = (\mathbf{y} - X\boldsymbol{\theta})^T(\mathbf{y} - X\boldsymbol{\theta}) = \|\mathbf{y} - X\boldsymbol{\theta}\|^2$$

The learning problem:

$$\min_{\boldsymbol{\theta}} \|\mathbf{y} - X\boldsymbol{\theta}\|^2$$

This admits an explicit solution!
RSS($\boldsymbol{\theta}$) is a quadratic function with derivatives

$$\nabla RSS(\theta) = -2X^T(\mathbf{y} - X\boldsymbol{\theta}), \qquad \nabla^2 RSS(\theta) = 2X^T X.$$

Assuming X full column rank, $X^T X$ is SPD and the unique solution satisfies:

$$X^T(\mathbf{y} - X\boldsymbol{\theta}) = 0 \to \hat{\boldsymbol{\theta}} = (X^T X)^{-1} X^T \mathbf{y} = X^\dagger \mathbf{y}$$

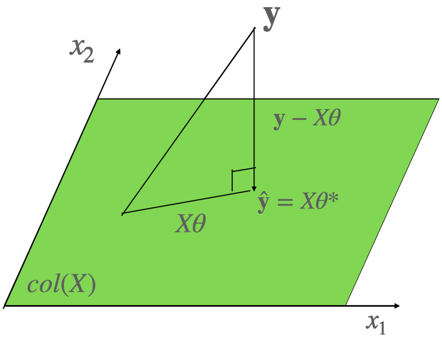
$\hat{\boldsymbol{\theta}}$ OLS estimator (ordinary least squares)
$X^\dagger$ Moore-Penrose pseudoinverse, linked to Singular Value Decomposition
`https://mathurinm.github.io/assets/2022_ens/class.pdf`, Def 0.12 & 0.13

Geometrical interpretation

The fitted (predicted) values are:

$$\hat{\mathbf{y}} = f_{\hat{\boldsymbol{\theta}}}(\mathbf{x})$$
$$\hat{\mathbf{y}} = X\hat{\boldsymbol{\theta}} = X(X^TX)^{-1}X^T\mathbf{y} = XX^{\dagger}\mathbf{y}$$

Geometrical interpretation



$$\mathbf{y} = \mathbf{y} - X\boldsymbol{\theta} + \underbrace{X\boldsymbol{\theta}}_{\in col(X)}$$

Orthogonality condition on the residual:

$$X^T(\mathbf{y} - X\boldsymbol{\theta}) = 0$$

$\rightarrow$ $\hat{\mathbf{y}}$ projection of $\mathbf{y}$ onto $col(X)$

What if X is not full (column) rank?

- If X is not full rank, it $\approx$ means that two inputs are *correlated*
- $\hat{\boldsymbol{\theta}}$ is not unique
- Fitted values are still projection of $\mathbf{y}$ onto *col*(X), but no unique way of expressing $\boldsymbol{\theta}$
- $\hat{\boldsymbol{\theta}}_\dagger = X^\dagger \mathbf{y}$ gives the minimum norm solution: among all parameters $\boldsymbol{\theta}$ that minimize $\|\mathbf{y} - X\boldsymbol{\theta}\|^2$, $\hat{\boldsymbol{\theta}}_\dagger$ has minimal ℓ_2 norm, i.e.:

$$\hat{\boldsymbol{\theta}}_\dagger = \operatorname{argmin} \|\boldsymbol{\theta}\| \quad \text{subject to} \quad \boldsymbol{\theta} \in \operatorname{argmin} \|\mathbf{y} - X\boldsymbol{\theta}\|^2$$

How to measure the quality of the model?

- Mean squared error:

$$MSE = \frac{RSS}{n} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- R squared:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

  We can see the R^2 as the error of the model divided by the error of a basic model who predicts the mean for all inputs.

MSE-R^2 a comparison

Properties and interpretation of R^2

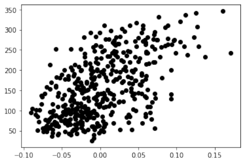
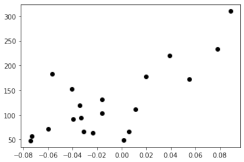
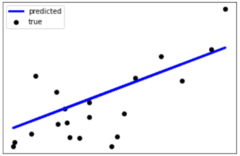
- The higher the better, the max is 100% (all predictions are exact).
- No minimum: basic model gets 0%, a negative R^2 is really bad.

Advantages and disadvantages

- Facilitate comparisons between models. The MSE depends on the absolute value of the variable to predict: cannot say if it is large or not, R^2 is normalized.
- No information on the mean error of the model on the predictions

$\rightarrow$ good to couple it with the MSE.

Linear regression with Scikit-learn

Principle

- First split the data into train and test
- LinearRegression() builds a linear model
- lin.fit fits the model to the data
- lin.coef_ gives the $\boldsymbol{\theta}$
- r2_score computed the R^2

Python code

```
from sklearn.model_selection import
        train_test_split
from sklearn.linear_model import
        LinearRegression
from sklearn.metrics import r2_score

X_train, X_test, y_train, y_test =
        train_test_split(X, y)
lin = LinearRegression()
lin.fit(X_train, y_train)
lin.coef_


r2_score(y_test, lin.predict(X_test))
```

Example: diabetes dataset

Objective

Predict, based on diagnostic measurements on the patients with diabetes, a quantitative measure of disease progression after one year

Predictors: $d = 10$ variables

- age
- sex
- body mass index
- average blood pressure
- six blood serum measurements

Data

$n = 442$ diabetes patients

Reference

`https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html`

Linear regression in statistical learning theory

- $\mathcal{X}$ and $\mathcal{Y}$ vector spaces of all possible inputs/outputs.
- Unknown probability distribution $p(z) = p(\mathbf{x}, y)$ on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$
- The training set $\mathcal{S} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ is made up of n samples from this probability distribution
- Inference problem: finding a function $f : \mathcal{X} \to \mathcal{Y}$ such that $f(\mathbf{x}) \sim y$ and the hypothesis space is the set of linear functions $\mathcal{L}$ parametrized by $\boldsymbol{\theta}$
- Let $\ell(f(\mathbf{x}), y)$ be the loss function, a metric for the difference between the predicted value $f(\mathbf{x})$ and the actual value y.
- The expected risk is defined to be
  $$I[f] = \int_{\mathcal{X}\times\mathcal{Y}} \ell(f(\mathbf{x}), y)\, p(\mathbf{x}, y)\, d\mathbf{x}\, dy$$
- The target function $f = \mathrm{argmin}_{\{h \in \mathcal{L}\}} I[h]$
- Because $p(\mathbf{x}, y)$ is unknown, a proxy measure for the expected risk must be used: the empirical risk $I_S[f] = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} \ell(f(\mathbf{x}_i), y_i)$

Linear regression in statistical learning theory

- A learning algorithm that chooses $f_S = \arg\min_{f \in \mathcal{L}} I_S[f]$ is called empirical risk minimization: least-squares in our case
- Learning algorithm $\equiv$ find the best $\boldsymbol{\theta}$.
- Learning algorithm gives a value for $\boldsymbol{\theta}$ that depends on the training sample, but $\boldsymbol{\theta}$ is actually a random variable.
- What are the statistical properties of this random variable?
- Statistical learning theory studies $\boldsymbol{\theta}$ as an estimator
- In particular we will study the properties of the OLS (ordinary least squares) estimator $\hat{\boldsymbol{\theta}} = (X^T X)^{-1} X^T \mathbf{y}$

Statistical estimators

- Let θ be a parameter that needs to be estimated.
- An estimator is a rule for calculating an estimate of a given quantity based on observed data.
  Example: sample mean $\hat{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is an estimator for the population mean (we don't have access to data of the full population)
- Algebra of random variables: if X is used to denote a random variable corresponding to the observed data, the estimator (itself treated as a random variable) is symbolised as a function of that random variable, $\widehat{\theta}(X)$.
- The estimate for a particular observed data value x (i.e. for $X = x$) is then $\widehat{\theta}(x)$, which is a fixed value.

Statistical properties of the least squares estimator (I)

Assumptions:

- $\mathbf{x}_i$ are fixed (X is nonrandom)
- $X^T X$ is invertible (the regressors in X must all be linearly independent)
- $\mathbf{y} = f(\mathbf{x}) = X\boldsymbol{\theta} + \boldsymbol{\epsilon}$
- $\mathbb{E}(\boldsymbol{\epsilon}|X) = 0$ (meaning $\mathbb{E}(\epsilon_i|X) = 0$ for all i), and so $\mathbb{E}(\boldsymbol{\epsilon}) = 0$ and $\mathbb{E}(X^T\boldsymbol{\epsilon}) = 0$.

Lemma: unbiased estimator

If $\mathbf{y} = X\boldsymbol{\theta} + \boldsymbol{\epsilon}$ where and $\mathbb{E}(\boldsymbol{\epsilon}) = 0$, then $\mathbb{E}(\hat{\boldsymbol{\theta}}) = \boldsymbol{\theta}$.

Proof

$$\hat{\boldsymbol{\theta}} = (X^T X)^{-1} X^T \mathbf{y} = (X^T X)^{-1} X^T (X\boldsymbol{\theta} + \boldsymbol{\epsilon}) = \boldsymbol{\theta} + (X^T X)^{-1} X^T \boldsymbol{\epsilon}$$

$$\mathbb{E}(\hat{\boldsymbol{\theta}}) = \mathbb{E}(\boldsymbol{\theta}) + (X^T X)^{-1} X^T \mathbb{E}(\boldsymbol{\epsilon}) = \boldsymbol{\theta}$$

Statistical properties of the least squares estimator (II)

The covariance matrix Σ: definition

Is a square matrix giving the covariance between each pair of elements of a given random vector $\boldsymbol{v}$

$$\Sigma_{\boldsymbol{v}} = \text{cov}(\boldsymbol{v}, \boldsymbol{v}) = \mathbb{E}[(\boldsymbol{v} - \mathbb{E}(\boldsymbol{v}))(\boldsymbol{v} - \mathbb{E}(\boldsymbol{v}))^T] = \mathbb{E}(\boldsymbol{v}\boldsymbol{v}^T) - \mathbb{E}(\boldsymbol{v})\mathbb{E}(\boldsymbol{v})^T$$
$$\Sigma_{\boldsymbol{v}}(i,j) = \text{cov}(v_i, v_j) = \mathbb{E}[(v_i - \mathbb{E}(v_i))(v_j - \mathbb{E}(v_j))]$$
$$\Sigma_{\boldsymbol{v}}(i,i) = \text{cov}(v_i, v_i) = \mathbb{E}[(v_i - \mathbb{E}(v_i))^2] = \text{Var}(v_i)$$

The covariance matrix: property

Under the assumptions, and assuming also that the errors are uncorrelated with common variance, that is $\Sigma_{\epsilon} = \sigma^2 I$, the covariance matrix for the LS estimator is:

$$\Sigma_{\boldsymbol{\theta}} = \sigma^2 (X^T X)^{-1}$$

Statistical properties of the least squares estimator

Proof

As before
$\hat{\boldsymbol{\theta}} = (X^T X)^{-1} X^T \mathbf{y} = (X^T X)^{-1} X^T (X\boldsymbol{\theta} + \boldsymbol{\epsilon}) = \boldsymbol{\theta} + (X^T X)^{-1} X^T \boldsymbol{\epsilon}$.
Then $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta} = (X^T X)^{-1} X^T \boldsymbol{\epsilon}$ and

$$\begin{aligned} \mathbb{E}((\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T) &= \mathbb{E}((X^T X)^{-1} X^T \boldsymbol{\epsilon}\boldsymbol{\epsilon}^T X (X^T X)^{-1}) \\ &= (X^T X)^{-1} X^T \mathbb{E}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T) X (X^T X)^{-1} \end{aligned}$$

and since $\mathbb{E}(\boldsymbol{\epsilon}) = 0$

$$\mathbb{E}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T) = \mathrm{cov}(\boldsymbol{\epsilon}, \boldsymbol{\epsilon}) = \Sigma_{\boldsymbol{\epsilon}} = \sigma^2 I$$

and the result follows.

Commonly used unbiased estimator of σ^2

$$\hat{\sigma}^2 = \frac{1}{N - p - 1} \sum_{i=1}^{d} (y_i - \hat{y}_i)^2, \quad \mathbb{E}(\hat{\sigma}^2) = \sigma^2$$

Distribution of the LS estimator

Theorem

Suppose that X^TX is invertible, and $\mathbf{y} = X\boldsymbol{\theta} + \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim N(0, \sigma^2 I)$. Then

$$\begin{aligned}
\hat{\boldsymbol{\theta}} &\sim N(\boldsymbol{\theta}, (X^TX)^{-1}\sigma^2), \\
\hat{\mathbf{y}} := X\hat{\boldsymbol{\theta}} &\sim N(X\theta, XX^\dagger\sigma^2), \\
\hat{\boldsymbol{\epsilon}} := \mathbf{y} - X\hat{\boldsymbol{\theta}} &\sim N(0, (I - XX^\dagger)\sigma^2).
\end{aligned}$$

$(X^\dagger = (X^TX)^{-1}X^T)$

The Gauss-Markov theorem

Definition - Linear unbiased estimators

Linear estimator: $\tilde{\boldsymbol{\theta}} = C\mathbf{y}$, i.e., $\tilde{\theta}_j = C_{1,j} y_1 + \cdots + C_{n,j} y_n$ for all j
Unbiased: $\mathbb{E}(\tilde{\boldsymbol{\theta}}) = \boldsymbol{\theta}$

The assumptions

- Regression model: $\mathbf{y} = X\boldsymbol{\theta} + \boldsymbol{\epsilon}$
- X has full-rank
- $\mathbb{E}(\boldsymbol{\epsilon}|X) = 0$
- $\mathrm{Var}(\boldsymbol{\epsilon}|X) = \sigma^2 I$

The theorem

Under the assumptions, the ordinary least squares (OLS) estimator $\hat{\boldsymbol{\theta}} = (X^T X)^{-1} X^T \mathbf{y}$ of the coefficients $\boldsymbol{\theta}$ of a linear regression model is the *best linear unbiased estimator* (BLUE), that is, the estimator that has the smallest variance among those that are unbiased and linear in the observed output variables $\mathbf{y}$.

The Gauss-Markov theorem

Best in which sense?

- Scalar case (one regressor, $\hat{\theta} \in \mathbb{R}$): $\mathrm{Var}(\hat{\theta}|X) \leq \mathrm{Var}(\tilde{\theta}|X)$ for any other linear unbiased estimator $\tilde{\theta}$
- Multivariate case (multiple regressors, $\hat{\boldsymbol{\theta}} \in \mathbb{R}^d$): $\mathrm{Var}(\mathbf{a}^T\hat{\theta}|X) \leq \mathrm{Var}(\mathbf{a}^T\tilde{\boldsymbol{\theta}}|X)$ for any other linear unbiased estimator $\tilde{\boldsymbol{\theta}}$ and vector $\mathbf{a} \in \mathbb{R}^d$. This is equivalent to

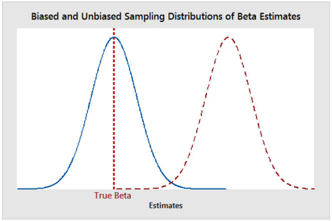
$$\mathrm{cov}(\tilde{\boldsymbol{\theta}}|X) - \mathrm{cov}(\hat{\boldsymbol{\theta}}|X) \quad \text{SPD}$$

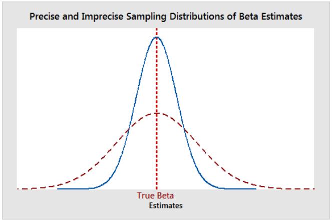
In other words, OLS is BLUE if and only if any linear combination of the regression coefficients is estimated more precisely by OLS than by any other linear unbiased estimator.

Towards biased estimators

$$MSE(\hat{\theta}) = \mathbb{E}(\hat{\theta} - \theta)^2 = \underbrace{\text{Var}(\hat{\theta})}_{variance} + \underbrace{[\mathbb{E}(\hat{\theta}) - \theta]^2}_{squared\ bias}$$

- GM theorem $\rightarrow$ LS estimator has the smallest MSE of all unbiased estimators BUT it may exist a biased estimator with lower MSE
- Trade a little bit of bias for a larger reduction in variance
- Models are distortions of the truth: why not?

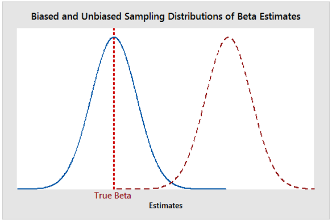


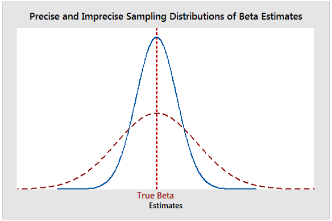


Towards biased estimators

$$MSE(\hat{\theta}) = \mathbb{E}(\hat{\theta} - \theta)^2 = \underbrace{\mathrm{Var}(\hat{\theta})}_{variance} + \underbrace{[\mathbb{E}(\hat{\theta}) - \theta]^2}_{squared\ bias}$$

- GM theorem $\rightarrow$ LS estimator has the smallest MSE of all unbiased estimators BUT it may exist a biased estimator with lower MSE
- Trade a little bit of bias for a larger reduction in variance
- Models are distortions of the truth: why not?





Two directions

- Subset selection methods: discrete methods
- Shrinkage methods: continuous methods

Subset selection methods

$$\mathbf{y} = f(\mathbf{x}) \sim f_{\boldsymbol{\theta}}(x),\ \boldsymbol{\theta} \in \mathbb{R}^d$$

What?

Select a subset of variables in $\{\theta_1, \ldots, \theta_d\}$ and set them to zero

Why?

- Improves prediction accuracy: setting some coefficients to zero may reduce the variance, helps generalization
- Improve interpretability: reduce number of predictors

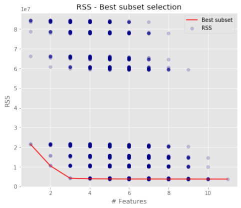
How?

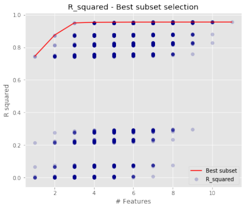
- Best subset selection (BSS)
- Forward-Stepwise selection
- Backward-Stepwise selection

Best subset selection

Goal:
For a well-chosen k, find the subset of size k that gives the lowest error

- For each $k \in \{1, \ldots, d\}$ enumerate all subsets of size k: $\binom{d}{k}$
- For each $k \in \{1, \ldots, d\}$ choose the subset that gives the smallest RSS or the largest R^2
- Form the best subset curve (red) and choose k

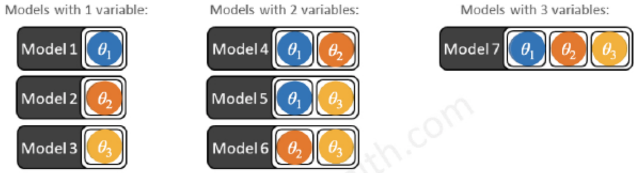




Best Subset Selection: Example with 3 Variables

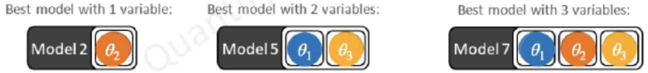
Step 1: Consider All Possible Models

By listing all possible combination of variables


Models with 1 variable:
Model 1 θ_1
Model 2 θ_2
Model 3 θ_3
Models with 2 variables:
Model 4 θ_1 θ_2
Model 5 θ_1 θ_3
Model 6 θ_2 θ_3
Models with 3 variables:
Model 7 θ_1 θ_2 θ_3

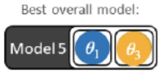
Step 2: Identify the Best Model of Each Size

By choosing the one with the lowest sum of squared errors or the highest R^2


Best model with 1 variable:
Model 2 θ_2
Best model with 2 variables:
Model 5 θ_1 θ_3
Best model with 3 variables:
Model 7 θ_1 θ_2 θ_3

Step 3: Identify the Best Overall Model

By choosing the one with the lowest AIC (or BIC) or the highest adjusted R^2



Best subset selection: step 2

Requires to consider a total of 2^d subsets
For all k we need to solve

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \|\mathbf{y} - X\boldsymbol{\theta}\|^2 \quad \text{s.t.} \quad \|\boldsymbol{\theta}\|_0 \leq k$$

where $\|\boldsymbol{\theta}\|_0$ is the number of non-zero components of $\boldsymbol{\theta}$.

How to solve this ?

- The Lagrangian function is not continuous and not smooth
- Cannot use "standard" optimization methods
- There are variants of branch-and-bound methods that are efficient but still limited to rather small d

How to choose k? (I)

- In step 2 all the sets have the same size, we use the RSS
- In step 3 we cannot use the RSS: the BSS curve is always decreasing
- Choice of k should give a good compromise between parsimony and accuracy
- Also, we want to minimize the test error, not the training error.
- We could use cross-validation (very expensive)!

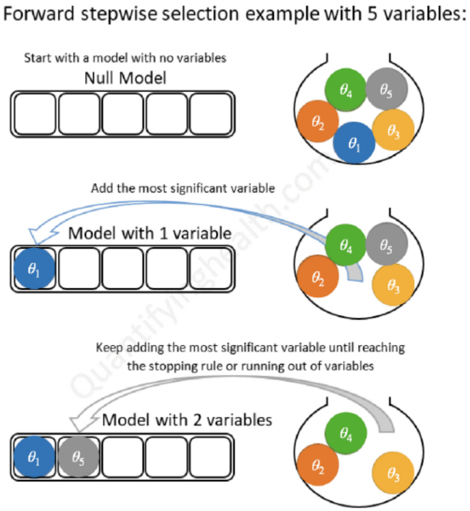
How to choose k? (II)

- Alternative estimates of test error:
  - (lowest) Akaike Information Criterion (AIC) $AIC = 2k - 2\ln(\hat{L})$
  - (lowest) Bayesian Information Criterion (BIC) $BIC = k\ln(n) - 2\ln(\hat{L})$
  - (highest) Adjusted R^2 $R_a^2 = 1 - (1 - R^2)\frac{n-1}{n-k-1}$

  where $\hat{L}$ is the likelihood of the model.
- They are motivated by asymptotic information theory arguments and rely on model assumptions (eg. normality of the errors).
- They are statistics that imposes some sort of penalty on bigger models and estimate the generalization error

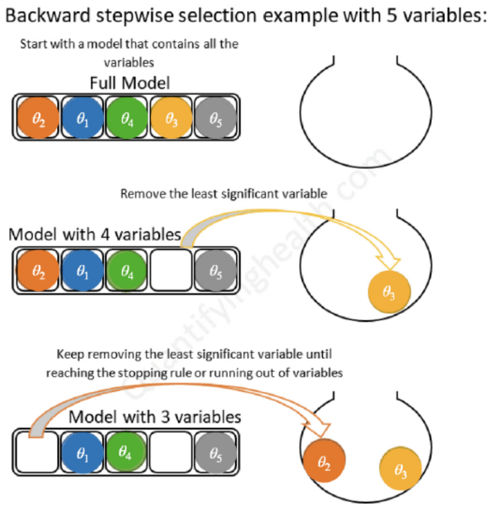
Forward-stepwise selection

Starts with one parameter and sequentially adds the predictor that most reduces the fit


Forward stepwise selection example with 5 variables:
Start with a model with no variables
Null Model
θ_4
θ_5
θ_2
θ_1
θ_3
Add the most significant variable
Model with 1 variable
θ_1
Keep adding the most significant variable until reaching the stopping rule or running out of variables
Model with 2 variables
θ_1
θ_5
QuantifyingHealth.com

Backward-stepwise selection

Starts with all parameters and sequentially removes the predictors that impact the fit less


Backward stepwise selection example with 5 variables:
Start with a model that contains all the variables
Full Model
θ_2 θ_1 θ_4 θ_3 θ_5
Remove the least significant variable
Model with 4 variables
θ_2 θ_1 θ_4 θ_5
θ_3
Keep removing the least significant variable until reaching the stopping rule or running out of variables
Model with 3 variables
θ_1 θ_4 θ_5
θ_2 θ_3

Shrinkage methods: Ridge and Lasso

- Based on a smooth approximation of the BSS problem
- Continuous methods: easier to use

Ridge regression

Shrinks regression coefficients by imposing a penalty on their size:

$$\hat{\theta}_R = \arg\min_{\theta} \sum_{i=1}^{n} (y_i - \theta_0 - \sum_{j=1}^{d} X_{i,j}\theta_j)^2 + \lambda \sum_{j=1}^{d} \theta_j^2$$

Used in neural networks $\rightarrow$ *weight decay*

Lasso regression

Drives some coefficients to zero by penalizing the sum of absolute values:

$$\hat{\theta}_L = \arg\min_{\theta} \sum_{i=1}^{n} (y_i - \theta_0 - \sum_{j=1}^{d} X_{i,j}\theta_j)^2 + \lambda \sum_{j=1}^{d} |\theta_j|$$

Used in signal processing $\rightarrow$ *basis pursuit*

Or equivalently...

From now on, denote

$$X = \begin{pmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{pmatrix} \in \mathbb{R}^{n\times d}, \; \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n$$

Ridge regression

$$\hat{\boldsymbol{\theta}}_R = \arg\min_{\theta} \|\mathbf{y} - X\boldsymbol{\theta}\|^2 \quad \text{subject to } \|\boldsymbol{\theta}\|_2^2 \leq t$$

Lasso regression

$$\hat{\boldsymbol{\theta}}_L = \arg\min_{\theta} \|\mathbf{y} - X\boldsymbol{\theta}\|^2 \quad \text{subject to } \|\boldsymbol{\theta}\|_1 \leq t$$

Subset selection

$$\hat{\boldsymbol{\theta}}_S = \arg\min_{\theta} \|\mathbf{y} - X\boldsymbol{\theta}\|^2 \quad \text{subject to } \|\boldsymbol{\theta}\|_0 \leq t$$

$\hookrightarrow$ Three different norms on the constraints
Used in practice: *penalized* form, $\hat{\boldsymbol{\theta}}_R = \arg\min_{\boldsymbol{\theta}} \|\mathbf{y} - X\boldsymbol{\theta}\|^2 + \lambda\|\boldsymbol{\theta}\|_2^2$

Similarities and differences

- The best subset selection (BSS) and the lasso estimators have sparse solutions, i.e., at a solution $\boldsymbol{\theta}$ we will have $\theta_j = 0$ for many components $j \in \{1, \ldots, d\}$.
  - For BSS k directly controls the sparsity level, for Lasso we get a higher degree of sparsity the smaller the value of $t \geq 0$ or the larger the value of $\lambda \geq 0$
- The lasso and ridge regression problems are convex, BSS is very far from being convex

Interpretation

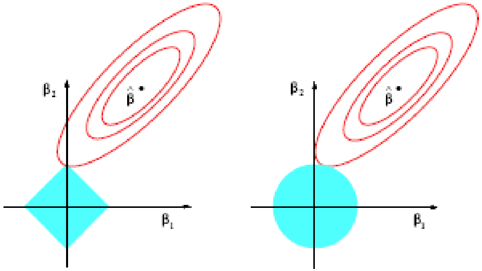


Figure 3.12: *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Relation between the three estimators

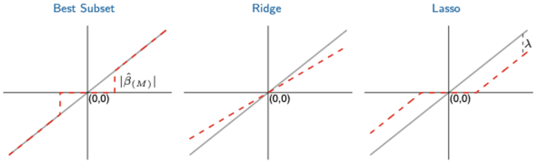
If X has orthonormal columns $\hat{\boldsymbol{\theta}} = X^T \mathbf{y}$ and

$$\hat{\boldsymbol{\theta}}_S = H_{\sqrt{2\lambda}}(\hat{\boldsymbol{\theta}}), \quad \hat{\boldsymbol{\theta}}_L = S_\lambda(\hat{\boldsymbol{\theta}}), \quad \hat{\boldsymbol{\theta}}_R = \frac{\hat{\boldsymbol{\theta}}}{1+2\lambda}$$

where

$$S_t(x) = sign(x)(|x| - t)_+, \quad H_t(x) = x \cdot I(|x| > t)$$

are the *Soft and Hard thresholding functions*.



($\hat{\boldsymbol{\theta}}$ in the x axis)

Elastic net

- The ridge regression problem is always strongly convex
- The lasso problem is not always strictly convex
- A compromise between the two: elastic net (Zou & Hastie 2005):

$$\min_{\boldsymbol{\theta}} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1 + \delta\|\boldsymbol{\theta}\|_2^2$$

  where now both λ, δ are hyperparameters.
- The problem is always strictly convex, the solution is unique, the elastic net combines some of the desirable predictive properties of ridge regression with the sparsity properties of the lasso.

Ridge regression: standardisation

The ridge solution is not equivariant under scaling of the input.
Sometimes standardising the inputs before solving the problem improves the results

Standardisation

Given x, μ, σ (variable, mean and standard deviation) $\rightarrow x_s = \frac{x-\mu}{\sigma}$

How to find $\hat{\theta}_R$?

The intercept θ_0 is not penalized and $\hat{\theta}_0 = \frac{1}{n}\sum_{i=1}^{n} y_i$
We can write the residual sum of squares as:

$$RSS_\lambda(\boldsymbol{\theta}) = (\mathbf{y} - X\boldsymbol{\theta})^T(\mathbf{y} - X\boldsymbol{\theta}) + \lambda\boldsymbol{\theta}^T\boldsymbol{\theta}$$

this is still a quadratic function.

$$\begin{aligned} \nabla RSS_\lambda(\boldsymbol{\theta}) &= -2X^T(\mathbf{y} - X\boldsymbol{\theta}) + 2\lambda\boldsymbol{\theta}, \\ \nabla^2 RSS_\lambda(\boldsymbol{\theta}) &= 2(X^TX + \lambda I). \end{aligned}$$

For any λ, $X^TX + \lambda I$ is SPD and the unique solution satisfies:

$$X^T(\mathbf{y} - X\boldsymbol{\theta}) - \lambda\boldsymbol{\theta} = 0$$

so that

$$\hat{\boldsymbol{\theta}}_R = (X^TX + \lambda I)^{-1}X^T\mathbf{y}.$$

How to find $\hat{\theta}_L$?

$$\hat{\boldsymbol{\theta}}_L = \arg\min_{\theta} \sum_{i=1}^{n} (y_i - \theta_0 - \sum_{j=1}^{d} X_{i,j}\theta_j)^2$$

$$\text{subject to} \sum_{j=1}^{d} |\theta_i| \leq t$$

- The solution is nonlinear in $\mathbf{y}$ and there is no closed form expression of $\hat{\boldsymbol{\theta}}_L$, unless X has orthonormal columns
- t small : some coefficients will be zero: continuous subset selection
- Non-smooth optimization problem solved by proximal methods
- State of the art for Lasso: Coordinate Descent, Celer algorithm

`https://mathurinm.github.io/celer/`

Proximal methods

$$\min_x f(x) + \lambda g(x)$$

- $f + g$ admits a minimizer
- f, g are convex
- f is β-smooth: $\|\nabla f(x) - \nabla f(y)\| \leq \beta \|x - y\|$ for all x, y
- g is possibly non-differentiable

If there is no g

Gradient descent: $x_{k+1} = x_k - \frac{1}{\beta} \nabla f(x_k)$

Why?

Gradient step minimizes an upper bound on the function:

$$f(x) \leq f(y) + \nabla f(y)^T (x - y) + \frac{\beta}{2} \|x - y\|^2, \quad \forall x, y$$

$$f(x_{k+1}) \leq f(x_k) + \nabla f(x_k)^T p_k + \frac{\beta}{2} \|p_k\|^2 \overbrace{\rightarrow}^{\arg\min_{x_{k+1}}} p_k = -\frac{\nabla f(x_k)}{\beta}$$

Proximal methods

If there is g: add λg to the upper bound:

$$f(x) + \lambda g(x) \leq f(y) + \nabla f(y)^T(x-y) + \frac{\beta}{2}\|x-y\|^2 + \lambda g(x), \quad \forall x, y$$

Can we minimize the upper bound?

$$\begin{aligned}
&\arg\min_x f(y) + \nabla f(y)^T(x-y) + \frac{\beta}{2}\|x-y\|^2 + \lambda g(x) = \\
&\arg\min_x \nabla f(y)^T(x-y) + \frac{\beta}{2}\|x-y\|^2 + \lambda g(x) = \\
&\arg\min_x \frac{1}{2}\|x - (y - \frac{1}{\beta}\nabla f(y))\|^2 + \frac{\lambda}{\beta} g(x) := \\
&\text{prox}_{\frac{\lambda}{\beta} g}(y - \frac{1}{\beta}\nabla f(y)).
\end{aligned}$$

Example

If $g(x) = \|x\|_1$, $\text{prox}_{\lambda g/\beta}(x) = S_{\lambda/\beta}(x)$.

Gradient methods

Differentiable case

$$x_{k+1} = x_k - \frac{1}{\beta}\nabla f(x_k)$$

Convergence

If f is differentiable, β-smooth and convex:

$$f(x_K) - f(x^*) \leq \frac{2\beta\|x_1 - x^*\|}{K-1}$$

Proximal gradient descent

$$x_{k+1} = \mathrm{prox}_{\lambda g/\beta}(x_k - 1/\beta\nabla f(x_k))$$

Convergence

If f is differentiable, β-smooth, convex and g is convex:

$$f(x_K) - f(x^*) \leq \frac{\beta\|x_1 - x^*\|}{2K}$$

In both cases $f(x_K) - f(x^*) = O\left(\frac{1}{K}\right)$.

Tuning of the hyperparameters

- β : easy for a linear function $f(\theta) = \|X\theta + b\|^2$, $\beta \sim \|X^TX\|_2$, can be computed by the power method
- λ : essential for good results: cross-validation

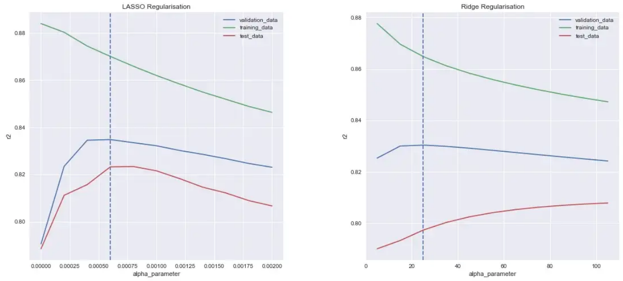
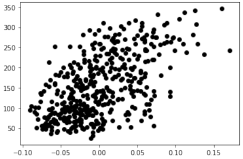
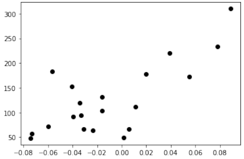
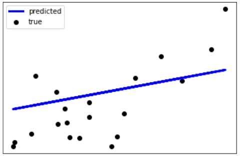


Figure: alpha_parameter=λ

Ridge and lasso regression with Scikit-learn

Principle

- $\lambda = 1$ by default
- Ridge() creates a Ridge model
- Lasso() creates a Lasso model*
- beware of datafit scaling ($1/n$ or not!)

Python code

```
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.metrics import r2_score

 # Ridge
 rid = Ridge()
 rid.fit(X_train, y_train)
 r2_score(y_test, rid.predict(X_test))

 # Lasso
 las = Lasso()
 las.fit(X_train, y_train)
 r2_score(y_test, rid.predict(X_test))
```

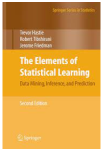
References I

Some online references for this lesson

- https://www.stat.cmu.edu/~ryantibs/statml/lectures/sparsity.pdf
- https://artowen.su.domains/courses/305a/ch2.pdf
- https://perso.telecom-paristech.fr/rgower/pdf/M2_statistique_optimisation/optimization_II_prox_LASSO-expanded.pdf

Book

The elements of statistical learning, Hastie, Tibshirani, Friedman, Spriengér (2009)


Springer Series in Statistics
Trevor Hastie
Robert Tibshirani
Jerome Friedman
The Elements of Statistical Learning
Data Mining, Inference, and Prediction
Second Edition
Springer