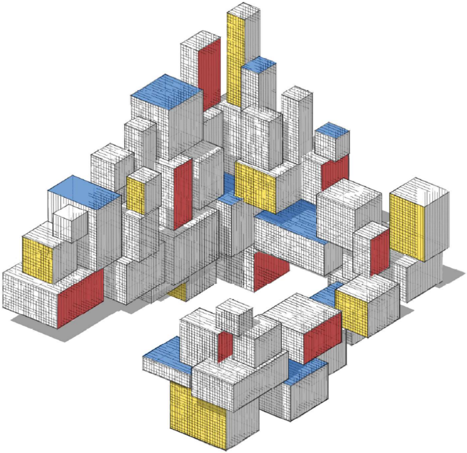
Regression



Purpose

In this lecture we discuss the mathematical ideas behind regression models.

Regression Framework

Regression refers to a broad class of supervised learning techniques where the aim is to predict a quantitative response (output) variable y via a function $g(\boldsymbol{x})$ of an explanatory (input) vector $\boldsymbol{x} = [x_1, \ldots, x_p]^\top$.

For instance, regression could be used to predictw the birth weight of a baby (the response variable) from the weight of the mother, her socio-economic status, and her smoking habits (the explanatory variables).

Recall the framework of supervised learning:

- The aim is to find a prediction function g that best guesses what the random output Y will be for a random input vector $\boldsymbol{X}$.
- The joint pdf $f(\boldsymbol{x}, y)$ of $\boldsymbol{X}$ and Y is unknown, but a training set $\tau = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ is available, which is thought of as the outcome of a random training set $\mathcal{T} = \{(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)\}$ of iid copies of $(\boldsymbol{X}, Y)$.

Regression Framework

Once we have selected a loss function $\text{Loss}(y, \widehat{y})$, such as the squared-error loss

$$\text{Loss}(y, \widehat{y}) = (y - \widehat{y})^2,$$

then the "best" prediction function g is defined as the one that minimizes the risk $\ell(g) = \mathbb{E}\,\text{Loss}(Y, g(\boldsymbol{X}))$.

For the squared-error loss this optimal prediction function is the conditional expectation

$$g^*(\boldsymbol{x}) = \mathbb{E}[Y \mid \boldsymbol{X} = \boldsymbol{x}].$$

As the squared-error loss is the most widely-used loss function for regression, we will assume it from now on.

Learner

The optimal prediction function g^* has to be learned from the training set τ by minimizing the training loss

$$\ell_\tau(g) = \frac{1}{n}\sum_{i=1}^{n}(y_i - g(\boldsymbol{x}_i))^2 \tag{1}$$

over a suitable class of functions $\mathcal{G}$.

The function $g_\tau^{\mathcal{G}}$ that minimizes the training loss is the function we use for prediction — the so-called learner.

When the function class $\mathcal{G}$ is clear from the context, we drop the superscript in the notation.

Model Assumptions

Conditional on $\boldsymbol{X} = \boldsymbol{x}$, the response Y can be written as

$$Y = g^*(\boldsymbol{x}) + \varepsilon(\boldsymbol{x}), \quad \text{where} \quad \mathbb{E}\,\varepsilon(\boldsymbol{x}) = 0.$$

This motivates a standard modeling assumption where $Y_1, \ldots, Y_n$, conditional on $\boldsymbol{X}_1 = \boldsymbol{x}_1, \ldots, \boldsymbol{X}_n = \boldsymbol{x}_n$, are assumed to be of the form

$$Y_i = g(\boldsymbol{x}_i) + \varepsilon_i, \quad i = 1, \ldots, n,$$

where the $\{\varepsilon_i\}$ are independent with $\mathbb{E}\,\varepsilon_i = 0$ and $\mathbb{V}\mathrm{ar}\,\varepsilon_i = \sigma^2$ for some function $g \in \mathcal{G}$ and variance σ^2.

The function g is further assumed to be completely known up to an unknown parameter vector; that is,

$$Y_i = g(\boldsymbol{x}_i \,|\, \boldsymbol{\beta}) + \varepsilon_i, \quad i = 1, \ldots, n. \tag{2}$$

A final model simplification is to view (2) as if the $\{\boldsymbol{x}_i\}$ were fixed, while the $\{Y_i\}$ are random.

Model Assumptions

For the remainder of this chapter, we assume that the training feature vectors $\{\boldsymbol{x}_i\}$ are fixed and only the responses are random; that is, $\mathcal{T} = \{(\boldsymbol{x}_1, Y_1), \ldots, (\boldsymbol{x}_n, Y_n)\}$.

Advantage: estimating the *function* g from the training data is reduced to the (much simpler) problem of estimating the *parameter vector* $\boldsymbol{\beta}$.

Disadvantage: functions of the form $g(\cdot \mid \boldsymbol{\beta})$ may not accurately approximate the true unknown g^*.

If the function $g(\cdot \mid \boldsymbol{\beta})$ is *linear*, the analysis proceeds through the class of linear models.

If, also, the error terms $\{\varepsilon_i\}$ are assumed to be *Gaussian*, this analysis can be carried out using the rich theory of normal linear models.

Simple Linear Regression

The most basic regression model involves a linear relationship between the response and a single explanatory variable. In particular, we have measurements $(x_1, y_1), \ldots, (x_n, y_n)$ that lie approximately on a straight line.

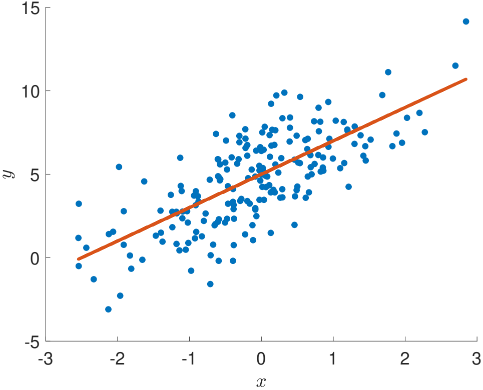


Figure: Data from a simple linear regression model.

Simple Linear Regression

Following the general scheme captured in (2), the simple linear regression model for these data is that the $\{x_i\}$ are fixed and variables $\{Y_i\}$ are random such that

$$Y_i = \beta_0 + \beta_1\, x_i + \varepsilon_i, \quad i = 1, \ldots, n,$$

for certain *unknown* parameters β_0 and β_1. The $\{\varepsilon_i\}$ are assumed to be independent with expectation 0 and unknown variance σ^2. The unknown line

$$y = \underbrace{\beta_0 + \beta_1\, x}_{g(x \mid \boldsymbol{\beta})}$$

is called the regression line.

Thus, we view the responses as random variables that would lie exactly on the regression line, were it not for some "disturbance" or "error" term represented by the $\{\varepsilon_i\}$. The extent of the disturbance is modeled by the parameter σ^2.

Multiple Linear Regression

In a multiple linear regression model the response Y depends on a d-dimensional explanatory vector $\boldsymbol{x} = [x_1, \ldots, x_d]^\top$, via the linear relationship

$$Y = \beta_0 + \beta_1\, x_1 + \cdots + \beta_d\, x_d + \varepsilon, \tag{3}$$

where $\mathbb{E}\,\varepsilon = 0$ and $\mathbb{V}\mathrm{ar}\,\varepsilon = \sigma^2$.
Thus, the data lie approximately on a d-dimensional affine hyperplane

$$y = \underbrace{\beta_0 + \beta_1 x_1 + \cdots + \beta_d x_d}_{g(\boldsymbol{x}\,|\,\boldsymbol{\beta})},$$

where we define $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_d]^\top$. The function $g(\boldsymbol{x}\,|\,\boldsymbol{\beta})$ is linear in $\boldsymbol{\beta}$, but not linear in the feature vector $\boldsymbol{x}$, due to the constant β_0. However, augmenting the feature space with the constant 1, the mapping $[1, \boldsymbol{x}^\top]^\top \mapsto g(\boldsymbol{x}\,|\,\boldsymbol{\beta}) := [1, \boldsymbol{x}^\top]\,\boldsymbol{\beta}$ becomes linear in the feature space and so (3) becomes a linear model.

Model Matrix

Note that in (3) we only specified the model for a single pair $(\boldsymbol{x}, Y)$. The model for the training set $\mathcal{T} = \{(\boldsymbol{x}_1, Y_1), \ldots, (\boldsymbol{x}_n, Y_n)\}$ is simply that each Y_i satisfies (3) (with $\boldsymbol{x} = \boldsymbol{x}_i$) and that the $\{Y_i\}$ are independent. Setting $\boldsymbol{Y} = [Y_1, \ldots, Y_n]^\top$, we can write the multiple linear regression model for the training data compactly as

$$\boldsymbol{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} = [\varepsilon_1, \ldots, \varepsilon_n]^\top$ is a vector of iid copies of ε and $\mathbf{X}$ is the model matrix given by

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{x}_1^\top \\ 1 & \boldsymbol{x}_2^\top \\ \vdots & \vdots \\ 1 & \boldsymbol{x}_n^\top \end{bmatrix}.$$

Example: Multiple Linear Regression

The figure depicts a realization of the multiple linear regression model

$$Y_i = x_{i1} + x_{i2} + \varepsilon_i, \quad i = 1, \ldots, 100,$$

where $\varepsilon_1, \ldots, \varepsilon_{100} \sim_{\text{iid}} \mathcal{N}(0, 1/16)$. The fixed feature vectors (vectors of explanatory variables) $\boldsymbol{x}_i = [x_{i1}, x_{i2}]^\top$, $i = 1, \ldots, 100$ lie in the unit square.

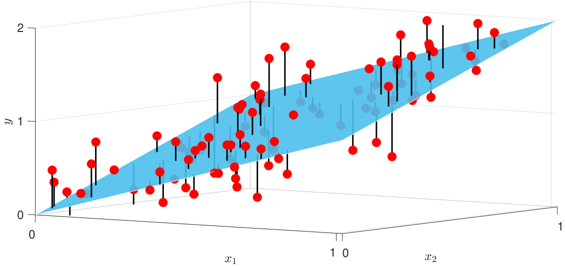


Figure: Data from a multiple linear regression model.

Analysis via Linear Models

Analysis of data from a linear regression model — including parameter estimation and model selection — is greatly simplified through the linear model representation

$$\boldsymbol{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{4}$$

where $\mathbf{X}$ is an $n \times p$ matrix, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_p]^\top$ a vector of p parameters, and $\boldsymbol{\varepsilon} = [\varepsilon_1, \ldots, \varepsilon_n]^\top$ an n-dimensional vector of independent error terms, with $\mathbb{E}\,\varepsilon_i = 0$ and $\mathbb{V}\mathrm{ar}\,\varepsilon_i = \sigma^2$, $i = 1, \ldots, n$.

Note that the model matrix $\mathbf{X}$ is assumed to be *fixed*, and $\boldsymbol{Y}$ and $\boldsymbol{\varepsilon}$ are *random*.

A specific outcome of $\boldsymbol{Y}$ is denoted by $\boldsymbol{y}$.

Note the different parameterization to the multiple linear regression model.

Parameter Estimation

The linear model $\boldsymbol{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ contains two unknown parameters, $\boldsymbol{\beta}$ and σ^2, which have to be estimated from the training data τ.

To estimate $\boldsymbol{\beta}$, we can repeat exactly the same reasoning used in our recurring polynomial regression example as follows.

For a linear prediction function $g(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\beta}$, the (squared-error) training loss can be written as

$$\ell_\tau(g) = \frac{1}{n} \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2,$$

and the optimal learner g_τ minimizes this quantity, leading to the least-squares estimate $\widehat{\boldsymbol{\beta}}$, which satisfies the normal equations

$$\mathbf{X}^\top \mathbf{X} \boldsymbol{\beta} = \mathbf{X}^\top \boldsymbol{y}. \tag{5}$$

Residuals

The corresponding training loss can be taken as an estimate of σ^2; that is,

$$\widehat{\sigma^2} = \frac{1}{n} \| \boldsymbol{y} - \mathbf{X}\widehat{\boldsymbol{\beta}} \|^2. \qquad (6)$$

The vector $\boldsymbol{e} := \boldsymbol{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}$ is called the vector of residuals and approximates the (unknown) vector of model errors $\boldsymbol{\varepsilon}$.

The quantity $\|\boldsymbol{e}\|^2 = \sum_{i=1}^{n} e_i^2$ is called the residual sum of squares (RSS).

Dividing the RSS by $n - p$ gives an unbiased estimate of σ^2, which we call the estimated residual squared error (RSE).

Supervised Learning Notation

In terms of our notation for supervised learning, we thus have:

1. The (observed) training data is $\tau = \{\mathbf{X}, \boldsymbol{y}\}$.
2. The function class $\mathcal{G}$ is the class of linear functions of $\boldsymbol{x}$; that is $\mathcal{G} = \{g(\cdot \mid \boldsymbol{\beta}) : \boldsymbol{x} \mapsto \boldsymbol{x}^\top \boldsymbol{\beta},\ \boldsymbol{\beta} \in \mathbb{R}^p\}$.
3. The (squared-error) training loss is $\ell_\tau(g(\cdot \mid \boldsymbol{\beta})) = \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2/n$.
4. The learner g_τ is given by $g_\tau(\boldsymbol{x}) = \boldsymbol{x}^\top \widehat{\boldsymbol{\beta}}$, where $\widehat{\boldsymbol{\beta}} = \operatorname{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^p} \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2$.
5. The minimal training loss is $\ell_\tau(g_\tau) = \|\boldsymbol{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2/n = \widehat{\sigma^2}$.

Model Selection and Prediction

Even if we restrict the learner to be a linear function, there is still the issue of which explanatory variables (features) to include.

While including too few features may result in large approximation error (underfitting), including too many may result in large statistical error (overfitting).

As discussed earlier, we need to select the features which provide the best tradeoff between the approximation and statistical errors, so that the (expected) generalization risk of the learner is minimized.

Depending on how the (expected) generalization risk is estimated, there are a number of strategies for feature selection:

Strategies for Model Selection

- Use test data $\tau' = (\mathbf{X}', \boldsymbol{y}')$ that are obtained independently from the training data τ, to estimate the generalization risk $\mathbb{E}\,\|Y - g_\tau(\boldsymbol{X})\|^2$ via the test loss.

  Then choose the collection of features that minimizes the test loss.

  When there is an abundance of data, part of the data can be reserved as test data, while the remaining data is used as training data.

- When there is a limited amount of data, we can use cross-validation to estimate the expected generalization risk $\mathbb{E}\,\|Y - g_{\mathcal{T}}(\boldsymbol{X})\|^2$ (where $\mathcal{T}$ is a random training set).

  This is then minimized over the set of possible choices for the explanatory variables.

Strategies for Model Selection

- Rather than using computer-intensive techniques, one can use theoretical estimates of the expected generalization risk, such as the in-sample risk, AIC, and BIC, and minimize this to determine a good set of explanatory variables.

- If the error terms are assumed to have a normal (Gaussian) distribution, then the inclusion and exclusion of variables can be decided by means of hypotheses tests. This is the classical approach to model selection.

- Finally, when using a Bayesian approach, comparison of two models can be achieved by computing their so-called *Bayes factor*.

Occam’s Razor

All of the above strategies can be thought of as specifications of a simple rule formulated by William of Occam, which can be interpreted as:

> *When presented with competing models, choose the simplest one that explains the data.*

This age-old principle, known as Occam’s razor, is mirrored in a famous quote of Einstein:

> *Everything should be made as simple as possible, but not simpler.*

In linear regression, the number of parameters or predictors is usually a reasonable measure of the simplicity of the model.