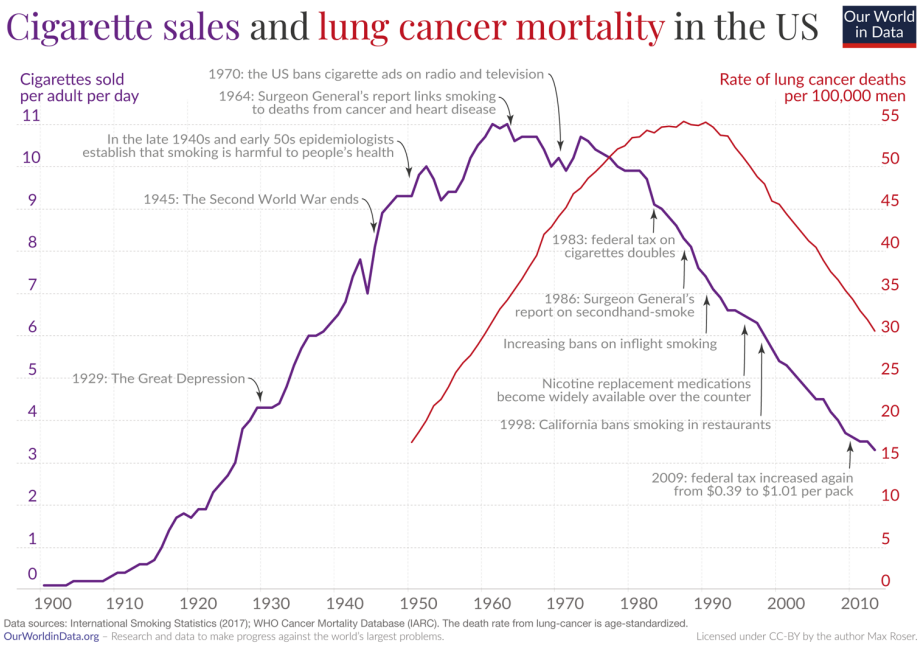
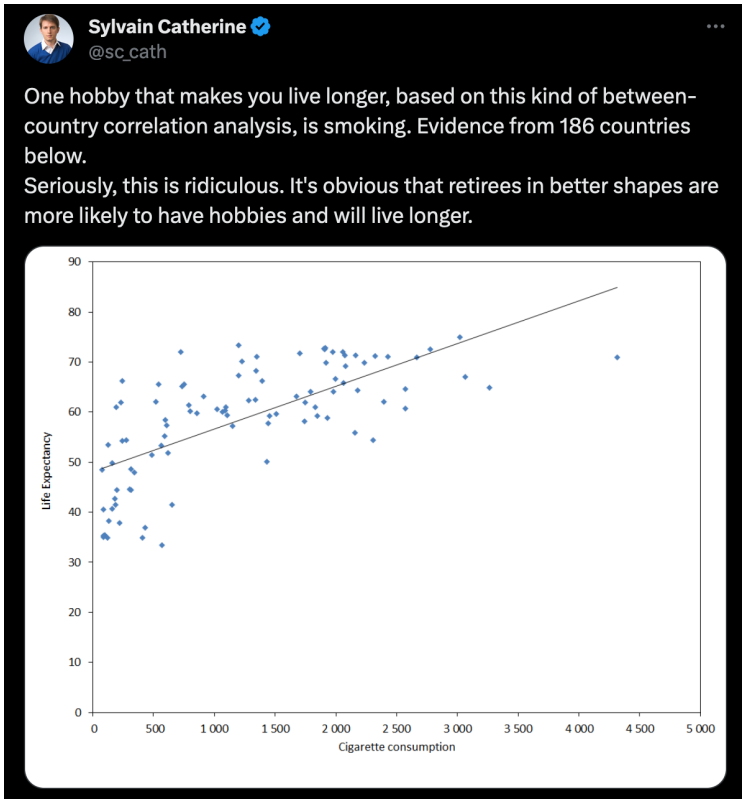
NBE-4070 : Basics of Biomedical Data Analysis

Stéphane Deny
Prof. in Neuroscience and Biomedial Engineering and Computer Science
Aalto University

Lecture 6: Linear Regression / Logistic Regression

How to establish causality between two variables?





Ideas?

How to establish causality between two variables?

- <u>For drugs and cures:</u> randomized trials (drug vs. placebo)

- <u>For dangerous substances:</u>
  - correlational studies, controlling for all other factors as much as possible (e.g. socio-economic status, diet, environmental factors, profession)
  - randomized trials on animals (substance vs. placebo)

- Demonstrate a plausible mechanism explaining the cause-and-effect

How to establish causality between two variables?

- Example of a plausible mechanism for how smoking can cause lung cancer

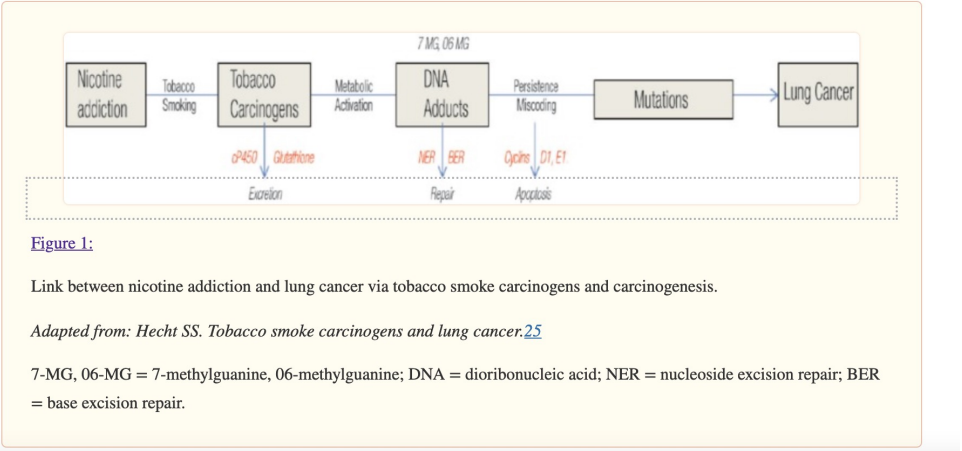


Figure 1:

Link between nicotine addiction and lung cancer via tobacco smoke carcinogens and carcinogenesis.

*Adapted from: Hecht SS. Tobacco smoke carcinogens and lung cancer.*25

7-MG, 06-MG = 7-methylguanine, 06-methylguanine; DNA = dioribonucleic acid; NER = nucleoside excision repair; BER = base excision repair.

Tobacco carcinogens are metabolised by cytochrome P-450 enzymes to make them readily excretable. Lipoxygenase, cyclooxygenase, myeloperoxidases, and monoamine oxidases may also be involved, although infrequently. The oxygenated intermediate metabolites undergo subsequent transformations (detoxification and secretion) by glutathiones, sufatases, or uridine-5′-diphosphate-glucuronosyltransferases (U5′DPGT).25 A few of the metabolites generated during these processes react with the deoxyribonucleic acid (DNA) to form covalent binding products called DNA adducts in a process called metabolic activation. Carcinogens like polcyclic aromatic hydrocarbons (PAH) and 4-(methylnitrosamino)-1-(3-pyridyl)-1-butanone (NNK) require metabolic activation to exert their carcinogenic effects. The carcinogenic metabolites of PAH-benzopyrenes (i.e. 7,8 diol 9,10 epioxides) and nicotine-derived nitrosamine ketone (NNK or NNAL) react with DNA to form adducts. Alpha-hydroxylase converts methyl adducts from the former agent to form 7-methylguanine or O6 methylguanine. The damage may be repaired, or apoptosis may ensue. Miscoding may result in permanent mutations, including *K-Ras*, *p53, p16*, fragile histidine triad protein (F-HIT), or unknown mutations, which results in either the suppression of tumour suppressor genes or the activation of oncogenes. Not all smokers get lung cancer, but under 20% do. Susceptibility to the development of cancer depends on the balance between metabolic activation and detoxification of potential carcinogens in smokers [Figure 1].25

Outline of the course

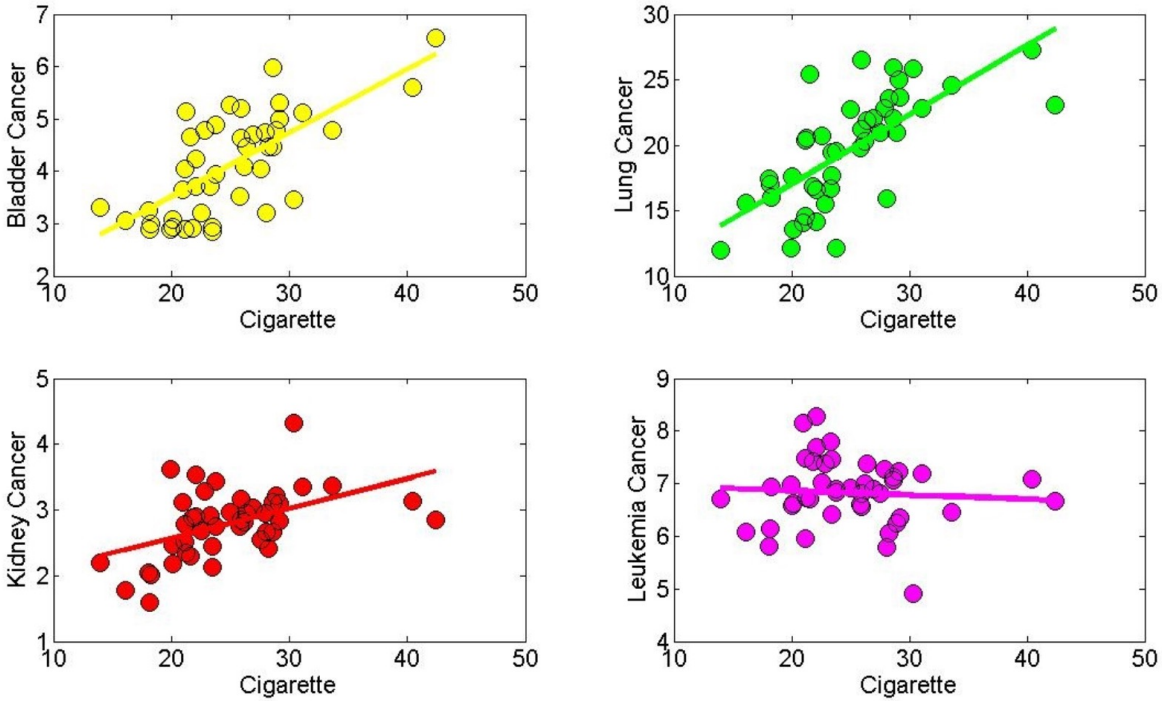
1. Mean, Standard Deviation, Standard Error, Confidence Intervals, T-test
2. Fourier Transform, Wavelet Transforms, Spectrograms, High-pass, Low-pass filters
3. Covariance and Principal Component Analysis (PCA)
4. Clustering Methods
5. Pearson Correlation, PCA and SVD
6. Linear Regression / Logistic Regression
7. Non-linear Methods: Independant Component Analysis, t-Stochastic Neighbour Embedding, Random Forests, Deep Networks
8. Oral exam preparation / Invited lectures from the biomedical industry

Example of linear regression: predicting state-wide cancer prevalance from cigarette consumption

Data

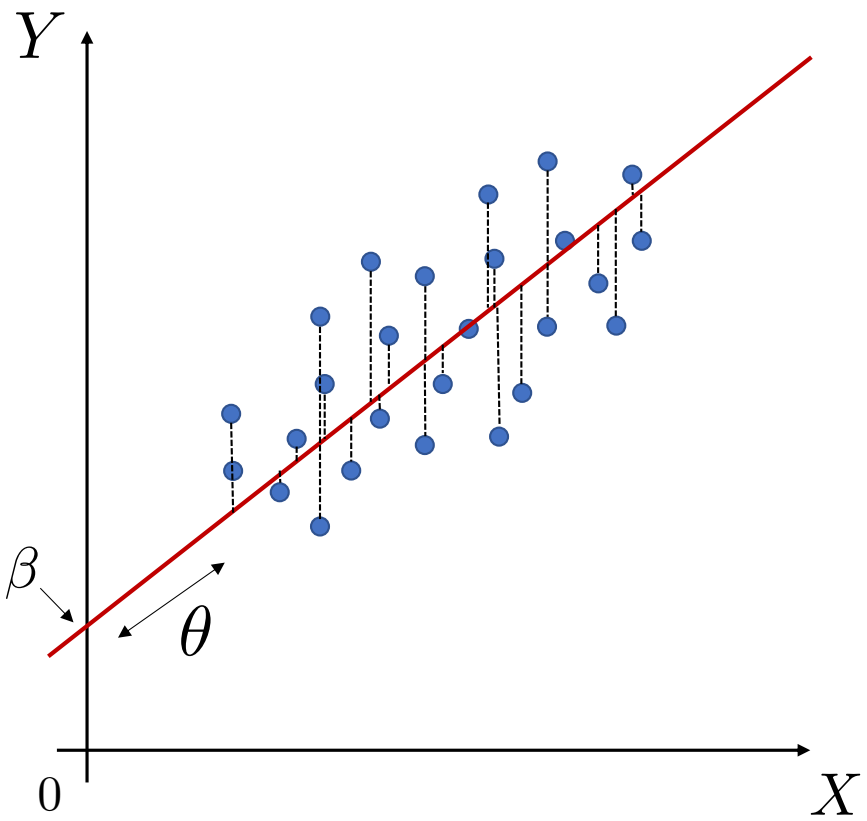
STATE	CIG	BLAD	LUNG	KID	LEUK
AL	18.20	2.90	17.05	1.59	6.15
AZ	25.82	3.52	19.80	2.75	6.61
AR	18.24	2.99	15.98	2.02	6.94
CA	28.60	4.46	22.07	2.66	7.06
CT	31.10	5.11	22.83	3.35	7.20
DE	33.60	4.78	24.55	3.36	6.45

STATE in 1960s
CIG = Number of cigarettes smoked (hds per capita)
BLAD = Deaths per 100K population from bladder cancer
LUNG = Deaths per 100K population from lung cancer
KID = Deaths per 100K population from bladder cancer
LEUK = Deaths per 100 K population from leukemia



source: https://prezi.com/mqzonna0ebvu/regression-in-biomedical-engineering/

Linear regression : definition



Linear regression predicts the value of an output variable Y based on the value of an input variable X, assuming a linear relationship between X and Y:

$$\widehat{y_i} = \theta * x_i + \beta$$

where x_i is the **input** variable for sample *i*

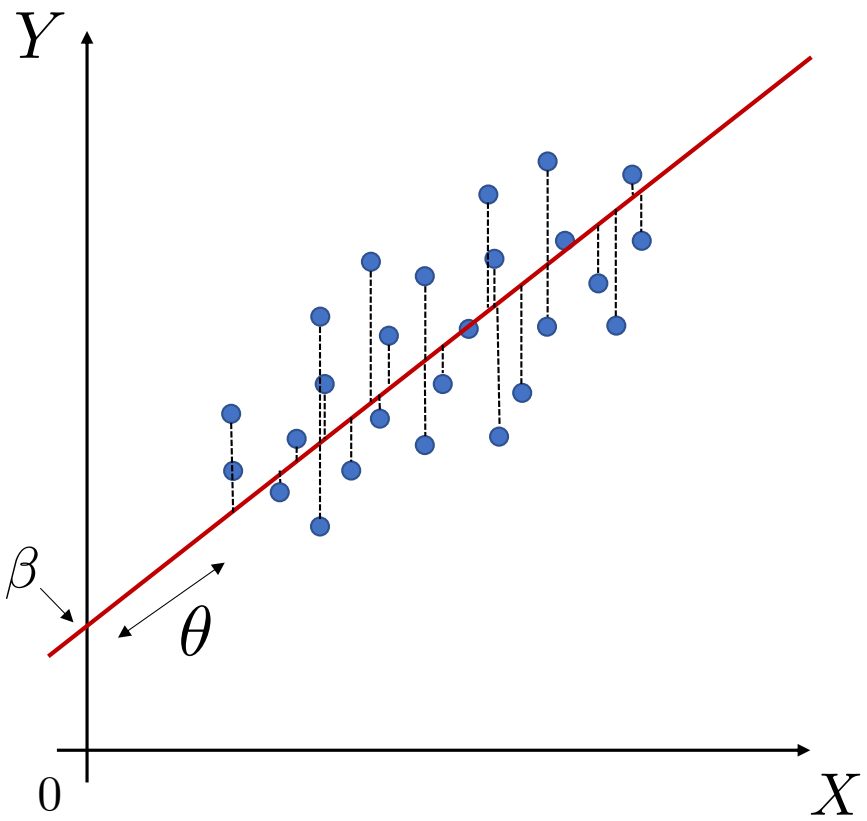
y_i is the **output** variable for sample *i*

$\widehat{y_i}$ is the **prediction** for sample *i*

β is the **intercept** of the linear fit

θ is the **slope** of the linear fit

Linear regression : optimization criterion



Linear regression finds β and θ such that the **mean squared error (MSE)** is minimized* :

$$\mathcal{L}_{\theta,\beta} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2$$

where $\mathcal{L}_{\theta,\beta}$ is the **loss function** (to be minimized)

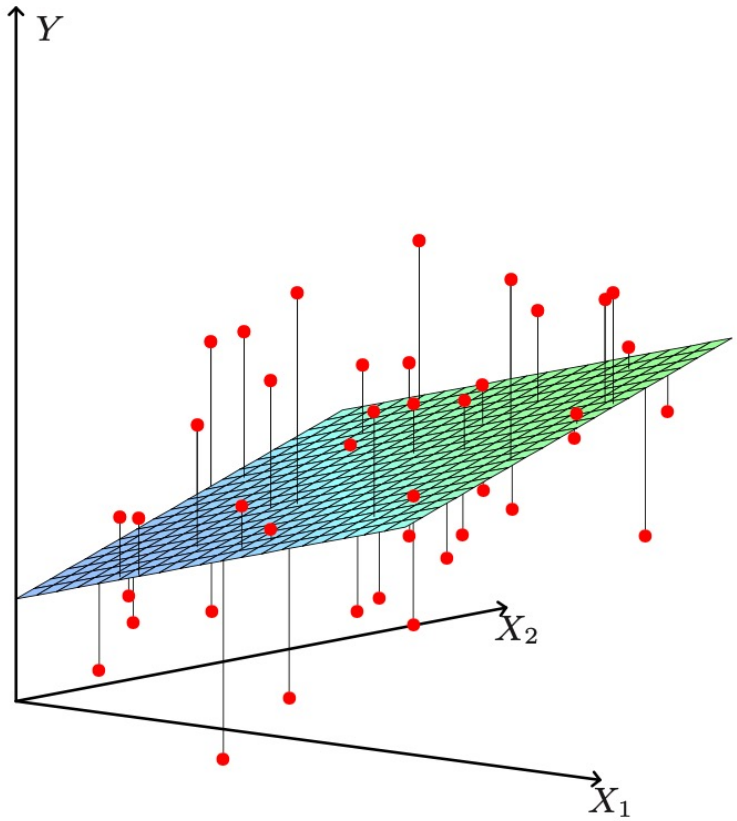
y_i is the **output** variable for sample *i*

$\widehat{y_i}$ is the **prediction** for sample *i*

N is the **number** of samples

*cf. supplementary slides for solution

Multiple linear regression



If there are more than one input variables, the process is called **multiple linear regression.** Like in simple linear regression, we seek the linear function of the inputs:

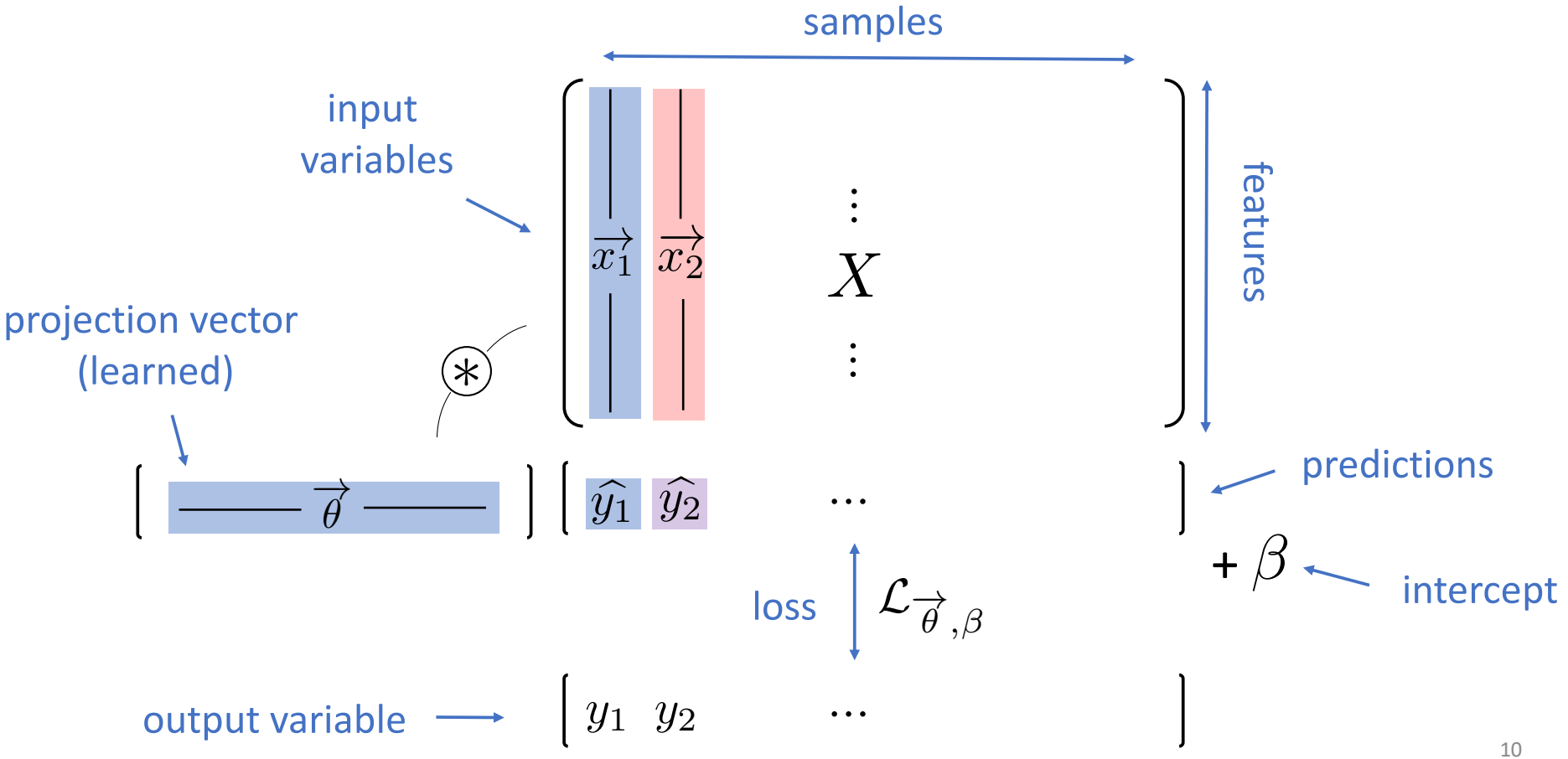
$$\widehat{y_i} = \langle \overrightarrow{\theta}, \overrightarrow{x_i} \rangle + \beta$$

that minimizes the mean squared error loss function*:

$$\mathcal{L}_{\overrightarrow{\theta},\beta} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \widehat{y_i})^2$$

*cf. supplementary slides for solution

Multiple linear regression (algebraic view)



Logistic regression : history

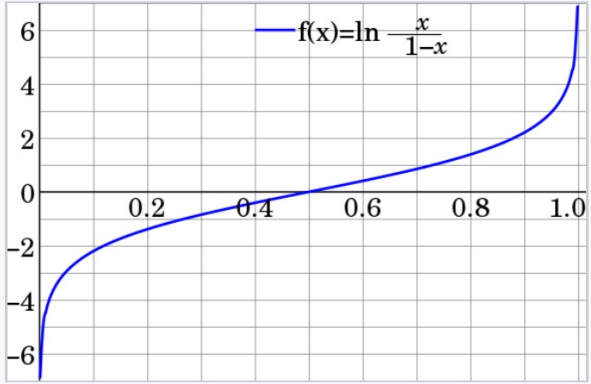
History [edit]

There have been several efforts to adapt linear regression methods to a domain where the output is a probability value, $(0, 1)$, instead of any real number $(-\infty, +\infty)$. In many cases, such efforts have focused on modeling this problem by mapping the range $(0, 1)$ to $(-\infty, +\infty)$ and then running the linear regression on these transformed values. In 1934 Chester Ittner Bliss used the cumulative normal distribution function to perform this mapping and called his model probit an abbreviation for "**prob**ability un**it**";.[2] However, this is computationally more expensive. In 1944, Joseph Berkson used log of odds and called this function *logit,* abbreviation for "**log**istic un**it**" following the analogy for probit:[3]

Mathematically, the logit is the inverse of the standard logistic function $\sigma(x) = 1/(1 + e^{-x})$, so the logit is defined as

$$\operatorname{logit} p = \sigma^{-1}(p) = \ln \frac{p}{1-p} \quad \text{for} \quad p \in (0, 1).$$

Because of this, the logit is also called the **log-odds** since it is equal to the logarithm of the odds $\frac{p}{1-p}$ where p is a probability. Thus, the logit is a type of function that maps probability values from $(0, 1)$ to real numbers in $(-\infty, +\infty)$,[1] akin to the probit function.



Plot of logit(*x*) in the domain of 0 to 1, where the base of the logarithm is *e*.

source: https://en.wikipedia.org/wiki/Logit

Logistic regression : definition

Logistic regression models the probability of an event taking place by having the **log-odds for the event** be a **linear combination of the input variables:**

$$log\left[\frac{p(\widehat{y_i}=1)}{1-p(\widehat{y_i}=1)}\right] = \langle \overrightarrow{\theta}, \overrightarrow{x_i} \rangle + \beta$$

The loss function to optimize is given by the **negative log-likelihood of the observed data as a function of the predicted distribution***:

$$\mathcal{L}_{\overrightarrow{\theta},\beta} = -\sum_{i=1}^{N} [y_i \log(p(\widehat{y_i}=1)) + (1-y_i)\log(1-p(\widehat{y_i}=1))]$$

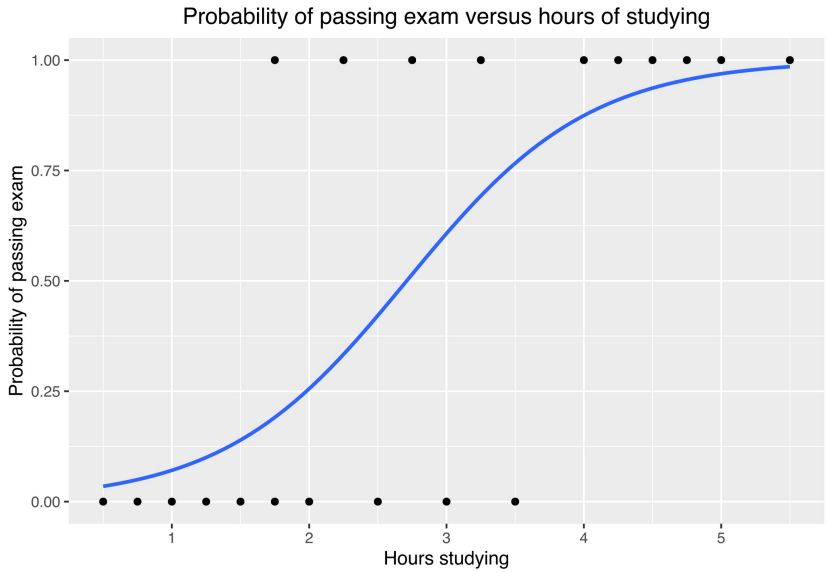
When the output variable is binary {0,1}, **logistic regression** is the correct tool to use

*cf. supplementary slides for algorithm

Example of simple logistic regression

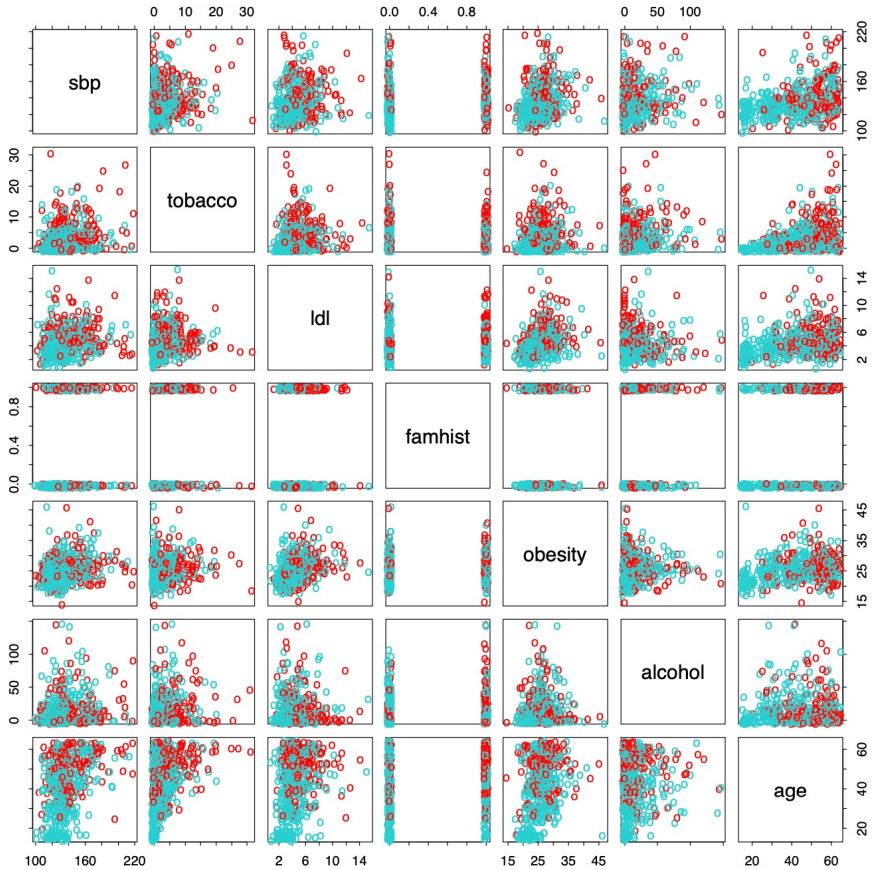
The table shows the number of hours each student spent studying, and whether they passed (1) or failed (0).

Hours (x_k)	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass (y_k)	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1



$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Case study for multiple logistic regression: the South African heart disease data



Here we present an analysis of binary data to illustrate the traditional statistical use of the logistic regression model. The data in Figure 4.12 are a subset of the Coronary Risk-Factor Study (CORIS) baseline survey, carried out in three rural areas of the Western Cape, South Africa (Rousseauw et al., 1983). The aim of the study was to establish the intensity of ischemic heart disease risk factors in that high-incidence region. The data represent white males between 15 and 64, and the response variable is the presence or absence of myocardial infarction (MI) at the time of the survey (the overall prevalence of MI was 5.1% in this region). There are 160 cases in our data set, and a sample of 302 controls. These data are described in more detail in Hastie and Tibshirani (1987).

FIGURE 4.12. *A scatterplot matrix of the South African heart disease data. Each plot shows a pair of risk factors, and the cases and controls are color coded (red is a case). The variable* family history of heart disease (`famhist`) *is binary (yes or no).*

Case study

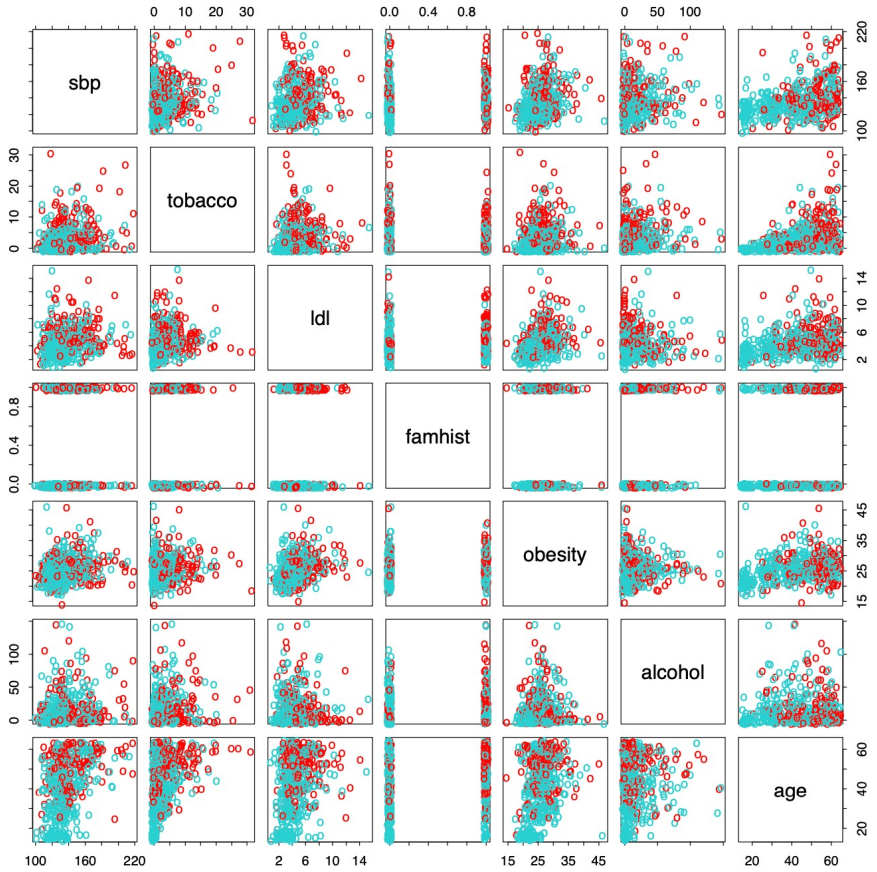


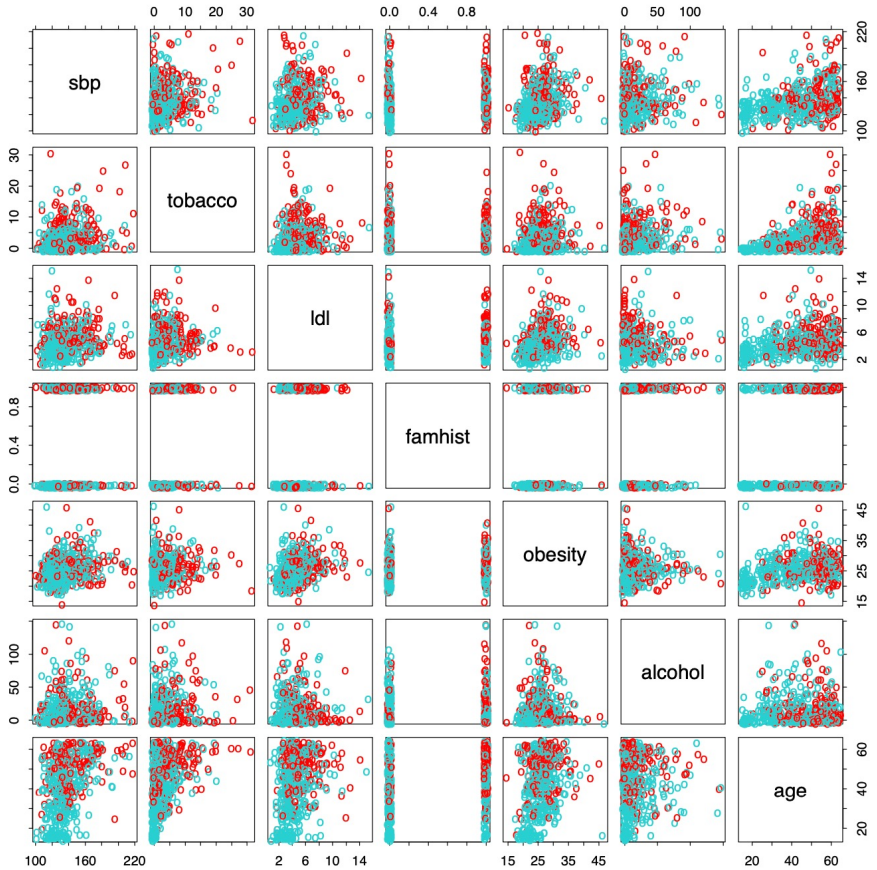
TABLE 4.2. *Results from a logistic regression fit to the South African heart disease data.*

	Coefficient	Std. Error	Z Score
`(Intercept)`	−4.130	0.964	−4.285
`sbp`	0.006	0.006	1.023
`tobacco`	0.080	0.026	3.034
`ldl`	0.185	0.057	3.219
`famhist`	0.939	0.225	4.178
`obesity`	-0.035	0.029	−1.187
`alcohol`	0.001	0.004	0.136
`age`	0.043	0.010	4.184

We fit a logistic-regression model by maximum likelihood, giving the results shown in Table 4.2. This summary includes Z scores for each of the coefficients in the model (coefficients divided by their standard errors); a nonsignificant Z score suggests a coefficient can be dropped from the model. Each of these correspond formally to a test of the null hypothesis that the coefficient in question is zero, while all the others are not (also known as the Wald test). A Z score greater than approximately 2 in absolute value is significant at the 5% level.

There are some surprises in this table of coefficients, which must be interpreted with caution. Systolic blood pressure (`sbp`) is not significant! Nor is `obesity`, and its sign is negative. This confusion is a result of the correlation between the set of predictors. On their own, both `sbp` and `obesity` are significant, and with positive sign. However, in the presence of many other correlated variables, they are no longer needed (and can even get a negative sign).

Case study (3)



At this stage the analyst might do some model selection; find a subset of the variables that are sufficient for explaining their joint effect on the prevalence of `chd`. One way to proceed by is to drop the least significant coefficient, and refit the model. This is done repeatedly until no further terms can be dropped from the model. This gave the model shown in Table 4.3.

TABLE 4.3. *Results from stepwise logistic regression fit to South African heart disease data.*

	Coefficient	Std. Error	Z score
`(Intercept)`	−4.204	0.498	−8.45
`tobacco`	0.081	0.026	3.16
`ldl`	0.168	0.054	3.09
`famhist`	0.924	0.223	4.14
`age`	0.044	0.010	4.52

How does one interpret a coefficient of 0.081 (Std. Error = 0.026) for `tobacco`, for example? Tobacco is measured in total lifetime usage in kilograms, with a median of 1.0kg for the controls and 4.1kg for the cases. Thus an increase of 1kg in lifetime tobacco usage accounts for an increase in the odds of coronary heart disease of $\exp(0.081) = 1.084$ or 8.4%. Incorporating the standard error we get an approximate 95% confidence interval of $\exp(0.081 \pm 2 \times 0.026) = (1.03, 1.14)$.

Sources of misinterpretation of the results of a regression analysis

1. Positive linear coefficient θ on an input variable ≠ causation (because in general correlation does not imply causation)

2. Conversely, a small regression coefficient for an input variable ***does not mean*** that this variable is not causally related to the output variable. It might simply be **less correlated to the output variable** than other input variables and thus ignored in the regression.

3. Statistical significance of the predictions ≠ strong predictive power (because statistical significance ≠ effect size significance)

The Overfitting Problem

Definition of overfitting:

- Predictions are good on the training dataset, but poor on new unseen data

Overfitting typically happens when:

- the dataset does not contain enough samples to properly learn the regression and/or
- the number of input variables (i.e. features) is too large to properly learn the regression

To **detect** overfitting:

- split your dataset into a training and validation set (see next slides)
- compute **cross-validation** predictions and compare them to training set predictions (worse performance on the validation set means overfitting)

To **prevent** overfitting:

- use **regularization techniques**, which consist in adding extra constraints on the regression problem based on what you know from the data (see examples in next slides)

To detect overfitting, separate data into training, validation and testing sets

- **Training Dataset**: the sample of data used to fit the model.
- **Validation Dataset**: the sample of data used to provide an evaluation of a model fit on the dataset while tuning model hyperparameters.
- **Test Dataset**: The sample of data used to provide an evaluation of a final model fit on the dataset.

source: https://machinelearningmastery.com/difference-test-validation-datasets/

Strategies to create a validation set when you don’t have a lot of data

- **k-fold strategy:** randomly divide the dataset into k groups or folds of approximately equal size. The first fold is kept for testing and the model is trained on k-1 folds. The process is repeated K times and each time different fold or a different group of data points are used for validation.

- **Leave-one-out strategy:** K-fold cross validation taken to its logical extreme, with K equal to N, the number of data points in the set. That means that N separate times, the function approximator is trained on all the data except for one point and a prediction is made for that point.

Examples of regularization techniques

1. Ridge regression (i.e. L2 penalty)
2. LASSO regression (i.e. L1 penalty)
3. Principal component regression

1. Ridge regression (L2 penalty)

We seek to mimize the loss:

$$\mathcal{L}_{\overrightarrow{\theta},\beta} = \underbrace{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2}_{\text{Mean squared error}} + \lambda \underbrace{\sum_{j=1}^{p}\theta_j^2}_{\text{L2 penalty}}$$

where

y_i is the **output** variable for sample *i*

$\widehat{y_i}$ is the **prediction** for sample *i*

N is the number of **samples**

p is the number of **input variables**

θ_j is the **coefficient of the regression** for each input variable (i.e. slope)

λ is the **trade-off parameter**, balancing the minimization of the MSE and the penalty term on the coefficients (also called Lagrangian)

1. Ridge regression (L2 penalty)

We seek to mimize the loss:

$$\mathcal{L}_{\overrightarrow{\theta},\beta} = \underbrace{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2}_{\text{Mean squared error}} + \underbrace{\lambda \sum_{j=1}^{p}\theta_j^2}_{\text{L2 penalty}}$$

where

y_i is the **output** variable for sample *i*

$\widehat{y_i}$ is the **prediction** for sample *i*

N is the number of **samples**

p is the number of **input variables**

θ_j is the **coefficient of the regression** for each input variable (i.e. slope)

λ is the **trade-off parameter**, balancing the minimization of the MSE and the penalty term on the coefficients (also called Lagrangian)

How to choose the hyperparameter lambda?
=> by trial-and-error on the validation set performance

2. LASSO regression (L1 penalty)

We seek to mimize the loss:

$$\mathcal{L}_{\overrightarrow{\theta},\beta} = \underbrace{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2}_{\text{Mean squared error}} + \underbrace{\lambda \sum_{j=1}^{p} |\theta_j|}_{\text{L1 penalty}}$$

where

y_i is the **output** variable for sample *i*

$\widehat{y_i}$ is the **prediction** for sample *i*

N is the number of **samples**

p is the number of **input variables**

θ_j is the **coefficient of the regression** for each input variable (i.e. slope)

λ is the **trade-off parameter**, balancing the minimization of the MSE and the penalty term on the coefficients (also called Lagrangian)

When to prefer LASSO vs. Ridge regression

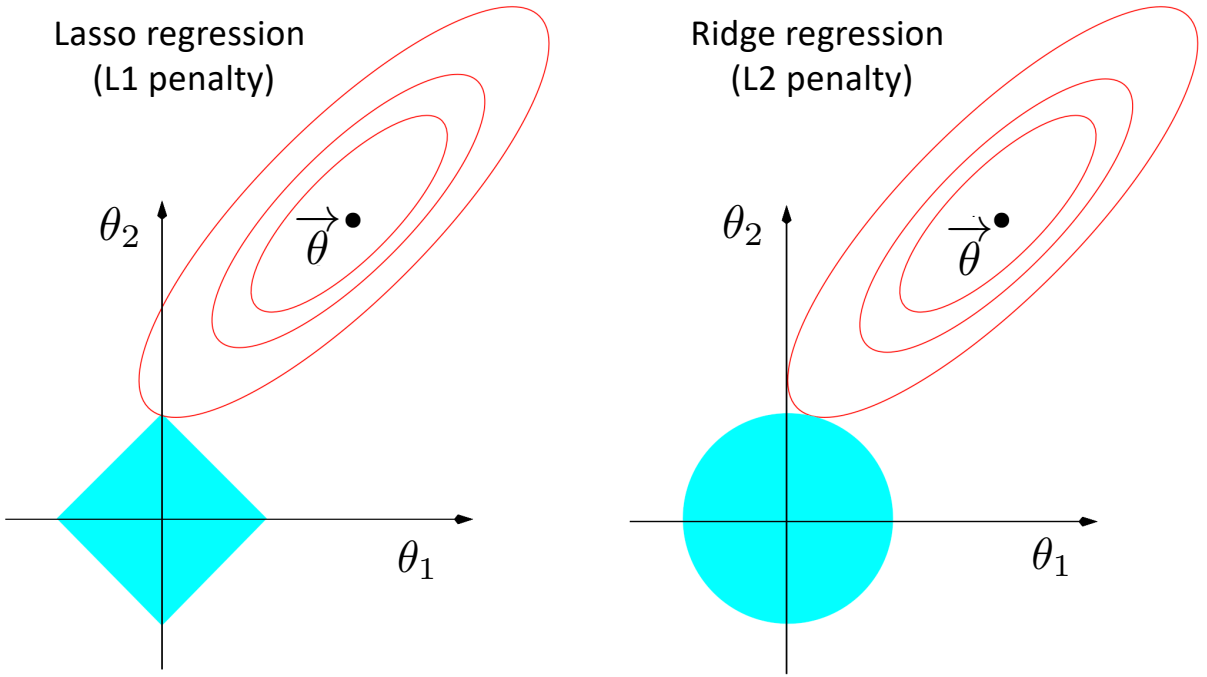


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

source: The Elements of Statstical Learning, Hastie, Tibshirani, Friedman (2017)

When to prefer LASSO vs. Ridge regression

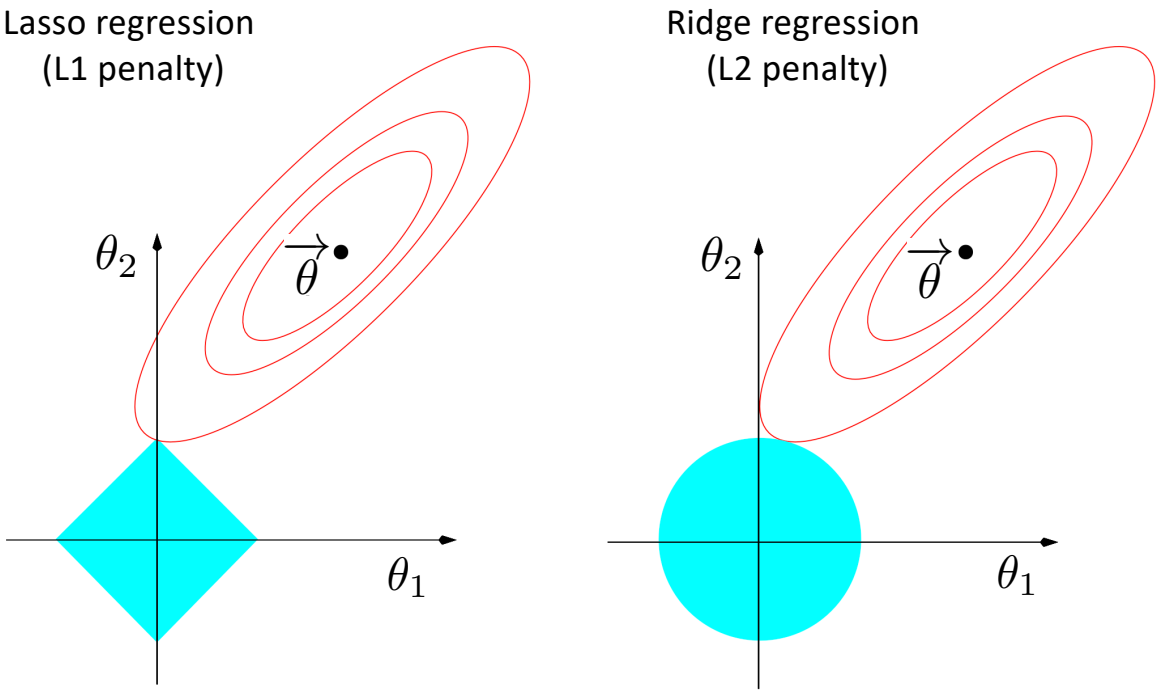


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

- Prefer LASSO regression when you believe that some features are not predictive of the output and can be ignored.
- Prefer ridge regression when you believe that directions in feature space will little variance are not very predictive of the output variable (e.g., too noisy).
- **Tip: try both regularizations and see what works best on the validation set!**

3. Principal Component Regression

A. Apply PCA to the dataset X, and retain only the scores corresponding to the k first dimensions of most variance (k can be chosen by cross-validation).

B. Fit regression to these truncated scores.

Use principal component regression as an alternative to ridge regression, when you believe that directions in feature space with little variance are not very meaningful or noisy.

Steps to perform a regression analysis on a computer

1. Separate the data X and Y into a training set and a validation set

2. Compute the regression fit using an existing function from the the computer and obtain the slope and intercept parameters of the fit

3. Test your regression performance on the validation set

4. if you find that the regression overfits to the training set, try different regularization methods (e.g., LASSO, ridge, principal component regression)

Next lecture on Oct 24

- Non-linear methods for data analysis

Supplementary material

Solution to linear regression

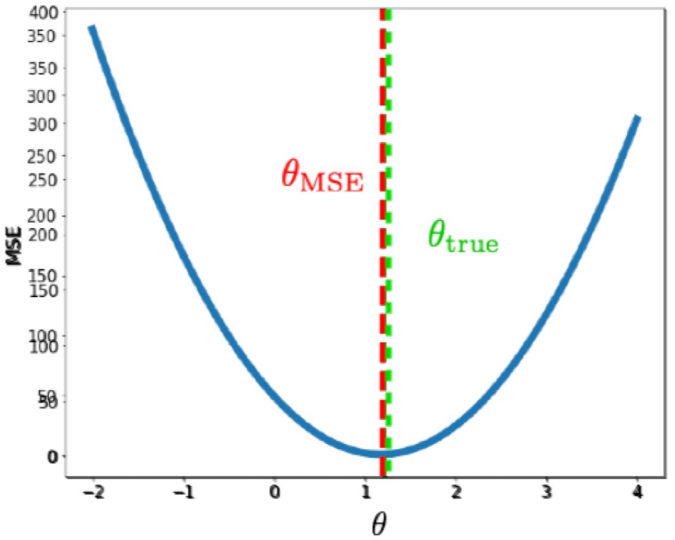
$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \theta x_i)^2$$

Optimal $$\theta^* = \underset{\theta}{\text{argmin}}\ \text{MSE} = \underset{\theta}{\text{argmin}}\ \frac{1}{N}\sum_{i=1}^{N}(y_i - \theta x_i)^2$$

To minimize MSE, we solve for where its gradient is 0:

$$\frac{\partial \text{MSE}}{\partial \theta} = -\frac{2}{N}\sum_{i=1}^{N}(y_i - \theta x_i)x_i = 0 \Rightarrow \theta\sum_{i=1}^{N}x_i^2 - \sum_{i=1}^{N}x_i y_i = 0$$

$$\Rightarrow \theta_{\text{MSE}} = \frac{\sum_{i=1}^{N}x_i y_i}{\sum_{i=1}^{N}x_i^2}$$



source: Neuromatch academy,
Anqi Wu: https://compneuro.neuromatch.io/tutorials/W1D2_ModelFitting/student/W1D2_Tutorial2.html

Solution to multiple linear regression

How do we minimize (3.2)? Denote by $\mathbf{X}$ the $N \times (p+1)$ matrix with each row an input vector (with a 1 in the first position), and similarly let $\mathbf{y}$ be the N-vector of outputs in the training set. Then we can write the residual sum-of-squares as

$$\mathrm{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta). \tag{3.3}$$

This is a quadratic function in the $p+1$ parameters. Differentiating with respect to β we obtain

$$\begin{aligned} \frac{\partial \mathrm{RSS}}{\partial \beta} &= -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) \\ \frac{\partial^2 \mathrm{RSS}}{\partial \beta \partial \beta^T} &= 2\mathbf{X}^T\mathbf{X}. \end{aligned} \tag{3.4}$$

Assuming (for the moment) that $\mathbf{X}$ has full column rank, and hence $\mathbf{X}^T\mathbf{X}$ is positive definite, we set the first derivative to zero

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0 \tag{3.5}$$

to obtain the unique solution

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{3.6}$$

source: The Elements of Statstical Learning, Hastie, Tibshirani, Friedman (2017)

Fitting logistic regression

4.4.1 Fitting Logistic Regression Models

Logistic regression models are usually fit by maximum likelihood, using the conditional likelihood of G given X. Since $\Pr(G|X)$ completely specifies the conditional distribution, the *multinomial* distribution is appropriate. The log-likelihood for N observations is

$$\ell(\theta) = \sum_{i=1}^{N} \log p_{g_i}(x_i;\theta), \tag{4.19}$$

where $p_k(x_i;\theta) = \Pr(G = k|X = x_i;\theta)$.

We discuss in detail the two-class case, since the algorithms simplify considerably. It is convenient to code the two-class g_i via a 0/1 response y_i, where $y_i = 1$ when $g_i = 1$, and $y_i = 0$ when $g_i = 2$. Let $p_1(x;\theta) = p(x;\theta)$, and $p_2(x;\theta) = 1 - p(x;\theta)$. The log-likelihood can be written

$$\begin{aligned}\ell(\beta) &= \sum_{i=1}^{N}\Big\{y_i \log p(x_i;\beta) + (1-y_i)\log(1-p(x_i;\beta))\Big\} \\ &= \sum_{i=1}^{N}\Big\{y_i\beta^T x_i - \log(1+e^{\beta^T x_i})\Big\}. \end{aligned} \tag{4.20}$$

Here $\beta = \{\beta_{10}, \beta_1\}$, and we assume that the vector of inputs x_i includes the constant term 1 to accommodate the intercept.

To maximize the log-likelihood, we set its derivatives to zero. These *score* equations are

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^{N} x_i(y_i - p(x_i;\beta)) = 0, \tag{4.21}$$

which are $p+1$ equations *nonlinear* in β. Notice that since the first component of x_i is 1, the first score equation specifies that $\sum_{i=1}^{N} y_i = \sum_{i=1}^{N} p(x_i;\beta)$; the *expected* number of class ones matches the observed number (and hence also class twos.)

To solve the score equations (4.21), we use the Newton–Raphson algorithm, which requires the second-derivative or Hessian matrix

$$\frac{\partial^2 \ell(\beta)}{\partial\beta\partial\beta^T} = -\sum_{i=1}^{N} x_i x_i^T p(x_i;\beta)(1-p(x_i;\beta)). \tag{4.22}$$

Starting with β^{old}, a single Newton update is

$$\beta^{\text{new}} = \beta^{\text{old}} - \left(\frac{\partial^2 \ell(\beta)}{\partial\beta\partial\beta^T}\right)^{-1}\frac{\partial \ell(\beta)}{\partial\beta}, \tag{4.23}$$

where the derivatives are evaluated at β^{old}.

It is convenient to write the score and Hessian in matrix notation. Let $\mathbf{y}$ denote the vector of y_i values, $\mathbf{X}$ the $N \times (p+1)$ matrix of x_i values, $\mathbf{p}$ the vector of fitted probabilities with ith element $p(x_i;\beta^{\text{old}})$ and $\mathbf{W}$ a $N \times N$ diagonal matrix of weights with ith diagonal element $p(x_i;\beta^{\text{old}})(1-p(x_i;\beta^{\text{old}}))$. Then we have

$$\frac{\partial \ell(\beta)}{\partial\beta} = \mathbf{X}^T(\mathbf{y}-\mathbf{p}) \tag{4.24}$$

$$\frac{\partial^2 \ell(\beta)}{\partial\beta\partial\beta^T} = -\mathbf{X}^T\mathbf{W}\mathbf{X} \tag{4.25}$$

The Newton step is thus

$$\begin{aligned}\beta^{\text{new}} &= \beta^{\text{old}} + (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{y}-\mathbf{p}) \\ &= (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\left(\mathbf{X}\beta^{\text{old}} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})\right) \\ &= (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{z}. \end{aligned} \tag{4.26}$$

In the second and third line we have re-expressed the Newton step as a weighted least squares step, with the response

$$\mathbf{z} = \mathbf{X}\beta^{\text{old}} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p}), \tag{4.27}$$

sometimes known as the *adjusted response*. These equations get solved repeatedly, since at each iteration $\mathbf{p}$ changes, and hence so does $\mathbf{W}$ and $\mathbf{z}$. This algorithm is referred to as *iteratively reweighted least squares* or IRLS, since each iteration solves the weighted least squares problem:

$$\beta^{\text{new}} \leftarrow \arg\min_{\beta}(\mathbf{z}-\mathbf{X}\beta)^T\mathbf{W}(\mathbf{z}-\mathbf{X}\beta). \tag{4.28}$$

It seems that $\beta = 0$ is a good starting value for the iterative procedure, although convergence is never guaranteed. Typically the algorithm does converge, since the log-likelihood is concave, but overshooting can occur. In the rare cases that the log-likelihood decreases, step size halving will guarantee convergence.

source: The Elements of Statistical Learning, Hastie, Tibshirani, Friedman (2017)

Blog post on linear regression

- https://towardsdatascience.com/machine-learning-for-biomedical-data-linear-regression-7d43461cdfa9