Statistical Signal Processing & Inference

Linear and Nonlinear Regression

Danilo Mandic

room 813, ext: 46271

Department of Electrical and Electronic Engineering
Imperial College London, UK

d.mandic@imperial.ac.uk, URL: www.commsp.ee.ic.ac.uk/∼mandic

Outline

Regression analysis examines the association between a **dependent variable**, y, and one or more **independent variables**, $x_1, x_2, \ldots, x_p$.

○ It determines whether and "how much" of the variation in the dependent variable can be explained by independent variables (relationship strength)

○ Regression analysis covers a range of linear and nonlinear models, from univariate regression to multiple regression, polynomial regression, and regression with multiplicative variables (Volterra series, Recurrent NNs)

Advantages of linear regression:

Interpretability: Regression models clearly establish how each independent variable affects the dependent variable

Simplicity: The concept of regression is relatively simple and intuitive, compared to most established machine learning models

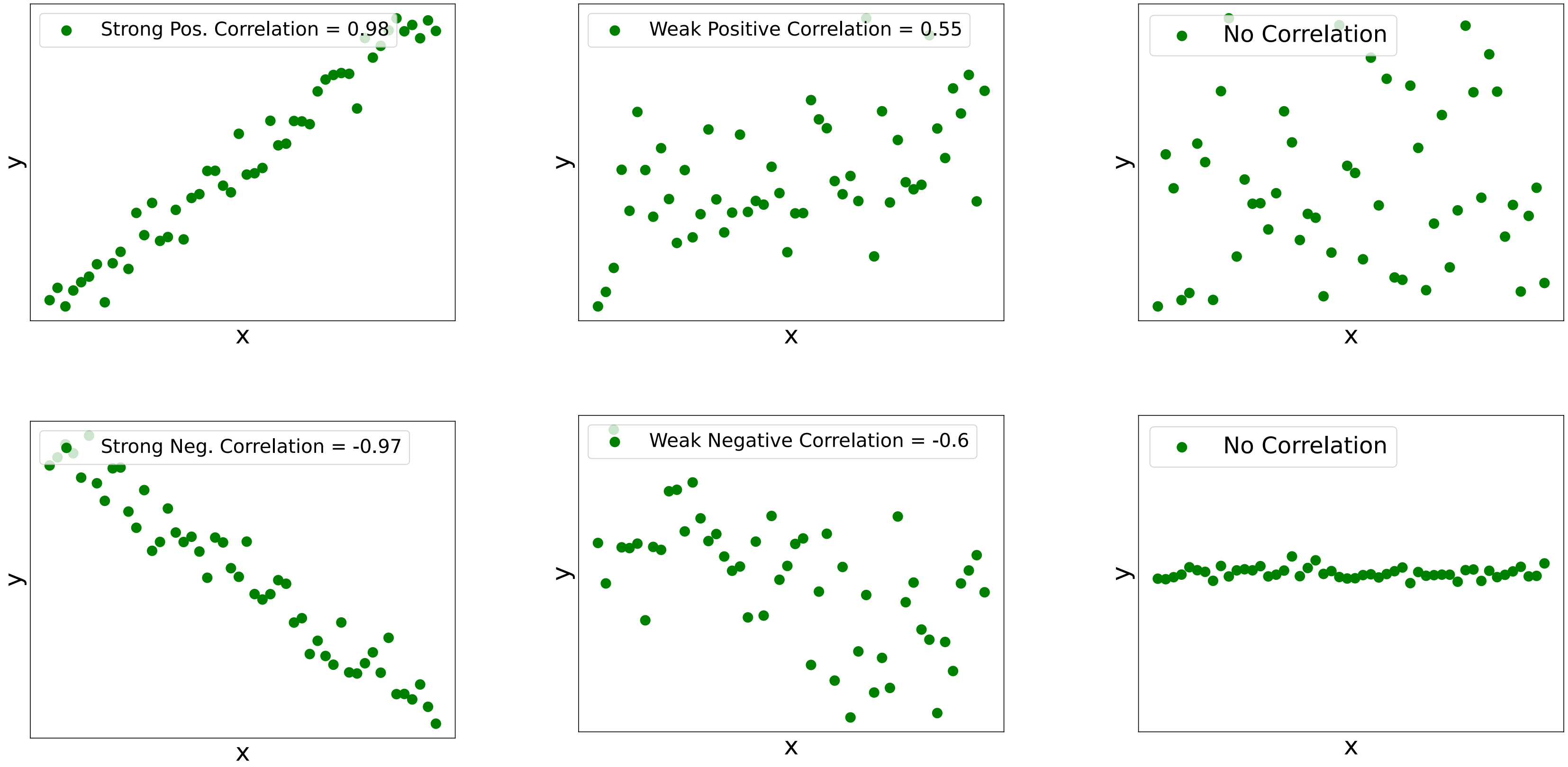
Applicability: It is an indispensable "must-try" tool in manifold fields, including finance, biomedicine, science and engineering

☞ **Logistic Regression** (which is linear in the "logit" space) is used for **classification problems** based on either binary or multiple categories

Visualising and quantifying relations between variables

Example 1: Scatter plots and correlation

Correlation quantifies the strength (scatter) and direction of the **linear relationship** between two variables, x and y, in both the x-direction and y-direction, as illustrated in the scatter plots below.



☞ In addition to the correlation analysis, it is very useful to quantify how two or more variables (co–)vary together $\looparrowright$ basis of regression

Correlation, partial correlation, non-metric correlation

○ The standard **bivariate or product moment correlation**, $r_{xy} = E\{xy\}$, measures the (linear) association between two numerical variables, x and y

○ This may be interpreted as examining the existence and strength of a linear "straight-line" relationship between the two variables, x and y

○ When normalised as $\rho_{xy} = \frac{cov(x,y)}{\sigma_x \sigma_y}$, it is known as the *Pearson correlation coefficient, or correlation coefficient*

○ In a multivariate case, the corr. coeff., ρ_{xy}, does not remove the effects of other variable(s) when quantifying the association between x and y

○ On the other hand, the **partial correlation coefficient** measures the association between two variables after adjusting for (or eliminating) the effects of additional variables, and has the form $\quad r_{xy\cdot z} = \frac{r_{xy} - r_{xz} r_{yz}}{\sqrt{1 - r_{xz}^2}\sqrt{1 - r_{yz}^2}}$

○ Recall from Lecture 2 that partial correlations are typically used to estimate the order of Autoregressive (AR processes); see the next slide

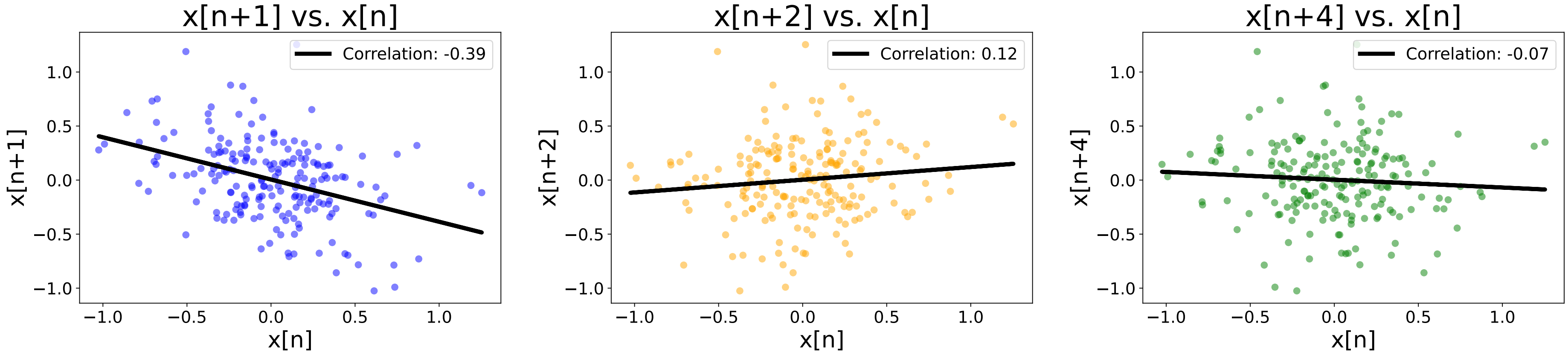
○ The coefficient of association, R^2 signifies the proportion of the total variation in y that is accounted for by the variation in x (see Appendix)

○ Spearman's rho, ρ_s, and Kendall's tau, τ, correlation coefficients can be used to measure the association between ordinal (categorical) variables

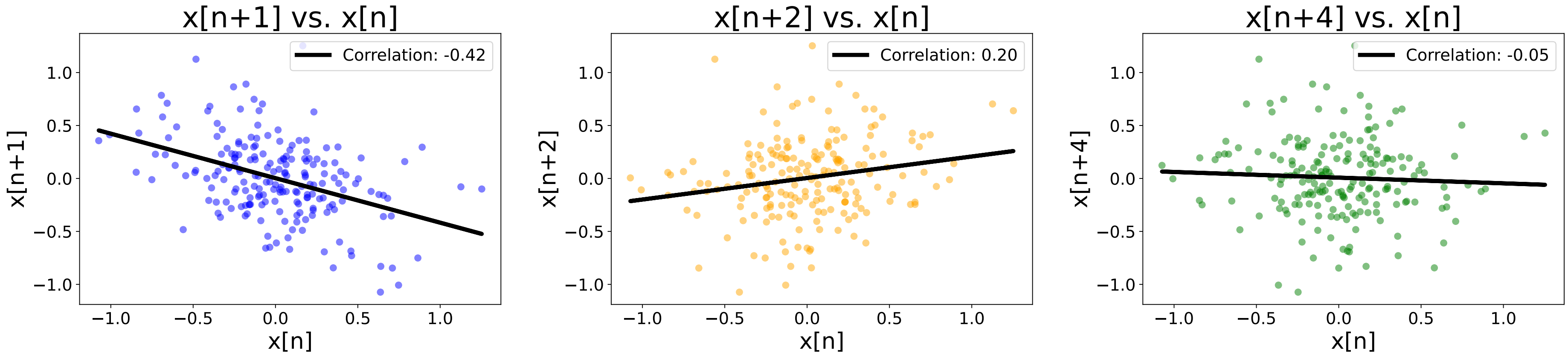
Example 2: Scatter plot, AutoRegressive process (Lecture 2)

Consider an AR(1) process with $a_1 = -0.3$, and an AR(2) with $a_1 < 0, a_2 > 0$

The nature of an AR process may be inferred through scatter plots of pairs $x[n], x[m+n]$, separated by an interval (lag), m.



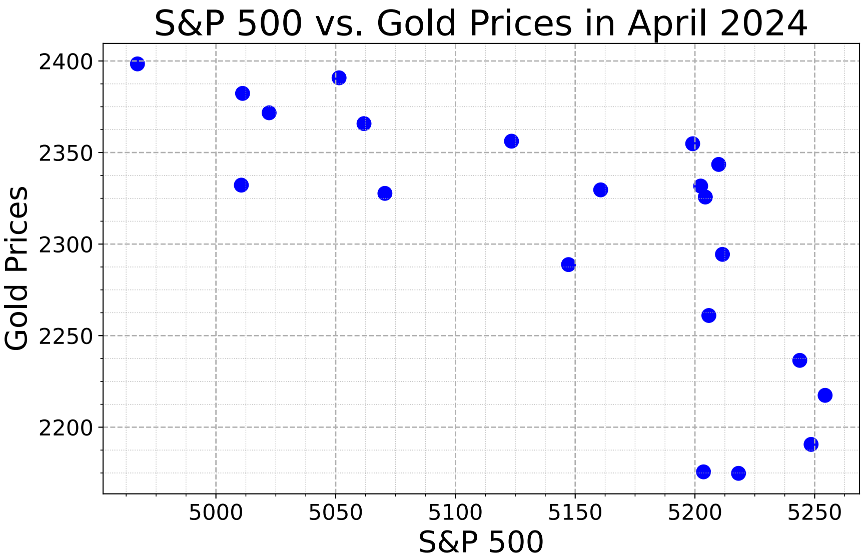
AR(1) process: $x[n] = -0.3x[n-1] + w[n]$

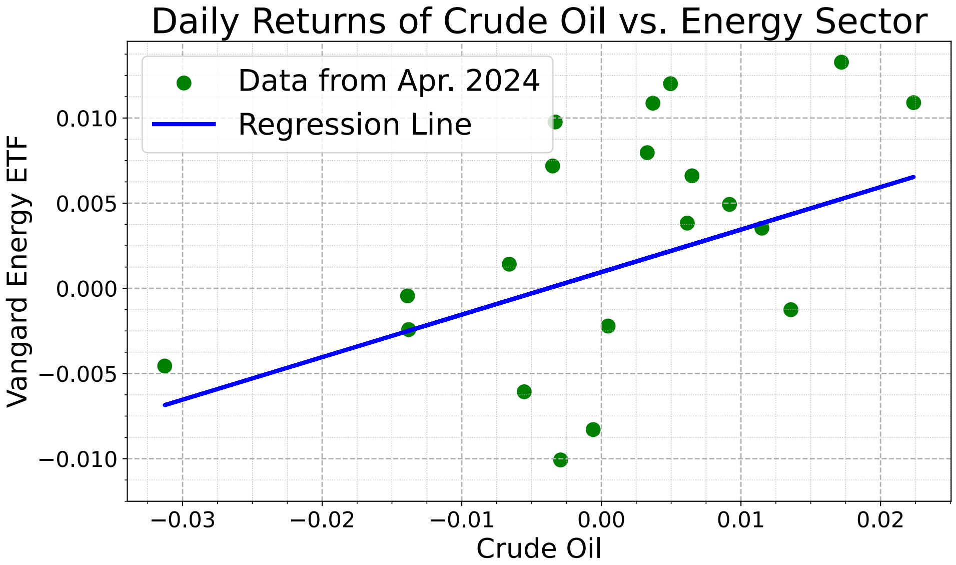


AR(2) process: $x[n] = -0.3x[n-1] + 0.1x[n-2] + w[n]$

Relationship between variables
Correlation versus Regression

After establishing how two variables co-vary, interpolation and prediction can be performed through **linear regression** ↬ **Regression line is unique**



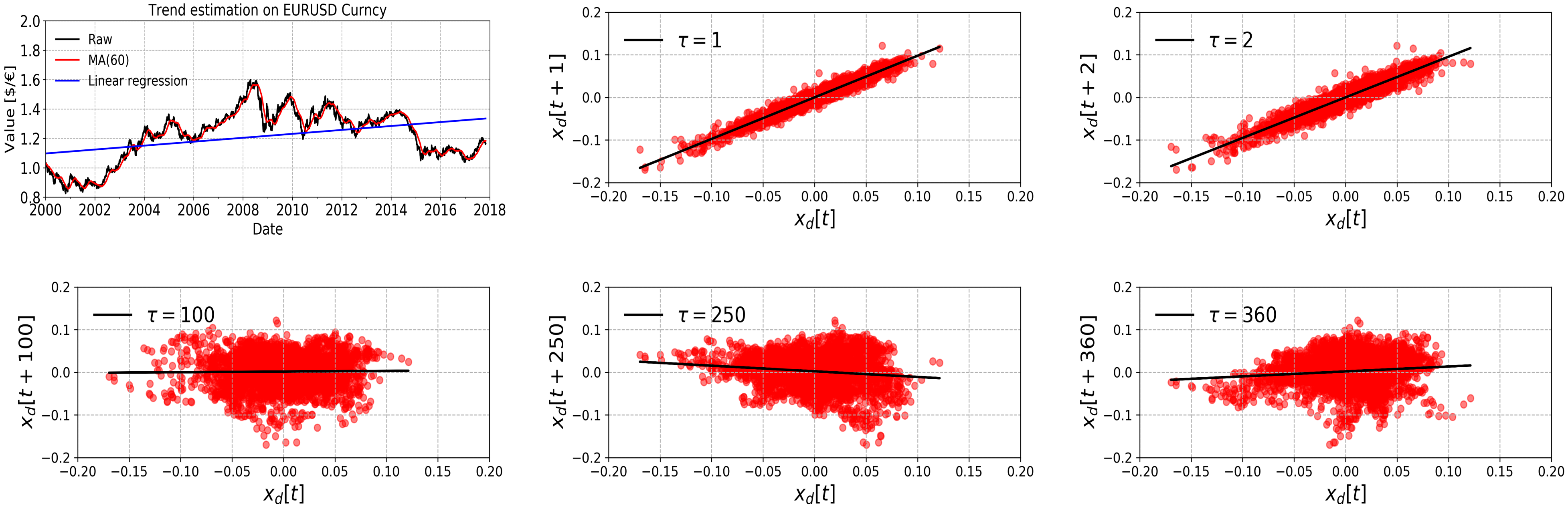


Regression examines the cumulative distance between all data points and the regression line in the y-direction only ↬ it models the variation of the explained variable, y, in response to a change in the explanatory variable, x

- Variable x is also called a regressor, independent variable, or predictor
- Variable y is also called response, dependent var., criterion or true label

Example 3: Scatter plots of the Euro vs USD exchange

Scatter plots of a detrended EUR/USD exchange rate vs its τ days lagged version



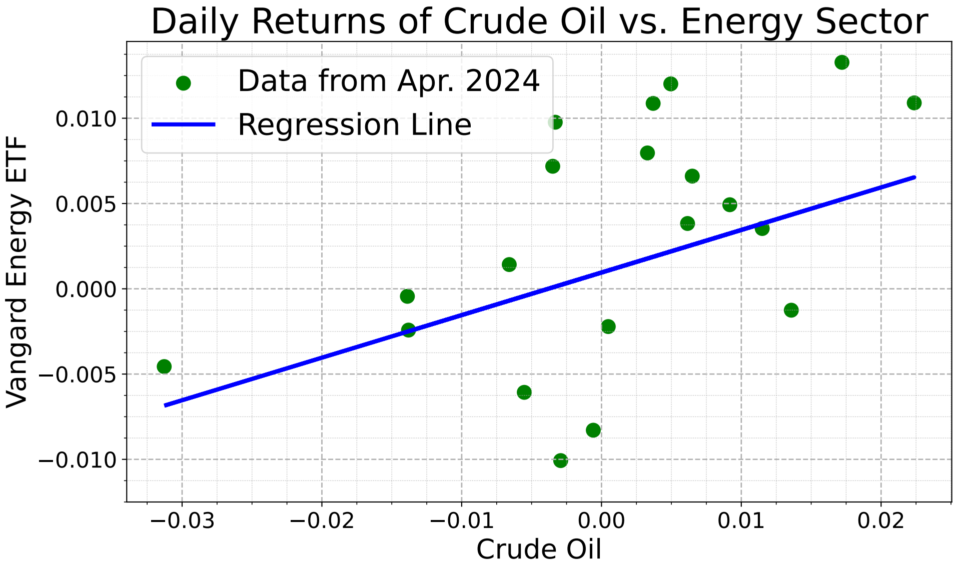
○ The detrended Euro to USD currency exchange time series shows strong correlation for small correlation lags ($\tau = 1, \tau = 2$), as evidenced by a narrow shape of the concentration of data points (the narrower the better)

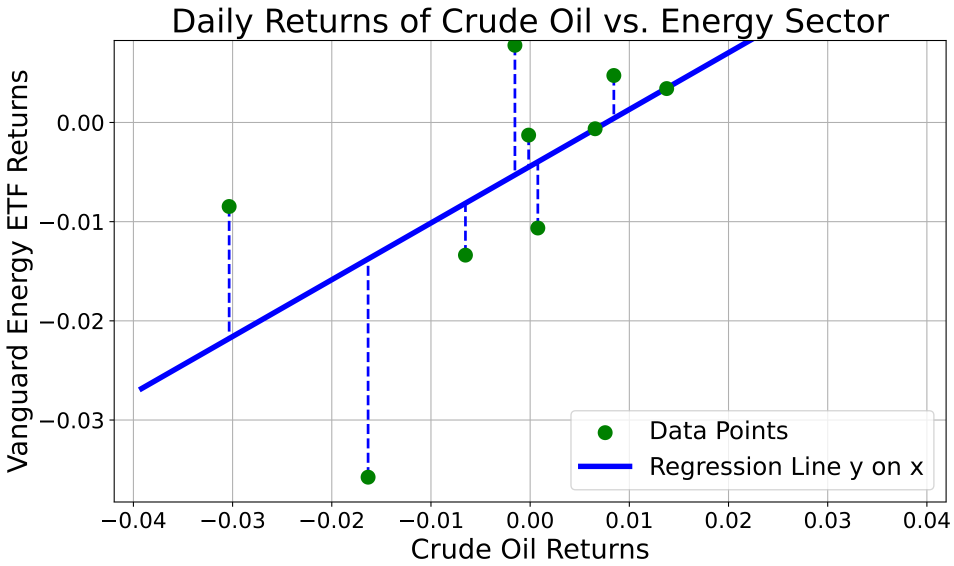
○ For large correlation lags ($\tau = 100, \tau = 250, \tau = 360$), the scatter plots indicate a lack of correlation or weak spurious correlations (see Slide 3)

Q: If x and y are correlated, can we infer the value of y based on x?

A: Yes, **we can regress y onto x and establish such inference!**

Linear Regression: The analysis framework


Daily Returns of Crude Oil vs. Energy Sector


Daily Returns of Crude Oil vs. Energy Sector

Regression models the "rate of change" of the explained variable, y, in response to the change in the explanatory variable, x, in the form

Univariate regression: $$y[n] = \alpha + \beta x[n] + e[n]$$

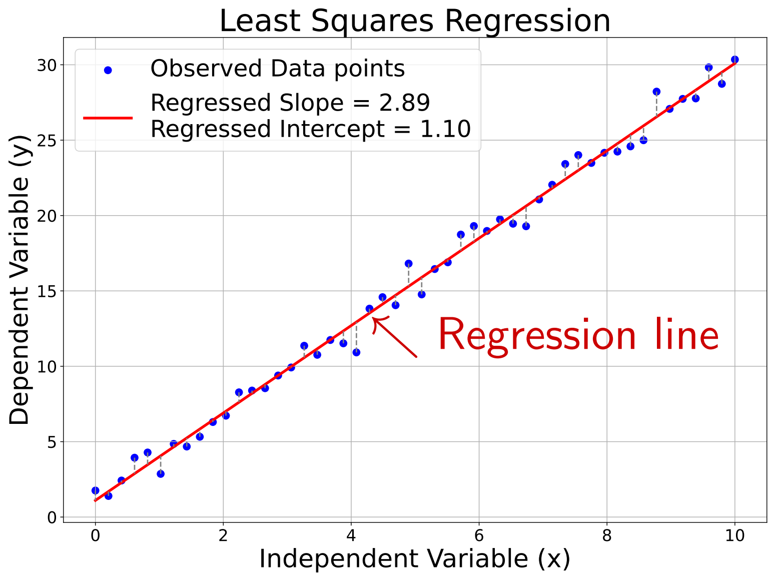
- The **error (or residual)**, $e[n]$, accounts for the stochastic nature of regression
- The **slope** of the regression line, $\beta = r\frac{\sigma_y}{\sigma_x}$, with r as the correlation coefficient between x and y, and σ_x and σ_y as the standard deviations in the x and y
- The **intercept** $\alpha = \bar{y} - \beta\bar{x}$ where $\bar{x}$ and $\bar{y}$ are the sample means of x and y

The regression line passes through the centroid $(\bar{x}, \bar{y})$

Finding regression coefficients: Least Squares Regression

Linear regression ↬ relationship between two variables based on a line of best fit

Consider a line fit: $\boldsymbol{y} = \beta\,\boldsymbol{x} + \boldsymbol{e} \Longleftrightarrow y_i = \beta\,x_i + e_i \quad i \in \{1, \ldots, N\}$



○ Least Squares regression (LSR) aims to minimise the sum of the squares of the differences between the observed and predicted values

$$\underset{\beta}{\operatorname{argmin}}\, ||\boldsymbol{y} - \beta\boldsymbol{x}||_2^2 \Longleftrightarrow \underset{\beta}{\operatorname{argmin}}\, ||\boldsymbol{e}||_2^2$$

○ We say that we regress y onto x, with β as the regression coefficient.

Common terminologies for Least Squares Regression

	Econometrics	**Statistics**	**Machine Learning**
$\boldsymbol{y}$	Dependent Var., Estimate	Explained V., Response, Regressand	True Label, Criterion
β	Coefficients	Coefficients	Parameters
$\boldsymbol{x}$	Independent Var., Predictor	Explanatory Var. Regressor	Features, Predictors
$\boldsymbol{e}$	Residual	Error	Prediction Error

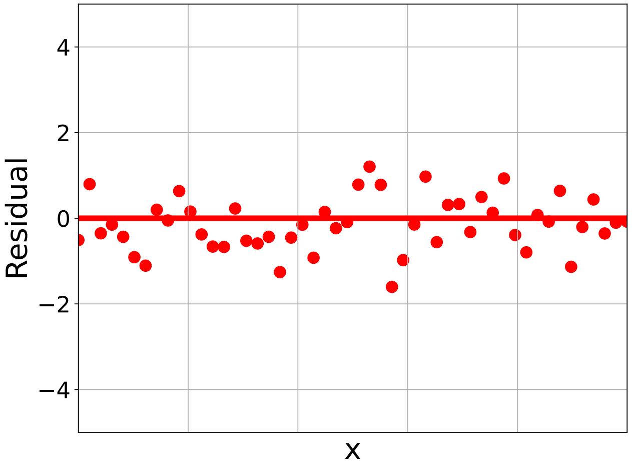
Goodness of regression fit: Examination of residuals

The Regression Analysis involves the following assumptions:

Linearity. The relationship between the dependent and independent variable(s) should be close to a straight line (hyper-plane).

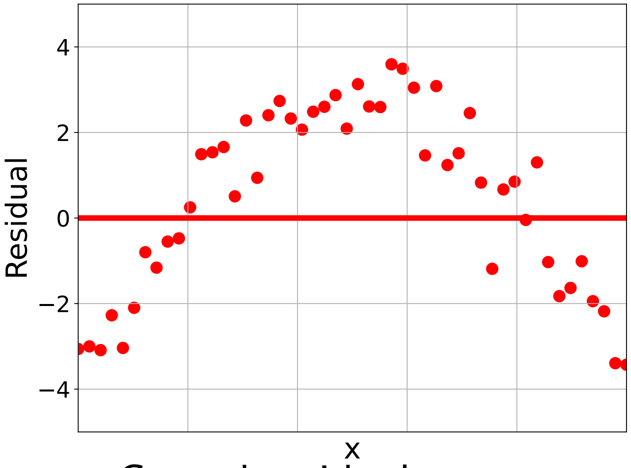
Homo-scedasticity. The residuals exhibit a zero-mean, $E\{e\} = 0$, and constant variance, $E\{e^2\} = \sigma^2$, which does not depend on the value of x.

☞ The error terms are independent and normally distributed (i.i.d.) around the regression line. For each fixed value of x the distribution of y is normal.



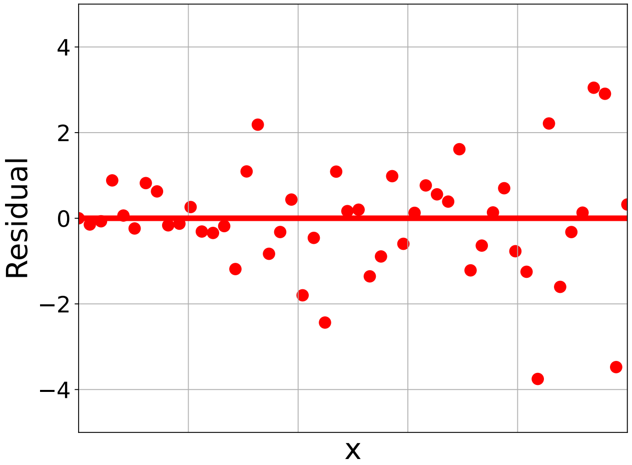
Randomly scattered residuals

Good: Assumptions satisfied



Curved residual pattern

Bad: Non-linear relation



Variance increase with x

Bad: Non-constant variance

☞ **Regression does not imply or assume any causality**

Example 4: Capital Asset Pricing Model (CAPM)

W. Sharpe was awarded the Nobel Prize for economy in 1990 for CAPM

The CAPM is given by the following linear regression model

$$E(R_i) = R_f + \beta\left(E(R_m) - R_f\right) + e$$

expected return of asset i ↗ risk-free ↑ ↑ exposure to market ↖ residual (unpredictable)

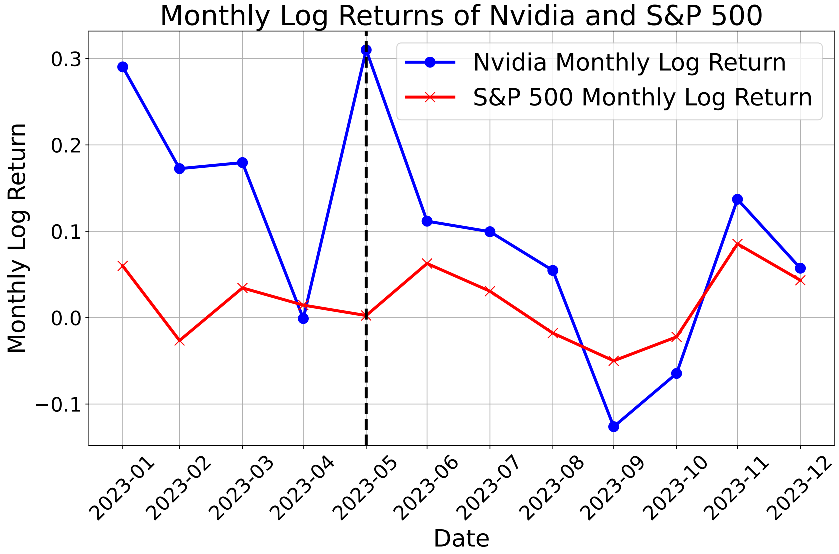
○ R_f is the **risk-free** rate of interest, e.g. interest arising from government bonds; R_f is assumed to be 3% Annual Percentage Rate (APR);

○ β (the beta) ↬ sensitivity of the expected excess asset returns, $E(R_i)$-R_f, to excess market returns, $E(R_m)$-R_f, **(β=exposure to market)**.

○ $\left(E(R_i) - R_f\right)$ is known as the **risk premium**;

○ $E(R_m)$ is the expected return of the market;

○ $\left(E(R_m) - R_f\right)$ is the **market premium** or **excess return of the market** (difference between the expected market return and the risk free).



We assume that the market is the S&P 500 index and regress for β.

☞ **So CAPM is actually fitting a line to noisy data!** ↬ LS regression

Large β ↬ a less resilient company. Small β ↬ lower exposure to market risk.

Example 4: Capital Asset Pricing Model (CAPM), cntd.

Notice that we employ a block-LS approach, over blocks of 22 days

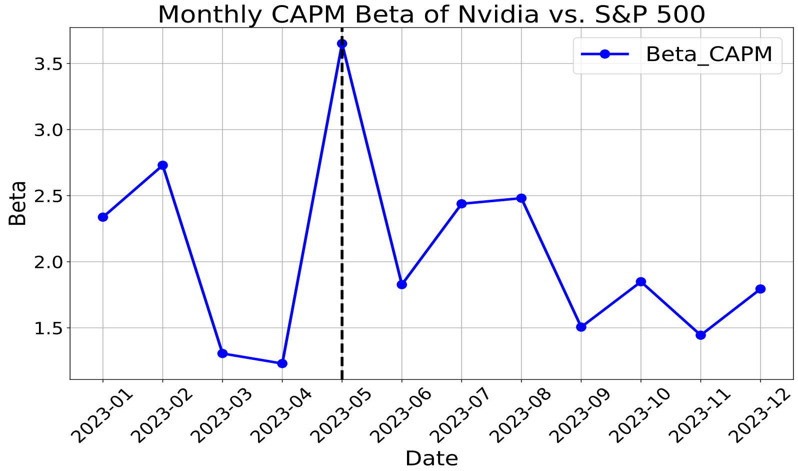
Asset return, R_i, risk-free interest rate, R_f, and market return, R_m, (S&P500 return) are all known. We consider log-returns.

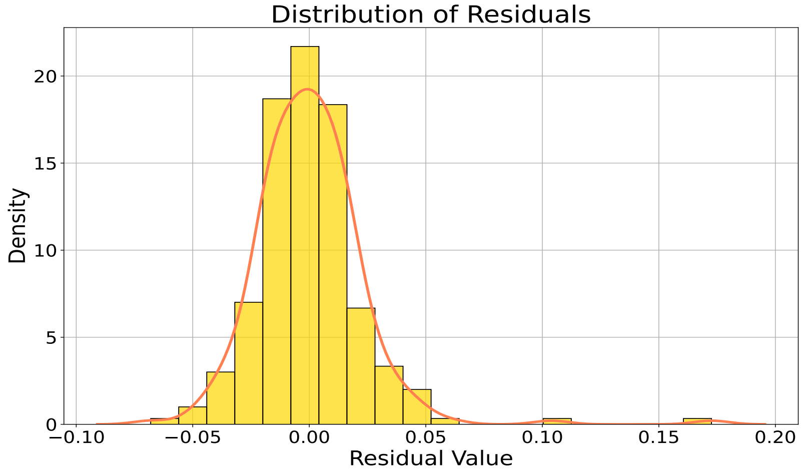
☞ We can now perform LS regression to obtain the value of β.

Each month has 22 trading days. Then, the CAPM states that

$$\begin{bmatrix} R_{i;day1} - R_f \\ R_{i;day2} - R_f \\ \vdots \\ R_{i;day22} - R_f \end{bmatrix} = \beta \begin{bmatrix} R_{m;day1} - R_f \\ R_{m;day2} - R_f \\ \vdots \\ R_{m;day22} - R_f \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_{22} \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{r_i} = \beta\, \boldsymbol{r_m} + \boldsymbol{e}$$

Therefore, the LS estimate: $\hat{\beta} = (\boldsymbol{r_m}^{\mathbf{T}} \boldsymbol{r_m})^{-1} \boldsymbol{r_m}^{\mathbf{T}}\, \boldsymbol{r_i}$





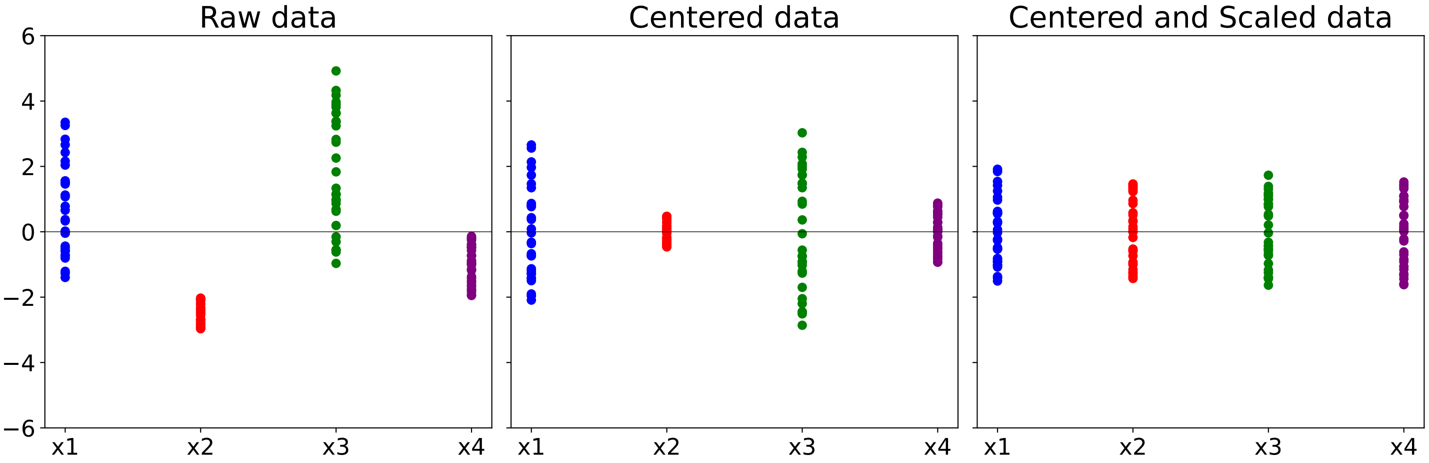
Standardised linear regression

It is often convenient to examine regression with zero intercept, as e.g in Financial Engineering in Example 4. In addition, the analysis may benefit from standardisation to zero mean and unit variance, especially in nonlinear regression and for visualisation purposes.

○ The parameter, $\hat{\beta}$, when estimated from raw data is termed the non-standardised regression coefficient

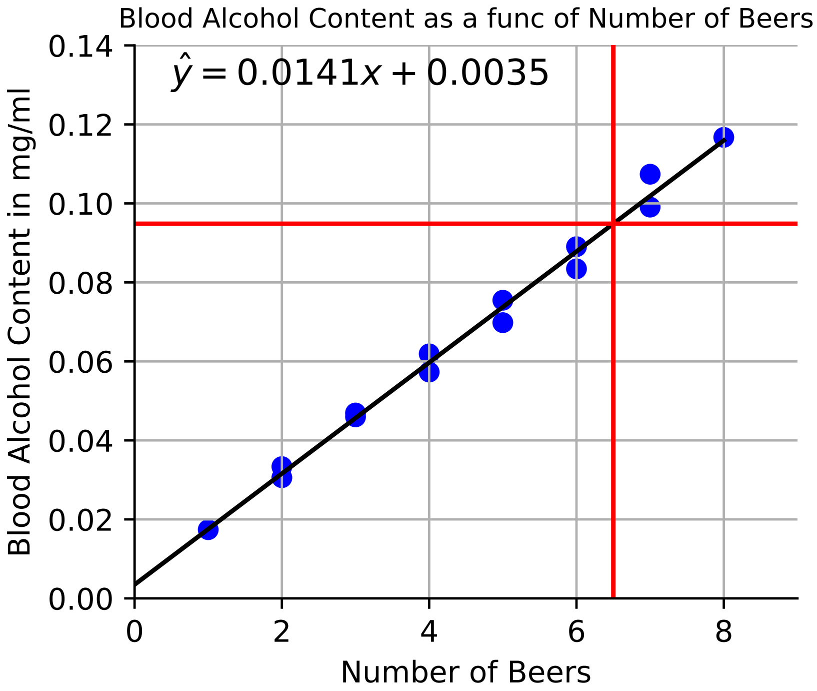
○ The standardised regression coefficient, also termed the beta coefficient or **the beta** is the slope obtained by the regression of y on x when the data are standardised to zero mean and unit variance.

○ When the data are standardised, the intercept, α, assumes the value of 0



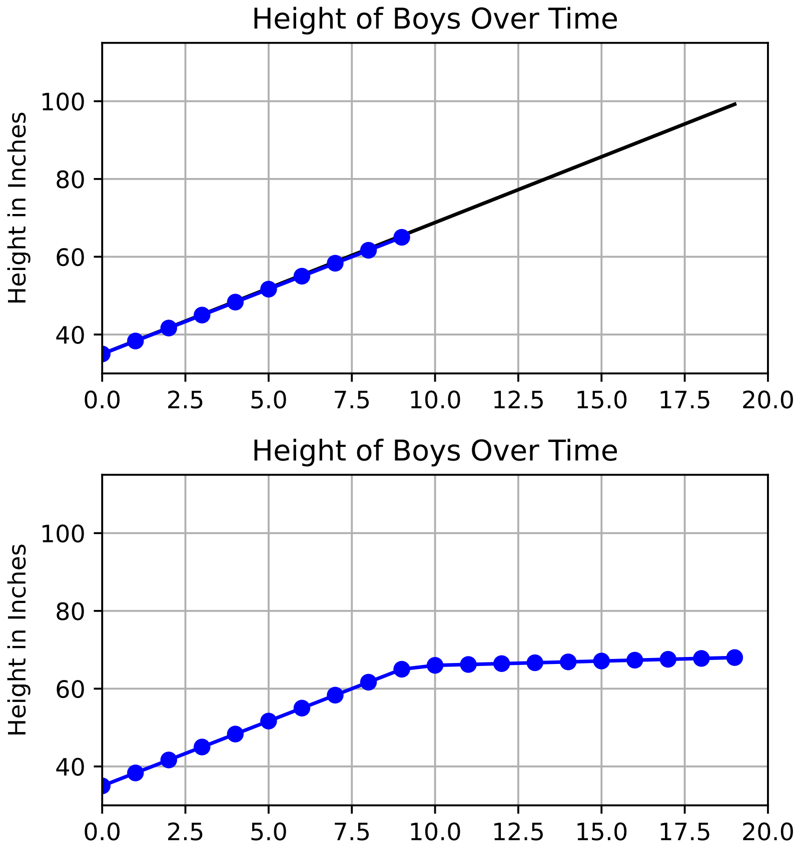
Example 5: Within-sample (interpolation) and out-of-sample (extrapolation) inference using regression

Interpolation: Quite accurate, as a linear fit matches the data range which the model has seen.



Nobody in the study drunk 6.5 pints of beer, but we can still use regression to interpolate and find the estimated blood alcohol level.

Extrapolation: Needs to be considered much more carefully.

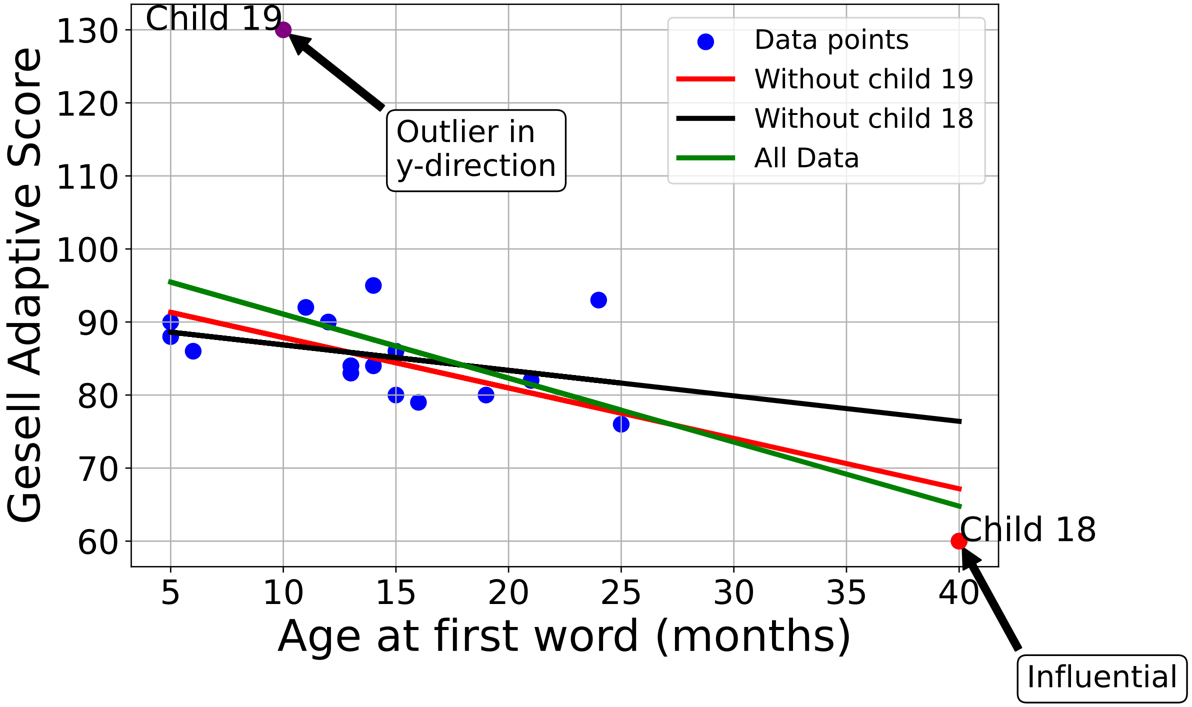


A "piece-wise" linear fit would be more appropriate (or quadratic).

Real world: Outliers and influential points in regression

Outlier: A perfectly good observation that lies outside the overall pattern of observations, that is, it is in the tails of the distribution.

Influential point aka leverage outlier: An observation that markedly changes the regression if removed. This is often an outlier along the x-axis.



Remedy: Robust regression estimators can deal easily with vertical outliers (explained variable, y, outside main concentration of data); see Appendix

Explaining the total variation in the dependent variable

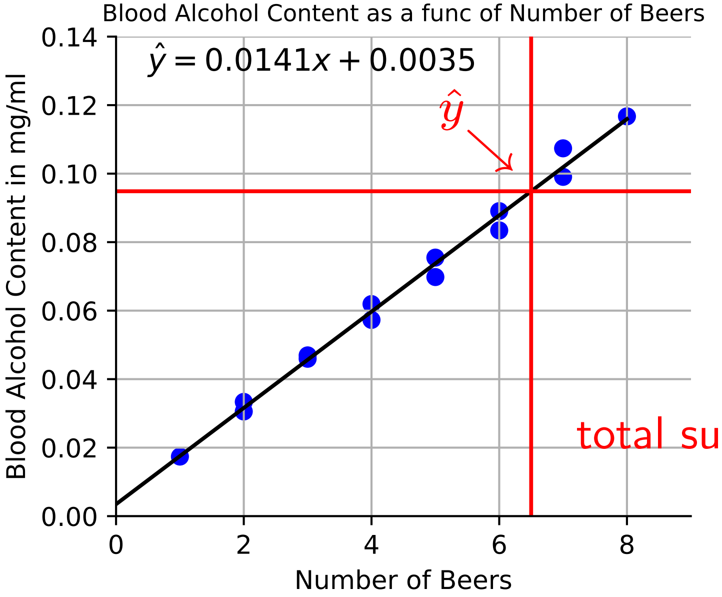
for other measures, such as coefficient of determination, see Appendix

For simplicity, consider the univariate regression model

$$y = \beta_0 + \beta_1 x + e \qquad \rightarrow \qquad \hat{y} = E\{y|x\} = \beta_0 + \beta_1 x \qquad \text{as} \qquad E\{e\} = 0$$

prediction ↗ ↖ mean, $\bar{y}$

$\hat{y}$ lies on the regression line!



This means that:

○ we are predicting the means, $\bar{y} = E\{y|x\}$

The **total variation** or the **total sum of squares (SST)** for the dependent variable, y, is therefore made up of two parts

↙ sum of squares error

$$\textbf{SST} = \textbf{SSE} + \textbf{SSR}$$

total sum of squares ↗ sum of squares regression ↗

The SSE is the **unexplained** part

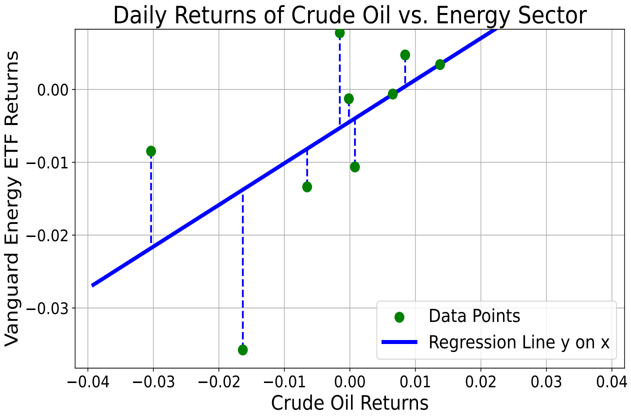
$$\text{SST} = \sum (y - \bar{y})^2 \qquad \text{SSE} = \sum (y - \hat{y})^2 \qquad \text{SSR} = \sum (\hat{y} - \bar{y})^2$$

Regressing y onto x is different from regressing x onto y

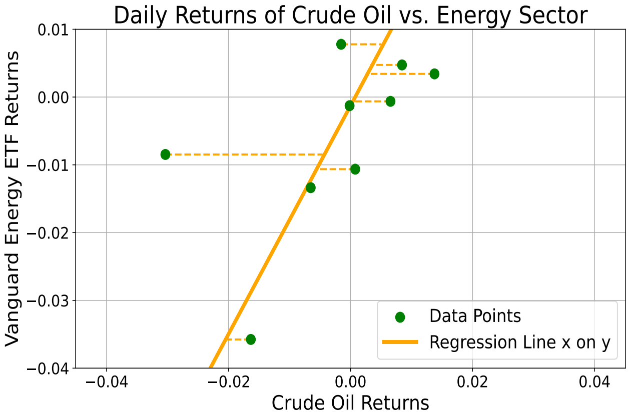
Regression examines the distance of all points in the y direction only

We always regress the explained variable, y, onto an explanatory variable, x

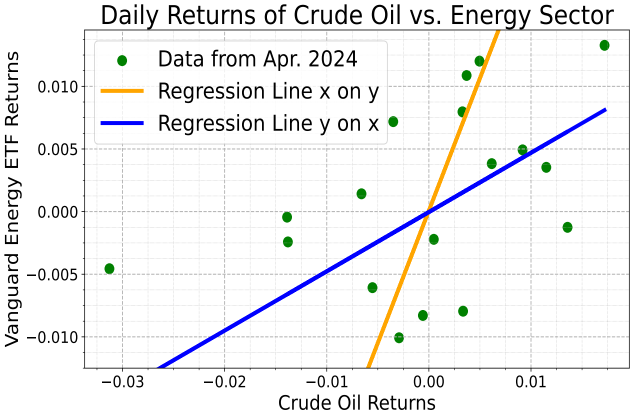
Correct: Regressing y onto x



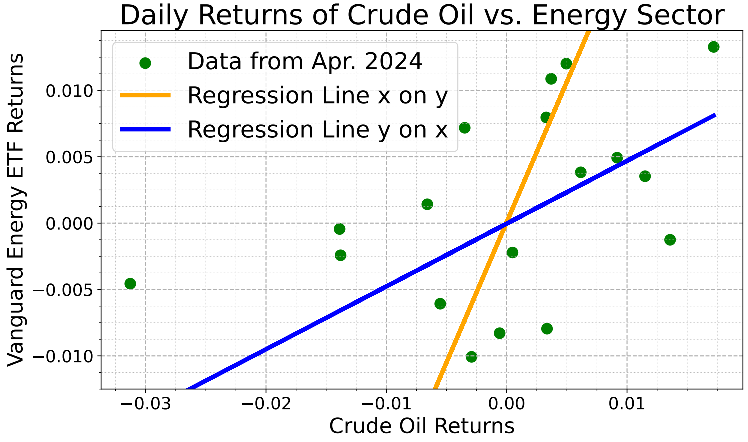
Incorrect: Regressing x onto y

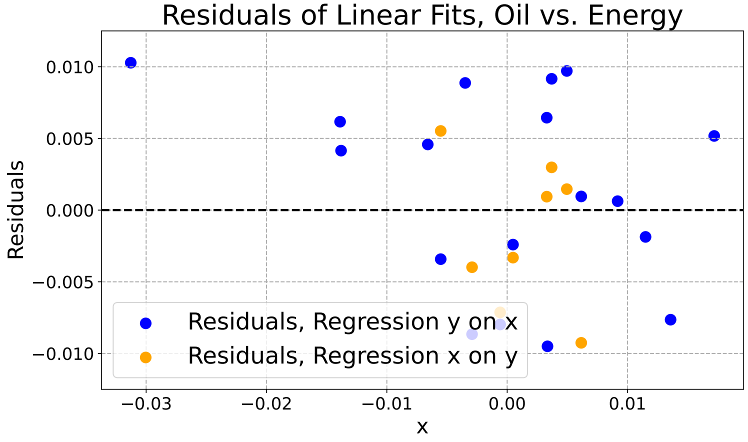


Both together



In the x on y regression (in orange), the residuals represent a horizontal distance between the observed data points and the x on y regression line.

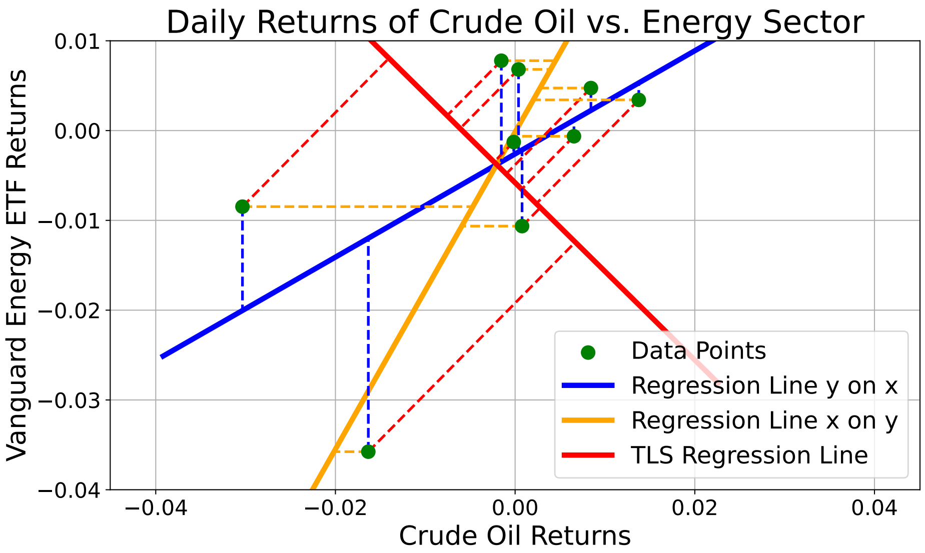


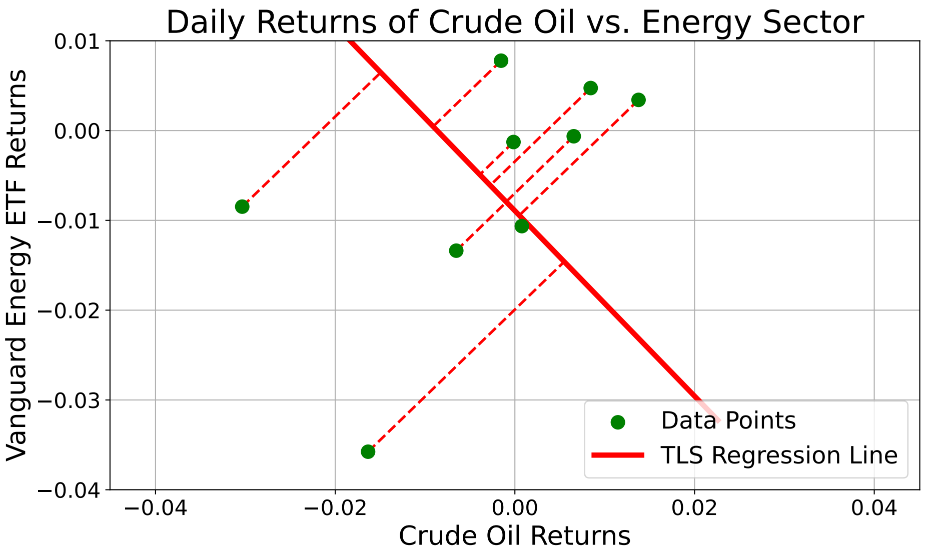


The random residuals indicate that linear regression is appropriate

The Total Least Squares (TLS) method

Instead of the "vertical distance" (regression of y onto x), we can also use the "shortest distance" (orthogonal projection) between the observed data and the regression line $\looparrowright$ the method of Total Least Squares (TLS).





The projection operator (see Lecture 6), that is, modelling based on the orthogonal distance, is more complicated than the modelling based on the vertical and horizontal distance but generally yields more accurate models.

In the 2D case, the TLS regression line is equivalent to the first principal component of the data matrix.

Multiple linear regression analysis

Same idea as univariate regression, just several explanatory variables, $x_1, \ldots, x_p$

The **multiple linear regression model** has the general form

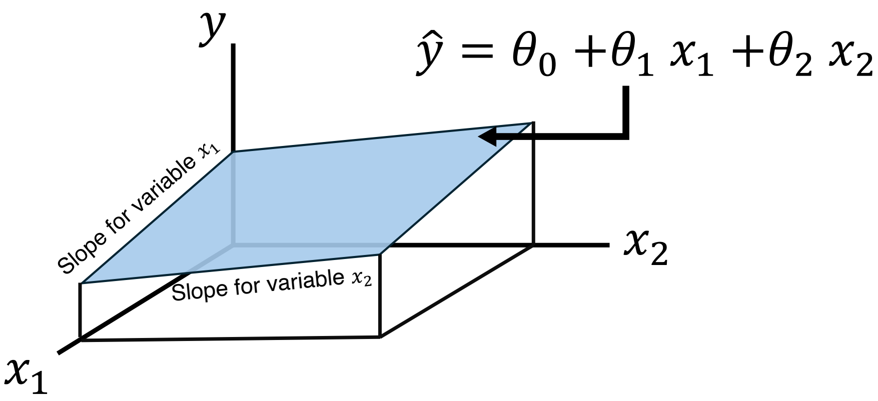
$$y[n] = \beta_0 + \beta_1 x_1[n] + \beta_2 x_2[n] + \cdots + \beta_p x_p[n] + e[n]$$

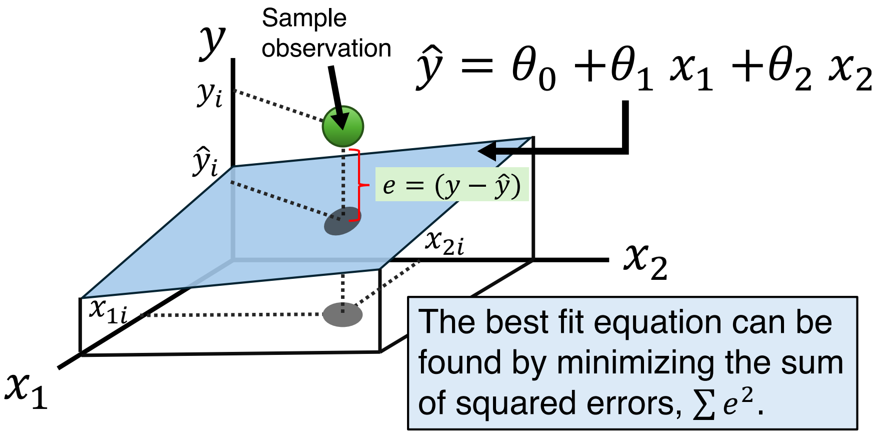
intercept ↗ ↑ partial correlation coefficients ↑

with $\mathbf{x}[n] = [x_1[n], \ldots, x_p[n]]^T$ as the explanatory variables, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_n]^T$ as the corresponding regression coefficients, and β_0 as the intercept.

The estimate $\hat{y}[n]$ based on the **multiple regression model** is given by

$$\hat{y}[n] = \hat{\beta}_0 + \hat{\beta}_1 x_1[n] + \hat{\beta}_2 x_2[n] + \cdots + \hat{\beta}_p x_p[n]$$





The best fit equation can be found by minimizing the sum of squared errors, $\sum e^2$.

☞ **Bivariate linear regression:** $\hat{y}[n] = \hat{\beta}_0 + \hat{\beta}_1 x_1[n] + \hat{\beta}_2 x_2[n]$

The coefficients $\hat{\theta}_i = \hat{\beta}_i$ are found using the method of Least Squares

Example 6: Bivariate regression in economics

Consider an example of the prediction of online sales of frozen pies.

○ **Dependent variable, y,** is the sales of pies (number sold per week)

○ **Independent variables, x_1 & x_2,** are

x_1: Price of a pie in local currency

x_2: Advertising costs in local currency

The bivariate linear regression model of pie sales now becomes

$$sales = \beta_0 + \beta_1 \times price + \beta_2 \times advertising\ cost$$

The parameters β_1 and β_2 indicate the "rate of change" wrt the corresponding independent variables: price and advertising costs.

Slope β_1. Indicates that the average value of sales changes by β_1 for each 1 GBP increase in the price, while all other parameters are held constant.

Slope β_2. Indicates that the average value of sales changes by β_2 for each 1 GBP spent on advertising, while all other parameters are held constant.

For example, for $\beta_1 = 10$, and if the income is to remain fixed after a change in price, then the sales would be expected to decrease by 10 pies per week for each 1 GBP increase in the selling price.

Example 7: Fama-French three-factor model (Problem Sets)

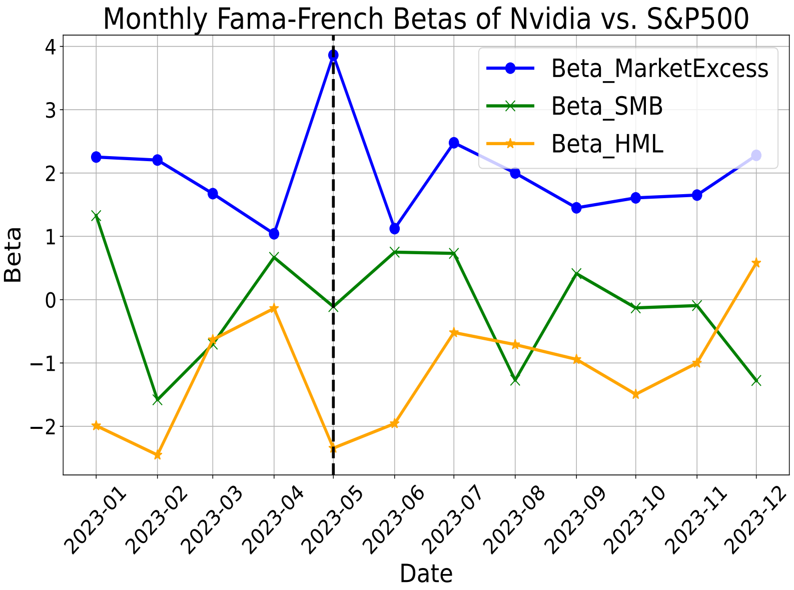
NB: β here is not equal to β in CAPM, due to two additional factors

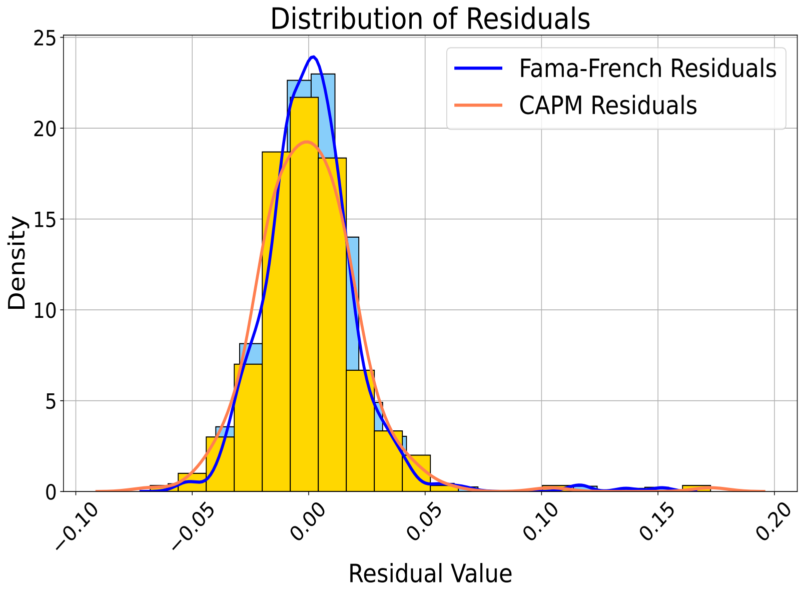
The model is given by (E. Fama won Nobel Prize in Economics in 2013)

$$R_i = R_f + \beta\left(R_m - R_f\right) + bs \cdot SMB + bv \cdot HML + e$$

where SMB measures the historic excess returns of small caps over big caps and HML the value stocks over growth stocks. bs and bv are coeffs.

LS Regression of Fama-French: We regress for the three beta's: The market is the S&P 500 index; R_f is assumed to be 3% APR; Intercept $= 0$.





Goodness of a regression model: Back to the residuals

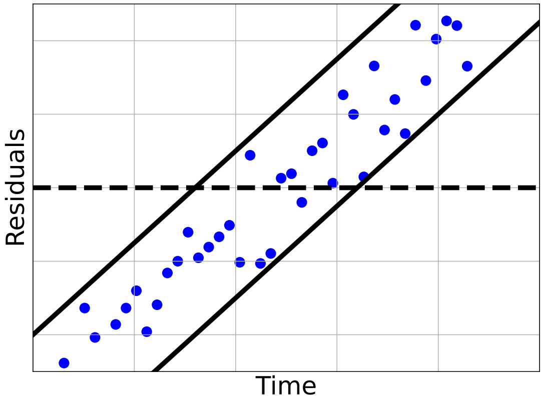
The examination of the residuals, $e[n]{=}y[n]$ - $\hat{y}[n]$, is very useful as e.g. their histogram will reveal whether the assumption of their normality holds.

Scatter plots (scattergrams) of the residuals are also very useful:

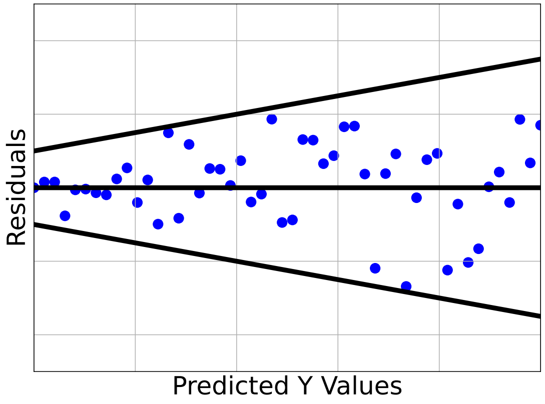
○ A plot of $e[n]$ against time can indicate whether the assumptions of a constant variance of the residuals and their uncorrelatedness hold

○ A plot of $e[n]$ against the predicted dependent variable, $\hat{y}[n]$, examines the assumption of constant variance of the residuals

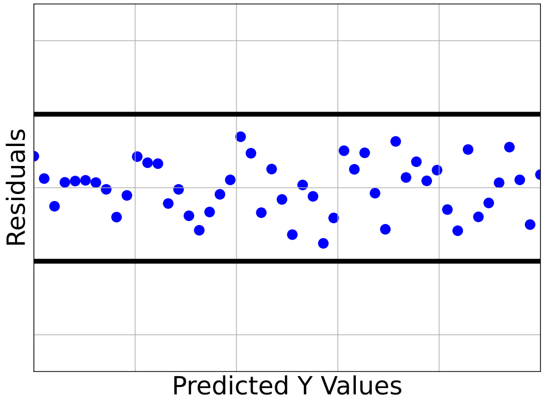
○ A plot of $e[n]$ against an independent variable $x_i[n]$ indicates whether a linear model is appropriate; it should exhibit a random pattern (Slide 10)



Residuals grow with time



Variance not constant



Appropriate model

Regression with categorical variables (see also Appendix)

Consider a task which involves both numerical and non–numerical data:

○ Yes or No ○ On or Off ○ Male or Female ○ Like or Dislike

○ Exercise per week: 1 - every day, 2 - two or more times, 3 - never

In such cases, linear regression should involve **qualitative variables**, also known as **dummy, indicator or categorical explanatory variables**.

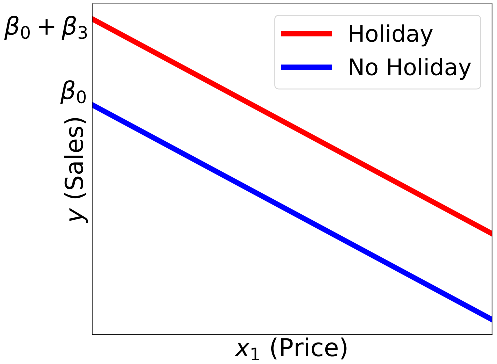
Example 8: Consider again Example 6 (pie sales), with $\hat{y}$ = pies sold

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 \qquad x_3 \in \{0, 1\}$$

where $x_1 = price$, $x_2 = advertising$, and dummy variable $x_3 = holiday$ ($x_3 = 1$ if there was holiday that week, $x_3 = 0$ if no holiday that week). So

Holiday: $\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 \times 1 = (\beta_0 + \beta_3) + \beta_1 x_1 + \beta_2 x_2$

No Holiday: $\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 \times 0 = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

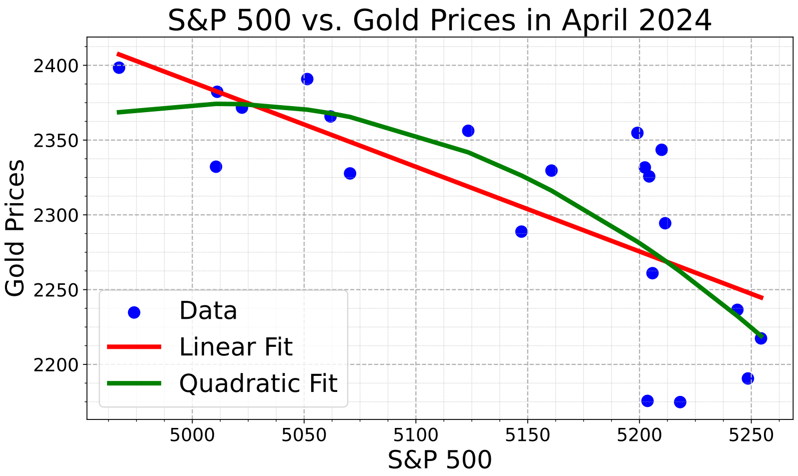


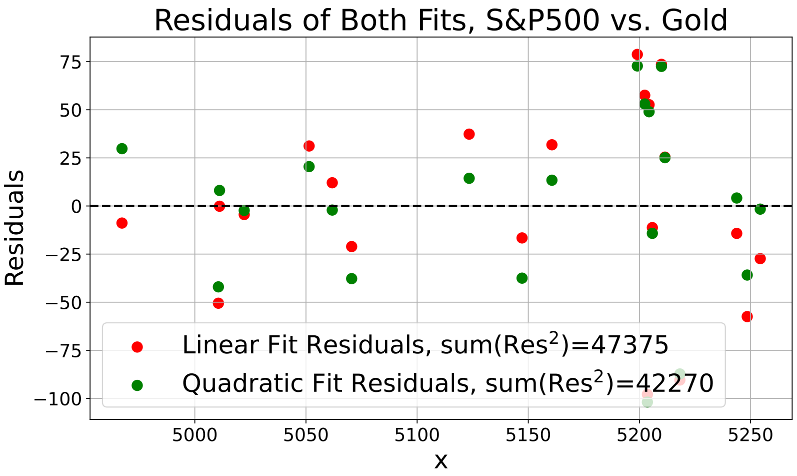
○ For $x_3 = 1$ (weeks with holiday) we have a different intercept $\beta = \beta_0 + \beta_3$

○ The slopes β_1 and β_2 remain the same as for weeks with no holiday ($x_3 = 0$)

Towards nonlinear regression: Polynomial regression

Q: Which of these two models is more appropriate: linear or quadratic?

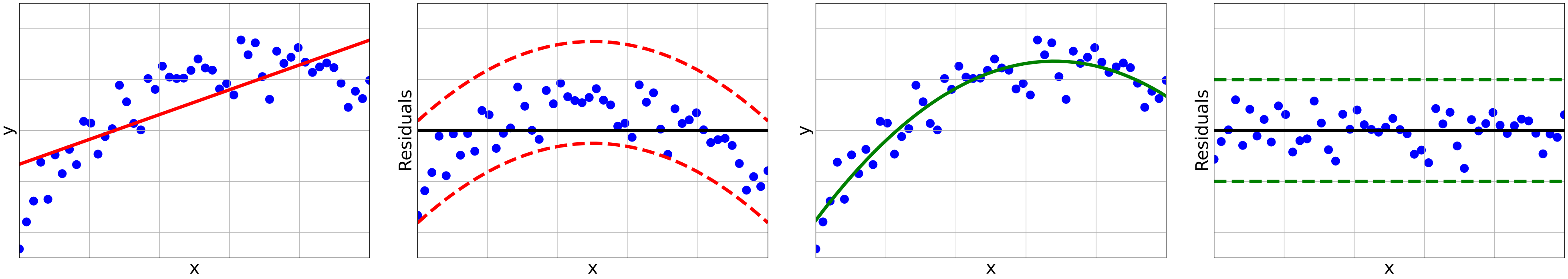




A: The scatter plot of residuals vs x slightly favours a **quadratic model**

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + e$$

☞ Again, the residuals help!



Polynomial regression model

Justification: Weierstrass theorem tells us that any continuous function can be approximated arbitrarily well with a high-enough order polynomial.

So, our polynomial regression model has the form

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_p x^p + e$$

where β_i are the regression coefficients (partial regression coefficients)

P: The powers of x appear to be highly correlated.
S: We should use centred variables x, that is $x \rightarrow (x - \bar{x})$, to remove the non-essential multi-collinearity in the data, to give

$$y = b_0 + b_1(x - \bar{x}) + b_2(x - \bar{x})^2 + \cdots + b_p(x - \bar{x})^p + \varepsilon$$

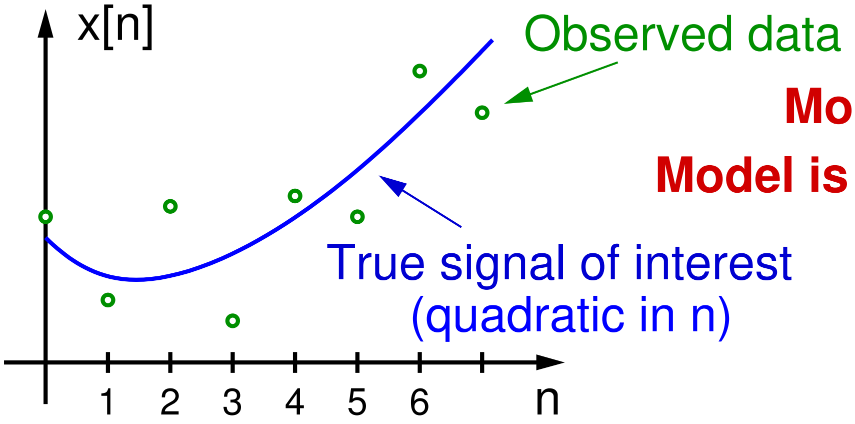
☞ The variables $(x - \bar{x}), (x - \bar{x})^2, \ldots, (x - \bar{x})^p$ are now linearly independent.

P: For a large p, the magnitudes of the powers of $(x - \bar{x})$ can be very large.
A: We can standardise data, as $\frac{x - \bar{x}}{\sigma_x}$.

P: The predictor variables are linearly dependent (**multi-collinearity**).
A: Use the first few principal components of the data matrix. (see Appendix)

The concept of “linear in the parameters” models

☞ A “Linear Model” **does not arise from fitting straight lines to data!**



Model is quadratic in time "n"

Model is "linear in the parameters!"

For N observations: $x[n] = \underbrace{\beta_0 + \beta_1 n + \beta_2 n^2}_{\text{linear in parameters } \beta} + e[n] \quad \Rightarrow \quad \mathbf{x} = \mathbf{H}\boldsymbol{\beta} + \mathbf{e}$

where $\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0^2 \\ 1 & 1 & 1^2 \\ \vdots & \vdots & \vdots \\ 1 & N-1 & (N-1)^2 \end{bmatrix}$

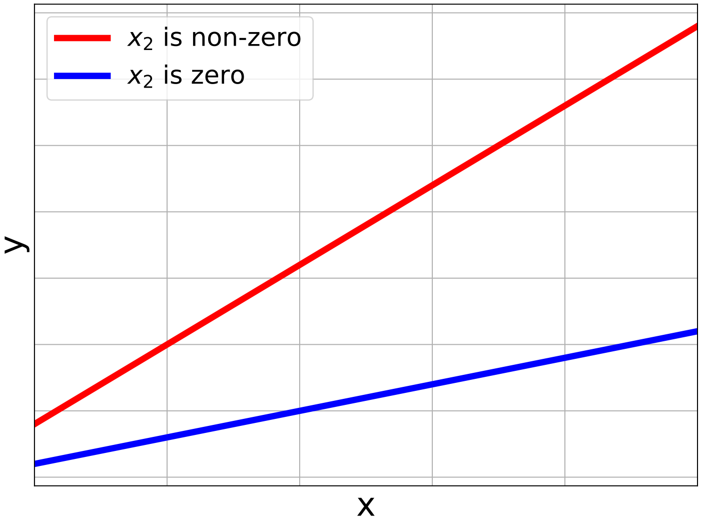
☞ So, both Multiple Regression (with $n \rightarrow x_1, n^2 \rightarrow x_2$) and Polynomial Regression ($n \rightarrow x, n^2 \rightarrow x^2$) are linear in the parameters.

Interaction effects between independent variables Building towards the Volterra approximation model

Interaction regression models contain cross-product terms, for example

$$y = \beta_0 + \underbrace{\beta_1 x_1 + \beta_2 x_1^2 + \beta_3 x_2 + \beta_4 x_2^2}_{\text{basic terms}} + \underbrace{\beta_5 x_1 x_2 + \beta_6 x_1^2 x_2 + \beta_7 x_1 x_2^2}_{\text{interactive terms}} + e$$

☞ Response to one x variable also depends on the values of other x variables



Example 8: Consider the interaction model

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + e$$

○ Without the interaction term, the effect of x_1 on y is measured by β_1.

○ With the interaction term, the effect of x_1 on y is measured by $\beta_1 + \beta_3 x_2$.

○ This effect changes according to the value of x_2.

Solution: Develop the LS regression in steps, by fitting a variety of models to the data, adding and removing variables according to their significance. In this process, the coefficient of partial determination provides a measure of the "marginal contribution" of each independent variable.

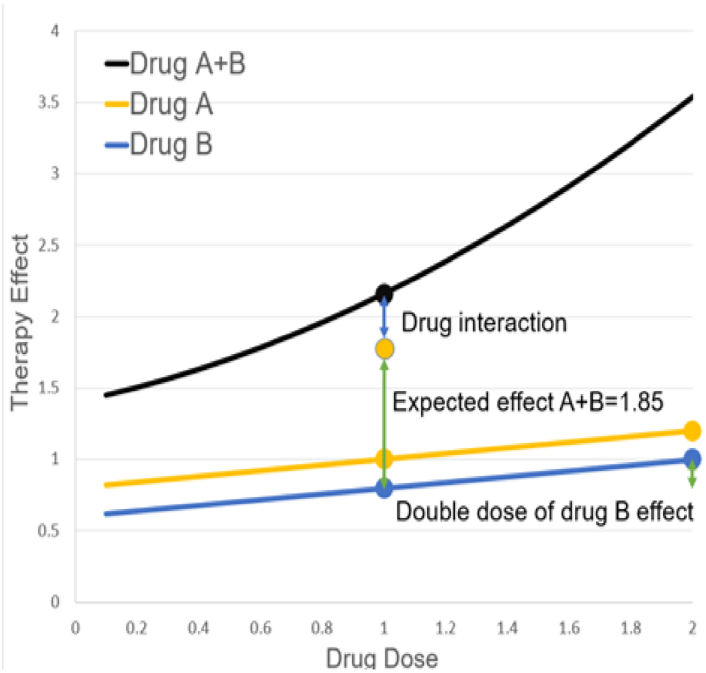
Example 9: Interaction models in medicine

Classes of antihypertensive medications include diuretics, calcium channel blockers, angiotensin converting enzyme inhibitors, and beta blockers.

Consider the interaction model in hypertension therapy based on two drugs

$$y_{A+B} = \underbrace{0.6 + 0.2\,x_1}_{\text{effect of Drug A}} \;+\; \underbrace{0.8 + 0.25\,x_2}_{\text{effect of Drug B}} \;+\; \underbrace{0.31 x_1 x_2}_{\text{interaction A+B}} \;+ e$$

where x_1 is the dose of Drug A and x_2 the dose of Drug B.



On average, a double of dose of a single drug gave only 16% of additional blood pressure reduction (expected 2×, observed 1.16×)

It has been shown that the effect of combining two different classes of drugs is approximately 5 times greater than that of doubling the dose of a single drug (≈ 1.85).

Wald, David S., et al.“Combination therapy versus monotherapy in reducing blood pressure”, American Journal of Medicine 122:3 (2009).

Calculating regression coefficients: Least Squares (LS)

Consider the following regression models

$$\begin{aligned} &\text{Multiple regression:} && y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + e \\ &\text{Polynomial regression:} && y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + e \\ &\text{Interaction regression:} && y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + e \end{aligned}$$

By employing linear independence between: (i) x and x^2 in polynomial regression, and (ii) when $x_1 \perp x_2$ in interaction regression, substitute

Polynomial regression: $x^2 \rightarrow x_2 \quad \& \quad x^3 \rightarrow x_3$

Interaction regression: $x_1 x_2 \rightarrow x_3$

☞ Polynomial and interaction regress. can be treated as multiple regressions

$$y[n] = \beta_0 + \beta_1 x_1[n] + \beta_2 x_2[n] + \cdots + \beta_p x_p[n] + e[n] \quad n = 0, 1, \ldots, N-1$$

$$\underbrace{\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[N-1] \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 1 & x_1[0] & \cdots & x_p[0] \\ 1 & x_1[1] & \cdots & x_p[1] \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1[N-1] & \cdots & x_p[N-1] \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}}_{\boldsymbol{\beta}} + \underbrace{\begin{bmatrix} e[0] \\ e[1] \\ \vdots \\ e[N-1] \end{bmatrix}}_{\boldsymbol{e}} \xrightarrow{\text{LS}} \hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

Recurrent Neural Networks (RNN) and Regression

The Volterra series (universal function approximation) is effectively a Taylor series expansion (TSE) with memory of the network output.

In Time-Delay Neural Networks (TDNN), the explanatory variables are separated by a time delay, for example, $x_1 = x(k-1)$ and $x_2 = x(k-2)$, so that the output of a **multiplicative NN** becomes an interaction model

$$\begin{aligned} y(k) &= c_0 + c_1 x(k-1) + c_2 x(k-2) + c_3 x^2(k-1) + c_4 x^2(k-2) + \\ & \quad c_5 x(k-1) x(k-2) + c_6 x^3(k-1) + c_7 x^3(k-2) + \cdots \end{aligned}$$

With the introduction of feedback, in the form $y(k-1), y(k-2), \ldots$, we arrive at Recurrent Neural Networks (RNN) as function approximators.

For example, the most frequently used **bilinear model** (first order truncated Volterra) model, given by

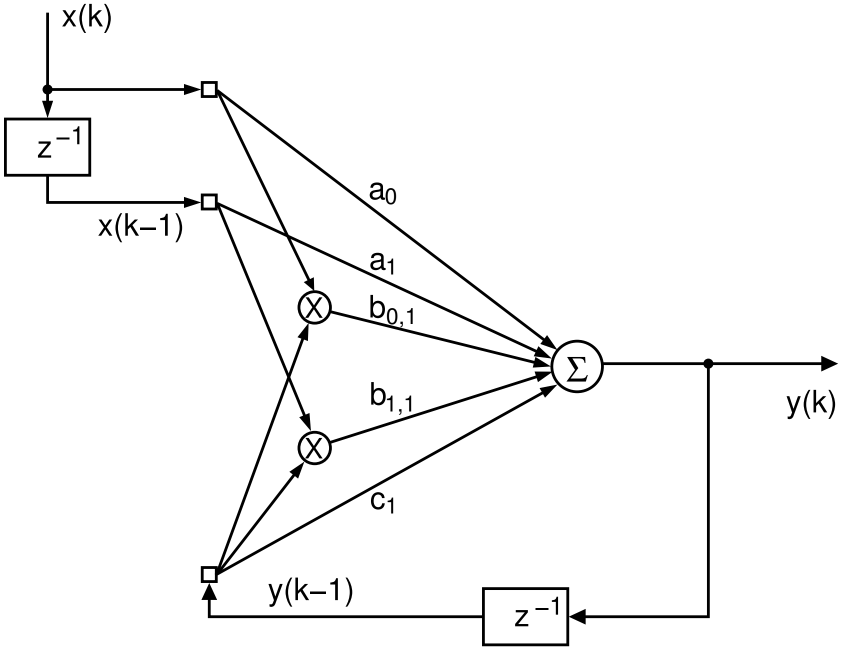
$$y(k) = \sum_{j=1}^{N-1} c_i y(k-j) + \sum_{i=0}^{N-1} \sum_{j=1}^{N-1} b_{i,j} y(k-j) x(k-i) + \sum_{i=0}^{N-1} a_i x(k-i)$$

is equivalent to an RNN with multiplicative synapses.

Example 10: Recurrent perceptron as an interaction regression model

Consider the following RNN (recurrent perceptron) with multiplicative synapses. Its output is given by

$$y(k) = c_1 y(k-1) + b_{0,1} x(k) y(k-1) + b_{1,1} x(k-1) y(k-1) + a_0 x(k) + a_1 x(k-1)$$

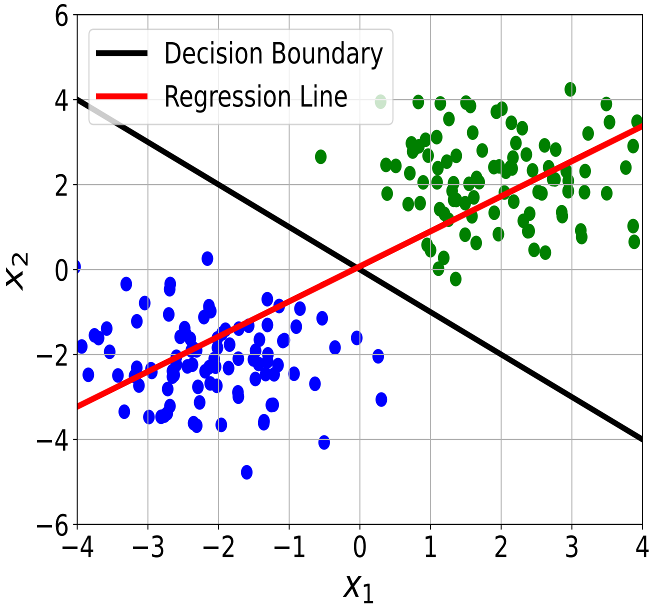


This is precisely the form of the bilinear model on the previous slide.

We can also involve different types of nonlinearity after the "summation" stage, such as the $\tanh$ or the logistic function elaborated next.

Logistic Regression: Motivation

Many situations require the output of a regression model to be discrete or even binary (classes C_1 and C_2) and not continuous-valued as in standard regression. This requires a **discriminative model** $p(C_1|\mathbf{x})$, $p(C_2|\mathbf{x})$.



P: Linear model does not output probabilities, it treats the classes as categories (here: 0 and 1) and fits the best hyperplane (here: a line) that minimises the distances between the observed points and the hyperplane. So, it simply interpolates between the points, and we cannot interpret this as probabilities.

P: The regression line is unbounded, while the probabilities, $p(C_1|\mathbf{x})$ and $p(C_2|\mathbf{x})$, are bounded to between 0 and 1 ↬ we need another method.

Solution: A function of probabilities which is linear in the data, $\mathbf{x}$, such as the **logistic mapping**, which for a binary classification example is given by

$$\ln \frac{p(y=C_1|\mathbf{x})}{p(y=C_2|\mathbf{x})} = \ln \frac{p(y=1)}{1-p(y=1)} = \ln \frac{p(y=Y)}{1-p(y=N)} = \underbrace{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}_{\text{linear in x}}$$

Logistic Regression: Formulation

Odds: The ratio of something occurring to something not occurring. Odds are different from the probability which is the ratio of something occurring to everything that could occur e.g. $\frac{0.8}{0.2}$ has the odds of 4:1. (see Appendix)

The **logit function** is the logarithm of the **odds** $= \frac{p(Y)}{p(N)} = \frac{p(1)}{p(0)}$, that is

$$\ln\frac{p(y=1)}{1-p(y=1)} = \ln\frac{p(y=Y)}{1-p(y=N)} = \underbrace{\beta_0+\beta_1x_1+\cdots+\beta_px_p}_{\text{linear in x}} = \boldsymbol{\beta}^T\mathbf{x}+\beta_0$$

Now: $\quad \log\frac{p(x)}{1-p(x)} = z = \boldsymbol{\beta}^T\mathbf{x}+\beta_0 \quad \rightarrow \quad \frac{p(x)}{1-p(x)} = \sigma(z) = e^z = e^{\boldsymbol{\beta}^T\mathbf{x}+\beta_0}$

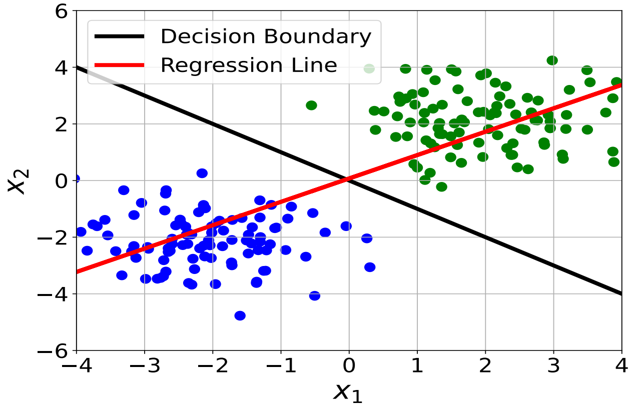
Then: $\quad p(x) = e^{\boldsymbol{\beta}^T\mathbf{x}+\beta_0}\big(1-p(x)\big) \quad \rightarrow \quad p(x) + e^{\boldsymbol{\beta}^T\mathbf{x}+\beta_0}p(x) = e^{\boldsymbol{\beta}^T\mathbf{x}+\beta_0}$

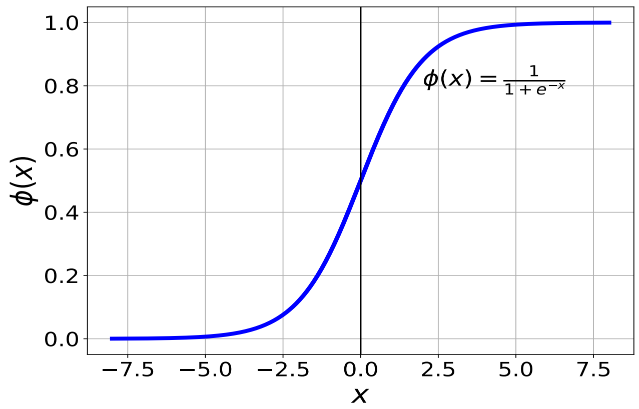
and $\quad p(x) = \frac{e^{\boldsymbol{\beta}^T\mathbf{x}+\beta_0}}{1+e^{\boldsymbol{\beta}^T\mathbf{x}+\beta_0}} \quad \rightarrow \quad p(x) = \frac{1}{1+e^{-\boldsymbol{\beta}^T\mathbf{x}-\beta_0}}$ **logistic function**

☞ For binary outcomes, Logistic Regression gives a linear classifier, with the decision boundary between the outcomes given by $\boldsymbol{\beta}^T\mathbf{x}+\beta_0=0$ (Appendix)

Logistic regression: Application in classification

Consider 2-class classification, based on logistic sigmoid $\Phi(x) = \frac{1}{1+e^{-\beta_1 x+\beta_0}}$





Logistic regression: Parameters are β_1 (slope) and β_0 (horizontal shift). Assume the class labels $C \in \{C_1, C_2\}$. Our aim is to model the conditional probabilities $p(C_1|\mathbf{x})$ and $p(C_2|\mathbf{x})$ as a function of $\mathbf{x}$.

$$\text{probability of class } C_1 : \quad P(C_1|x) = \Phi(\beta_1 x + \beta_0) = p$$

$$\text{probability of class } C_2 : \quad P(C_2|x) = 1 - \Phi(\beta_1 x + \beta_0) = 1 - p$$

☞ Logistic regression gives the probability that data belong to a category. It is an extension of the linear regression model for classification problems. In the simplest case, if the decision boundary $\beta x + \beta_0 \geq 0$ then $p = 1$, and if $\beta x + \beta_0 < 0$ then $p = 0$. The β's are found using Maximum Likelihood.

Logistic Regression: Intuition and scope

Consider a **probabilistic generative model** for classification, given by

$$\swarrow \text{ class probability}$$

$$p(\mathbf{x}, C) = p(\mathbf{x}|C)\, p(C) \qquad\qquad (*)$$

$$\nwarrow \text{ class-conditional probability}$$

where $\mathbf{x} \in \mathbb{R}^D$ is the sample and $C \in \{C_1, C_2\}$ is the class label.

We are interested in finding a **discriminative model for classification**, that is, **we aim to find** $p(C|\mathbf{x})$.

From eq. (*), we have (without loss of generality we can choose $C = C_1$)

$$p(C_1|\mathbf{x}) = \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_1)p(C_1) + p(\mathbf{x}|C_2)p(C_2)} \qquad \text{(Bayes theorem)}$$

$$= \frac{1}{1 + \frac{p(\mathbf{x}|C_2)p(C_2)}{p(\mathbf{x}|C_1)p(C_1)}} = \frac{1}{1 + e^{-ln\left(\frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)}\right)}} \qquad (**)$$

We are interested in cases where the term in the exponent takes a simple form, for example $-ln\left(\frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)}\right)$ is linear.

Logistic Regression: Intuition and scope

To see how eq. (**) simplifies, we shall choose parametric forms for the class- and class-conditional probabilities

$$p(C_1) = \pi \qquad\qquad p(C_2) = 1 - p(C_1) = 1 - \pi$$

$$p(\mathbf{x}|C_1) = \mathcal{N}(\mu_1, \Sigma) \qquad\qquad p(\mathbf{x}|C_2) = \mathcal{N}(\mu_2, \Sigma) \qquad (***)$$

☞ We have chosen the class-conditional probabilities to have different means but same covariances. Now, upon replacing (***) into (**) we arrive at

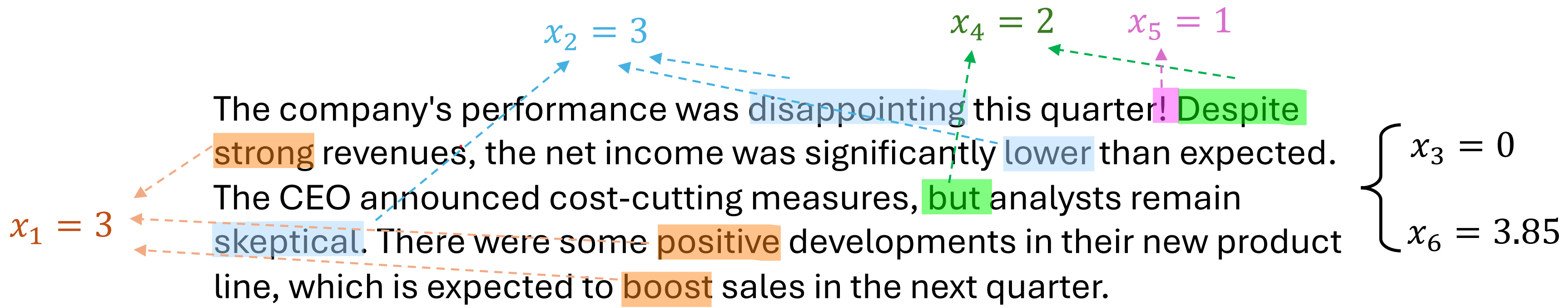
$$p(C_1|\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}+b)}} \qquad (****)$$

☞ By choosing Gaussian distributions with same covariances, the quadratic terms in the exponent of (**) vanish, and the **logistic regression model becomes the true classification model.**

Homework: Find the expressions for $\mathbf{w}$ and b as a function of μ_1, μ_2, Σ. Now, consider different covariances in (***), e.g. Σ_1, Σ_2; how does the expression in (****) change?

Example 11: Classification of financial sentiment with Logistic Regression

Consider binary sentiment classification of a financial news article



The company's performance was disappointing this quarter! Despite strong revenues, the net income was significantly lower than expected. The CEO announced cost-cutting measures, but analysts remain skeptical. There were some positive developments in their new product line, which is expected to boost sales in the next quarter.

☞ We wish to assign the sentiment class $+$ or $-$ to this document. To this end, let us represent each input observation by the 6 features $x_1, \ldots, x_6$.

Var	Definition	Value
x_1	count (positive financial words $\in$ doc)	3
x_2	count (negative financial words $\in$ doc)	3
x_3	$\begin{cases} 1 & \text{if no } \in \text{ doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_4	count (transition words $\in$ doc)	2
x_5	$\begin{cases} 1 & \text{if ! } \in \text{ doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_6	ln(word count of doc)	ln(47) $=$ 3.85

Example 11: Classification of financial sentiment with Logistic Regression, contd.

Each of these features, $x_1, \ldots, x_6$ is associated with a weight, $w_1, \ldots, w_6$.

Assume that, after training, the six weights corresponding to the six features are

$$\mathbf{w} = [2.5, -5.0, -1.2, 0.5, -2.0, 0.7]^T \qquad b = 0.1$$

A comparison of the weights $w_1 = 2.5$ (for x_1, positive words) and $w_2 = -5.0$ (for x_2, negative words) indicates that the "negative lexicon words" (disappointing, lower, sceptical), with negative sentiment, are two times as important as "positive lexicon words" (strong, positive, boost).

Based on these features of the input text x, the probabilities of the positive and negative sentiment classes

$$P(+ \mid x) \text{ and } P(- \mid x)$$

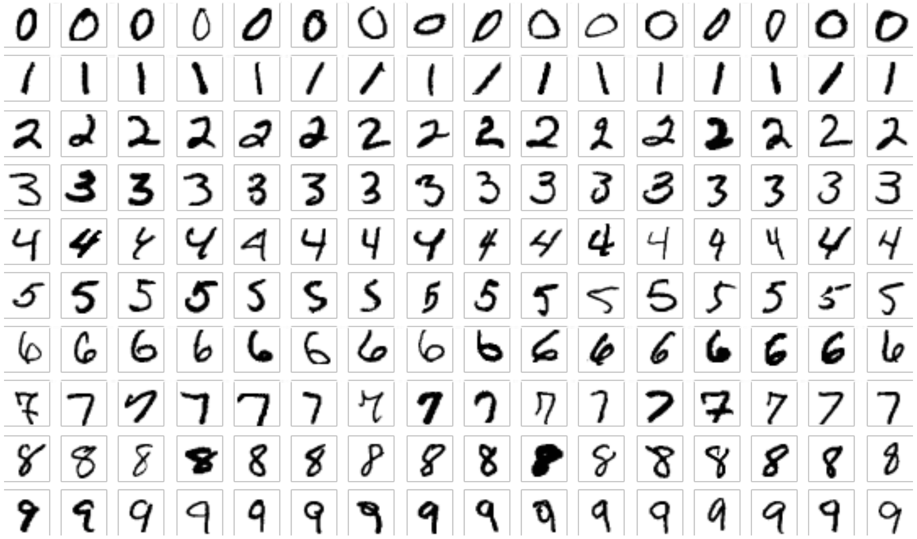
can be computed as

$$p(+ \mid x) = P(y = 1 \mid x) = \sigma(\mathbf{w}^T\mathbf{x} + b) = \sigma(-5.75) = 0.0033$$

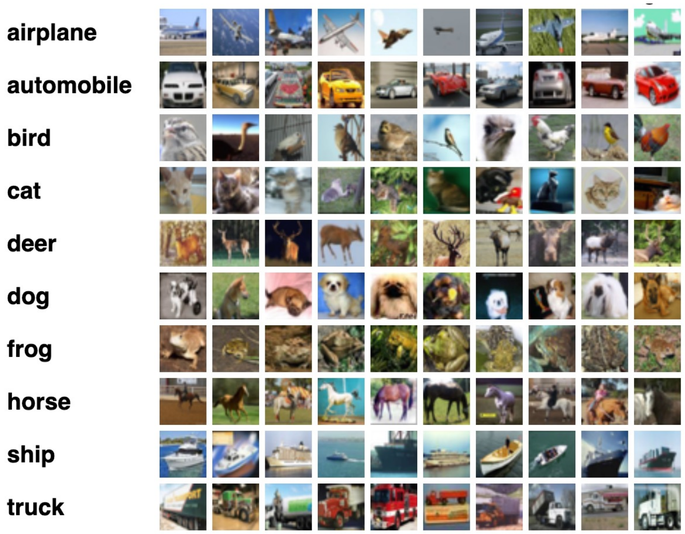
$$p(- \mid x) = P(y = 0 \mid x) = 1 - \sigma(\mathbf{w}^T\mathbf{x} + b) = 0.9967$$

Multi-class logistic regression classifier: Introduction

MNIST (handwritten digits)



CIFAR (image classification)



The output takes more than two values, so each class $c \in \mathcal{C}$ will have its own offset $\beta_0^{(c)}$ and parameters $\boldsymbol{\beta}^{(c)}$. Now, the conditional probabilities

$$\Pr(Y = c \mid \vec{X} = x) = \frac{e^{\beta_0^{(c)} + \mathbf{x}^T \boldsymbol{\beta}^{(c)}}}{\sum_c e^{\beta_0^{(c)} + \mathbf{x}^T \boldsymbol{\beta}^{(c)}}}$$

For only two classes (say, 0 and 1), we have $\frac{1}{1+e^{-(\beta_0 + x \cdot \beta)}}$, that is, we arrive at standard Logistic Regression.

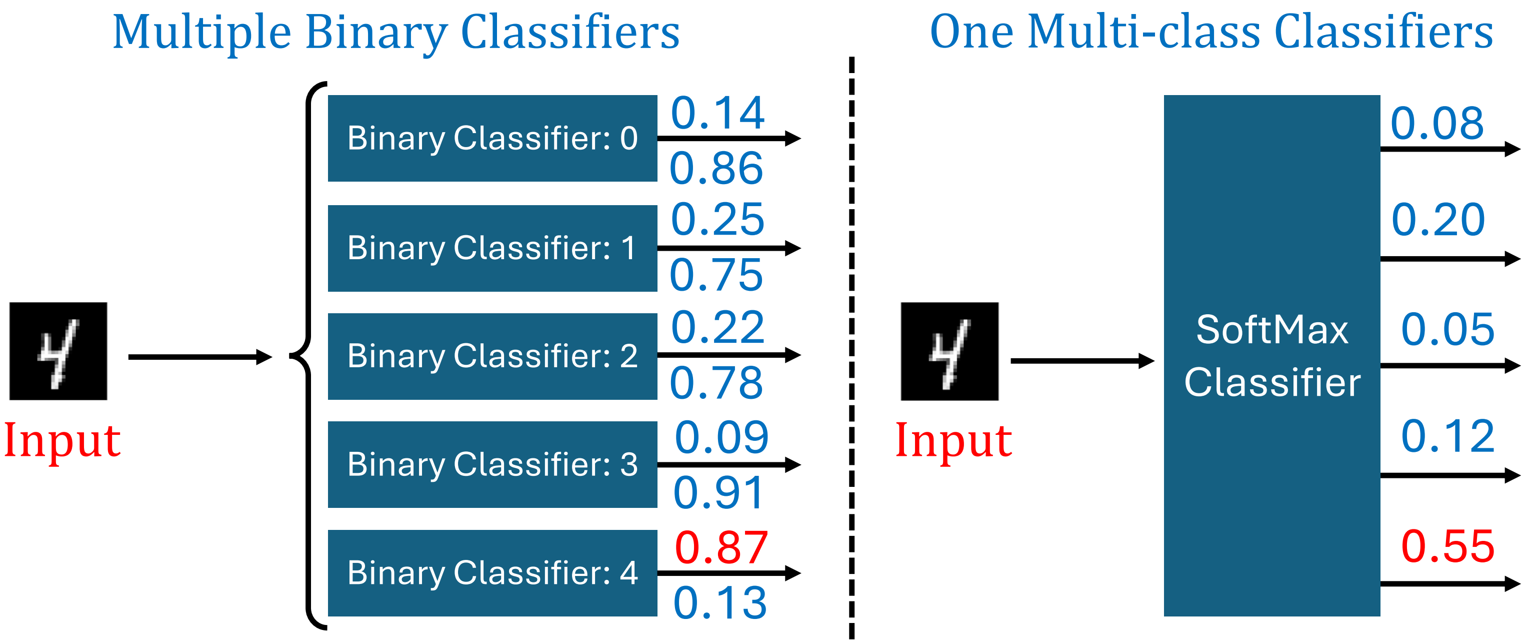
Multi-class logistic regression classifier: Principle

To categorise into multiple classes, we may adopt two strategies:

○ Use multiple binary classifiers for each class of the input data, and examine their outputs to make a class prediction

○ Use one multi-class classifier as a generalisation of logistic regression

$$\Pr(Y = c \mid \vec{X} = x) = \frac{e^{\beta_0^{(c)} + \mathbf{x}^T \boldsymbol{\beta}^{(c)}}}{\sum_c e^{\beta_0^{(c)} + \mathbf{x}^T \boldsymbol{\beta}^{(c)}}} \quad c \in \mathcal{C}$$



Multi-class logistic classifier: The SoftMax function

The SoftMax function represents a multi-class logistic classifier which uses one-hot output encoding to compute $p(y_k = 1|\mathbf{x})$ (with $\sum_{k=1}^{K} p(y_k|\mathbf{x}) = 1$)

$$p(y_k|\mathbf{x}) = \frac{\exp(\boldsymbol{z}_k)}{\sum_{j=1}^{K} \exp(\boldsymbol{z}_j)} = \frac{\exp(\mathbf{w}_k^T \mathbf{x} + b_k)}{\sum_{j=1}^{K} \exp(\mathbf{w}_j^T \mathbf{x} + b_j)}$$

y_k is one of K classes

where the model parameters for a class k are $\mathbf{w}_k$ and $b_k, k = 1, 2, \ldots, K$

For example, for reduced CIFAR data {cat, dog, bird}, we can calculate

$$P(y = \text{cat} \mid \mathbf{x}) = \frac{\exp\left(\boldsymbol{\beta}_{\text{cat}}^T \mathbf{x}\right)}{\exp\left(\boldsymbol{\beta}_{\text{cat}}^T \mathbf{x}\right) + \exp\left(\boldsymbol{\beta}_{\text{dog}}^T \mathbf{x}\right) + \exp\left(\boldsymbol{\beta}_{\text{bird}}^T \mathbf{x}\right)}$$

Relationship to Logistic Regression: For a special case $K = 2$ (Slide 34)

$$p(y_k|\mathbf{x}) = \frac{1}{\exp\left(\mathbf{w}_1^\top \mathbf{x} + b_1\right) + \exp\left(\mathbf{w}_2^\top \mathbf{x} + b_2\right)} \begin{bmatrix} \exp\left(\mathbf{w}_1^\top \mathbf{x} + b_1\right) \\ \exp\left(\mathbf{w}_2^\top \mathbf{x} + b_2\right) \end{bmatrix}$$

and the SoftMax regression reduces into the 2-class logistic regression. The SoftMax is a backbone of modern Neural Networks and Deep Learning.

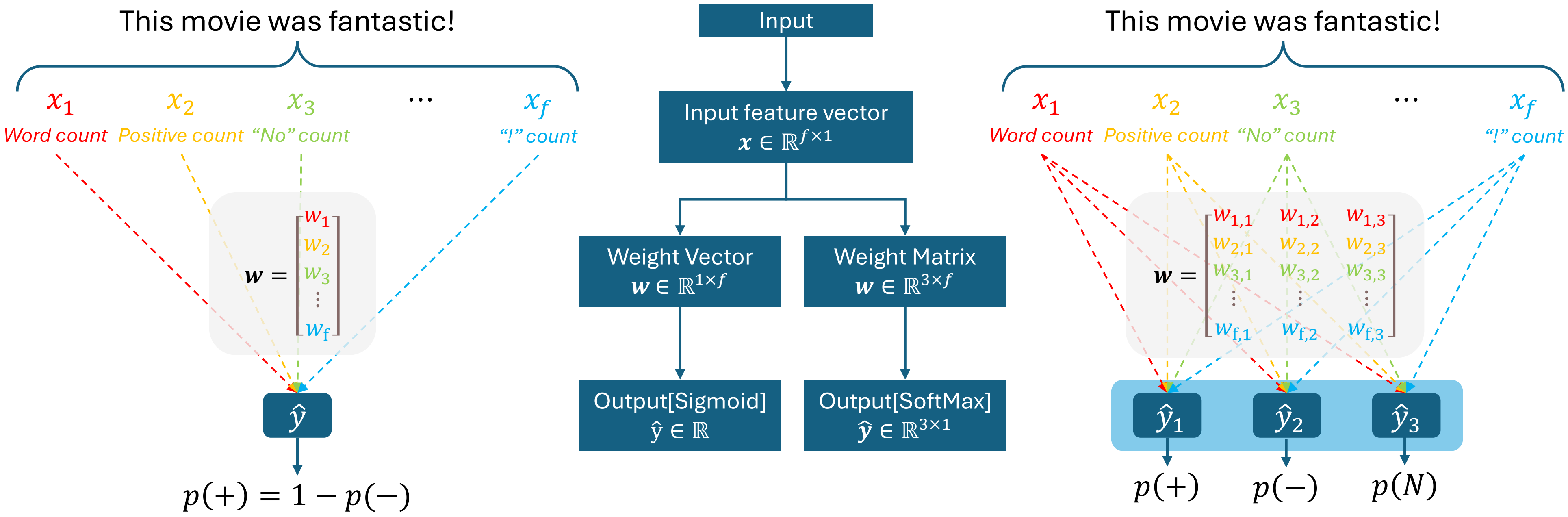
Employing SoftMax in Logistic Regression

Binary logistic regression employs a weight vector $\mathbf{w}$ and a scalar output $\hat{y}$. Multinomial logistic regression $\looparrowright$ weight matrix $\mathbf{W}$ and vector output $\hat{\mathbf{y}}$. Each row k of $\mathbf{W}_{K\times F}$ is the weight vector $\mathbf{w}_k$, with K as the number of classes and F as number of input features

$$\hat{\mathbf{y}} = \mathrm{SoftMax}(\mathbf{W}\mathbf{x}+\mathbf{b}) = \left[\frac{\exp(\boldsymbol{z}_1)}{\sum_{j=1}^{K}\exp(\boldsymbol{z}_j)}, \frac{\exp(\boldsymbol{z}_2)}{\sum_{j=1}^{K}\exp(\boldsymbol{z}_j)}, \ldots, \frac{\exp(\boldsymbol{z}_K)}{\sum_{j=1}^{K}\exp(\boldsymbol{z}_j)}\right]$$

Binary Logistic Regression

SoftMax Logistic Regression



This movie was fantastic!

x_1 Word count

x_2 Positive count

x_3 "No" count

…

x_f "!" count

$\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_f \end{bmatrix}$

$\hat{y}$

$p(+) = 1 - p(-)$

Input

Input feature vector $\boldsymbol{x} \in \mathbb{R}^{f\times 1}$

Weight Vector $\boldsymbol{w} \in \mathbb{R}^{1\times f}$

Weight Matrix $\boldsymbol{w} \in \mathbb{R}^{3\times f}$

Output[Sigmoid] $\hat{y} \in \mathbb{R}$

Output[SoftMax] $\hat{\boldsymbol{y}} \in \mathbb{R}^{3\times 1}$

This movie was fantastic!

x_1 Word count

x_2 Positive count

x_3 "No" count

…

x_f "!" count

$\boldsymbol{w} = \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \\ w_{3,1} & w_{3,2} & w_{3,3} \\ \vdots & \vdots & \vdots \\ w_{f,1} & w_{f,2} & w_{f,3} \end{bmatrix}$

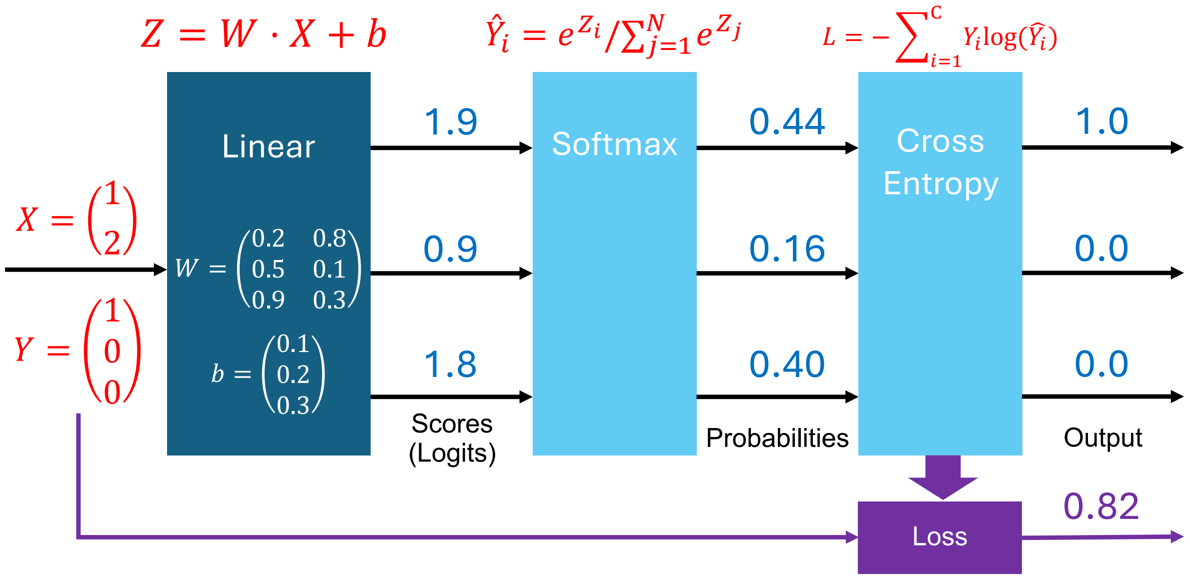
$\hat{y}_1$ $\hat{y}_2$ $\hat{y}_3$

$p(+)$ $p(-)$ $p(N)$

Multi-class logistic classifier: The SoftMax, contd.

Parameters of SoftMax are calculated using Negative Log-Likelihood (NLL) loss

The parameters of SoftMax are typically found by Maximum Likelihood



$Z = W \cdot X + b$
$\hat{Y}_i = e^{Z_i} / \sum_{j=1}^{N} e^{Z_j}$
$L = -\sum_{i=1}^{C} Y_i \log(\hat{Y}_i)$
$X = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$
$Y = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$
Linear
$W = \begin{pmatrix} 0.2 & 0.8 \\ 0.5 & 0.1 \\ 0.9 & 0.3 \end{pmatrix}$
$b = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix}$
1.9
0.9
1.8
Scores (Logits)
Softmax
0.44
0.16
0.40
Probabilities
Cross Entropy
1.0
0.0
0.0
Output
Loss
0.82

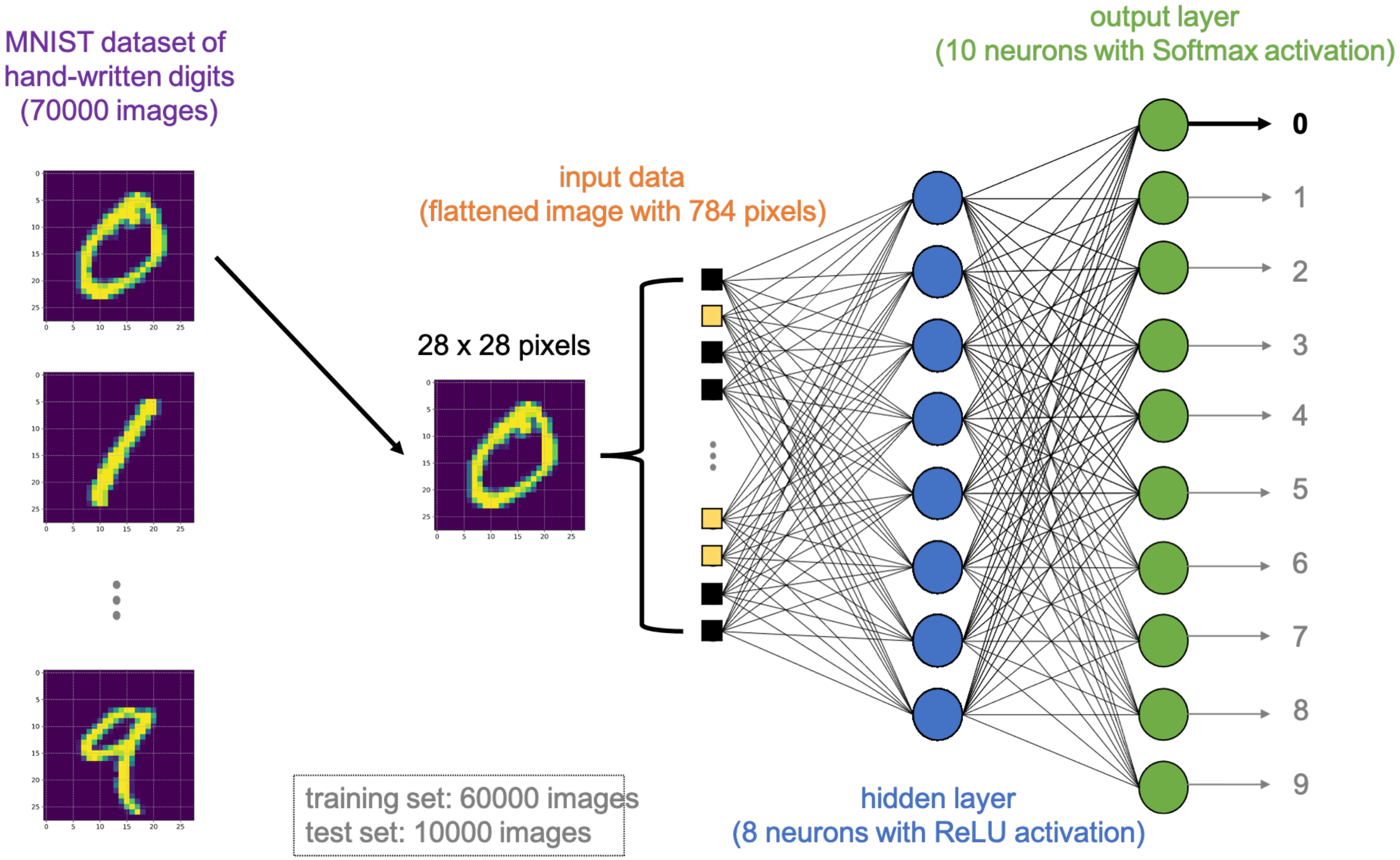
Denote by $\mathbf{x}_m^k$ the m-th sample from class k. Then, the likelihood function

$$l\big((\mathbf{w}_k, b_k)|\mathbf{X}\big) = \sum_k \sum_m ln\, p(y_k|\mathbf{x}_k^m) = \sum_k \sum_m ln \frac{\exp(\mathbf{w}_k^T \mathbf{x}_k^m + b_k)}{\sum_j \exp(\mathbf{w}_j^T \mathbf{x}_k^m + b_j)}$$

and $$\frac{\partial l(\cdot)}{\partial \mathbf{w}_j} = \sum_m \Big[1 - p(y_j|\mathbf{x}_j^m)\Big]\mathbf{x}_j^m - \sum_{k \neq j} \sum_m p(y_k|\mathbf{x}_k^m)\mathbf{x}_k^m$$ (see Appendix)

SoftMax in action: Deep Neural Networks

classification of hand–written numbers ↬ a 10–class discriminative problem



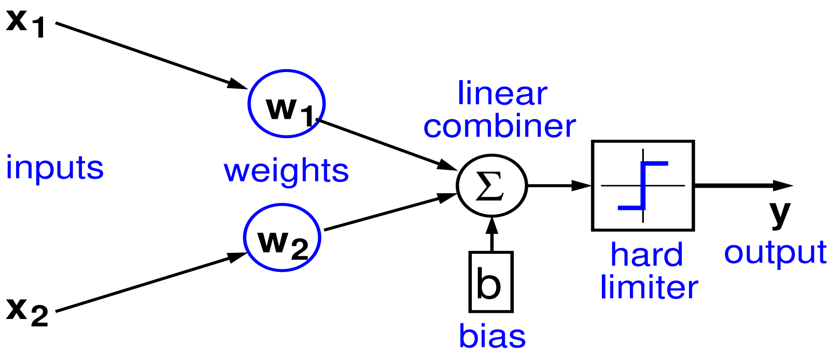
☞ **Our images are 28×28 pixels, so we have $28 \times 28 = 784$ inputs to the network!**

☞ MLP has 2 layers of neurons (hidden & output), and in total **6,370 trainable parameters**.

Logistic regression and neural networks: The Perceptron

With a slight abuse in notation ($\Phi \rightarrow 2\Phi - 1$), and for $\beta \rightarrow \infty$, we have

$$y = \Phi(\mathbf{x}) = \frac{1}{1 + e^{-\beta(\mathbf{w}^T\mathbf{x}+b)}} \quad \xrightarrow{\beta\rightarrow\infty} \quad y = \begin{cases} +1, & \mathbf{w}^T\mathbf{x} + b \geq 0 \\ -1, & \mathbf{w}^T\mathbf{x} + b < 0 \end{cases} = \text{sign}(\mathbf{w}^T\mathbf{x} + b)$$



So, for $\beta \rightarrow \infty$, logistic regression becomes standard linear regression followed by a hard limiter, that is, the sign of our standard regression

For two classes, as in the figure, the decision boundary is given by:

$$w_1 x_1 + w_2 x_2 - b = 0$$

Perceptron learning: With d_n as a teaching signal, the weight update is

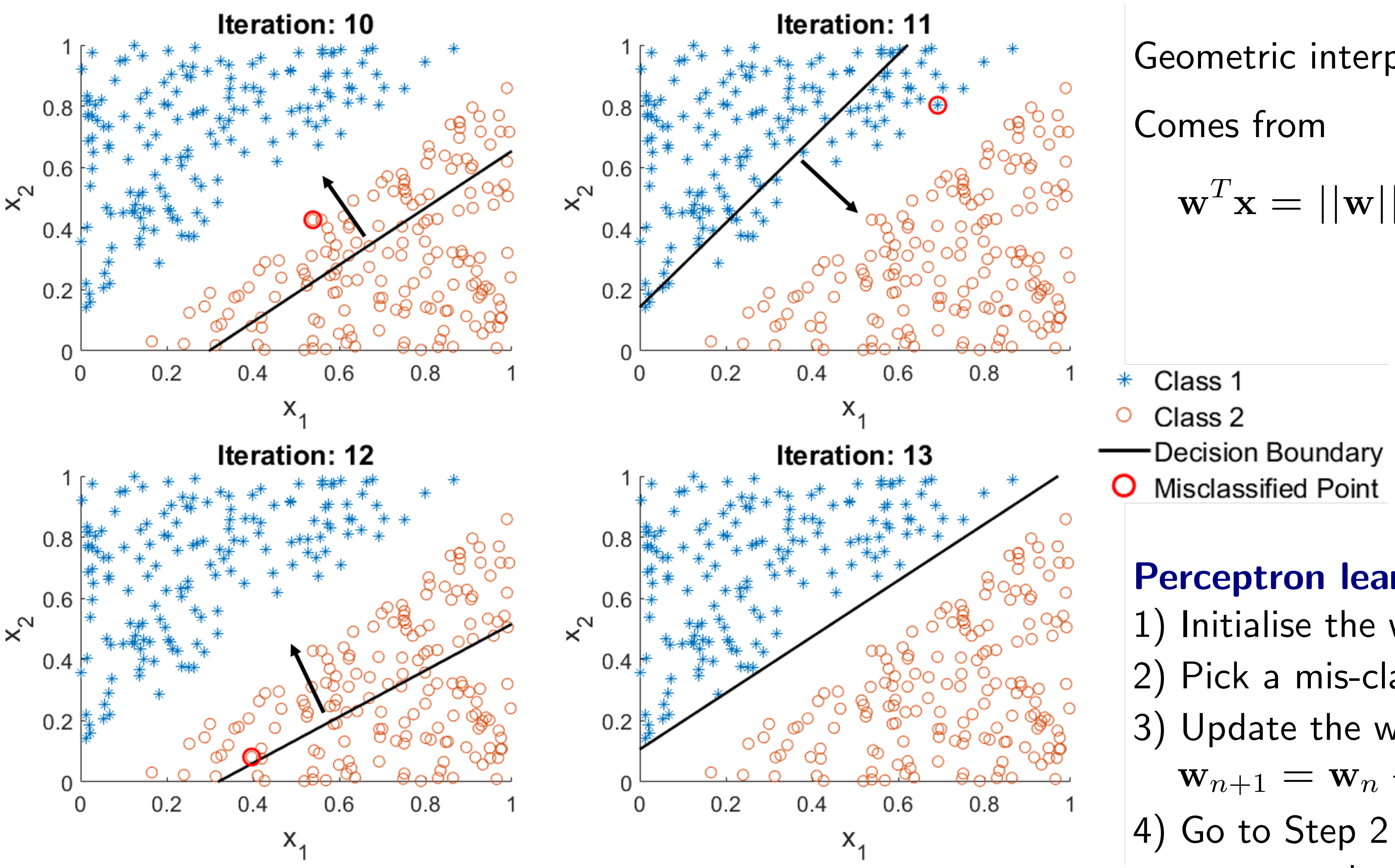
$$\mathbf{w}_{n+1} = \begin{cases} \mathbf{w}_n & if \;\; y = d_n \\ \mathbf{w}_n + \eta\,\mathbf{x}_n & if \;\; y < d_n \\ \mathbf{w}_n - \eta\,\mathbf{x}_n & if \;\; y > d_n \end{cases}$$

Now, because $y_n = \text{sign}(\mathbf{w}_n^T\mathbf{x}_n + b) = \pm 1$ we have $\mathbf{w}_{n+1} = \mathbf{w}_n + \eta\, y_n \mathbf{x}_n$

where η is the stepsize (learning rate), a small positive number.

Perceptron in Action: Binary classification

perceptron.m



Geometric interpretation

Comes from

$$\mathbf{w}^T\mathbf{x} = ||\mathbf{w}||\ ||\mathbf{x}||\ \cos\theta$$

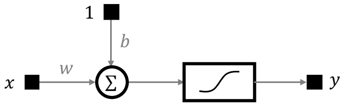
Perceptron learning:

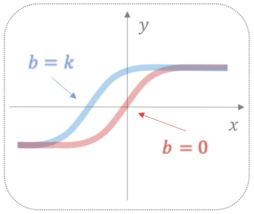
1) Initialise the weights
2) Pick a mis-classified point
3) Update the weights as
   $\mathbf{w}_{n+1} = \mathbf{w}_n + \eta\, y_n \mathbf{x}_n$
4) Go to Step 2 until all points are correctly classified

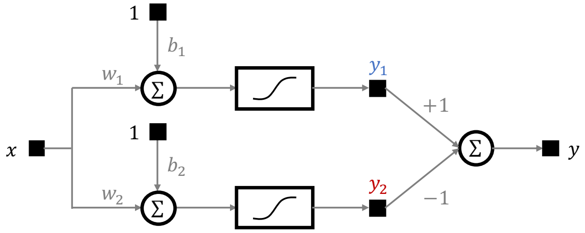
Guaranteed to converge if data is linearly separable

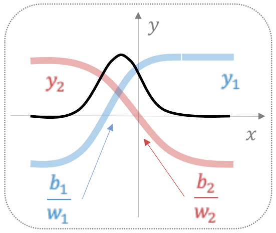
Logistic function for temporal data: Universal function approximation property of Neural Networks (NN)

Consider the output of a single "logistic neuron," the bias b provides a temporal shift

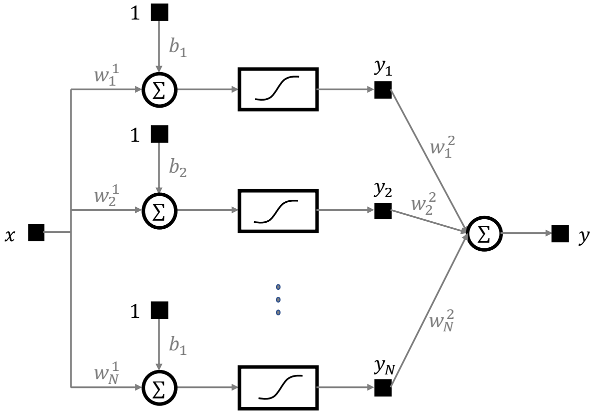


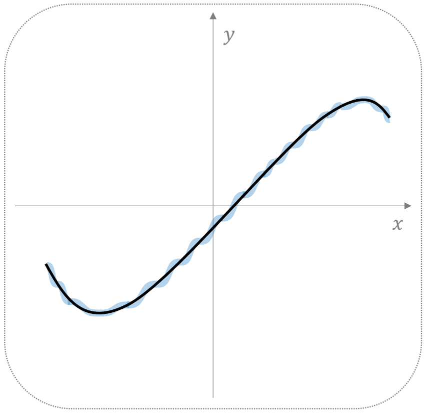






Output of a combination of two "logistic neurons" (blue and red) ↬ **Gaussian–like**





☞ **A layer of such neurons ↬ smooth Universal Function Approximation**

Summary: Logistic Regression

Assumptions:

- Independent observations
- Linear relationship between independent variables and log odds
- Little or no collinearity among independent variables
- Residuals do not have to be normally distributed
- Sensitive to outliers
- Homoscedasticity is not required

Pro's and Con's

- It is not only a classification model, but also gives probabilities
- The interpretation is more difficult because the the weights are multiplicative and not additive

Advantages over naive Bayes:

○ Does not require strong conditional independence assumptions

○ Much more robust to correlated features, it will assign part of the weight to one feature and the other part to another feature

○ Naive Bayes works well on small datasets and logistic regression works better on large datasets

Summary: Regression vs Logistic Regression

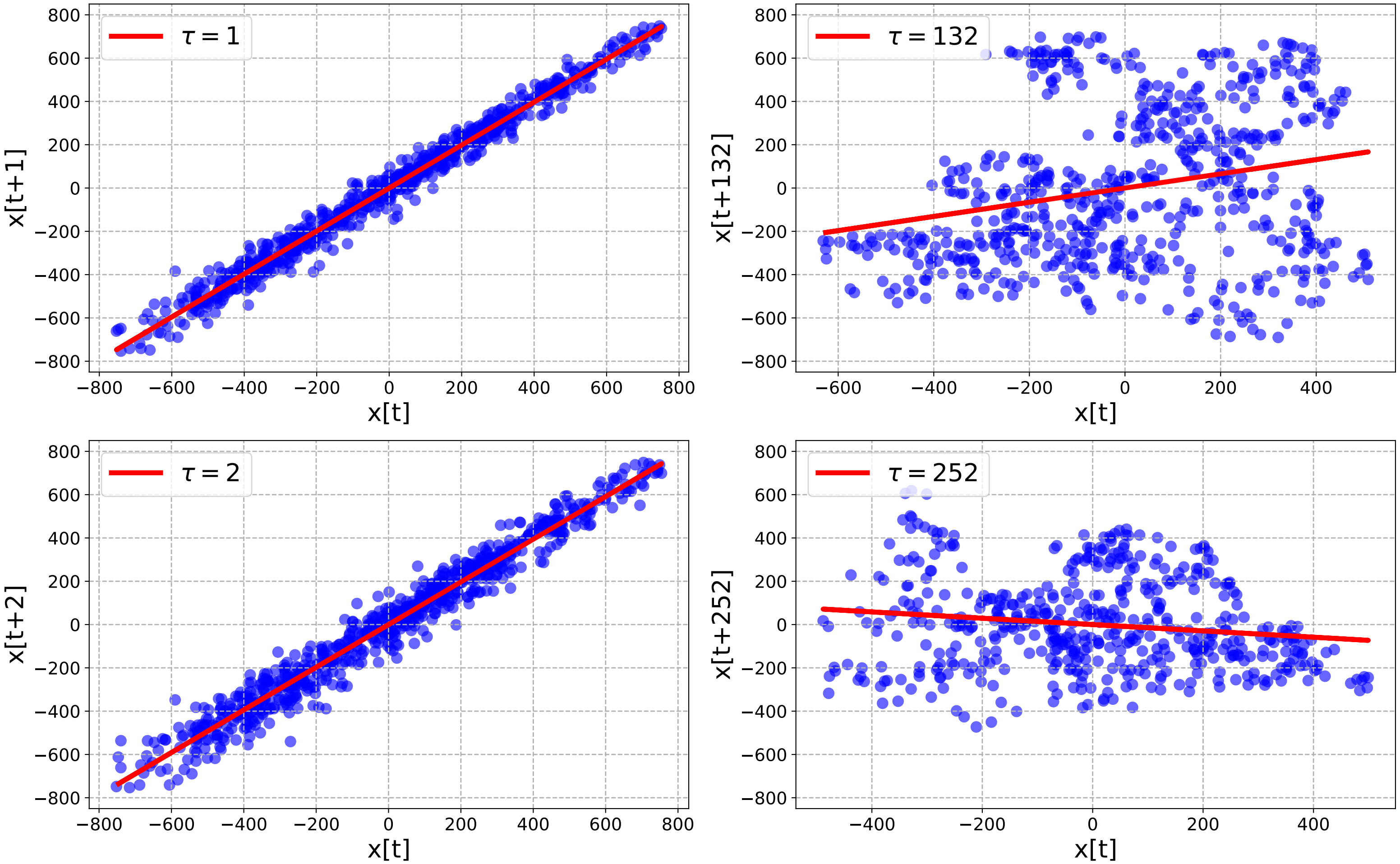
Linear Regression

- Establishes a relationship between a continuous dependent variable and one or more independent variables
- Does not require large sample size for successful operation
- Easily interpretable and intuitive
- Applications across disciplines
- Polynomial regression $\looparrowright$ universal function approximation
- Robust regression required in the presence of outliers
- 'Linear in the parameters' family
- Parameters typically found using Least Squares methods

Logistic Regression

- Estimates a relationship between a categorical dependent variable and one or more continuous independent variables
- Requires large sample size to represent values across all response categories
- A discriminative model, aims to distinguish between categories
- Multinomial logistic regression uses the SoftMax function to compute probabilities
- Parameters typically found using Maximum Likelihood Estimation
- Linear in the logit space

Appendix: Scatter plots of the detrended S&P 500 financial index



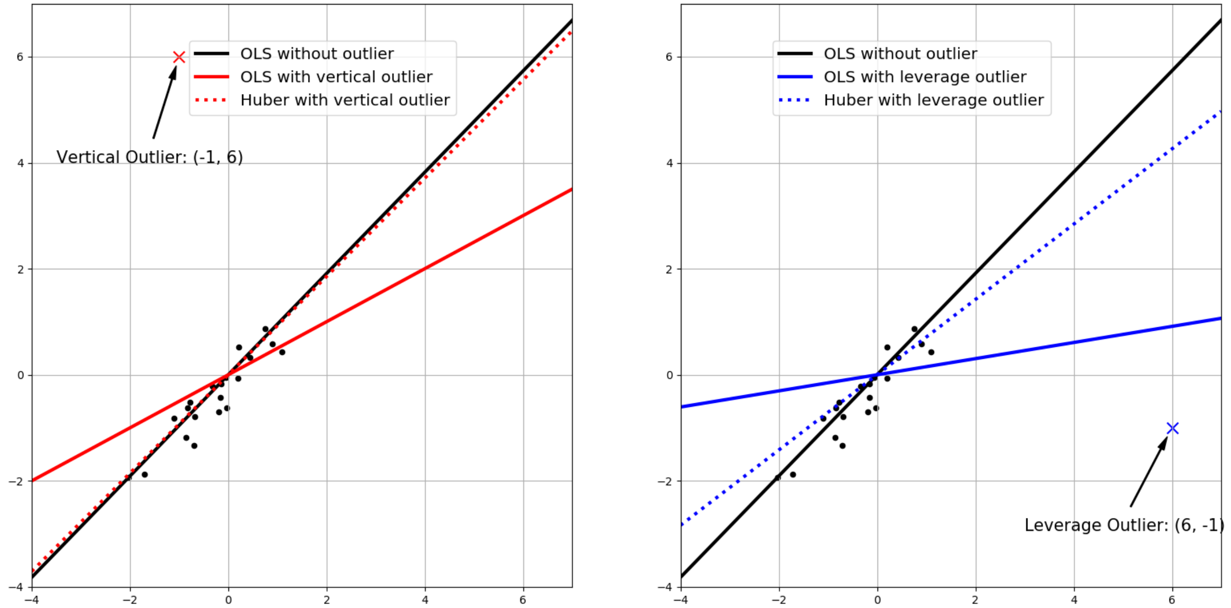
The detrended S&P 500 time series shows strong correlations for small lags in the scatter plot, and spurious correlations for large lags.

Appendix: Influence of outliers in linear regression

There are two general types of outlying observations:

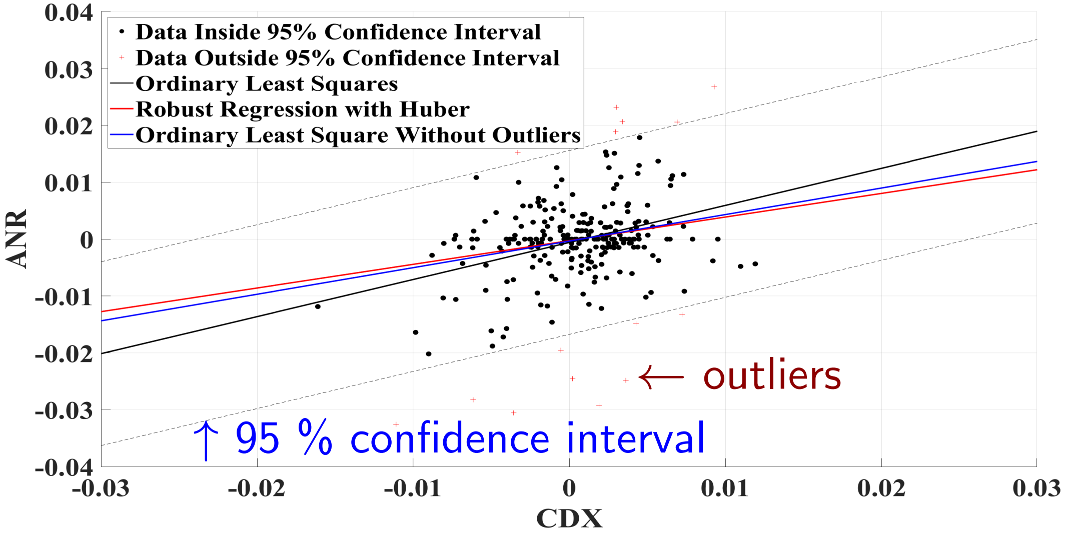
- ○ **Vertical outliers:** y_n is outlying while x_n is not outlying (easy to fix)
- ○ **Leverage points:** y_n is not outlying while x_n is outlying (complicated)

Problems caused by leverage points: As the outlier in the explanatory variable, it has an unbounded influence (full weighting of leverage points).



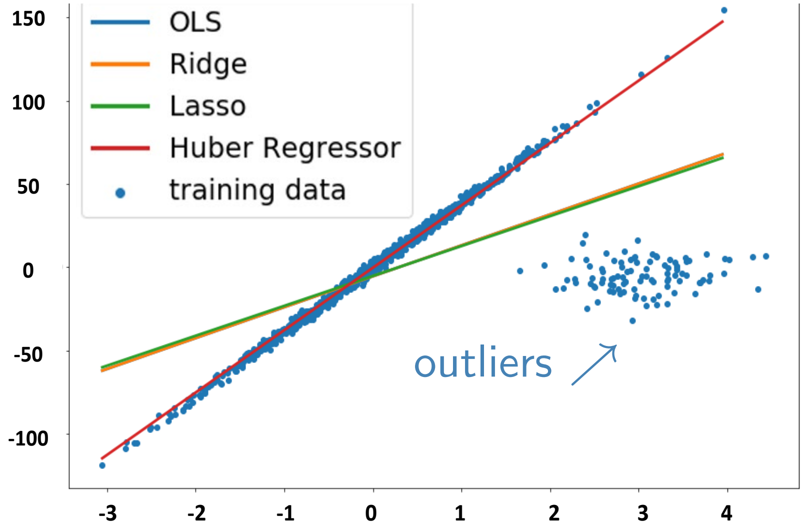
☞ Robust (e.g. Huber) estimators are almost unaffected by vertical outliers (deviation from concentration of explained variable, y), but are vulnerable to leverage points (outside the concentration of explanatory variable, x)

Appendix: Sensitivity to outliers of the ordinary Least Squares (OLS) (role of regularisation and robust estimators)



← outliers

↑ 95 % confidence interval



outliers ↗

Regression of daily returns of Altona Energy (ANR) corporate bond on the credit default swap (CDX).

Regression under outliers

Huber is a robust estimator

Ridge regression:

$$\mathcal{J}_n(\mathbf{w}) = \underbrace{(d_n - \mathbf{w}_n^T\mathbf{x}_n)^2}_{\text{standard cost}} + \underbrace{\lambda_1\|\mathbf{w}_n\|_2^2}_{L_2 \text{ penalty}} = e_n^2 + \lambda_1\mathbf{w}_n^T\mathbf{w}_n$$

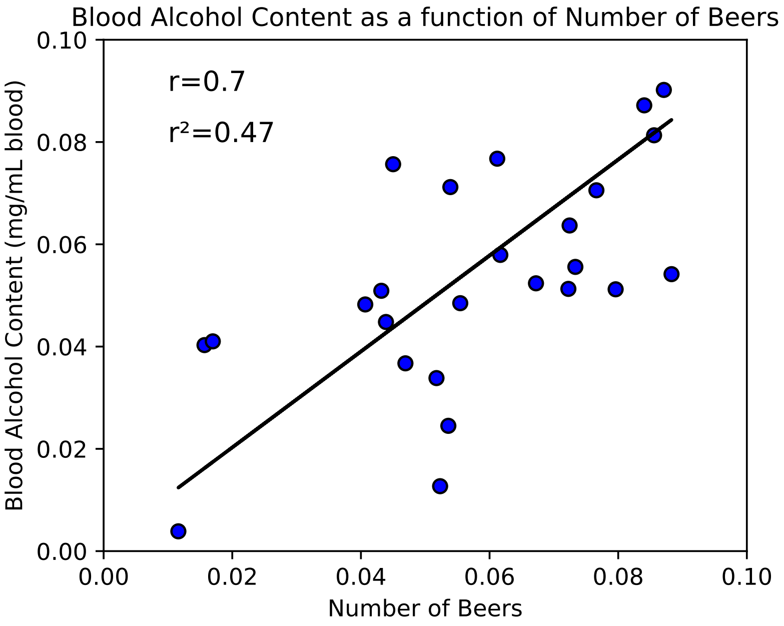
LASSO (sparsity promoting):

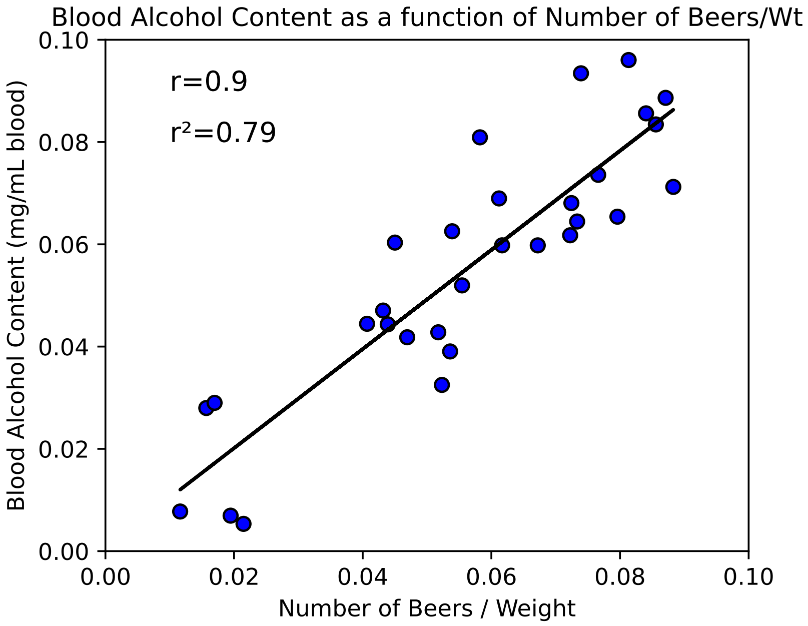
$$\mathcal{J}_n(\mathbf{w}) = \underbrace{(d_n - \mathbf{w}_n^T\mathbf{x}_n)^2}_{\text{standard cost}} + \underbrace{\lambda_2\|\mathbf{w}_n\|_1}_{L_1 \text{ penalty}}$$

○ Ridge: Penalises for large weights (but does not reduce system dimensionality)

○ Least absolute shrinkage and selection operator (LASSO) enforces insignificant weights to go to zero, and thus promotes sparsity and aids **interpretability** ($\lambda_1, \lambda_2 \looparrowright$ param's.)

Appendix: Coefficient of determination, R^2


Blood Alcohol Content as a function of Number of Beers


Blood Alcohol Content as a function of Number of Beers/Wt

Coefficient of determination $R^2 = \frac{SSR}{SST} = \frac{\text{sum of squares regression}}{\text{total sum of squares}} = \frac{\sum(\hat{y}-\bar{y})^2}{\sum(y-\bar{y})^2}$

Coef. of determ. represents the portion of total variation in the dependent variable that is explained by variation in the independent variable.

∘ R^2 behaves like a square of the correlation coef. and ranges from 0 to 1

∘ R^2 does not decrease when a new x variable is added to the model (this may be a disadvantage when comparing models)

Appendix: Multicollinearity

Multi-collinearity refers to the existence of high correlations among the independent variables (predictors).

◦ This means that the the correlated explanatory (predictor) variables provide redundant information to the multiple regression model.

◦ Numerical difficulties in Least Squares solutions (ill-conditioned matrix of predictors) unless “extra” predictor variables are removed.

◦ Difficult to assess the relative importance of independent variables when explaining the variation in the dependent variable, e.g. a previously significant independent variable becomes insignificant.

◦ Difficult to make inferences about the effects of individual regression coefficients on the dependent variable y (lack of intepretability).

◦ The partial regression coefficients may not be estimated precisely.

◦ The estimated standard deviation of the model increases when a variable is added to the model.

Q: Does multi-collinearity mean that multiple regression does not work?

A: Multi-collinearity does not affect the ability of multiple regression to predict the dependent variable, y, but affects stability and interpretability of regression coefficients.

Appendix: Multicollinearity and metrics related to regression

Practical indicators of collinearity: (i) When we add/remove an independent variable, the values of the remaining regression coefficients undergo a drastic change. (ii) From domain knowledge: an independent variable which is known to be an important predictor is associated with a small regression coefficient. (iii) Domain knowledge: a regression coefficient which should be positive becomes negative, and vice versa

○ A rule-of-thumb is that if the correlation between two independent variables is between -0.70 to 0.70, keep both independent variables

○ A more precise test of multi-collinearity is the Variance Inflation Factor VIF $= \frac{1}{1-R_j^2}$, with R_j^2 as coef. of determination after the j-th independent variable is regressed against the remaining (p - 1) independent variables

○ Rule-of-thumb: If VIF > 10 we should remove the considered independent variable from the analysis (see the next slide)

○ Rule-of-thumb: Calculate correlation between the independent variables and use only one of the highly correlated variables

○ Alternatively, transform the existing independent variables into a new set of mutually independent predictors (PCR, latent root regression)

Appendix: Multicollinearity $\looparrowright$ how to detect it

Start from the standard multiple regression model

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

Then, regress each independent variable against the (p-1) other independent variables

$$\begin{aligned}
\hat{x}_1 &= \beta_0 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_p x_p \\
\hat{x}_2 &= \beta_0 + \beta_1 x_1 + \beta_3 x_3 + \cdots + \beta_p x_p \\
\vdots \;\; & \;\; \vdots \\
\hat{x}_p &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{p-1} x_{p-1}
\end{aligned}$$

☞ If an independent variable can be expressed via a linear combination of other independent variables, then it is redundant.
We can measure this dependence via the Tolerance, $T = 1 - R^2$
or the Variance Inflation Factor (VIF), $VIF = \frac{1}{1-R^2}$.
In practice, an independent var. can be removed if $T < 0.1$ or $VIF < 10$.

Appendix: Dealing with categorical explanatory variables

We need to transform categorical variables into a format suitable for algorithms

For categories of an ordinal variable, such a condition of a car (poor, average, good), we can assign numerical scores to the categories, e.g. (poor=1, average=2, good=3), which makes perfect sense.

Problem: Often, we cannot establish rank between categorical variables, e.g. "red" is not greater than "blue", "male" is not greater than "female".
Solution: Resort to a "dummy" (indicator) variable in the form of e.g. **1** if the category is true and **0** it the category is false (see Slide 23).

Example: Find an average weight wrt gender via categorical variables

$$weight_i = \beta_1 \cdot female_i + \beta_2 \cdot male_i + \alpha$$

P: Direct use of variables $male$ and $female$ does not make sense, while the dummy variables $x_1{=}female \in \{0,1\}$ and $x_2{=}male \in \{0,1\}$, exhibit a linear relationship, as $x_1 + x_2 = 1 \rightarrow x_2 = 1 - x_1$, causing collinearity.

S: Encode our $N{=}2$ categories into N-1 **dummy var.** $\delta^{male} = 1$ if true,

so that $$weight_i = \beta \cdot \delta_i^{male} + \alpha \quad \rightarrow \quad y = \beta x_2 + \alpha$$

α = mean weight of category 0 (female), β = weight difference between cat. 0 and 1.

☞ Categorical information has been encoded by a binary **indicator variable**

Appendix: Dealing with multiple categorical (qualitative) explanatory variables $\looparrowright$ indicator variables, contd.

Example: Find an average weight wrt gender and exercise level by extending the previous example with the binary category "exercise level",
$N = 3$ categories: E_1=x_3=*daily*, E_2=x_4=*often*, E_3=x_5=*sometimes*
Now, $x_3 + x_4 + x_5 = 1 \quad \rightarrow \quad x_3 = 1 - x_4 - x_5$, and

$$weight_i = \beta_1 \cdot \delta_i^{male} + \beta_2 \cdot \delta_i^{E_2} + \beta_3 \cdot \delta_i^{E_3} + \alpha$$

We now have 3 parameters describing 5 categories, where α is the average weight of a female who exercises daily (Cat E_1), β_1 models the effects of gender on weight (without accounting for exercise), while β_2 and β_3 give the effects of exercise level on mean weight (without accounting for gender), all relative to α= weight of a female who exercises daily.

Summary: A dummy indicator variable converts a categorical variable with N categories into $(N - 1)$ binary variables which have the value of **1** if an observation belongs to a certain category, L, and **0** otherwise.

E.g. we wish to predict the price of a car based on their category (1 or 0) x_1=hatch, x_2=saloon, x_3=SUV, x_4=estate. For these $N = 4$ categories we need N-1=3 indicator variables, as $\sum_{i=1}^{4} x_i = 1$ and $x_4 = 1 - \sum_{1}^{3} x_i$, so

$$price = \beta_0 + \beta_1 \times hatch + \beta_2 \times saloon + \beta_3 \times SUV$$

Appendix: Dealing with multiple categorical explanatory variables ↬ one-hot encoding

While a ‘dummy’ indicator variable models the presence or absence of a particular category, in this way one category is not explicitly represented.

One-hot encoding is a specific method of creating indicator variables, whereby each of N categorical variables is converted into an N-dimensional binary ’indicator vector’, so that each category is explicitly represented.

☞ In this ‘1-of-N’ scheme, each category is treated equally without implying any ordinal relationship, e.g. red $= [1, 0, 0]$, green $= [0, 1, 0]$, blue $= [0, 0, 1]$ or hatch$=[1, 0, 0, 0]$, saloon$=[0, 1, 0, 0]$, SUV$=[0, 0, 1, 0]$, estate$=[0, 0, 0, 1]$.

In this way, one-hot encoding provides a sparse representation which:

◦ Is straightforward, through a vector of N variables for N categories,

◦ Handles uniformly both nominal (no natural order) and ordinal (natural order but treated as nominal) categorical var., enhancing interpretability,

◦ Simplifies the data pre-processing pipeline through this uniformity.

One-hot encoding: Commonly used to handle high-dimensional feature spaces in ML, e.g. at the tokenisation stage in Natural Lang. Proc. (NLP).

Dummy indicator variables: Typically used in linear regression with categorical variables, and to avoid multi-collinearity.

Appendix: Logistic Regression $\looparrowright$ Historical notes

○ Logistic function was introduced by the Belgian mathematician Pierre Francois Verhulst in 1838 to model population growth

○ It was later popularised by Pearl and Reed whose solution is in the form of the logistic function we know today

○ In 1944, Joseph Berkson introduced the term 'logit' (in analogy to the 'probit' model) and developed a logistic model for use in medical statistics.

○ Logistic regression was further developed in statistics for the analysis of binary data in the 1960s, and was common in medicine

○ The book "The Analysis of Binary Data" by D. Cox and J. Snell elaborated on the proportional hazards model and logistic regression

○ In the late 1970s it became prominent in linguistics (linguistic variation)

○ It was used in Natural Language Processing (NLP) since the 1990s, also under the names **maximum entropy modelling** or **maxent**, for language modelling, text classification, and speech tagging)

○ It has become fundamental in machine learning for binary classification (spam detection, credit scoring), as it is simple, interpretable and effective

Appendix: Odds versus probabilities

Problem: Probabilities are not linear so an increase 10% to 20% doubles the probability, but increase 80% to 90% only slightly improves probability.

Odds vs probabilities

Probability$=\frac{\text{number of favourable outcomes}}{\text{total number of outcomes}}$, so for flipping a coin $p(heads) = 0.5$

Odds $= \frac{\text{prob. of event happening}}{\text{prob. of event not happening}}$ so for flipping a coin $\frac{p(heads)}{1-p(heads)} = \frac{0.5}{1-0.5}$=1:1

Example: Rolling a dice

The probability of rolling 1 on a 6-sided die is $p(1) = \frac{1}{6} \approx 16.7\%$

The odds of rolling a 1 on a die are $\frac{p(1)}{1-p(1)} = \frac{1/6}{1-1/6} = \frac{1}{5}$ or Odds $=$ 1:5

Converting between odds and probability

$$Odds = \frac{p}{1-p} \qquad\qquad p = \frac{Odds}{1+Odds}$$

So, if the Odds of an event are 2:1, then $p = \frac{2}{1+2} = 0.667 = 66.7\%$

If the probability of an event is $p = 0.25$, then $Odds = \frac{0.25}{1-0.25} = \frac{1}{3}$, so 1:3

Example: Covid19 affects 1 in 1000 people. The diagnostic test has sensitivity (true positive rate) $= 99\%$ specificity (true negative rate) 99%

Appendix: Odds versus probabilities

☞ The odds of a randomly selected person having Covid are $\frac{0.001}{0.999} = 1 : 999$

Scenario: Positive test (PT)

False positive rate = (1- specificity) = 1 %
False negative rate = (1 - sensitivity) = 1 %

$$p(Covid|PT) = p(PT|Covid) \times p(Covid) + p(PT|No\ Covid) \times p(No\ Covid)$$

where $p(PT|Covid) = 0.99$, $p(Covid) = 0.001$, $p(PT|NoCovid) = 0.01$, $p(NoCovid) = 0.999$

$$\text{so:} \quad p(PT) = 0.99 \cdot 0.001 + 0.01 \cdot 0.999 = 0.01098$$

$$\text{and} \quad p(Covid|PT) = \frac{0.99 \cdot 0.001}{0.01098} \approx 0.0902 \quad or \quad 9.02\%$$

So, the probability that a person has Covid given that they have tested positive is 9.02%

Now, the $Odds(Covid|PT) = \frac{0.0902}{1-0.0902} \approx 0.099$ or 1:10

Interpretation: Probability provides a direct measure of the likelihood of an event occurring. Odds offers a comparative measure of the likelihood of an event occurring versus it not occurring.

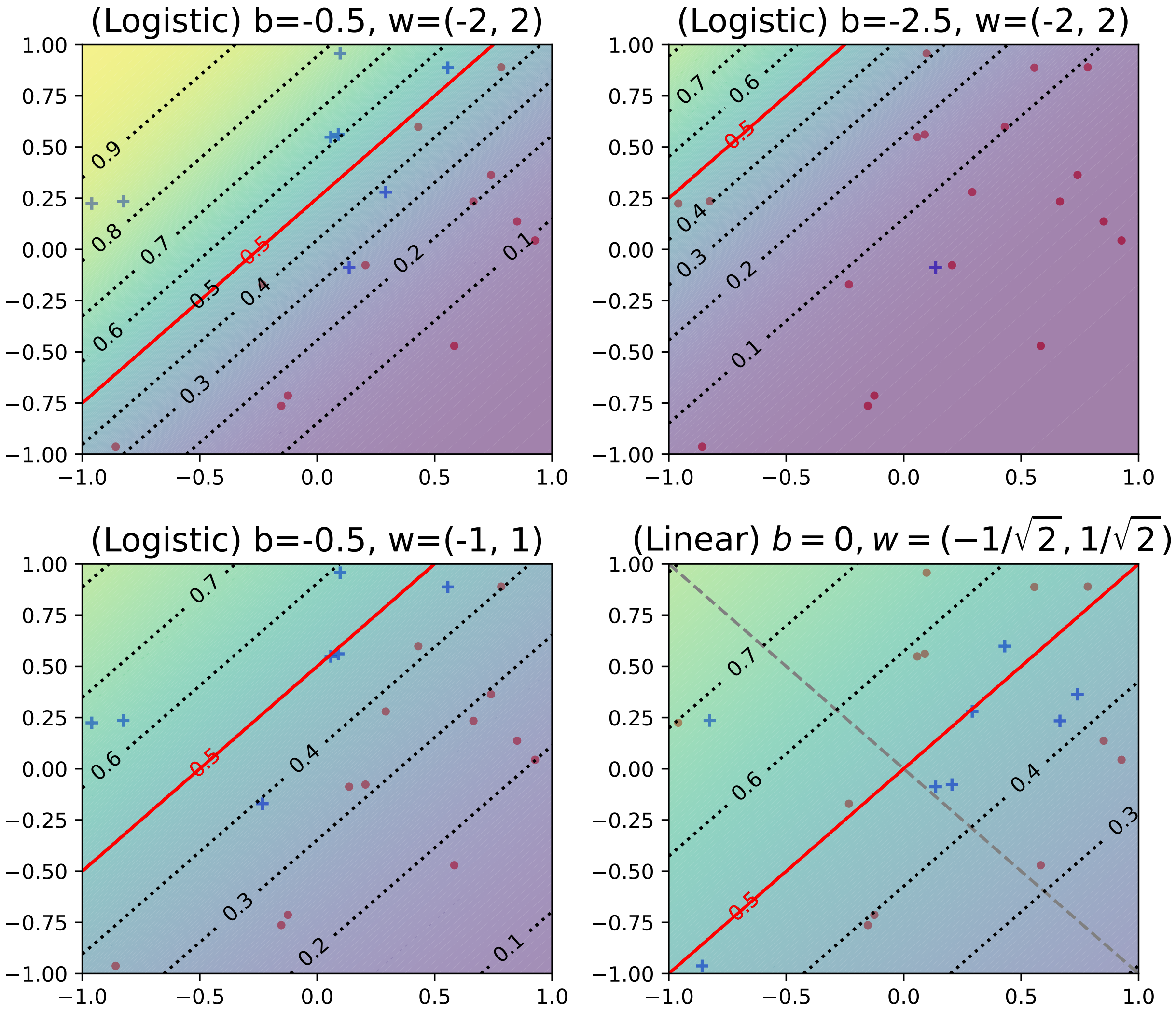
Appendix: Effects of Scaling Logistic Regression Parameters

Effects of scaling the parameters of logistic regression:

The values of x and y were the same in all plots and were drawn from the $\mathcal{U}(-1,1)$ distribution.

The labels were generated randomly from logistic regressions with different w and β, and from a perfect linear classifier with the same boundary.

The red line designates a contour of 0.5.



(Logistic) b=-0.5, w=(-2, 2)

(Logistic) b=-2.5, w=(-2, 2)

(Logistic) b=-0.5, w=(-1, 1)

(Linear) $b = 0, w = (-1/\sqrt{2}, 1/\sqrt{2})$

Appendix: Finding parameters of Logistic Regression

For simplicity, we consider a binary classification task

For ○ $y = 1$, the predicted probabilities will be $\hat{y} = p(x; \beta_0, \boldsymbol{\beta}) = p(x)$
○ $y = 0$, the predicted probabilities will be $1 - \hat{y} = 1 - p(x)$

In other words, for $y = 1$ our aim is to estimate $\boldsymbol{\beta}$ and β_0 so that the product of all probabilities $\hat{y} = p(x)$ is close to 1, while for $y = 0$ the product of all probabilities, $1 - \hat{y} = 1 - p(x)$, should also be close to 1.

Upon combining these conditions into a Likelihood Function (Bernoulli)

$$p(y|\mathbf{x}) = \hat{y}^{y}(1-\hat{y})^{1-y} \quad \rightarrow \quad L(y|\mathbf{x}; \beta_0, \boldsymbol{\beta}) = \prod_{n=1}^{N} \hat{y}^{y_n}\big(1-\hat{y}\big)^{1-y_n}$$

Our goal becomes that of finding the parameters, $\boldsymbol{\beta}$ and β_0, which **maximise the likelihood,** $L(\cdot)$, and consequently the log-likelihood, $l(\cdot)$

$$l(y|\mathbf{x}; \beta_0, \boldsymbol{\beta}) = \sum_{n=1}^{N} \Big[y_n \log \hat{y} + (1-y_n)\log(1-\hat{y})\Big]$$

This is known as Maximum Likelihood Estimation (MLE). (see Lecture 5)

We shall now show that maximising the likelihood is equivalent to minimising the cross-entropy, a typical cost function in Neural Networks, given by $J_{CE}(\hat{y}, y) = -\log p(y|\mathbf{x})$

Appendix: Maximising the log-likelihood function is equivalent to minimising the cross-entropy loss in NNs

Since the $\log$ is a monotonic function, it is more convenient to maximise

$$
\begin{aligned}
l(y|\mathbf{x};\beta_0,\boldsymbol{\beta}) &= \sum_{n=1}^{N}\Big[y_n\log\hat{y}+(1-y_n)\log(1-\hat{y})\Big] \\
&= \sum_{n=1}^{N}\log(1-\hat{y})+\sum_{n=1}^{N}y_n log\frac{\hat{y}}{1-\hat{y}} \\
&= \sum_{n=1}^{N}\log\left(\frac{1}{1+e^{\beta_0+\boldsymbol{\beta}^T\mathbf{x}}}\right)+\sum_{n=1}^{N}y_n(\beta_0+\boldsymbol{\beta}^T\mathbf{x}) \\
&= \sum_{n=1}^{N}y_n(\beta_0+\boldsymbol{\beta}^T\mathbf{x})-\log\left(1+e^{\beta_0+\boldsymbol{\beta}^T\mathbf{x}}\right)
\end{aligned}
$$

With $\boldsymbol{\theta}=[\beta_0,\boldsymbol{\beta}]$ and the cross-entropy, $J_{CE}(\hat{y},y)$, we have

$arg\max_{\boldsymbol{\theta}} L = arg\min_{\boldsymbol{\theta}} J_{CE}(y|\mathbf{x};\boldsymbol{\theta}) = arg\min_{\boldsymbol{\theta}} -\sum_{n=1}^{N}\log L(y_n|\mathbf{x};\boldsymbol{\theta})$

In this way, the cross-entropy loss is smaller if the estimate $\hat{y}$ is close to the correct y and bigger if the estimate is further from the correct y.

Notes:

- ○

Notes:

○

Notes:

○