Linear regression

Mauricio A. Álvarez

Machine Learning and Adaptive Intelligence
The University of Sheffield

Contents

Scalar and vectors

标向量, 标量

- A **scalar** is just a numeric value like 0.9 or −18.7.

- Scalars are usually denoted as lower case letters like *x* or *a*.

- A **vector** is an ordered list of scalar values. Sometimes we refer to these scalar values of the vector as *attributes* or *entries* of the vector.

- Vectors are usually denoted by bold lowercase letters like **x** or **y**.

Vectors

- A vector can appear sometimes written as a row vector, e.g.

$$\mathbf{x} = [x_1, x_2, x_3, x_4, x_5]$$

  Or as a column vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

- In this module, ALL vectors will be column vectors by default. So, when you see a vector, e.g. $\mathbf{x}, \mathbf{y}, \mathbf{z}$ always think this vector has a column-wise shape.

Matrices

- A **matrix** is a rectangular array of scalars arranged in rows and columns.

- Matrices are usually denoted by bold uppercase letters, e.g. **X** or **Y**.

- The following matrix has three rows and two columns

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{bmatrix}$$

- The entries in the matrix above are of the form x_{ij}, where the first subindex i indicates the row of the element and the second subindex j indicates the column.

Matrix transpose

- Let **X** be a matrix with elements x_{ij}.

- The <u>transpose</u> of a matrix **X** is a new matrix $\mathbf{X}^\top$ with elements x_{ji}.

$$\mathbf{X} = \begin{bmatrix} 4.1 & -5.6 \\ -2.6 & 7.9 \\ 3.5 & 1.8 \end{bmatrix}, \quad \mathbf{X}^\top = \begin{bmatrix} 4.1 & -2.6 & 3.5 \\ -5.6 & 7.9 & 1.8 \end{bmatrix}$$

Matrix multiplication

- ❑ Let **A** be a matrix with entries a_{ik} of dimensions $p \times q$.

- ❑ Let **B** be a matrix with entries b_{kj} of dimensions $t \times s$.

- ❑ Matrix multiplication of the form **AB** is only possible if $q = t$.

- ❑ If this is the case, the matrix **C** = **AB** has dimensions $p \times s$ with entries

$$c_{ij} = \sum_k a_{ik} b_{kj}.$$

Transpose of a product

- Let **w** be a vector of dimensions $d \times 1$. Let **X** be a matrix with dimensions $n \times d$.

- The transpose of the product **Xw**, $(\mathbf{Xw})^\top$ is

$$(\mathbf{Xw})^\top = \mathbf{w}^\top \mathbf{X}^\top.$$

How to prove that? (Linear algebra)

- We can apply this result to a product of several matrices

$$
\begin{aligned}
(\mathbf{ABCD})^\top &= ((\mathbf{AB})(\mathbf{CD}))^\top \\
&= (\mathbf{CD})^\top (\mathbf{AB})^\top \\
&= \mathbf{D}^\top \mathbf{C}^\top \mathbf{B}^\top \mathbf{A}^\top.
\end{aligned}
$$

From a scalar operation to a vector operation

❑ It is usually desirable to transform a scalar operation into a vector operation.

❑ When coding scalar operations, we require making use of loops, which can be expensive.

❑ In contrast, vector operations are handled efficiently by low-level routines already included in modules like numpy.

Example

Write the following scalar operation into a vector/matrix form

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{d} x_{ij}w_j)^2.$$

$i=1 : \left(y_1 - \sum_{j=1}^{d} x_{1j}w_j \right)^2$

$\left(y_2 - \sum x_{2j}w_j \right)^2$

$\vdots$

$(y_n - \sum_{j=1}^{d} x_{nj}w_j)^2$

$y_1 - \begin{cases} x_{1,1} w_1 \\ + x_{1,2} w_2 \\ + x_{1,5} w_3 \\ + \vdots \\ x_{1,d} w_d \end{cases}$

$(\cdots) = [x_{1,1}, x_{1,2} \cdots x_{1,d}] \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$

$1 \times d \quad d \times 1$

$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

$y_1 - x_1 w$

$y_2 - x_2 w$

$\vdots$

$y_n - x_n w$

$\Rightarrow Y - w X.$

Answer (I)

❑ The sum above can be written as

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{d} x_{ij}w_j)^2 = (y_1 - \sum_{j=1}^{d} x_{1j}w_j)(y_1 - \sum_{j=1}^{d} x_{1j}w_j)$$
$$+ \cdots$$
$$+ (y_n - \sum_{j=1}^{d} x_{nj}w_j)(y_n - \sum_{j=1}^{d} x_{nj}w_j).$$

❑ Let us define a vector **v** of dimensions $n \times 1$ with entries given as

$$(y_i - \sum_{j=1}^{d} x_{ij}w_j).$$

Answer (II)

❑ The product of vectors $\mathbf{v}^\top \mathbf{v}$ gives the same result than the required sum,

$$\mathbf{v}^\top \mathbf{v} = \begin{bmatrix} (y_1 - \sum_{j=1}^{d} x_{1j} w_j) & \cdots & (y_n - \sum_{j=1}^{d} x_{nj} w_j) \end{bmatrix} \begin{bmatrix} (y_1 - \sum_{j=1}^{d} x_{1j} w_j) \\ \vdots \\ (y_n - \sum_{j=1}^{d} x_{nj} w_j) \end{bmatrix}$$

$$= \sum_{i=1}^{n} (y_i - \sum_{j=1}^{d} x_{ij} w_j)^2.$$

❑ How do we express the elements in $\mathbf{v}$ with vectors and matrices?

Answer (III)

- For a fixed i, $x_{i1}, \ldots, x_{id}$ can be grouped into a vector $\mathbf{x}_i^\top$.
- The internal sums in the entries of $\mathbf{v}$ can then be written as

$$\sum_{j=1}^{d} x_{ij} w_j = \mathbf{x}_i^\top \mathbf{w} = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{id} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

- We can now write $\mathbf{v}$ as
$$\mathbf{v} = \begin{bmatrix} y_1 - \mathbf{x}_1^\top \mathbf{w} \\ \vdots \\ y_n - \mathbf{x}_n^\top \mathbf{w}. \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} \mathbf{x}_1^\top \mathbf{w} \\ \vdots \\ \mathbf{x}_n^\top \mathbf{w}. \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix} \mathbf{w}$$

- We can group the scalars $y_1, \ldots, y_n$ into a vector $\mathbf{y}$.
- We can group the row vectors $\mathbf{x}_1^\top, \ldots, \mathbf{x}_n^\top$ into a matrix $\mathbf{X}$.

Answer (IV)

❑ It means that $\mathbf{v} = \mathbf{y} - \mathbf{Xw}$.

❑ Finally

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{d} x_{ij}w_j)^2 = \mathbf{v}^\top \mathbf{v} = (\mathbf{y} - \mathbf{Xw})^\top (\mathbf{y} - \mathbf{Xw}).$$

Two common types of products

- **Inner product**. The inner product between two vectors results in a scalar.
- Let **x** and **y** be vectors of dimension $m \times 1$. The inner product is given as

the first x determine the row
and
second y determine the column

inner

$$\mathbf{x}^{\top}\mathbf{y} = \sum_{i=1}^{m} x_i y_i,$$

$(1 \times m \cdot m \times 1)$ → (1×1)

- **Outer product**. The outer product between two vectors results in a matrix.
- Let **x** be a vector of dimension $m \times 1$ and **y** a vector of dimension $p \times 1$. The outer product is given as

outer

$$\mathbf{x}\mathbf{y}^{\top} = \begin{bmatrix} x_1 y_1 & \cdots & x_1 y_p \\ x_2 y_1 & \cdots & x_2 y_p \\ \vdots & \vdots & \vdots \\ x_m y_1 & \cdots & x_m y_p. \end{bmatrix}$$

$m \times 1 \cdot 1 \times p$
$= m \times p$

Differentiating a function in a vector/matrix form (I)

- ❏ We will see cases in which a function $f(\mathbf{w})$ depends on some parameters grouped in a vector $\mathbf{w}$.

- ❏ We would like to find the vector of parameters $\mathbf{w}$ that maximise $f(\mathbf{w})$.

- ❏ For example, suppose $f(\mathbf{w})$ is defined as

$$w_1 b_1 + w_2 x_2 + \cdots w_d x_d \leftrightarrow |x|$$
$$\begin{bmatrix} w_1 & \cdots & w_d \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \quad \text{or} \quad (x_1 \cdots x_d) \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$$

$$f(\mathbf{w}) = \sum_{i=1}^{d} w_i x_i. \qquad w \cdot x \qquad \text{or} \qquad x^\top w.$$

- ❏ We can group the scalars $x_1, \ldots, x_d$ into $\mathbf{x}$. Likewise for $\mathbf{w}$.

- ❏ According to what we saw before, we can write $f(\mathbf{w})$ as $f(\mathbf{w}) = \mathbf{x}^\top \mathbf{w}$.

Differentiating a function in a vector/matrix form (II)

❑ For a fixed **x**, we are interested in computing the gradient of $f(\mathbf{w})$ with respect to **w**

$$\frac{df(\mathbf{w})}{d\mathbf{w}} = \begin{bmatrix} \frac{\partial f(\mathbf{w})}{\partial w_1} \\ \vdots \\ \frac{\partial f(\mathbf{w})}{\partial w_d} \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_d. \end{bmatrix} = \mathbf{x}.$$

❑ Some useful identities when differentiating with respect to a vector

1. $x = \begin{bmatrix} x_1, x_2, \cdots x_d \end{bmatrix}^{\top}$

$w^{\top}x = w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots + w_d x_d$

$(w^{\top}x) \, dw = x_1 + x_2 + \cdots + x_d = x$

$f(\mathbf{w})$	$\frac{df(\mathbf{w})}{d\mathbf{w}}$
$\mathbf{w}^{\top}\mathbf{x}$	$\mathbf{x}$
$\mathbf{x}^{\top}\mathbf{w}$	$\mathbf{x}$
$\mathbf{w}^{\top}\mathbf{w}$	$2\mathbf{w}$
$\mathbf{w}^{\top}\mathbf{Cw}$	$2\mathbf{Cw}.$

easy to comprehend

3. $w^{\top}w = w_1 \cdot w_1 + \cdots + w_d \cdot w_d$

$(w^2 w) \, dw = w_1^2 \, dw + \cdots + w_d^2 \, dw = 2w_1 + 2w_2 + \cdots + 2w_d = 2w.$

Identity matrix and the inverse of a matrix

☑ The identity matrix of size *N* is a square matrix with ones on the main diagonal and zeros elsewhere, e.g.,

$$\mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

❑ The inverse matrix of a matrix **A** of dimensions $d \times d$, denoted as $\mathbf{A}^{-1}$, satisfies

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_d$$

Contents

Olympic 100m Data

Image from Wikimedia Commons `http://bit.ly/191adDC`.
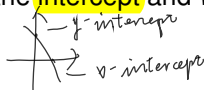
Dataset

Male 100 *mts*
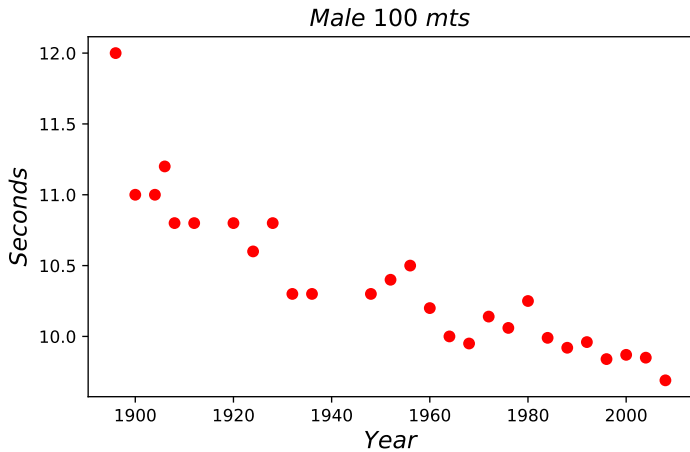
Model

- We will use a linear model $f(x, \mathbf{w})$ to predict y, where y is the time in seconds and x the year of the competition.

  1-dimension Gradient

- The linear model is given as

$$f(x, \mathbf{w}) = w_0 + w_1 x,$$
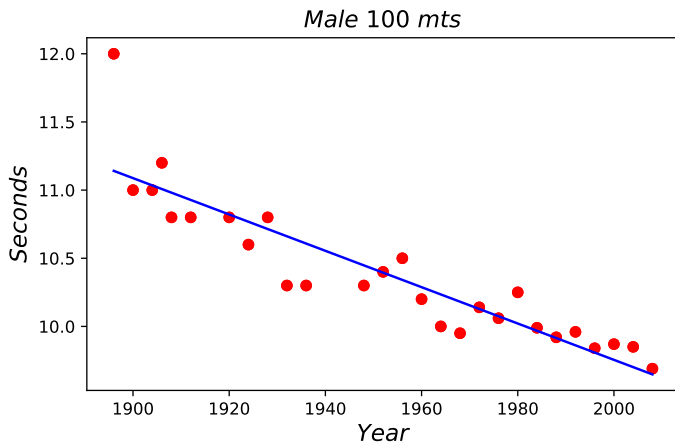
where w_0 is the **intercept** and w_1 is the **slope**.

  - y-intercept

  v-intercept

- We use $\mathbf{w}$ to refer both to w_0 and w_1.

$$\begin{Bmatrix} w_0 \\ w_1 \end{Bmatrix}$$

Data and model

Contents

Linear model

- a build-up model to lengthen the old data to make it able to accomplish prediction

□ A simple model for regression consists in using a <u>linear combination</u> of the attributes to predict the output

Very like the formula to calculate the Expectation Value
Similar to

$$f(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_D x_D, \rightarrow f(x)$$

where $w_0, w_1, \cdots, w_D$ are the parameters of the regression model.

□ The term w_0 is the bias term or intercept, e.g. $f(\mathbf{0}, \mathbf{w}) = w_0$.

□ The expression above can be written in a <u>vectorial form</u>

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^\top \mathbf{x}.$$
$$\underset{1 \times D}{} \underset{D \times 1}{} \rightarrow 1 \times 1$$

$$w^\top \cdot x \rightarrow [w_0, w_1 \cdots w_D] \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ : \\ x_D \end{pmatrix}$$

where we have defined $\mathbf{w} = [w_0, w_1, \cdots, w_D]^\top$ and $\mathbf{x} = [1, x_1, \cdots, x_D]^\top$.

□ Notice that $x_0 = 1$. $x_0 \cdot w_0 \rightarrow$ *intercept*

25/63

Parenthesis: Gaussian pdf

mass ↩ discrete

probability { density function
→ continous

μ: mean or expectation of the distribution.
σ: standard deviation
σ²: variance of the distribution

□ The Gaussian pdf has the form

高斯都师
正态都师

$$p(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(y-\mu)^2}{2\sigma^2} \right\}.$$

$$f(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2}$$

$P(Y=y)$: the probability of Event y, happen

□ A Gaussian pdf requires two parameters μ and σ^2, the mean and the variance of the RV Y.

or expectation

□ We denote the Gaussian pdf as $p(y|\mu, \sigma^2) = \mathcal{N}(y|\mu, \sigma^2)$ or $y \sim \mathcal{N}(\mu, \sigma^2)$.

normal distribution

$$P(y|\mu, \sigma^2) = \mathcal{N}(y|\mu, \sigma^2)$$

$$or = y \sim \mathcal{N}(\mu, \sigma^2) \checkmark$$

Parenthesis: Gaussian pdf

The mean of the three Gaussians is $\mu = 2$ and the variances are $\sigma^2 = 0.5$ (solid), and $\sigma^2 = 2$ (dashed).

$\sigma^2 \uparrow$, difference between x and μ ↑ the more wavy it is.

$\sigma^2 \uparrow$
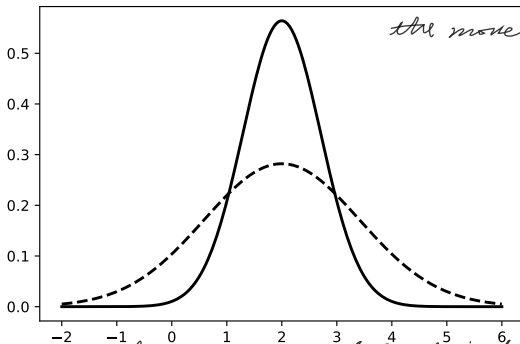
σ^2: describes, to how much degree each sample is different from the sum of the { mean value / expectation value

(variance

Gaussian regression model (I) _build connection_

- We use a Gaussian regression model to relate the inputs and outputs

$$y = f(\mathbf{x}, \mathbf{w}) + \epsilon,$$

Target attribution

$f(\mathbf{x}, \mathbf{w}) = w_*$ noises

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

- It assumes that each output y_i that we observe can be explained as the prediction of an underlying model, $f(\mathbf{x}_i, \mathbf{w})$ plus a noise term ϵ_i. _why add?_

- For a fixed $\mathbf{x}$ and a fixed $\mathbf{w}$, $f(\mathbf{x}, \mathbf{w})$ is a constant, then

$\underline{x_1 \cdot w_1}$, or $\underline{w_o}$ or $\underline{x_o w_s}$ or ... $\underline{x_{n} w_{n}}$ constant are all

$$y = \text{constant} + \epsilon,$$

where ϵ is a continuous RV.

- What is the pdf for y? (we are adding a constant to a Gaussian RV)

Expectation $E\{y\} = E\{\text{constant} + \epsilon\} = \text{constant}$ = μ : the constant. Exp value is itself.

μ $E\{y\} =$

σ^2 $\text{var}\{y\} = \text{var}\{\text{constant}\} + \text{var}\{\epsilon\} = \sigma^2.$ $- \sigma^2$

Gaussian regression model (II)

Handwritten annotations (top right):
about the μ: mean, average, or Expected Value
1. for continuous R.V.: $\mathcal{E}(X) = \int_R x \cdot f(x) \, dx$
2. for discrete case: (finite) $\mathcal{E}(X) = \sum_{i=1}^{k} x_i \cdot p_i$
$\hookrightarrow$ special Case. Usually refer to average.
$\emptyset$. μ: expectation value | when $p_i = \frac{1}{n}$, $\mathcal{E}(x) = \frac{\sum x_i}{n}$

- This means that

$$y \sim \mathcal{N}(constant, \sigma^2),$$

  where we said constant was $f(\mathbf{x}, \mathbf{w})$, this is,

$$y \sim \mathcal{N}(f(\mathbf{x}, \mathbf{w}), \sigma^2).$$
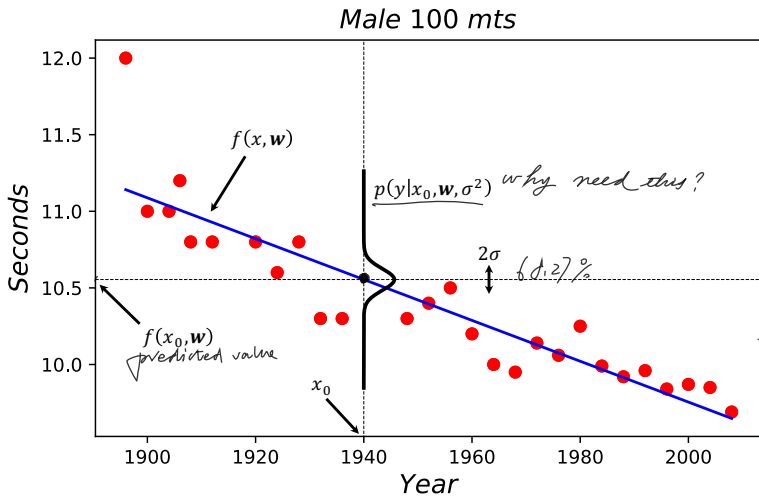
- Because we assumed that $\mathbf{x}$ and $\mathbf{w}$ are given, we can also write

$$p(y|\mathbf{x}, \mathbf{w}, \sigma^2) = \mathcal{N}(y|f(\mathbf{x}, \mathbf{w}), \sigma^2).$$

- If we knew the value for $\mathbf{w}$, once we have a new $\mathbf{x}_*$, we can predict the output as $f(\mathbf{x}_*, \mathbf{w})$.

- σ^2 tells us the noise variance.

Gaussian regression model (III)

Male 100 mts

How do we estimate **w**? (I)

- We start with a training dataset $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)$.

- We assume that the random variables $Y_1, \cdots, Y_N$ are *independent*, *they do not interfere with each other*

  $p(y_1/x_1, y_1/x_2, \cdots y_N/x_N)$

$$p(y_1, \cdots, y_N | \mathbf{x}_1, \cdots, \mathbf{x}_N) = p(y_1|\mathbf{x}_1) \cdots p(y_N|\mathbf{x}_N) = \prod_{n=1}^{N} p(y_n|\mathbf{x}_n).$$

  $p(y/x)$

- *why so huge this could from as/to a Normal distribution*

  We also assume that the RVs $Y_1, \cdots, Y_N$ follow an *identical* distribution, Gaussian in this case

  $\left[w_0, w_1, \cdots w_n \right] \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$

$$p(y_n|\mathbf{x}_n, \mathbf{w}, \sigma^2) = \mathcal{N}(y_n|f(\mathbf{x}_n, \mathbf{w}), \sigma^2) = \mathcal{N}(y_n|\mathbf{w}^\top \mathbf{x}_n, \sigma^2).$$

  $= w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$

  μ : the Expe Value

- Both assumptions go by the name of the *iid* assumption, *independent and identically distributed*.

How do we estimate **w**? (II)

(handwritten: $P(y/X, w, \sigma^2)$: we know y, X, σ^2: then how to get w?)

□ Putting both assumptions together, we get

(handwritten: $E y_n$; $\mu = (w_0 + w_1 x_1 + \cdots + w_n x_n)$)

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2) = \prod_{n=1}^{N} p(y_n|\underbrace{\mathbf{x}_n, \mathbf{w}}_{\mu}, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(y_n|\underbrace{\mathbf{w}^\top \mathbf{x}_n}_{\mu}, \sigma^2),$$

where $\mathbf{y} = [y_1, \cdots, y_N]^\top \in \mathbb{R}^{N \times 1}$ and $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times (D+1)}$.

(handwritten left of y: $\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$: N rows / 1 column)

(handwritten: what's the meaning)

□ The expression above can then be written as

(handwritten: here: $f(y) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1}{2}(\frac{\sigma-\mu}{\sigma})^2}$)

(handwritten right: $X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix}$ N rows, (D+1) columns in this case, D equal to 0.)

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(y_n|\mathbf{w}^\top \mathbf{x}_n, \sigma^2),$$

(handwritten: where: $\mu = w_0 + w_1 x_1 + \cdots + w_n x_n = f(x, w)$)

$$= \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(y_n - \mathbf{w}^\top \mathbf{x}_n)^2}{2\sigma^2} \right\}.$$

(handwritten: $\frac{1}{(2\pi\sigma^2)^{N \cdot \frac{1}{2}}} e$)

$$= \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{w}^\top \mathbf{x}_n)^2 \right\}.$$

How do we estimate **w**? (III)

❑ When we look at a Gaussian pdf, like

$$p(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(y-\mu)^2}{2\sigma^2} \right\},$$

we assume that both μ and σ^2 are given. In this case, the pdf follows all the properties we reviewed before.

❑ The same is true for

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2) = \prod_{n=1}^{N} p(y_n|\mathbf{x}_n, \mathbf{w}, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(y_n|\mathbf{w}^\top \mathbf{x}_n, \sigma^2).$$

Similar to μ

❑ Given $\mathbf{w}^\top \mathbf{x}_n$ and σ^2, then each $p(y_n|\mathbf{x}_n, \mathbf{w}, \sigma^2)$ is a pdf.

❑ A different approach would be to say: I have some data for $\{y_n\}_{n=1}^{N}$ and $\{\mathbf{x}_n\}_{n=1}^{N}$ but

– "I don't know what is $\mathbf{w}^\top$ (therefore I don't know what is $\mathbf{w}^\top \mathbf{x}_n$)"

– "I don't know what is σ^2".

How do we estimate **w**? (IV)

- With y_n and $\mathbf{x}_n$ given but with unknown values for $\mathbf{w}$ and σ^2, each $p(y_n|\mathbf{x}_n, \mathbf{w}, \sigma^2)$ is not a pdf anymore.
- In that case, the function

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(y_n|\mathbf{w}^\top \mathbf{x}_n, \sigma^2), \qquad 0 \leqslant p < 1$$

  probability

  receives the name of a *likelihood function*.
- We can think of a likelihood function as a function of the parameters $\mathbf{w}$ and σ^2,

$$g(\mathbf{w}, \sigma^2) = p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2),$$

- And subsequently, we can use *multivariate calculus* to find the values of $\mathbf{w}, \sigma^2$ that maximise $g(\mathbf{w}, \sigma^2)$.
- In statistics, this is known as the *maximum-likelihood* (ML) criterion to estimate parameters.

How do we estimate **w**? (V)

❑ Given **y**, **X**, we use the ML criterion to find the parameters **w** and σ^2 that maximise

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{w}^\top \mathbf{x}_n)^2 \right\}.$$

❑ In practice, we prefer to maximise the log of the likelihood $p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2)$,

$$\begin{aligned} LL(\mathbf{w}, \sigma^2) &= \log p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2) \\ &= -\frac{N}{2} \log(2\pi) - \frac{N}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{w}^\top \mathbf{x}_n)^2. \end{aligned}$$

❑ **Consistency of the ML criterion** If data was really generated according to the probability we specified, the correct parameters will be recovered in the limit as $N \to \infty$.

Connection with the sum of squared errors

❏ If we multiply $LL(\mathbf{w}, \sigma^2)$ by minus one, we get

$$E(\mathbf{w}, \sigma^2) = -\log p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2) \propto \sum_{n=1}^{N}(y_n - \mathbf{w}^\top \mathbf{x}_n)^2.$$

❏ The ML criterion for this model has a close connection with the sum-of-squared errors used in non-probabilitistic formulations of linear regression.

❏ Maximising the log-likelihood function is equivalent to minimising the sum-of-squares errors.

❏ Notice that the $\log$ is a monotonic function, meaning that if we find $\mathbf{w}, \sigma^2$ that maximise $g(\mathbf{w}, \sigma^2)$, those will also maximise $\log(g(\mathbf{w}, \sigma^2))$.

Normal equation (I)

□ Let us find an estimate for **w**.

□ From what we saw before,

$$LL(\mathbf{w}, \sigma^2) = -\frac{N}{2} \log(2\pi) - \frac{N}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{w}^\top \mathbf{x}_n)^2.$$

□ Using what we reviewed in the section on vector/matrix notation, it can be shown that this expression can be written in a vectorial form as

$$LL(\mathbf{w}, \sigma^2) = -\frac{N}{2} \log(2\pi) - \frac{N}{2} \log \sigma^2 - \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w})$$

□ Let us focus on the term $(\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w})$,

$$(\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{y}^\top \mathbf{y} - \mathbf{w}^\top \mathbf{X}^\top \mathbf{y} - \mathbf{y}^\top \mathbf{X}\mathbf{w} + \mathbf{w}^\top \mathbf{X}^\top \mathbf{X}\mathbf{w}$$

Normal equation (II)

❏ We can find the **w** that maximises $LL(\mathbf{w}, \sigma^2)$ by taking the gradient $\frac{dLL(\mathbf{w}, \sigma^2)}{d\mathbf{w}}$, equating to zero and solving for **w**.

❏ Taking the gradient of each term in $LL(\mathbf{w}, \sigma^2)$ wrt **w**, we get

$$\frac{d}{d\mathbf{w}}\left[-\frac{N}{2}\log(2\pi)\right] = 0, \quad \frac{d}{d\mathbf{w}}\left[-\frac{N}{2}\log\sigma^2\right] = 0, \quad \frac{d}{d\mathbf{w}}\left[-\frac{1}{2\sigma^2}\mathbf{y}^\top\mathbf{y}\right] = 0,$$

$$\frac{d}{d\mathbf{w}}\left[\frac{1}{2\sigma^2}\mathbf{w}^\top\mathbf{X}^\top\mathbf{y}\right] = \frac{1}{2\sigma^2}\mathbf{X}^\top\mathbf{y},$$

$$\frac{d}{d\mathbf{w}}\left[\frac{1}{2\sigma^2}\mathbf{y}^\top\mathbf{X}\mathbf{w}\right] = \frac{1}{2\sigma^2}\mathbf{X}^\top\mathbf{y}$$

$$\frac{d}{d\mathbf{w}}\left[-\frac{1}{2\sigma^2}\mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w}\right] = -\frac{1}{2\sigma^2}2\mathbf{X}^\top\mathbf{X}\mathbf{w}$$

Normal equation (III)

❏ Putting these terms together, we get

$$\frac{d}{d\mathbf{w}}LL(\mathbf{w}, \sigma^2) = \frac{1}{2\sigma^2}\mathbf{X}^\top\mathbf{y} + \frac{1}{2\sigma^2}\mathbf{X}^\top\mathbf{y} - \frac{1}{2\sigma^2}2\mathbf{X}^\top\mathbf{X}\mathbf{w}$$
$$= \frac{1}{\sigma^2}\mathbf{X}^\top\mathbf{y} - \frac{1}{\sigma^2}\mathbf{X}^\top\mathbf{X}\mathbf{w}$$

❏ Now, equating to zero and solving for **w**, we get

$$\frac{1}{\sigma^2}\mathbf{X}^\top\mathbf{y} - \frac{1}{\sigma^2}\mathbf{X}^\top\mathbf{X}\mathbf{w} = \mathbf{0}$$
$$\mathbf{X}^\top\mathbf{X}\mathbf{w} = \mathbf{X}^\top\mathbf{y}$$
$$\mathbf{w}_* = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y}.$$

❏ The expression for $\mathbf{w}_*$ is known as the *normal equation*.
❏ The solution for $\mathbf{w}^*$ exists if we can compute $\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}$.
❏ The inverse can be computed as long as $\mathbf{X}^\top\mathbf{X}$ is non-singular (e.g. determinant different from zero, or has full-rank).

Solving for σ_*^2

□ Following a similar procedure, it can be shown that the ML solution for σ_*^2 is given as

$$\sigma_*^2 = \frac{1}{N}(\mathbf{y} - \mathbf{X}\mathbf{w}_*)^\top (\mathbf{y} - \mathbf{X}\mathbf{w}_*).$$
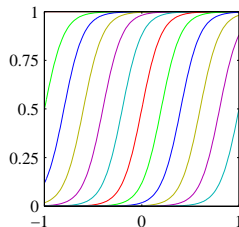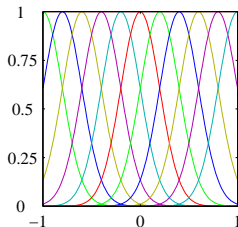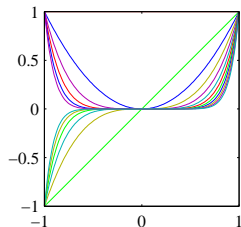
Basis functions

- The model that is linear in **x** only allows linear relationships between **x** and y.

- We can extend the model to describe non-linear relationships between the inputs and the output by using basis functions, non-linear mappings from inputs to outputs.

- However, we keep the linear relationship of y wrt **w** for tractability.

- The predictive model follows as $f(\mathbf{x}, \mathbf{w})$

$$f(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{M} w_i \phi_i(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}),$$

where $\phi_i(\mathbf{x})$ are basis functions and we have $M + 1$ parameters for the vector **w** and $\phi(\mathbf{x}) = [1, \phi_1(\mathbf{x}), \cdots, \phi_M(\mathbf{x})]^\top$.

Examples of basis functions

Polynomial: $\phi_i(x) = x^i$.

Exponential: $\phi_i(x) = \exp\left\{ -\frac{(x-\mu_i)^2}{2s^2} \right\}$

Sigmoidal: $\phi_i(x) = \sigma\left(\frac{x-\mu_i}{s}\right)$, $\sigma(a) = 1/(1 + \exp(-a))$.

Transforming the input using the basis functions

- As an example, let us use polynomial basis functions to predict y, the time in seconds in the 100 mt Olympics competition.

- For each x (year of the competition), we now compute the vector of polynomial basis functions

$$\phi(x) = \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ \vdots \\ x^M \end{bmatrix}$$

- We have converted the unidimensional input feature x into a higher dimensional feature representation $\phi(x) \in R^{M+1}$.

Normal equations with a design matrix

❏ Given **X**, we first compute a new design matrix Φ,

$$\Phi = \begin{bmatrix} \phi(\mathbf{x}_1)^\top \\ \phi(\mathbf{x}_2)^\top \\ \vdots \\ \phi(\mathbf{x}_N)^\top \end{bmatrix} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_M(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_M(\mathbf{x}_2) \\ \vdots & & & \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_M(\mathbf{x}_N) \end{bmatrix}$$

❏ We now can use $(\mathbf{y}, \Phi)$ and write the Gaussian linear regression problem
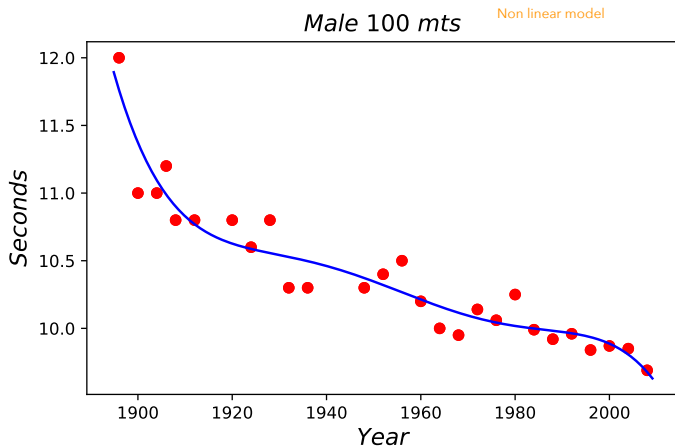
$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2) = \prod_{n=1}^N \mathcal{N}(y_n|\mathbf{w}^\top \phi_n, \sigma^2),$$

where $\phi_n = \phi(\mathbf{x}_n)$.

❏ Using the ML criterion, we arrive to the following normal equation

$$\mathbf{w}_* = \left(\Phi^\top \Phi\right)^{-1} \Phi^\top \mathbf{y}.$$

Olympic 100-mt data with $M = 5$

Alternative to find **w** different ways of finding w

- ❑ For solving the normal equation, we need to invert $\mathbf{X}^\top \mathbf{X}$.

- ❑ This inversion has a computational complexity between $\mathcal{O}((D+1)^{2.4})$ to $\mathcal{O}((D+1)^3)$ (depending on the implementation).

- ❑ The normal equation is linear regarding the number of instances in the training data, $\mathcal{O}(N)$.

- ❑ It can handle a large training set as long as it fits in memory.

- ❑ Alternatively, we can use iterative optimisation in cases with a large number of features and too many instances to fit in memory.

Contents

General problem

❑ We are given a function $h(\mathbf{w})$, where $\mathbf{w} \in \mathbb{R}^p$.

❑ Aim: to find a value for $\mathbf{w}$ that minimises $h(\mathbf{w})$.

❑ Use an iterative procedure

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \eta \mathbf{d}_k,$$

where $\mathbf{d}_k$ is known as the search direction and it is such that

$$h(\mathbf{w}_{k+1}) < h(\mathbf{w}_k).$$

❑ The parameter η is known as the **step size** or **learning rate**.

Gradient descent

❑ Perhaps, the simplest algorithm for unconstrained optimisation.

❑ It assumes that $\mathbf{d}_k = -\mathbf{g}_k$, where $\mathbf{g}_k = \mathbf{g}(\mathbf{w}_k)$.

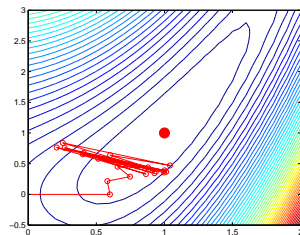❑ Also known as **steepest descent**.

❑ It can be written like

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta\mathbf{g}_k.$$

Step size

❑ The main issue in gradient descent is how to set the step size.

❑ If it is too small, convergence will be very slow. If it is too large, the method can fail to converge at all.
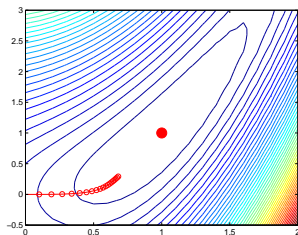
(a)

(b)

Figure: The function to optimise is $h(w_1, w_2) = 0.5(w_1^2 - w_2)^2 + 0.5(w_1 - 1)^2$. The minimum is at $(1, 1)$. In (a) $\eta = 0.1$. In (b) $\eta = 0.6$.

Alternatives to choose the step size η

- Line search methods (there are different alternatives).

- Line search methods may use search directions other than the steepest descent direction.

- Conjugate gradient (method of choice for quadratic objectives $g(\mathbf{w}) = \mathbf{w}^\top \mathbf{A}\mathbf{w}$).

- Use a *Newton* search direction.

Gradient descent for linear regression (I)

❏ For simplicity, let us assume that the objective function $h(\mathbf{w})$ corresponds to the mean squared error

$$E(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} (y_n - \mathbf{w}^\top \mathbf{x}_n)^2.$$

❏ We could also minimise the negative $LL(\mathbf{w})$ instead.

❏ We write the update equation as

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta \frac{d}{d\mathbf{w}} E(\mathbf{w}) \bigg|_{\mathbf{w}=\mathbf{w}_k}.$$

Gradient descent for linear regression (II)

❑ Computing the gradient for $E(\mathbf{w})$, we get

$$\frac{d}{d\mathbf{w}}E(\mathbf{w}) = \frac{2}{N}\sum_{n=1}^{N}\left(\mathbf{w}^{\top}\mathbf{x}_n - y_n\right)\mathbf{x}_n = \frac{2}{N}\mathbf{X}^{\top}\left(\mathbf{X}\mathbf{w} - \mathbf{y}\right).$$

❑ The update equation follows as

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta\frac{2}{N}\mathbf{X}^{\top}\left(\mathbf{X}\mathbf{w}_k - \mathbf{y}\right).$$

❑ The computation of the gradient involves using the whole dataset $(\mathbf{X}, \mathbf{y})$ at every step.

❑ For this reason, this algorithm is known as *batch gradient descent*.

Gradient descent and feature scaling

❑ Always normalise the features if using gradient descent.

❑ Gradient descent converges faster if all features have a similar scale.

❑ If the attributes are in very different scales, it may take a long time to converge.

Contents

Online learning and large datasets

- ❑ Traiditionally in machine learning, the gradient $\mathbf{g}_k$ is computed using the whole dataset $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$.

- ❑ There are settings, though, where only a subset of the data can be used.

- ❑ **Online learning**: the instances $(\mathbf{x}_n, y_n)$ appear one at a time.

- ❑ **Large datasets**: computing the exact value for $\mathbf{g}_k$ would be expensive, if not impossible.

Stochastic gradient descent (I)

- In stochastic gradient descent (SGD), the gradient $\mathbf{g}_k$ is computed using a subset of the instances available.

- update when new instances added
  The word stochastic refers to the fact that the value for $\mathbf{g}_k$ will depend on the subset of the instances chosen for computation.

Stochastic gradient descent (II)

❏ In the stochastic setting, a better estimate can be found if the gradient is computed using

$$\mathbf{g}_k = \frac{1}{|S|} \sum_{i \in S} \mathbf{g}_{k,i},$$

where $S \in \mathcal{D}$, $|S|$ is the cardinality of S, and $\mathbf{g}_{k,i}$ is the gradient at iteration k computed using the instance $(\mathbf{x}_i, y_i)$.

❏ This setting is called *mini-batch gradient descent*.

Step size in SGD

❑ Choosing the value of η is particularly important in SGD since there is no easy way to compute it.

❑ Usually the value of η will depend on the iteration k, η_k.

❑ It should follow the **Robbins-Monro** conditions

$$\sum_{k=1}^{\infty} \eta_k = \infty, \quad \sum_{k=1}^{\infty} \eta_k^2 < \infty.$$

❑ Various formulas for η_k can be used

$$\eta_k = \frac{1}{k}, \quad \eta_k = \frac{1}{(\tau_0 + k)^{\kappa}},$$

where τ_0 slows down early interations and $\kappa \in (0.5, 1]$.

Contents

What is regularisation?

- It refers to a technique used for preventing overfitting in a predictive model.

- It consists in adding a term (a regulariser) to the objective function that encourages simpler solutions.

- With regularisation, the objective function for linear regression would be

$$h(\mathbf{w}) = E(\mathbf{w}) + \lambda R(\mathbf{w}),$$

  where $R(\mathbf{w})$ is the regularisation term and λ the regularisation parameter.

- In the expression for $h(\mathbf{w})$, we can use the negative $LL(\mathbf{w})$ instead of $E(\mathbf{w})$.

- If $\lambda = 0$, we get $h(\mathbf{w}) = E(\mathbf{w})$.

Different types of regularisation

❏ The objective function for linear regression would be

$$h(\mathbf{w}) = E(\mathbf{w}) + \lambda R(\mathbf{w}),$$

where $R(\mathbf{w})$ follows as

$$R(\mathbf{w}) = \alpha \|\mathbf{w}\|_1 + (1 - \alpha) \frac{1}{2} \|\mathbf{w}\|_2^2,$$

where $\|\mathbf{w}\|_1 = \sum_{m=1}^{p} |w_m|$, and $\|\mathbf{w}\|_2^2 = \sum_{m=1}^{p} w_m^2$.

lawso relularisation

❏ If $\alpha = 1$, we get ℓ_1 regularisation.

❏ If $\alpha = 0$, we get ℓ_2 regularisation. Rich regression

❏ If $0 < \alpha < 1$, we get the elastic net regularisation.

Ridge regression or ℓ_2 regularisation

- In ridge regression, $\alpha = 0$,

$$h(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} (y_n - \mathbf{w}^\top \mathbf{x}_n)^2 + \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w},$$

- It can be shown that an optimal solution for $\mathbf{w}_*$ is given as

$$\mathbf{w}_* = \left(\mathbf{X}^\top \mathbf{X} + \frac{\lambda N}{2} \mathbf{I} \right)^{-1} \mathbf{X}^\top \mathbf{y}.$$

- Notice that we can also use iterative procedure for optimising $h(\mathbf{w})$ either through batch gradient decent, SGD or mini-batch SGD.