machine learning

→ reinforcement learning

· driving the agent in the 'right' direction

e.g. chess or go engine

supervised

input: data + "labels"

examples: linear regression
decision trees
neural networks
...

unsupervised

input: data

· clustering

· dimensionality reduction

labels are categorical or continuous
'classification' 'regression methods'

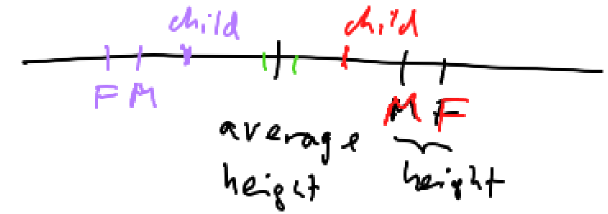
Data is usually split into 2-3 sets:

training data — test data — (verification data)

e.g. 80% 20%

Linear regression

sir Galton regression



$f: \mathbb{R}^m \to \mathbb{R}$

model: linear (affine)

$x_i = (x_{i1}, x_{i2}, \dots, x_{im})$

i.e.

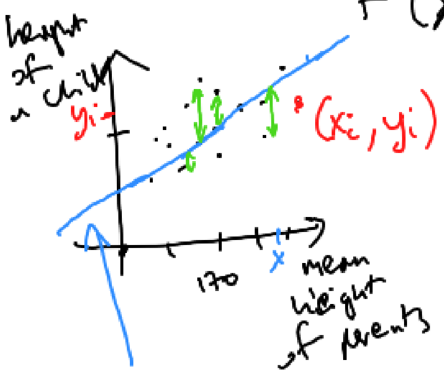
$$f(\underset{\in \mathbb{R}^m}{x_i}) = \vartheta_0 + \vartheta_1 x_{i1} + \dots + \vartheta_m x_{im} = \quad , \vartheta_j \in \mathbb{R} \text{ fixed} \quad , x \in \mathbb{R}^m$$

$$= \underbrace{\begin{bmatrix} 1 & x_{1i} & \dots & x_{mi} \end{bmatrix}}_{x_i^T} \cdot \begin{bmatrix} \vartheta_0 \\ \vartheta_1 \\ \vdots \\ \vartheta_m \end{bmatrix}$$

Convention: $[a] = a$

$$f(\cancel{X}) = \begin{bmatrix} 1 & \\ 1 & \\ 1 & X \\ \vdots & \\ 1 & \end{bmatrix} \begin{bmatrix} \vartheta_0 \\ \vartheta_1 \\ \vdots \\ \vartheta_m \end{bmatrix} = \begin{bmatrix} \vartheta_0 + \vartheta_1 x_{11} + \dots + \vartheta_m x_{1m} \\ \vdots \\ \vartheta_0 + \vartheta_1 x_{n1} + \dots + \vartheta_m x_{nm} \end{bmatrix}$$

$$\cancel{X} = \begin{bmatrix} x_{11} & \dots & x_{1n} \\ \vdots & & \\ x_{n1} & \dots & x_{nm} \end{bmatrix}$$



$f(x) = \vartheta_1 x + \vartheta_0$

How to choose the coefficients ϑ?

The most approach is to minimise the following loss function:

$$L(\underset{\substack{\uparrow \\ \text{actual}}}{y},\underset{\substack{\uparrow \\ \text{predicted}}}{\hat{y}}) = \sum_{i=1}^{n} (y_i - \underbrace{\hat{y}_i}_{= x_i^T \cdot \vartheta})^2 = \sum_{i=1}^{n} (y_i - x_i^T \vartheta)^2 = (y - x^T\vartheta)^T (y - x^T \vartheta)$$

X, y – given $\longrightarrow$ we want to find ϑ

$X \in \mathbb{R}^{n\times(m+1)}$, $y \in \mathbb{R}^n$

$$X = \begin{bmatrix} 1 & 163 & 179 \\ 1 & 174 & 188 \\ 1 & 141 & 155 \\ 1 & 156 & 152 \\ 1 & 180 & 178 \end{bmatrix}$$

$m = 2,\ n = 5$

If ~~...~~ $\bar{X}$ is of full rank (rank $\bar{X} = \min(n, m+1)$), then there is a formula for the optimal ϑ (i.e. the minimiser of L):

$$\vartheta^T = \underbrace{(X^T \cdot X)^{-1}}_{(m+1)\times(m+1)} \underbrace{\underbrace{X^T}_{(m+1)\times n} \underbrace{y}_{n}}_{(m+1)\times 1} \longrightarrow (m+1)\times 1$$

the formula for the 1st assignment

$X^T \sim (m+1)\times n$ $\qquad$ $X^T X \sim (m+1)\times(m+1)$

$X \sim n \times (m+1)$

Another approach:

$\lambda \geq 0$ is fixed

$\lambda = 0$ just linear regression

$$L_\vartheta^{\text{ridge}}(y, \hat{y}) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{i=1}^{m} \vartheta_i^2$$

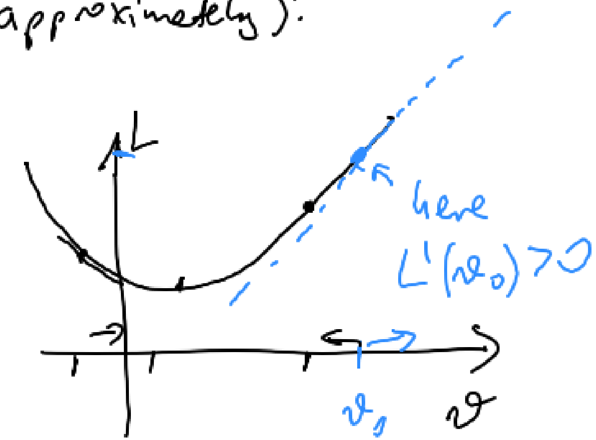
note that ϑ_0 is missing

Find ϑ to minimise that

'Ridge regression'

There is also an explicit formula for the minimiser in this setting.

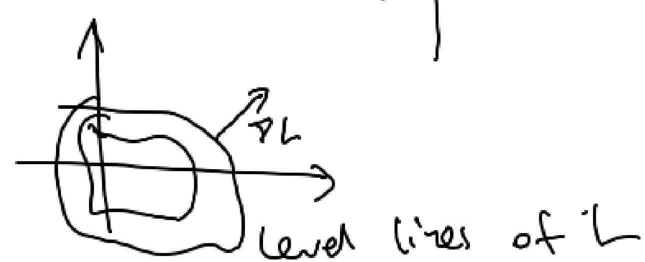
But we will use another method to find ϑ (approximately):



- Pick some initial $\vartheta^{(0)}$
- Given $\vartheta^{(n)}$ $(n = 0, \ldots)$

pick $\vartheta^{(n+1)} := \vartheta^{(n)} - c \cdot \nabla_\vartheta L$

small constant



$$L = \sum_{i=1}^{n} \left(y_i - (\vartheta_0 + \vartheta_1 x_{i1} + \dots + \vartheta_m x_{im}) \right)^2 + \lambda \left(\cancel{\vartheta_0^2} + \vartheta_1^2 + \dots + \vartheta_m^2 \right)$$

$$\nabla_\vartheta L = \left(\frac{\partial L}{\partial \vartheta_0}, \frac{\partial L}{\partial \vartheta_1}, \dots, \frac{\partial L}{\partial \vartheta_m} \right)$$

$(y^2)' = 2y$

e.g. $$\frac{\partial L}{\partial \vartheta_0} = 2 \sum_{i=1}^{n} \left(y_i - (\vartheta_0 + \dots + \vartheta_m x_{im}) \right) \cdot (-1)$$

$$\frac{\partial L}{\partial \vartheta_j} = 2 \sum_{i=1}^{n} \left(\text{— ''} \text{————} \right) \cdot (-x_{ij}) + 2 \lambda \vartheta_j$$