Machine Learning

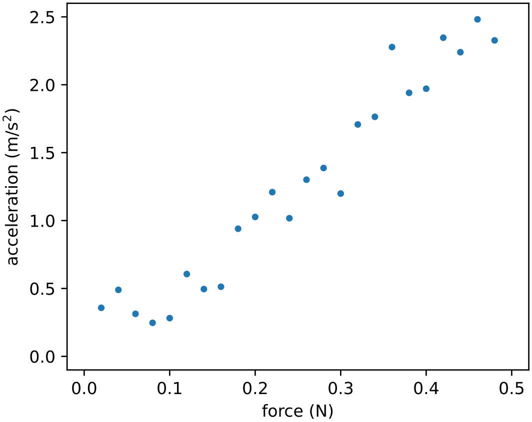
Linear Regression

Hiroshi Shimodaira and Hao Tang

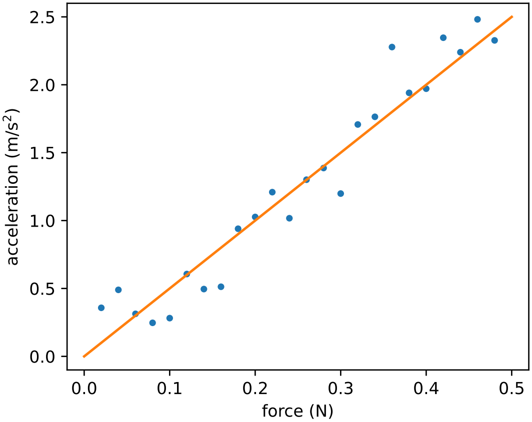
29 January 2024

Ver. 1.1

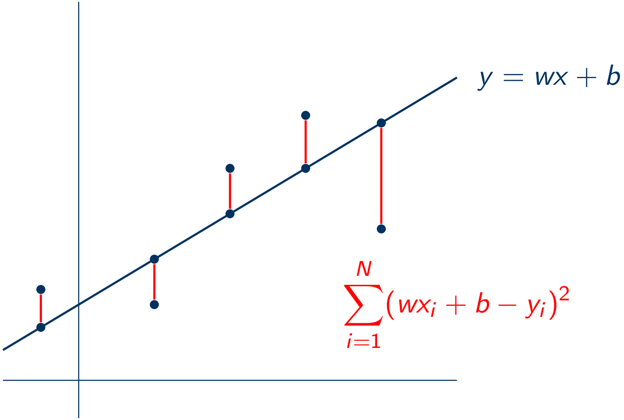
First example



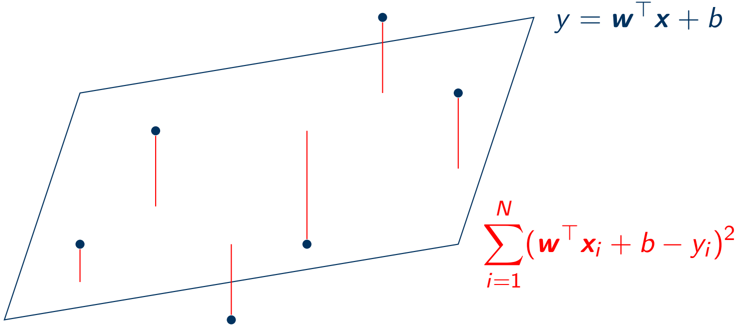
First example



Geometry of linear regression



Geometry of linear regression (*cont.*)



Linear regression

- $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$: data set
  - $\boldsymbol{x}_i = \begin{bmatrix} x_{i1} & \cdots & x_{id} \end{bmatrix}^\top$: input, features, independent variables
  - $y_i \in \mathbb{R}$: target/dependent variable, ground truth, for $\boldsymbol{x}_i$.
- $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$: linear predictor, hyperplane
  - $\boldsymbol{w} = \begin{bmatrix} w_1 & \cdots & w_d \end{bmatrix}^\top$: weights
  - $b \in \mathbb{R}$: bias
  - $\{\boldsymbol{w}, b\}$: parameters $\cdots$ $\boldsymbol{\theta} = [b \; \boldsymbol{w}^\top]^\top$

Linear regression

- Given $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$, find $\boldsymbol{\theta}$ such that the mean-squared error (MSE)

$$L = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2 \tag{1}$$

is minimised.

- The act of finding $\boldsymbol{w}$ is called training.

- c.f. "least squares" – a parameter estimation method based on MSE or minimising the sum of squares of errors/residuals.

Linear regression: training with MSE

- The goal of linear regression is to solve

$$\min_{\boldsymbol{w},b} \quad \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2. \tag{2}$$

- The optimal solution satisfies

$$\frac{\partial L}{\partial b} = 0\,, \qquad \frac{\partial L}{\partial \boldsymbol{w}} = \begin{bmatrix} \frac{\partial L}{\partial w_1} & \frac{\partial L}{\partial w_2} & \cdots & \frac{\partial L}{\partial w_d} \end{bmatrix} = \mathbf{0}. \tag{3}$$

(Is this global optimal? More on this in lectures on optimisation.)

Linear regression: finding the bias b

$$\frac{\partial}{\partial b}\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2 = \frac{2}{N}\sum_{i=1}^{N}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i) \tag{4}$$

$$= 2b - \frac{2}{N}\sum_{i=1}^{N}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i) = 0 \tag{5}$$

$$b = \frac{1}{N}\sum_{i=1}^{N}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i) = \frac{1}{N}\sum_{i=1}^{N} y_i - \boldsymbol{w}^\top \left(\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i\right) = \bar{y} - \boldsymbol{w}^\top \bar{\boldsymbol{x}} \tag{6}$$

Linear regression: data centring (mean normalisation)

$$\frac{\partial L}{\partial b} = 0 \implies b = \bar{y} - \boldsymbol{w}^\top \bar{\boldsymbol{x}} \tag{7}$$

$$L = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i)^2 = \frac{1}{N}\sum_{i=1}^{N}[\boldsymbol{w}^\top(\boldsymbol{x}_i - \bar{x}) - (y_i - \bar{y})]^2 \tag{8}$$

$$= \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{w}^\top \dot{\boldsymbol{x}}_i - \dot{y}_i)^2 \tag{9}$$

where $\dot{\boldsymbol{x}}_i = \boldsymbol{x}_i - \bar{\boldsymbol{x}}, \quad \dot{y}_i = y_i - \bar{y}$

Linear regression: finding the weights w

$$\frac{\partial}{\partial \boldsymbol{w}} \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{w}^\top \dot{\boldsymbol{x}}_i - \dot{y}_i)^2 = \frac{2}{N} \sum_{i=1}^{N} (\boldsymbol{w}^\top \dot{\boldsymbol{x}}_i - \dot{y}_i)(\dot{\boldsymbol{x}}_i) \tag{10}$$

$$= \frac{2}{N} \sum_{i=1}^{N} ((\boldsymbol{w}^\top \dot{\boldsymbol{x}}_i)\dot{\boldsymbol{x}}_i - \dot{y}_i \dot{\boldsymbol{x}}_i) \tag{11}$$

Linear regression: finding the weights w (*cont.*)

$$\frac{\partial}{\partial \boldsymbol{w}} \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{w}^\top \dot{\boldsymbol{x}}_i - \dot{y}_i)^2 = \frac{2}{N} \sum_{i=1}^{N} ((\boldsymbol{w}^\top \dot{\boldsymbol{x}}_i)\dot{\boldsymbol{x}}_i - \dot{y}_i \dot{\boldsymbol{x}}_i) \tag{12}$$

$$= \frac{2}{N} \left(\begin{bmatrix} \dot{\boldsymbol{x}}_1 & \dot{\boldsymbol{x}}_2 & \cdots & \dot{\boldsymbol{x}}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{w}^\top \dot{\boldsymbol{x}}_1 \\ \boldsymbol{w}^\top \dot{\boldsymbol{x}}_2 \\ \vdots \\ \boldsymbol{w}^\top \dot{\boldsymbol{x}}_N \end{bmatrix} - \begin{bmatrix} \dot{\boldsymbol{x}}_1 & \dot{\boldsymbol{x}}_2 & \cdots & \dot{\boldsymbol{x}}_N \end{bmatrix} \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \vdots \\ \dot{y}_N \end{bmatrix} \right) \tag{13}$$

$$= \frac{2}{N} \left(\mathbf{X}\mathbf{X}^\top \boldsymbol{w} - \mathbf{X}\dot{\boldsymbol{y}} \right) = \mathbf{0} \tag{14}$$

$$\longrightarrow \quad \boldsymbol{w} = \left(\mathbf{X}\mathbf{X}^\top \right)^{-1} \mathbf{X}\dot{\boldsymbol{y}} \tag{15}$$

NB: the definition of $\mathbf{X}$ (which is a $d \times N$ matrix) here is different from the one in the textbook LWLS.

Linear regression - training process

1. Centring

$$\dot{\boldsymbol{y}} = \begin{bmatrix} y_1 - \bar{y} \\ \vdots \\ y_N - \bar{y} \end{bmatrix}, \qquad \mathbf{X} = \begin{bmatrix} \boldsymbol{x}_1 - \bar{\boldsymbol{x}} & \cdots & \boldsymbol{x}_N - \bar{\boldsymbol{x}} \end{bmatrix} \tag{16}$$

2. Computing the weights $\boldsymbol{w}$ and b

$$\boldsymbol{w} = (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\dot{\boldsymbol{y}} \tag{17}$$

$$b = \bar{\boldsymbol{y}} - \boldsymbol{w}^\top \bar{\boldsymbol{x}} \tag{18}$$

NB: $(\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}$ is called a Moore-Penrose pseudoinverse of $\mathbf{X}$.
In practice, we find the solution $\boldsymbol{w}$ without calculating $(\mathbf{X}\mathbf{X}^\top)^{-1}$

What is $\mathbf{X}\mathbf{X}^\top$?

- $\mathbf{X} = \left[\boldsymbol{x}_1 - \bar{\boldsymbol{x}}, \ldots, \boldsymbol{x}_N - \bar{\boldsymbol{x}}\right]$
- $\mathbf{X}\mathbf{X}^\top$ is a $d \times d$ symmetric matrix
- $\mathbf{X}\mathbf{X}^\top$ is positive semi-definite, i.e. $\boldsymbol{x}^\top(\mathbf{X}\mathbf{X}^\top)\boldsymbol{x} \geq 0$ for any $\boldsymbol{x} \in \mathbb{R}^d$

  NB: Eigen values of a positive semi-definite matrix are non-negative, i.e. $\lambda_i \geq 0$ for $i = 1, \ldots, d$
- $C = \frac{1}{N}\mathbf{X}\mathbf{X}^\top$ is called a **covariance matrix**
  - $C = (\sigma_{ij})$: σ_{ii} is the (population) variance of i-th dimension of data, σ_{ij} is the covariance between i-th and j-th dimensions of data.
  - used in many areas, e.g. multivariate normal distributions, principal component analysis (PCA)
- $\det(C) = \prod_{i=1}^{d} \lambda_i$ and $tr(C) = \sum_{i=1}^{d} \lambda_i$, where λ_i is the i-the eigen value of C
- $\det(C) = 0$ and $\mathrm{rk}(C) < d$ if $N \leq d$

Features

$$y = \boldsymbol{w}^\top \boldsymbol{x} + b = \begin{bmatrix} \boldsymbol{w}^\top & b \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} = \boldsymbol{w}'^\top \boldsymbol{x}' \tag{19}$$

- Fitting $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$ is equivalent to appending 1 to $\boldsymbol{x}$ and fitting $f(\boldsymbol{x}') = \boldsymbol{w}'^\top \boldsymbol{x}'$.
- The 1 can be seen as a feature independent of the input.

Features

- Suppose we have a data point $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^\top$.
- The data point after appending 1 becomes

$$\begin{bmatrix} 1 & x_1 & x_2 & x_3 \end{bmatrix}^\top \tag{20}$$

- The data point after appending 1 and quadratic terms becomes

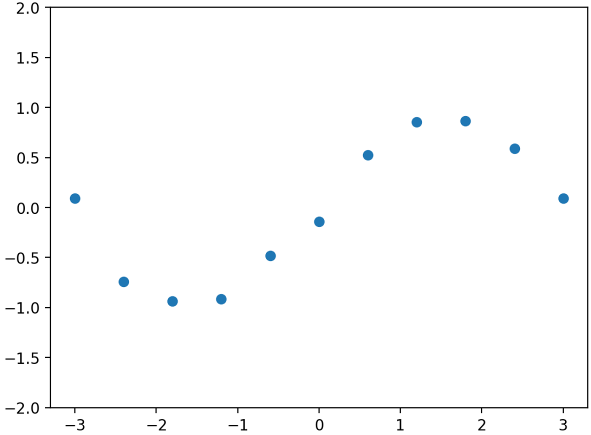
$$\phi(\boldsymbol{x}) = \begin{bmatrix} 1 & x_1 & x_2 & x_3 & x_1 x_2 & x_2 x_3 & x_1 x_3 & x_1^2 & x_2^2 & x_3^2 \end{bmatrix}^\top \tag{21}$$

- The function $f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x})$ is a polynomial.

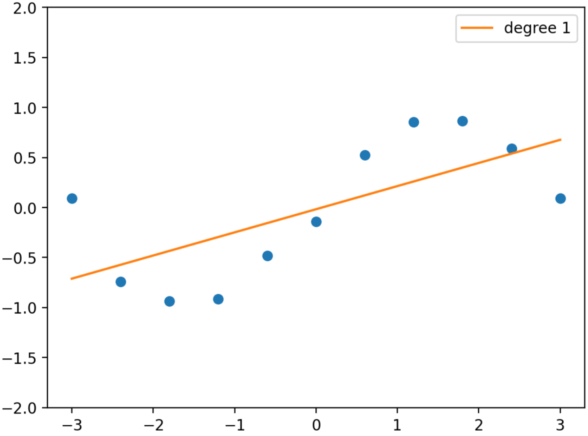
Linear regression with feature transformation

- We call ϕ a feature function.
- In general, ϕ can be any function.
- Instead of $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$, we now have $f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x})$.
- Instead of $\mathbf{X} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_N \end{bmatrix}$, we have $\Phi = \begin{bmatrix} \phi(\boldsymbol{x}_1) & \phi(\boldsymbol{x}_2) & \cdots & \phi(\boldsymbol{x}_N) \end{bmatrix}$
- The optimal solution for linear regression becomes $\boldsymbol{w} = (\Phi\Phi^\top)^{-1}\Phi\boldsymbol{y}$.

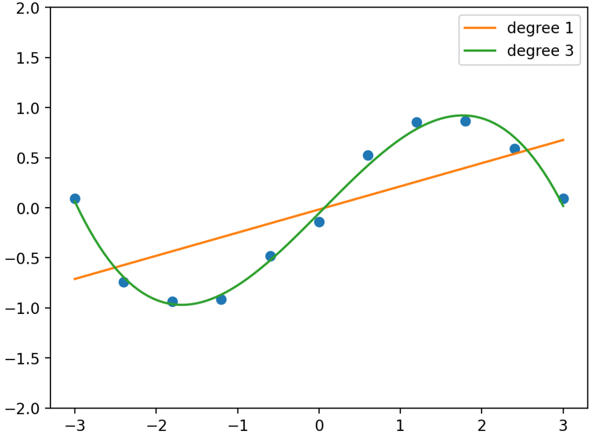
Examples



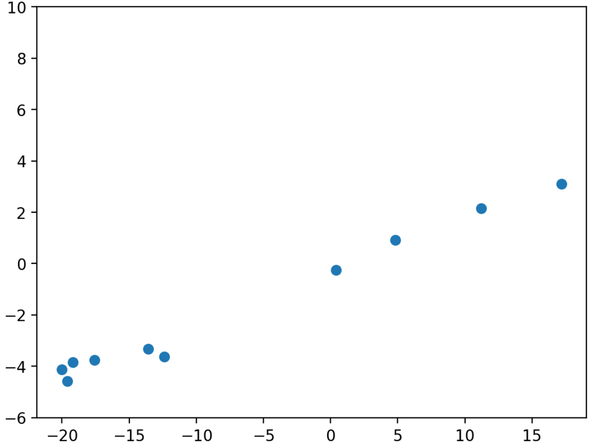
Examples



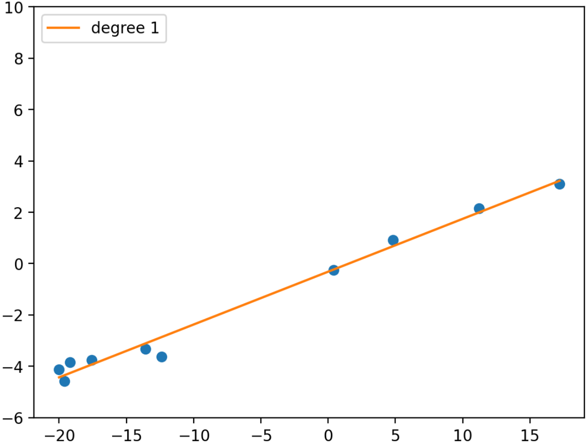
Examples



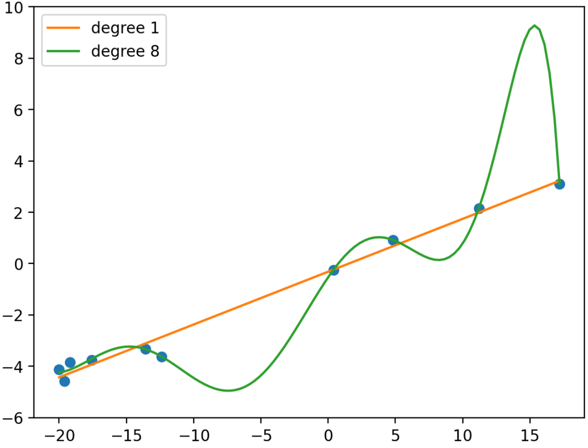
Examples



Examples



Examples



Linear regression

- A "linear" regression model is linear in the parameters $\boldsymbol{w}$ (i.e. linear combination between the parameters and features), **not** the features.

- A linear regression model can fit an arbitrary nonlinear function.

- What are the "right" features?

- What does it mean for the program $\boldsymbol{w}^\top \phi(\boldsymbol{x})$ we write with data to be "correct"? (Is it right to use a complex nonlinear transformation $\phi(\boldsymbol{x})$?)

A probabilistic interpretation

- Assume we cannot get a perfect fit because of noise.
- In particular, we assume the noise is additive and Gaussian.
- In other words, $y_i = \boldsymbol{w}^\top \phi(\boldsymbol{x}_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, 1)$.
- If $\epsilon_i \sim \mathcal{N}(0, 1)$, then $y_i \sim \mathcal{N}(\boldsymbol{w}^\top \phi(\boldsymbol{x}_i), 1)$.
- The log-likelihood of $\boldsymbol{w}$ is

$$\log \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(y_i - \boldsymbol{w}^\top \phi(\boldsymbol{x}_i))^2\right) \tag{22}$$

A probabilistic interpretation

- Log-likelihood of $\boldsymbol{w}$

$$\sum_{i=1}^{N} \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} (y_i - \boldsymbol{w}^\top \phi(\boldsymbol{x}_i))^2 \right] \tag{23}$$

- Mean-squared error

$$\frac{1}{N} \sum_{i=1}^{N} (y_i - \boldsymbol{w}^\top \phi(\boldsymbol{x}_i))^2 \tag{24}$$

- The maximum likelihood estimator is the optimal solution for MSE.

Practical issues

- The complexity of computing $(\Phi\Phi^\top)\Phi\boldsymbol{y}$ is $O(N^3)$, where N is the number of samples.
- The runtime is not particularly suitable for large data sets.
- Instead of solving $\min_w L$ exactly, could we find an approximate solution?
- In exchange, could we have an algorithm that scales better than $O(N^3)$?
- Not all problems have closed-form solutions for $\frac{\partial L}{\partial \boldsymbol{w}}$ anyways.
- What if there are outliers?

Linear regression

- We write a program $f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x})$ with $\boldsymbol{w} = (\Phi\Phi^\top)^{-1}\Phi\boldsymbol{y}$.
- In what sense is this program "correct"?

Linear regression using matrix calculus

- The mean-squared error can be written compactly as

$$L = \|\Phi^\top \boldsymbol{w} - \boldsymbol{y}\|_2^2. \tag{25}$$

- We can expand the mean-squared error as

$$L = \|\Phi^\top \boldsymbol{w} - \boldsymbol{y}\|_2^2 = (\Phi^\top \boldsymbol{w} - \boldsymbol{y})^\top(\Phi^\top \boldsymbol{w} - \boldsymbol{y}) = \boldsymbol{w}^\top \Phi\Phi^\top \boldsymbol{w} - 2\boldsymbol{y}^\top \Phi^\top \boldsymbol{w} + \boldsymbol{y}^\top \boldsymbol{y}. \tag{26}$$

- Solving the optimal solution gives

$$\frac{\partial L}{\partial \boldsymbol{w}}^\top = (\Phi\Phi^\top + (\Phi\Phi^\top)^\top)w - 2\Phi\boldsymbol{y} = \boldsymbol{0} \implies \boldsymbol{w} = (\Phi\Phi^\top)^{-1}\Phi\boldsymbol{y}. \tag{27}$$

Topics not covered

- Choices of features $\boldsymbol{x}$ (feature selection)
- Interpretations of the model parameters $\boldsymbol{\theta}$
- Collinearity
- Heteroscedasticity
- Other linear regression models (e.g. ridge regression, LASSO, Bayesian linear regression)
- Multiple linear regression
- Relationships with neural networks
- Relationships with principal component analysis (PCA)

Quizzes

1. What is the number of dimensions of the hyperplane formed by linear regression?
2. Give detailed derivations for Eqs. (13) and (14).
3. Show that $\mathbf{X}\mathbf{X}^\top$ is positive semi-definite.
4. Using $\boldsymbol{x}' = (1, \boldsymbol{x}^\top)^\top$ instead of $\boldsymbol{x}$, rewrite Eqs. $(1), \ldots, (15)$.