Workshop 3

COMP90051 Statistical Machine Learning

Semester 2, 2024

Learning outcomes

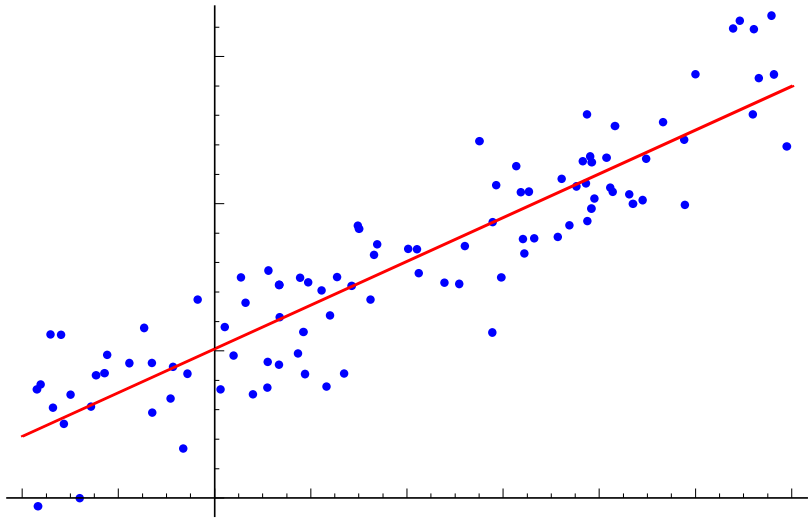
At the end of this workshop you should:

- be able to implement linear regression and logistic regression
- be able to explain how the optimisation problems for linear regression and logistic regression differ
- be able to implement gradient descent

Linear regression

Assume the response y is a *linear* function of the features $\mathbf{x} = [x_1, \ldots, x_m]^{\mathrm{T}}$:

$$y = w_0 + \sum_{i=1}^{m} w_i \cdot x_i$$



Write this more compactly as $y = \mathbf{x}^{\mathrm{T}}\mathbf{w}$ by redefining $\mathbf{x} = [x_0, x_1, \ldots, x_m]^{\mathrm{T}}$ with $x_0 = 1$ and defining $\mathbf{w} = [w_0, \ldots, w_m]^{\mathrm{T}}$

If we encodes noise: $y = \mathbf{x}^{\mathrm{T}}\mathbf{w} + \boldsymbol{\varepsilon}_{noise}$

Question: How do we choose the weights?

Solving linear regression

Decision theoretic view

Make decision that minimises the empirical risk

$$\hat{R} = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i)$$

and choose the square loss $L(y, \hat{y}) = (\hat{y} - y)^2$.

Optimal decision for $\mathbf{w}$ minimises the sum-squared error.

Probabilistic view

Assume

$$y|\mathbf{x}, \mathbf{w} \sim \mathcal{N}(\mathbf{x}^{\mathrm{T}}\mathbf{w}; \sigma^2)$$

Can write down the likelihood for the observations

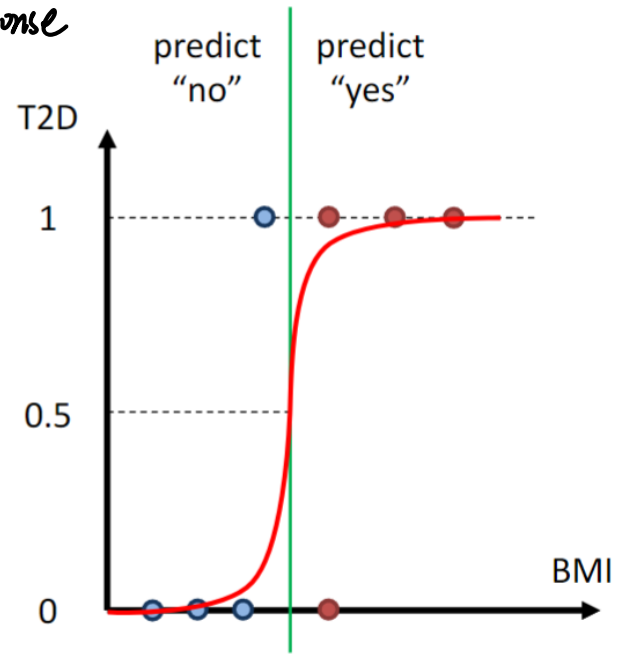
$$L(w|\boldsymbol{X}, \boldsymbol{Y}) = \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \mathbf{w}, \sigma)$$

MLE for $\mathbf{w}$ minimises the sum-squared error.

Logistic regression

- Logistic regression is a **linear binary (could be extend to multi-class) classifier** for **classification** task
- Linear regression: gives a continuous value of **output y** for a given input X.
- Logistic regression: gives a continuous value of **P(Y=1)** for a given input X, which is later converted to Y=0 or Y=1 based on a threshold value.

not continous response variable =7



Solving logistic regression

Logistic regression optimisation problem:

$$\mathbf{w}^\star \in \arg\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mu_i)$$

where $\mu_i = \frac{1}{1+\mathrm{e}^{-\mathbf{x}_i^\top \mathbf{w}}}$ and $\ell(y, \mu) = -y \log \mu - (1-y)\log(1-\mu)$

loss entropy

Unfortunately, no closed form solution, need to use optimization techniques

Gradient Descent: easy to compute slow to converge

IRLS: quick to convergence

Least square 的 optimal weights:

$$(y-Xw)^T(y-Xw)$$

$$= y^Ty - y^TXw - \quad -$$

$\nabla_w w^Tx = \nabla_w x^Tw = x^T$

$\nabla_w Aw = A$

$\nabla_w w^TAw = w^T(A^T+A)$

$$\frac{\partial L}{\partial w} =$$

$X: n \times p$

$w: p \times 1$

$$(Xw-y)^T(Xw-y)$$

$$= w^TX^TXw - w^TX^Ty - y^TXw - y^Ty$$

$$\nabla_w(R(w)) = \underbrace{w^T(X^TX + X^TX)}_{\text{二次型}} - y^TX - y^TX \Rightarrow$$

$$2W^TX^TX = 2X^Ty$$

$$w^T = y^TX(X^TX)^{-1} \quad \text{inversible}$$

$$w^* = (X^TX)^{-1}X^Ty$$

$\frac{1}{n}X^T(M-y)$ → logistic 的梯度 → update the weight at the opposite direction

$$W_t = W_{t-1} - \underbrace{\lambda}_{\text{步长}} \nabla_w R(W_{t-1})$$

Worksheet 3