Workshop 3

COMP90051 Machine Learning

Semester 2, 2018

Learning Outcomes

At the end of this workshop you should:

1. Be familiar with basic NumPy functionality
2. Be able to implement regularised logistic regression using a numerical optimisation solver
3. Be able to apply basis expansion to turn logistic regression into a non-linear classifier
4. Understand the purpose/effect of L2 regularisation

NumPy Basics

Worksheet 3

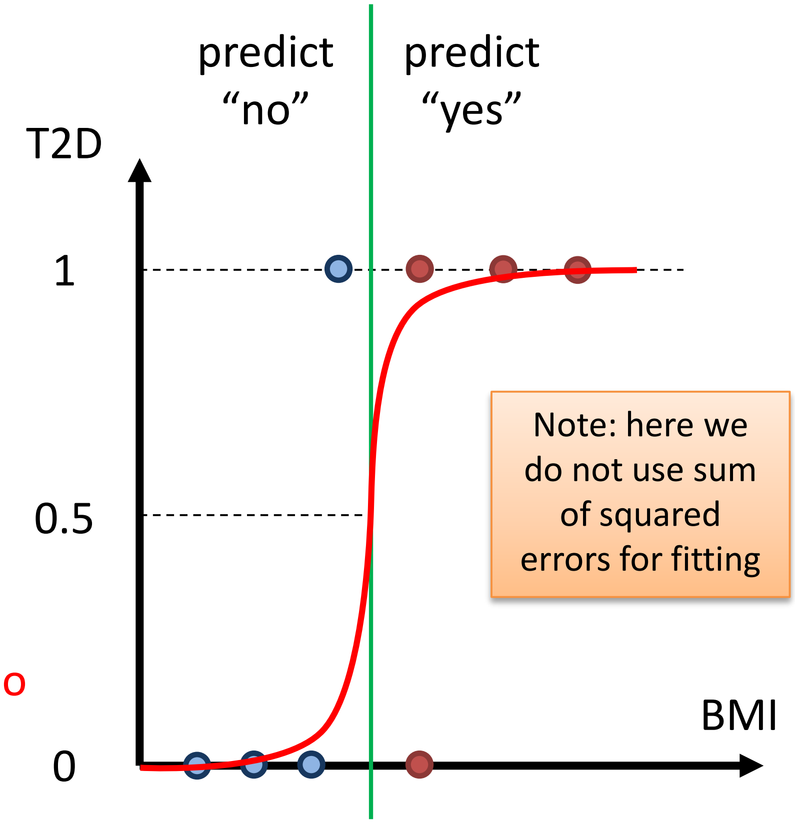
Logistic regression model

- Probabilistic approach to classification
  - $P(Y=1|\boldsymbol{x}) = f(\boldsymbol{x}) = ?$
  - Use a linear function? E.g., $s(\boldsymbol{x}) = \boldsymbol{x}'\boldsymbol{w}$
- Problem: the probability needs to be between 0 and 1.
- Logistic function $f(s) = \frac{1}{1+\exp(-s)}$
- Logistic regression model

$$P(Y=1|\boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{x}'\boldsymbol{w})}$$

- Equivalent to linear model for log-odds ratio

$$\log\frac{P(Y=1|\boldsymbol{x})}{P(Y=0|\boldsymbol{x})} = \boldsymbol{x}'\boldsymbol{w}$$



Logistic regression is a linear classifier

- Logistic regression model:

$$P(Y = 1|\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{x}'\boldsymbol{w})}$$

- Classification rule:

if $\left(P(Y = 1|\boldsymbol{x}) > \frac{1}{2}\right)$ then class “1”, else class “0”

- Decision boundary $\frac{1}{1+\exp(-\boldsymbol{x}'\boldsymbol{w})} = \frac{1}{2}$

Linear vs. logistic probabilistic models

- Linear regression assumes a Normal distribution with a fixed variance and mean given by linear model
$$p(y|\boldsymbol{x}) = Normal(\boldsymbol{x}'\boldsymbol{w}, \sigma^2)$$
- Logistic regression assumes a Bernoulli distribution with parameter given by logistic transform of linear model
$$p(y|\boldsymbol{x}) = Bernoulli(\text{logistic}(\boldsymbol{x}'\boldsymbol{w}))$$
- Recall that Bernoulli distribution is defined as
$$p(1) = \theta \text{ and } p(0) = 1 - \theta \text{ for } \theta \in [0,1]$$
- Equivalently $p(y) = \theta^y (1-\theta)^{(1-y)}$ for $y \in \{0,1\}$

Training as Max Likelihood Estimation

- Assuming independence, probability of data

$$p(y_1, \dots, y_n | \boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \prod_{i=1}^{n} p(y_i | \boldsymbol{x}_i)$$

- Assuming Bernoulli distribution we have

$$p(y_i | \boldsymbol{x}_i) = \left(\theta(\boldsymbol{x}_i)\right)^{y_i} \left(1 - \theta(\boldsymbol{x}_i)\right)^{1-y_i}$$

$$\text{where } \theta(\boldsymbol{x}_i) = \frac{1}{1+\exp(-\boldsymbol{x}_i{}'\boldsymbol{w})}$$

- Training: maximise this expression wrt weights $\boldsymbol{w}$

Apply log trick, simplify

- Instead of maximising likelihood, maximise its logarithm

$$\log\left(\prod_{i=1}^{n} p(y_i|\boldsymbol{x_i})\right) = \sum_{i=1}^{n} \log p(y_i|\boldsymbol{x_i})$$

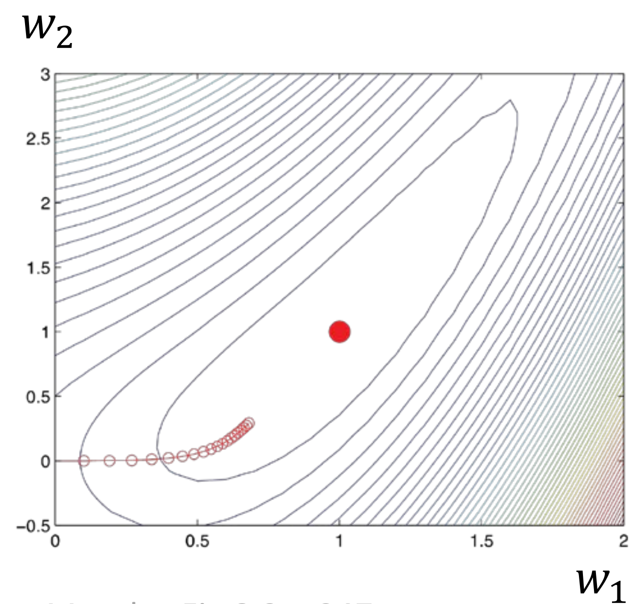
$$= \sum_{i=1}^{n} \log\left(\left(\theta(\boldsymbol{x}_i)\right)^{y_i}\left(1-\theta(\boldsymbol{x}_i)\right)^{1-y_i}\right)$$

$$= \sum_{i=1}^{n}\left(y_i \log\left(\theta(\boldsymbol{x}_i)\right) + (1-y_i)\log\left(1-\theta(\boldsymbol{x}_i)\right)\right)$$

$$= \sum_{i=1}^{n}\left((y_i - 1)\boldsymbol{x}_{\boldsymbol{i}}'\boldsymbol{w} - \log(1+\exp(-\boldsymbol{x}_{\boldsymbol{i}}'\boldsymbol{w}))\right)$$

Iterative optimisation

- Training logistic regression amounts to finding $\boldsymbol{w}$ that maximise log-likelihood
- Analytical approach: Set derivatives of objective function to zero and solve for $\boldsymbol{w}$
- **Bad news**: No closed form solution, iterative method necessary (e.g., gradient descent, Newton-Raphson, or iteratively-reweighted least squares)
- **Good news**: Problem is strictly convex (like a bowl) if there are no irrelevant features → optimisation guaranteed to work!



Murphy, Fig 8.3, p247

Look ahead (L5): regularisation helps with irrelevant features