Lecture 4b. Logistic Regression.

COMP90051 Statistical Machine Learning

Semester 2, 2020
Lecturer: Ben Rubinstein

THE UNIVERSITY OF MELBOURNE

This lecture

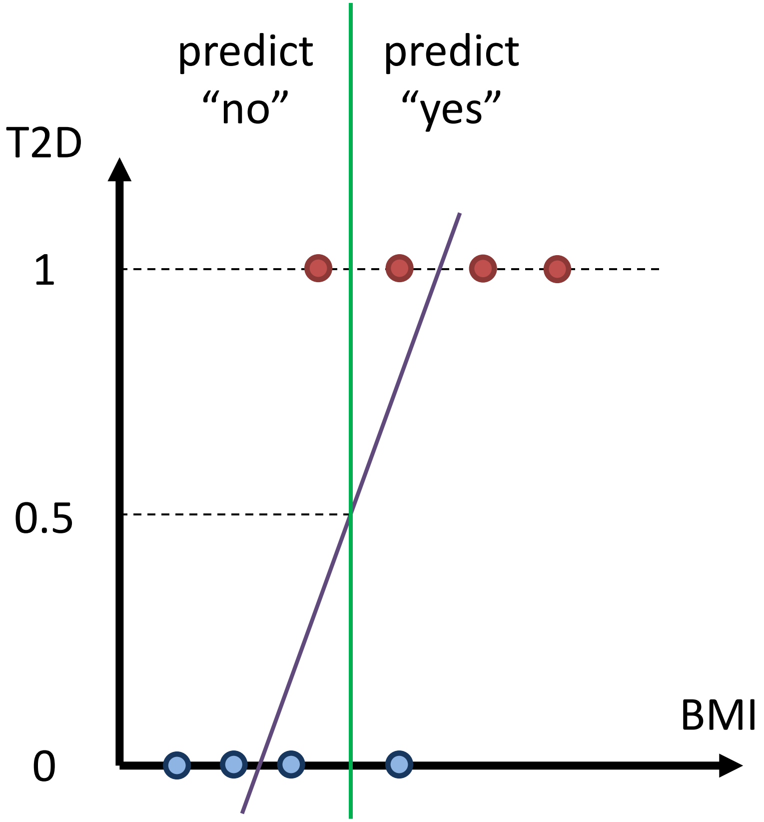
- Iterative optimisation for extremum estimators
  - First-order method: Gradient descent
  - Second-order: Newton-Raphson method
  - Later: Lagrangian duality
- **Logistic regression**: workhorse linear classifier
  - Possibly familiar derivation: **frequentist**
  - **Decision-theoretic** derivation
  - Training with Newton-Raphson looks like repeated, weighted linear regression

Logistic Regression Model

A workhorse linear, binary classifier;
(A review for some of you; new to some.)

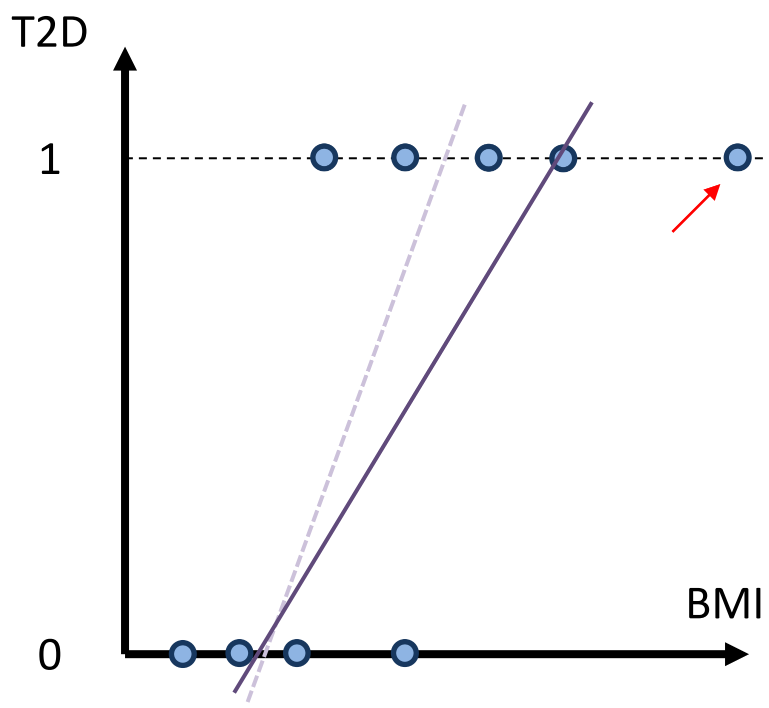
Binary classification: Example

- Example: given body mass index (BMI) does a patient have type 2 diabetes (T2D)?
- This is (supervised) binary classification
- One *could* use linear regression
  - Fit a line/hyperplane to data (find weights $\boldsymbol{w}$)
  - Denote $s \equiv \boldsymbol{x}'\boldsymbol{w}$
  - Predict "Yes" if $s \geq 0.5$
  - Predict "No" if $s < 0.5$



Why not linear regression

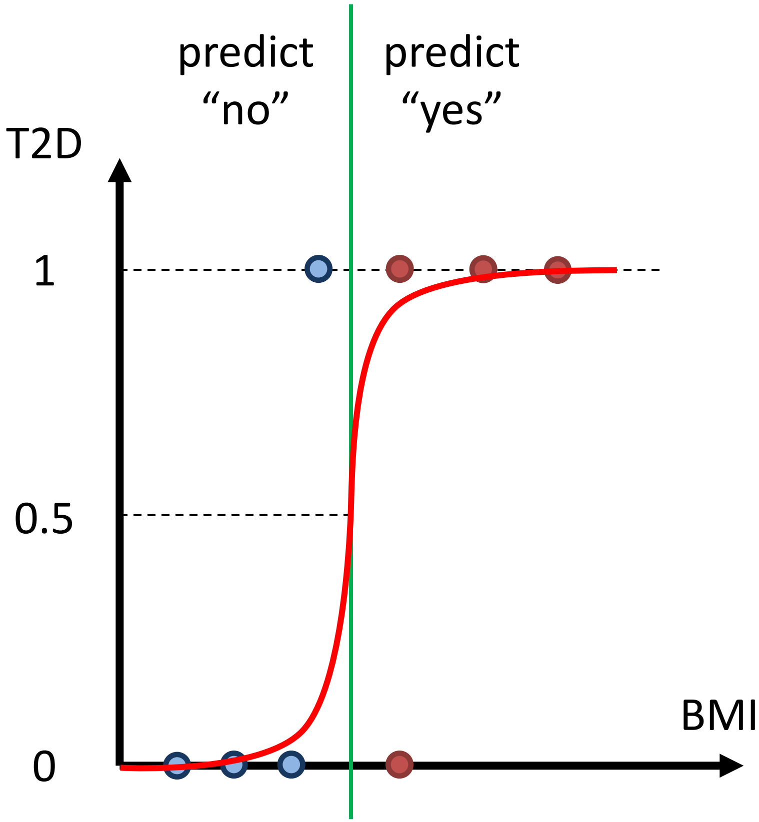
- Due to the square loss, points far from boundary have loss squared – even if they’re confidently correct!
- Such “outliers” will “pull at” the linear regression
- Overall, the least-squares criterion looks unnatural in this setting



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

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



How is logistic regression *linear*?

- Logistic regression model:

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

- Classification rule:

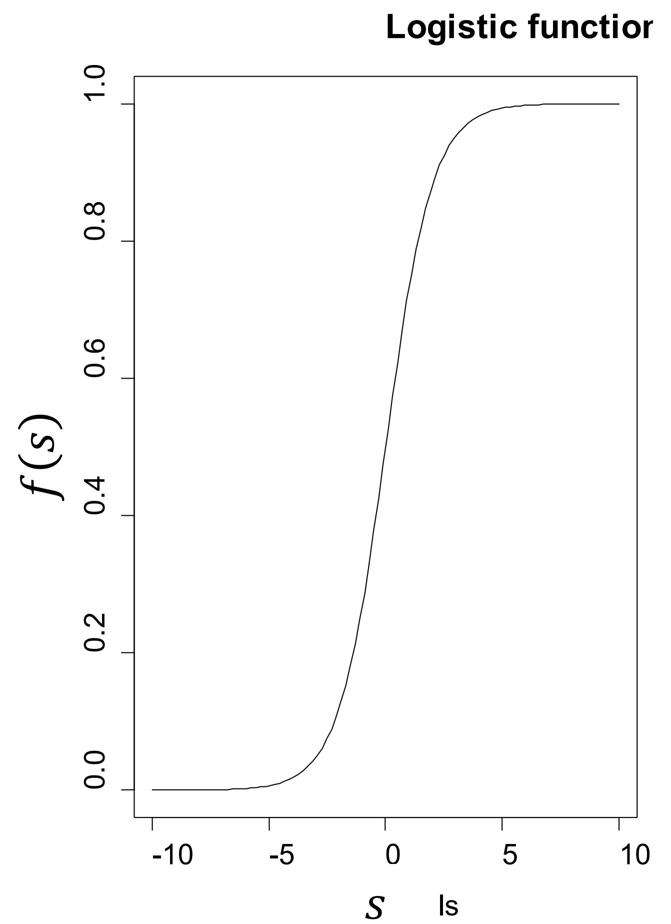
if $\left(P(Y = 1|\boldsymbol{x}) > \frac{1}{2}\right)$ then class "1", else class "0"

- Decision boundary is the set of **x**'s such that:

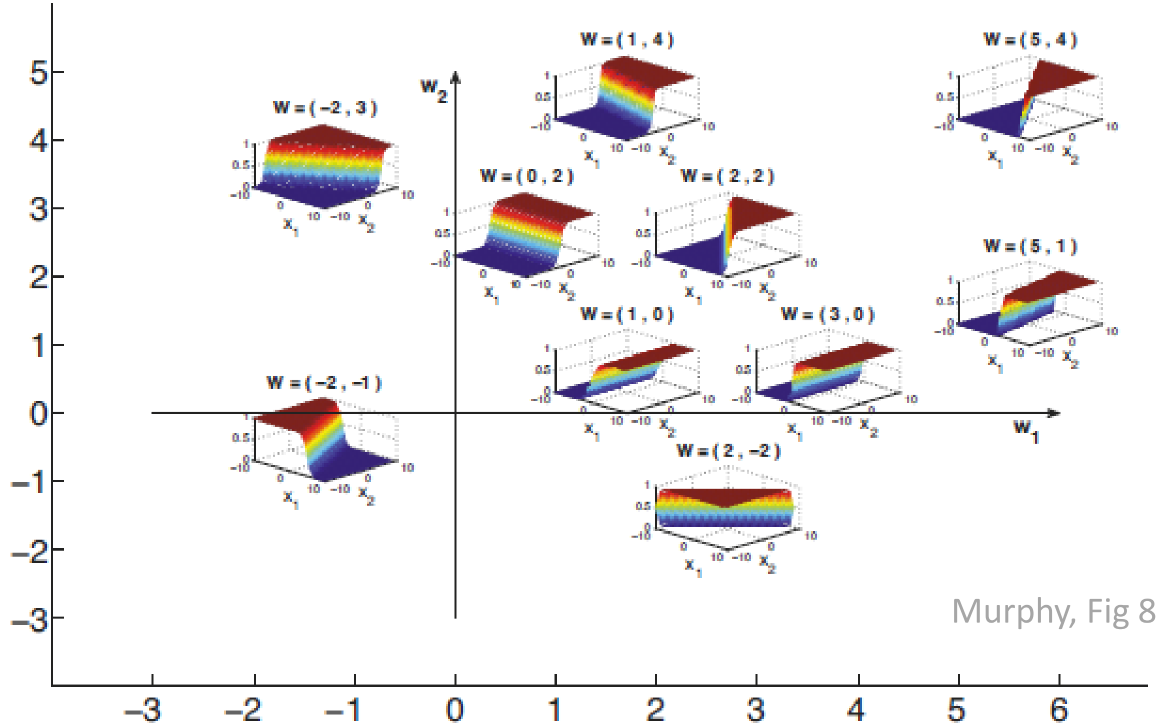
$$\frac{1}{1+\exp(-\boldsymbol{x}'\boldsymbol{w})} = \frac{1}{2}$$

$$\exp(-\underline{x}'\underline{w}) = 1$$

$$\underline{x}'\underline{w} = 0$$



Effect of parameter vector (2D problem)



Murphy, Fig 8.1, p246

- Decision boundary is the line where $P(Y=1|\boldsymbol{x})=0.5$
  - In higher dimensional problems, the decision boundary is a plane or hyperplane
- Vector $\boldsymbol{w}$ is perpendicular to the decision boundary (see linear algebra review topic)
  - That is, $\boldsymbol{w}$ is a normal to the decision boundary
  - Note: in this illustration we assume $w_0=0$ for simplicity

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

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

- 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

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

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

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

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

Can't do this analytically

Logistic Regression: Decision-Theoretic View

Via cross-entropy loss

Background: Cross entropy

- Cross entropy is an information-theoretic method for comparing two distributions

- Cross entropy is a measure of a divergence between reference distribution $g_{ref}(a)$ and estimated distribution $g_{est}(a)$. For discrete distributions:

$$H\left(g_{ref}, g_{est}\right) = -\sum_{a \in A} g_{ref}(a) \log g_{est}(a)$$

A is support of the distributions, e.g., $A = \{0,1\}$

Training as cross-entropy minimisation

- Consider log-likelihood <u>for a single data point</u>

$$\log p(y_i|\boldsymbol{x}_i) = y_i \log\big(\theta(\boldsymbol{x}_i)\big) + (1-y_i)\log\big(1-\theta(\boldsymbol{x}_i)\big)$$

- Cross entropy $H\left(g_{ref}, g_{est}\right) = -\sum_a g_{ref}(a) \log g_{est}(a)$
  - * If reference (true) distribution is

$$g_{ref}(1) = y_i \text{ and } g_{ref}(0) = 1 - y_i$$

  - * With logistic regression estimating this distribution as

$$g_{est}(1) = \theta(\boldsymbol{x}_i) \text{ and } g_{est}(0) = 1 - \theta(\boldsymbol{x}_i)$$

It finds $\boldsymbol{w}$ that minimises sum of cross entropies per training point

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

Cost function

Linear regression: $J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \cancel{\frac{1}{2}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2}$

$\mathrm{cost}(h_\theta(x^{(i)}), y)$

$$\mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Summary

- Logistic regression *formulation*
  - A workhorse linear binary classifier
  - Frequentist: Bernoulli label with coin bias logistic-linear in $\mathbf{x}$
  - Decision theory: Minimising cross entropy with labels

Next time: Training quickly with Newton-Raphson, and how that is repeated (weighted) linear regression under the hood!