Lecture 4. Logistic Regression. Basis Expansion

COMP90051 Statistical Machine Learning

Semester 2, 2018
Lecturer: Ben Rubinstein

THE UNIVERSITY OF
MELBOURNE

POSTERA CRESCAM LAUDE

This lecture

- ## Logistic regression
  - * Workhorse of binary classification

- ## Basis expansion
  - * Extending model expressiveness via data transformation
  - * Examples for linear and logistic regression
  - * Theoretical notes

Logistic Regression Model

A workhorse of binary classification

Binary classification: Example

- <u>Example</u>: given body mass index (BMI) does a patient have type 2 diabetes (T2D)?

- This type of problem is called binary classification

- One can use linear regression
  - ∗ Fit a line/hyperplane to data (find weights w)
  - ∗ Denote $s \equiv x'w$
  - ∗ Predict "Yes" if $s \geq 0.5$
  - ∗ Predict "No" if $s < 0.5$

predict "no" | predict "yes"

T2D

1

0.5
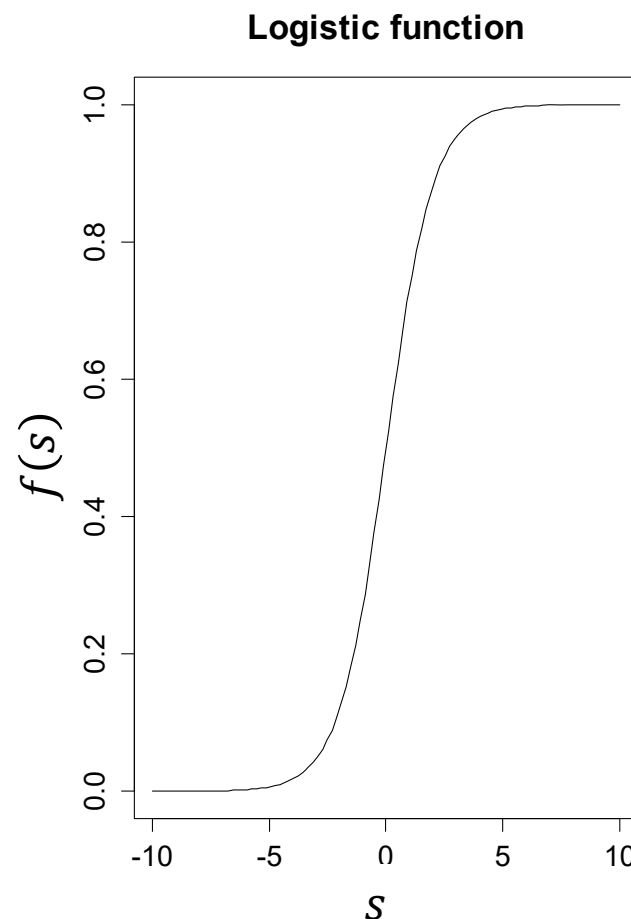
0

BMI
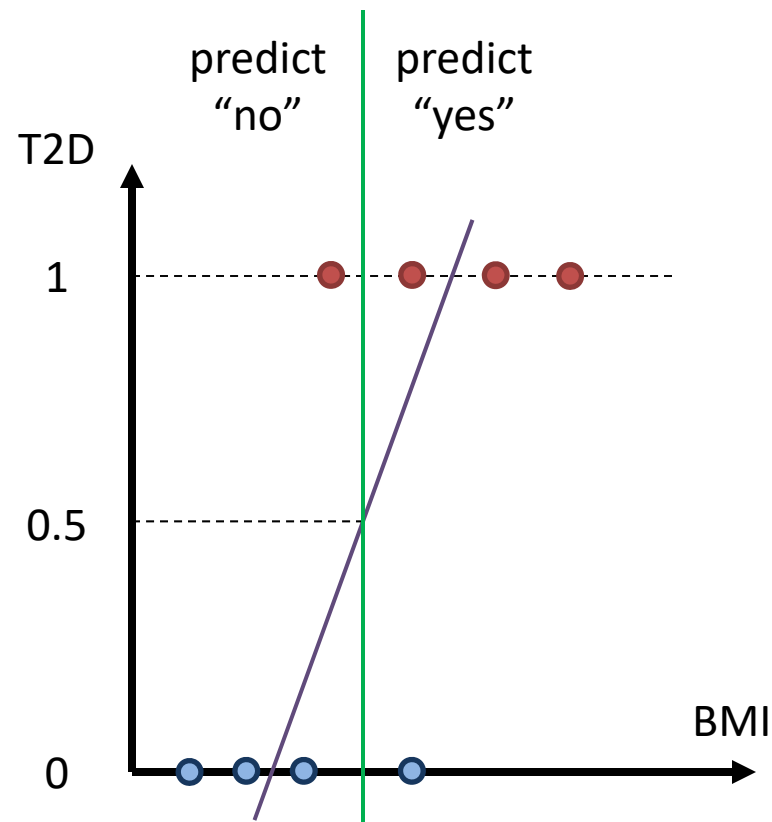
Approaches to classification

- This approach can be susceptible to outliers

- Overall, the least-squares criterion looks unnatural in this setting

- There are many methods developed specifically with binary classification in mind

- Examples include logistic regression, perceptron, support vector machines (SVM)
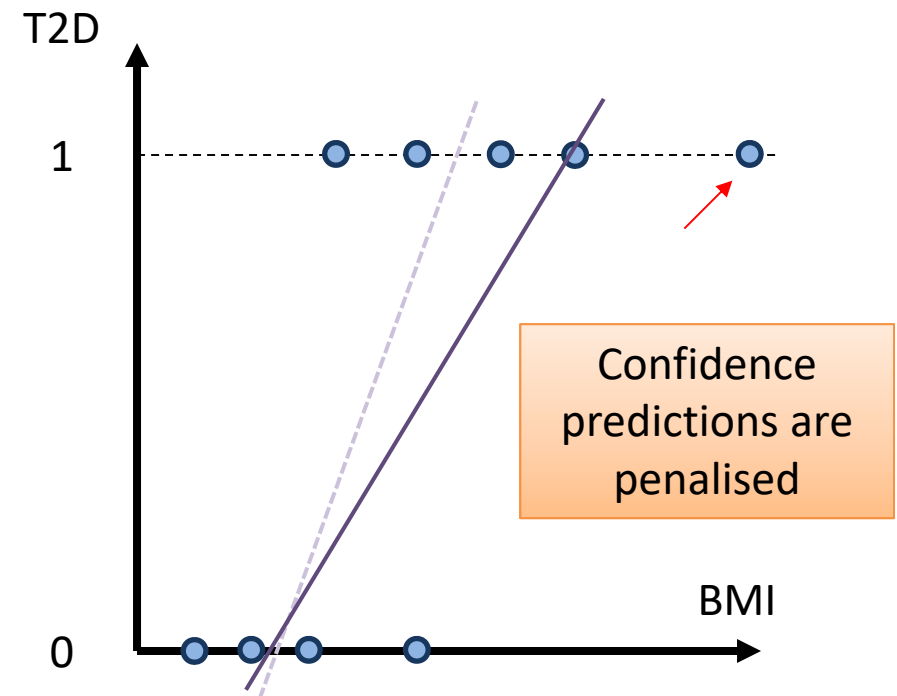
T2D

1

Confidence predictions are penalised

BMI

0

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 $f(s) = \dfrac{1}{1+\exp(-s)}$

- Logistic regression model

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

- Equivalent to linear model for log-odds ratio

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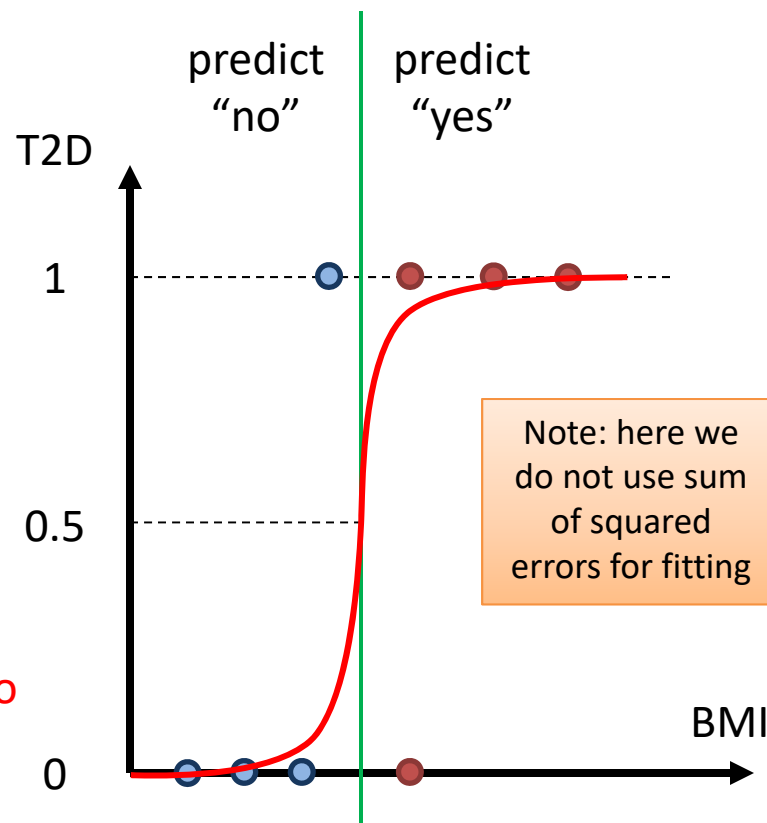
Logistic function

6

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 $f(s) = \dfrac{1}{1+\exp(-s)}$

- Logistic regression model

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

- Equivalent to linear model for log-odds ratio

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

predict "no"   predict "yes"

T2D

Note: here we do not use sum of squared errors for fitting

BMI

Is logistic regression a linear method?

Logistic function
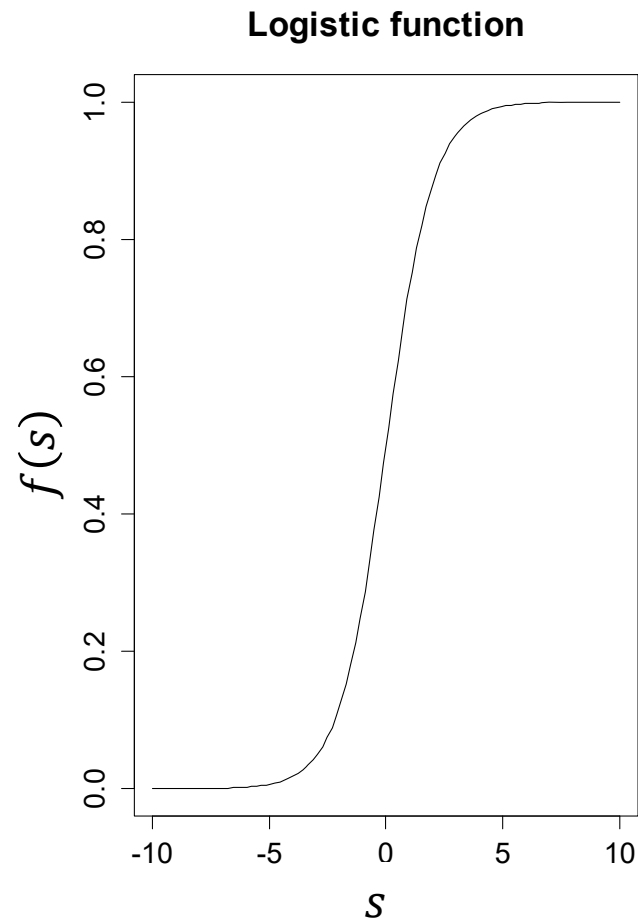
Logistic regression is a linear classifier

- Logistic regression model:

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

- Classification rule:

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

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

$1 + \exp(-x \cdot w) = 2$

$\exp(-x \cdot w) = 1$

$-x \cdot w = \log(1)$

$x \cdot w = 0$

$\boxed{\text{linear}}$

or at thresholds
other than $\frac{1}{2}$

get $x \cdot w + b = 0$

$\Rightarrow$ affine level sets

linear but not through
origin

9

Effect of parameter vector (2D problem)

Murphy, Fig 8.1, p246
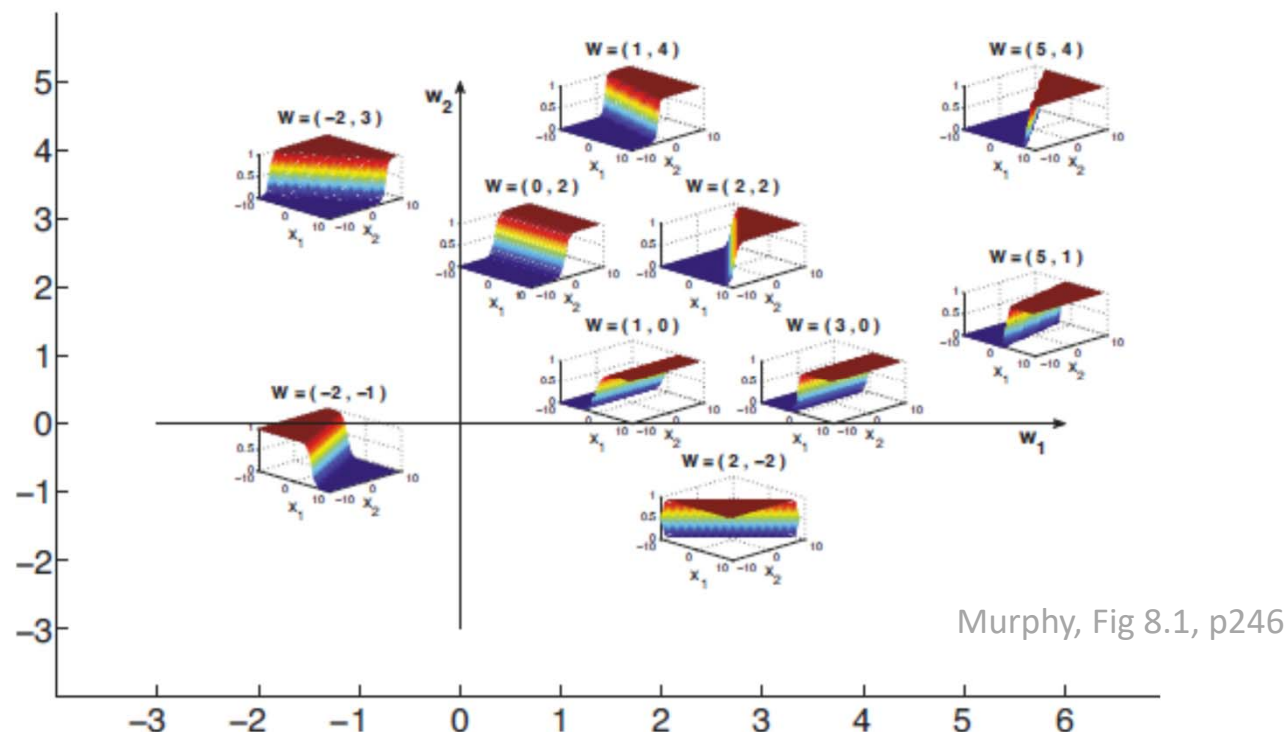
- Decision boundary is the line where $P(Y = 1 | x) = 0.5$
    - In higher dimensional problems, the decision boundary is a plane or hyperplane

- Vector w is perpendicular to the decision boundary (see supplemental LMS vector slides)
    - That is, w is a normal to the decision boundary
    - Note: in this illustration we assume $w_0 = 0$ for simplicity

10

Linear vs. logistic probabilistic models

- **Linear regression** assumes a <u>Normal distribution</u> 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 <u>Bernoulli distribution</u> 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

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

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

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

Apply log trick, simplify

- Instead of maximising likelihood, maximise its logarithm

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

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

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

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

Iterative optimisation

- Training logistic regression amounts to finding w that maximise log-likelihood

- Analytical approach: Set derivatives of objective function to zero and solve for 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!

- Look ahead (L5): regularisation helps with irrelevant features

w_2

Murphy, Fig 8.3, p247

w_1
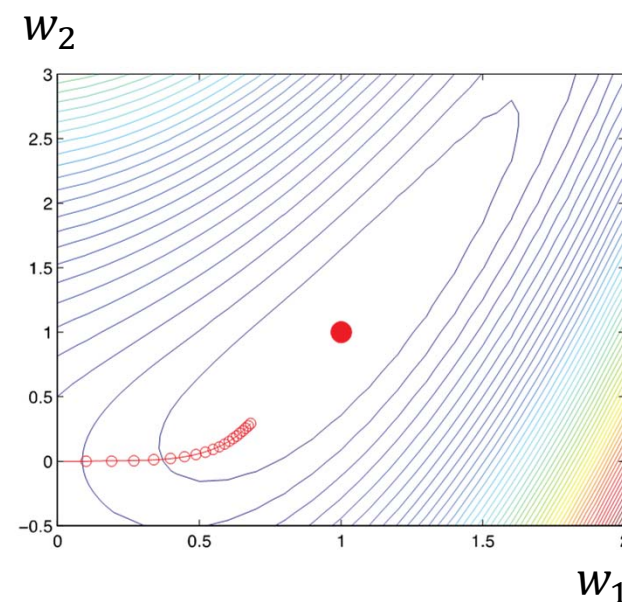
Logistic Regression: Decision-Theoretic View

Where loss is cross entropy

Side note: Cross entropy

- Cross entropy is a 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>

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

- This expression is the negative cross entropy

- Cross entropy $H\big(g_{ref}, g_{est}\big) = -\sum_a g_{ref}(a)\log g_{est}(a)$

- The reference (true) distribution is

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

- Logistic regression aims to estimate 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 pt

Basis Expansion

Extending the utility of models via
data transformation

Basis expansion for linear regression

- Let's take a step back. Back to linear regression and least squares

- Real data is likely to be non-linear

- What if we still wanted to use a linear regression?
  - ∗ It's simple, easier to understand, computationally efficient, etc.

- How to marry non-linear data to a linear method?

- *if the mountain won't come to Muhammad then Muhammad must go to the mountain*
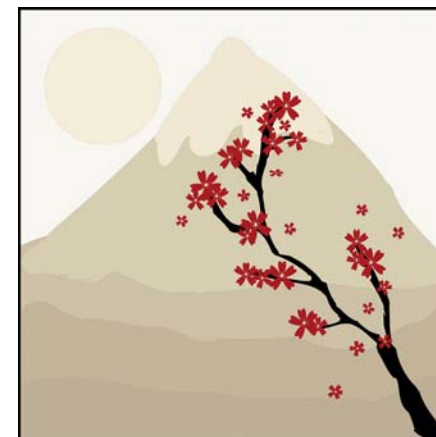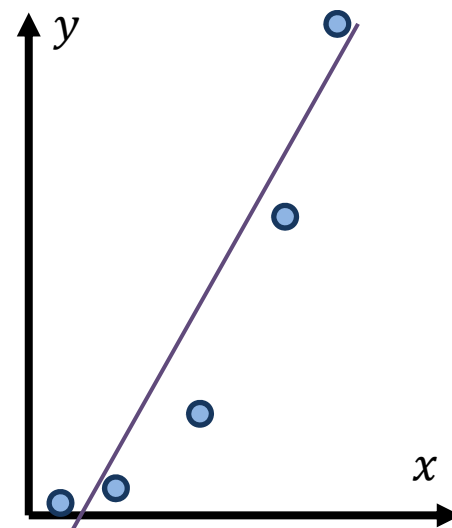
art: OpenClipartVectors at pixabay.com (CC0)

19

Transform the data

- The trick is to transform the data: Map data onto another features space, s.t. data is linear in that space

- Denote this transformation $\varphi: \mathbb{R}^m \rightarrow \mathbb{R}^k$. If $\boldsymbol{x}$ is the original set of features, $\varphi(\boldsymbol{x})$ denotes new feature set

- Example: suppose there is just one feature x, and the data is scattered around a parabola rather than a straight line
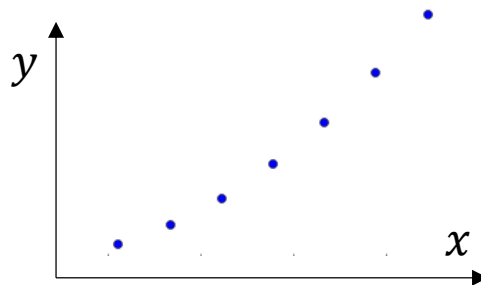
Example: Polynomial regression

- No worries, mate: define
$$\varphi_1(x) = x$$
$$\varphi_2(x) = x^2$$
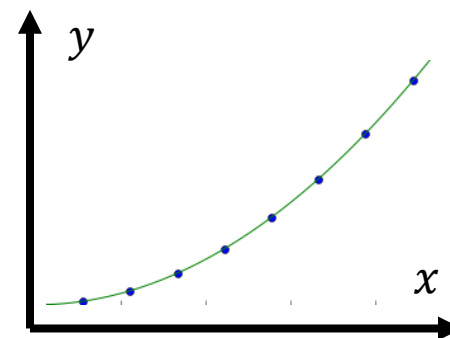
- Next, apply linear regression to φ_1, φ_2

$$y = w_0 + w_1\varphi_1(x) + w_2\varphi_2(x) = w_0 + w_1 x + w_2 x^2$$

   and here you have **quadratic regression**

- More generally, obtain **polynomial regression** if the new set of attributes are powers of x

Basis expansion

- Data transformation, also known as basis expansion, is a general technique
    * We'll see more examples throughout the course

- It can be applied for both regression and classification
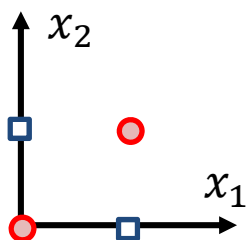
- There are many possible choices of φ
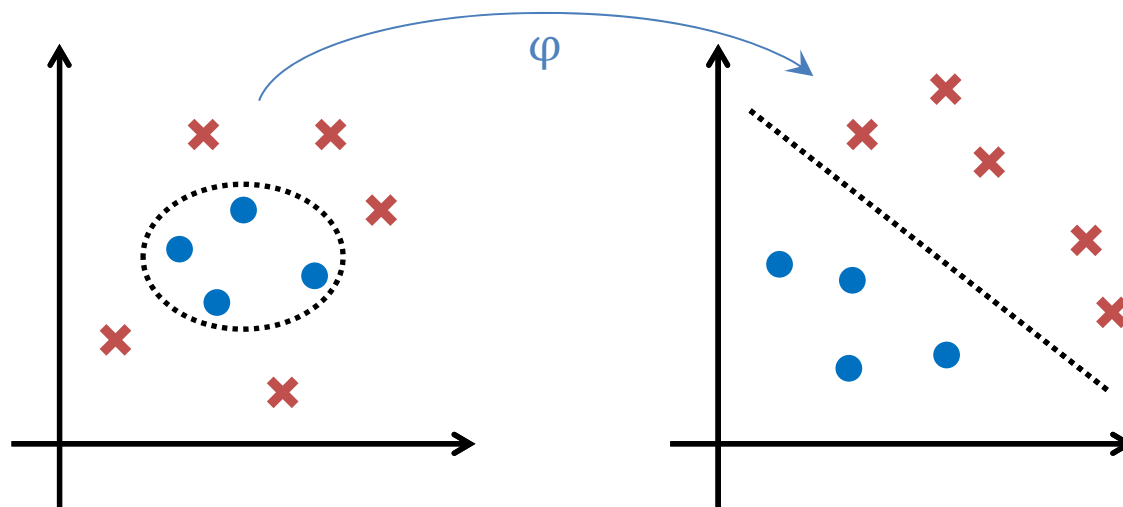
Basis expansion for logistic regression

- Example binary classification problem: Dataset not linearly separable

- Define transformation as

$$\varphi_i(x) = \|x - z_i\|, \text{ where } z_i \text{ some pre-defined constants}$$

- Choose $z_1 = [0,0]'$, $z_2 = [0,1]'$, $z_3 = [1,0]'$, $z_4 = [1,1]'$

there exist weights that make new data separable, e.g.:

w_1	w_2	w_3	w_4
1	0	0	1

The transformed data is linearly separable!

x_1	x_2	y
0	0	Class A
0	1	Class B
1	0	Class B
1	1	Class A

φ_1	φ_2	φ_3	φ_4
0	1	1	$\sqrt{2}$
1	0	$\sqrt{2}$	1
1	$\sqrt{2}$	0	1
$\sqrt{2}$	1	1	0

$\varphi'w$	y
$\sqrt{2}$	Class A
2	Class B
2	Class B
$\sqrt{2}$	Class A

23

Radial basis functions

- The above transformation is an example of the use of radial basis functions (RBFs)
  * Their use has been motivated from approximation theory, where sums of RBFs are used to approximate given functions

- A radial basis function is a function of the form $\varphi(\boldsymbol{x}) = \psi(\|\boldsymbol{x} - \boldsymbol{z}\|)$, where $\boldsymbol{z}$ is a constant

- Examples:

- $\varphi(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{z}\|$

- $\varphi(\boldsymbol{x}) = \exp\left(-\frac{1}{\sigma}\|\boldsymbol{x} - \boldsymbol{z}\|^2\right)$
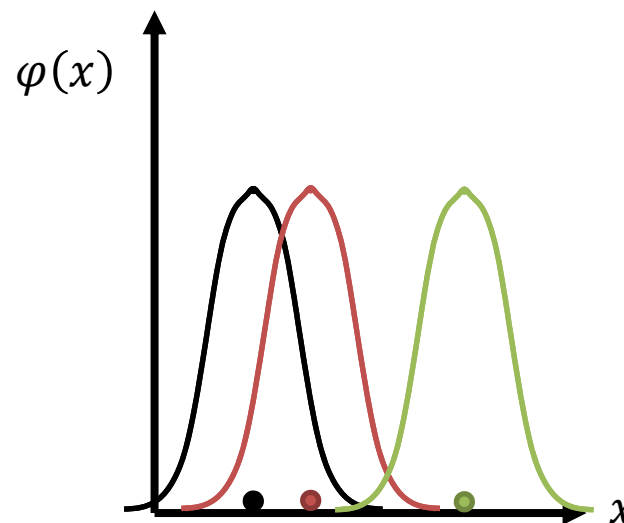
Challenges of basis expansion

- Basis expansion can significantly increase the utility of methods, especially, linear methods

- In the above examples, one limitation is that the transformation needs to be defined beforehand
  * Need to choose the size of the new feature set
  * If using RBFs, need to choose z_i

- Regarding z_i, one can choose uniformly spaced points, or cluster training data and use cluster centroids

- Another popular idea is to use training data $z_i \equiv x_i$
  * E.g., $\varphi_i(x) = \psi(\|x - x_i\|)$
  * However, for large datasets, this results in a large number of features → computational hurdle

Further directions

- There are several avenues for taking the idea of basis expansion to the next level
  - ∗ Will be covered later in this subject

- One idea is to *learn* the transformation φ from data
  - ∗ E.g., Artificial Neural Networks

- Another powerful extension is the use of the kernel trick
  - ∗ "Kernelised" methods, e.g., kernelised perceptron

- Finally, in sparse kernel machines, training depends only on a few data points
  - ∗ E.g., SVM

Summary

- ## Logistic regression
    - * Workhorse linear binary classifier

- ## Basis expansion
    - * Extending model expressiveness via data transformation
    - * Examples for linear and logistic regression
    - * Theoretical notes

Next time: regularisation for avoiding overfitting and ill-posed optimisation; with example algorithms