

# Lecture 4. Logistic Regression. Basis Expansion

COMP90051 Statistical Machine Learning

Semester 2, 2018  
Lecturer: Ben Rubinstein



Copyright: University of Melbourne

# This lecture

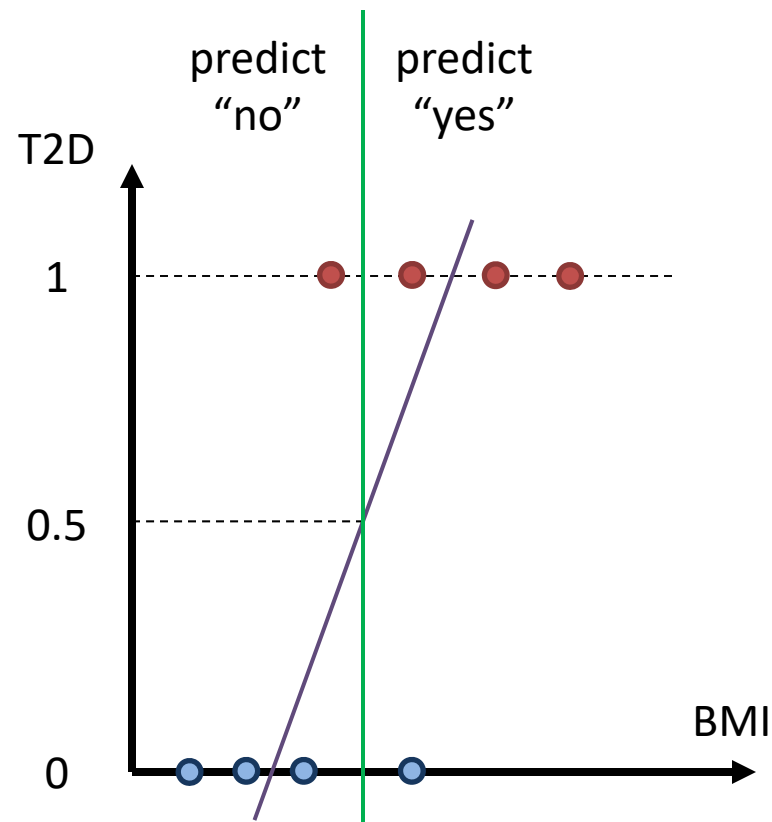
- Logistic regression
  - \* Workhorse of binary classification
- Basis expansion
  - \* Extending model expressiveness via data transformation
  - \* Examples for linear and logistic regression
  - \* Theoretical notes

# Logistic Regression Model

A workhorse of binary classification

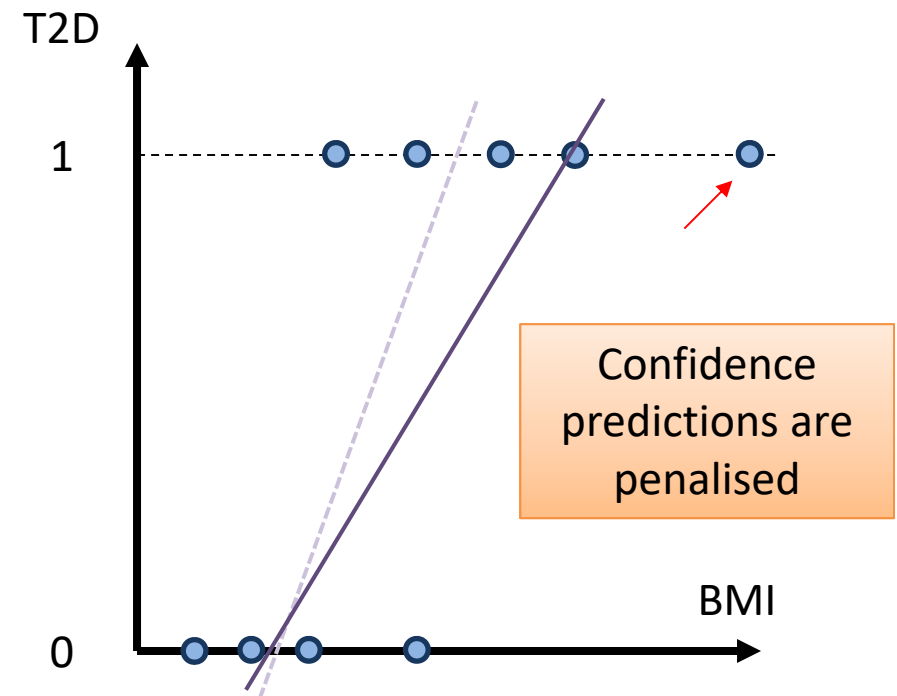
# Binary classification: Example

- Example: 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  $\mathbf{w}$ )
  - \* Denote  $s \equiv \mathbf{x}'\mathbf{w}$
  - \* Predict "Yes" if  $s \geq 0.5$
  - \* Predict "No" if  $s < 0.5$



# Approaches to classification

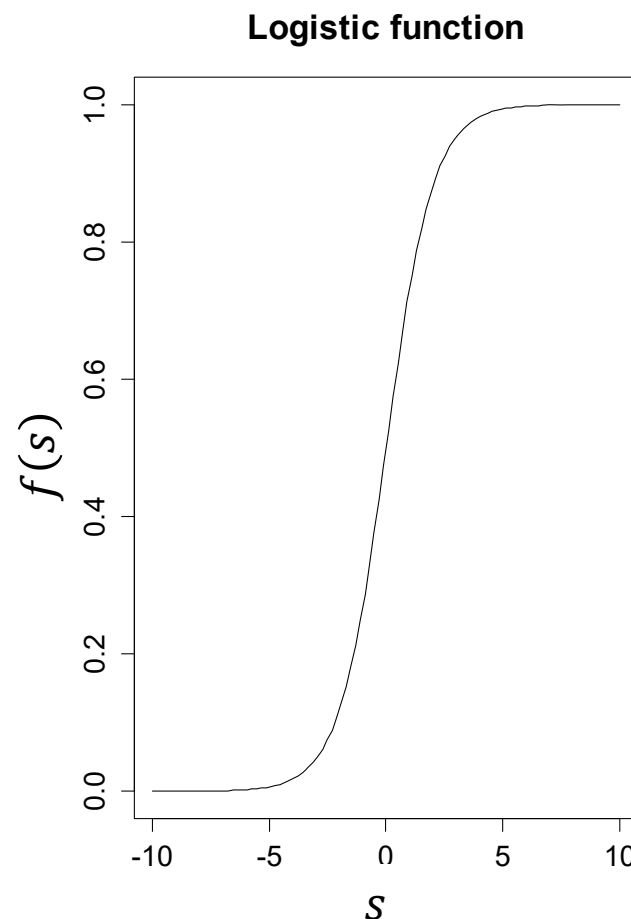
- This approach can be susceptible to outliers
- Overall, the least-squares criterion looks unnatural in this setting
- There are many methods developed specifically with binary classification in mind
- Examples include logistic regression, perceptron, support vector machines (SVM)



# Logistic regression model

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

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

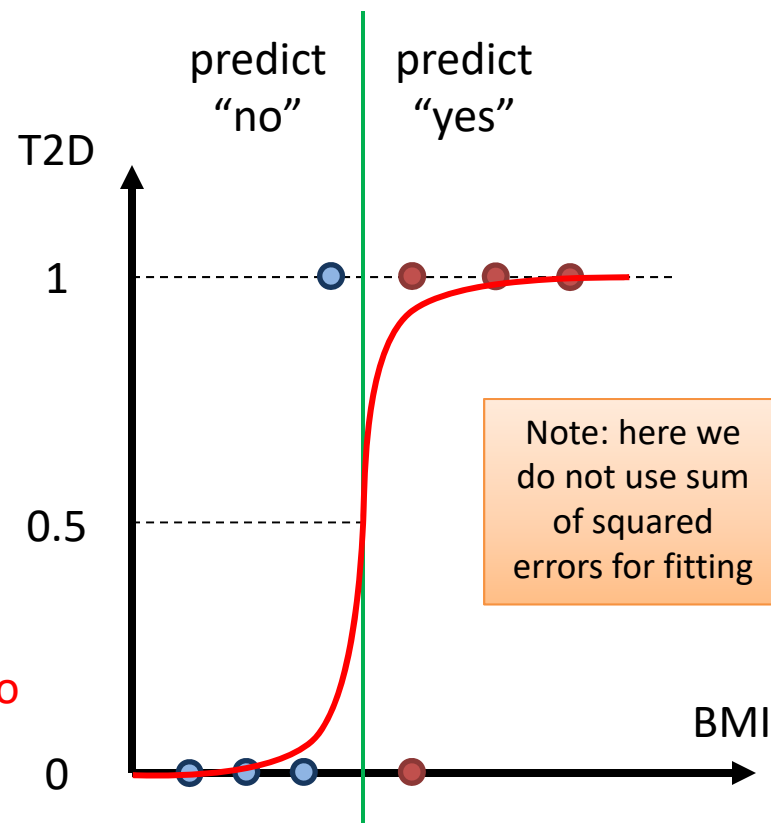


# Logistic regression model

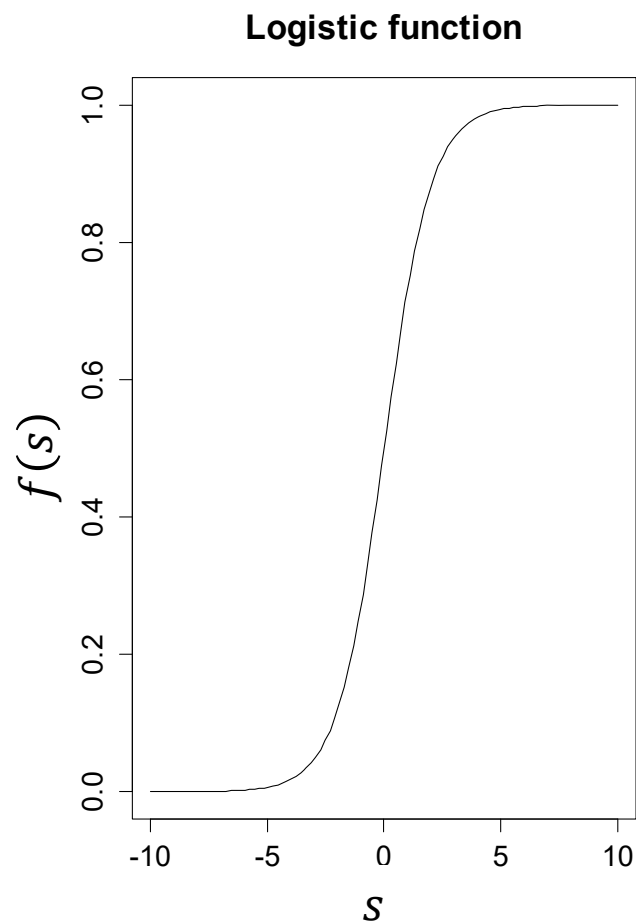
- Probabilistic approach to classification
  - \*  $P(Y = 1|\mathbf{x}) = f(\mathbf{x}) = ?$
  - \* Use a linear function? E.g.,  $s(\mathbf{x}) = \mathbf{x}'\mathbf{w}$
- Problem: the probability needs to be between 0 and 1.
- **Logistic** function  $f(s) = \frac{1}{1 + \exp(-s)}$
- **Logistic regression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

$$P(Y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{x}'\mathbf{w})}$$
- Equivalent to linear model for **log-odds ratio**

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



# Is logistic regression a linear method?





# Logistic regression is a linear classifier

- Logistic regression model:

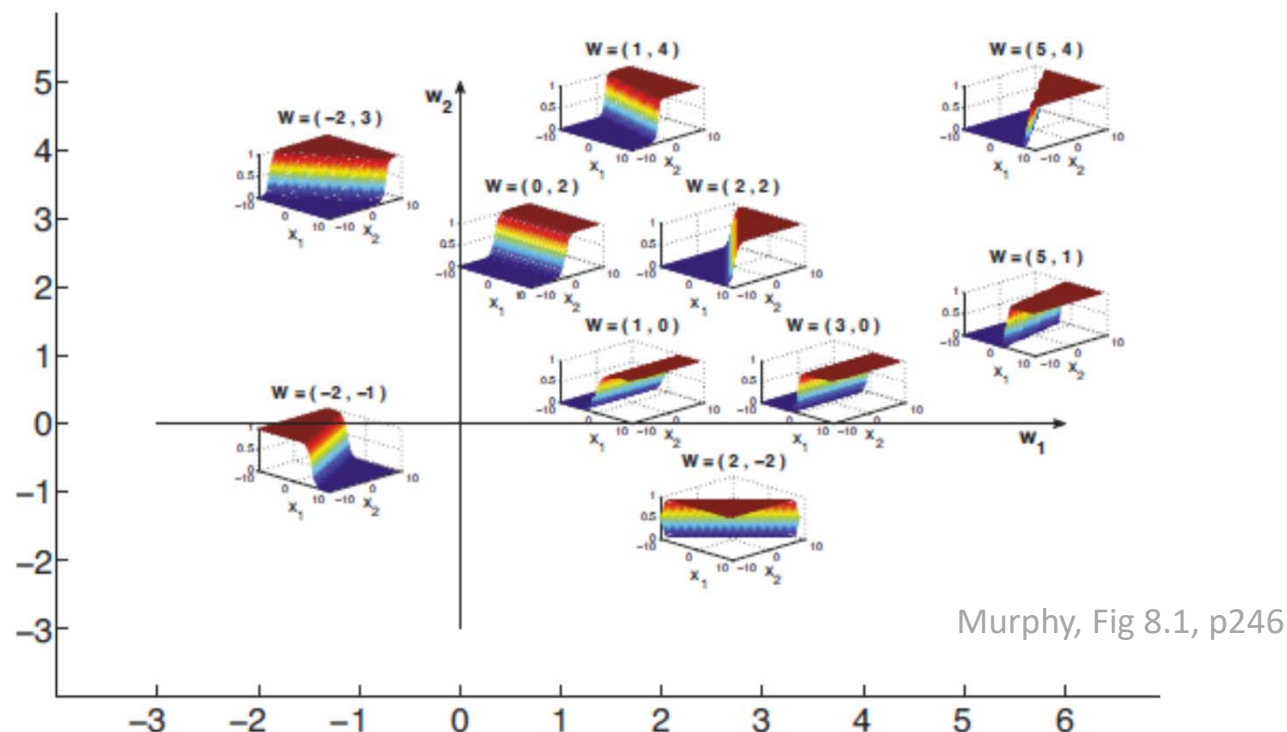
$$P(Y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{x}'\mathbf{w})}$$

- Classification rule:

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

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

# Effect of parameter vector (2D problem)



- **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  $\mathbf{w}$  is perpendicular to the decision boundary (see supplemental LMS vector slides)
  - \* That is,  $\mathbf{w}$  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

$$p(y|\mathbf{x}) = \text{Normal}(\mathbf{x}'\mathbf{w}, \sigma^2)$$

- **Logistic regression** assumes a Bernoulli distribution with parameter given by logistic transform of linear model

$$p(y|\mathbf{x}) = \text{Bernoulli}(\text{logistic}(\mathbf{x}'\mathbf{w}))$$

- 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

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

- Assuming Bernoulli distribution we have

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

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

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

## Apply log trick, simplify

- Instead of maximising likelihood, maximise its logarithm

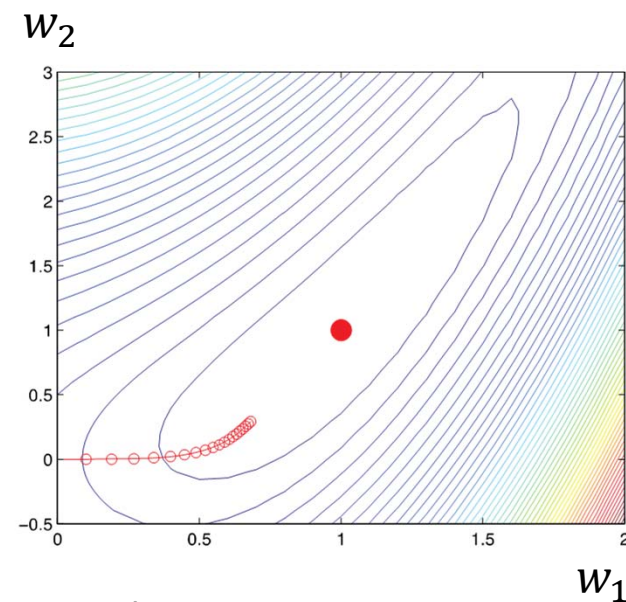
$$\begin{aligned}\log\left(\prod_{i=1}^n p(y_i|\mathbf{x}_i)\right) &= \sum_{i=1}^n \log p(y_i|\mathbf{x}_i) \\ &= \sum_{i=1}^n \log\left((\theta(\mathbf{x}_i))^{y_i} (1 - \theta(\mathbf{x}_i))^{1-y_i}\right) \\ &= \sum_{i=1}^n (y_i \log(\theta(\mathbf{x}_i)) + (1 - y_i) \log(1 - \theta(\mathbf{x}_i))) \\ &= \sum_{i=1}^n ((y_i - 1)\mathbf{x}_i' \mathbf{w} - \log(1 + \exp(-\mathbf{x}_i' \mathbf{w})))\end{aligned}$$

# Iterative optimisation

- Training logistic regression amounts to finding  $\mathbf{w}$  that maximise log-likelihood
- Analytical approach: Set derivatives of objective function to zero and solve for  $\mathbf{w}$
- **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  $\rightarrow$  optimisation guaranteed to work!



Look ahead (L5): regularisation helps with irrelevant features



Murphy, Fig 8.3, p247

# 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:

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

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



## Training as cross-entropy minimisation

- Consider log-likelihood for a single data point  
$$\log p(y_i | \mathbf{x}_i) = y_i \log(\theta(\mathbf{x}_i)) + (1 - y_i) \log(1 - \theta(\mathbf{x}_i))$$
- This expression is the **negative cross entropy**
- Cross entropy  $H(g_{ref}, g_{est}) = -\sum_a g_{ref}(a) \log g_{est}(a)$
- 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

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

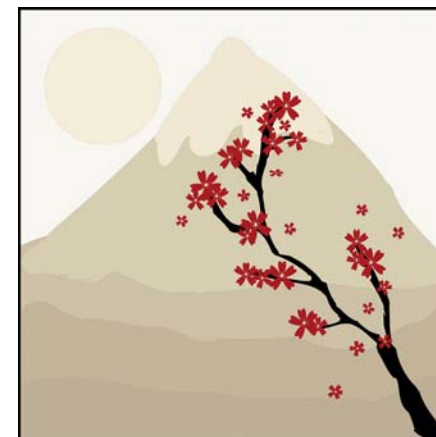
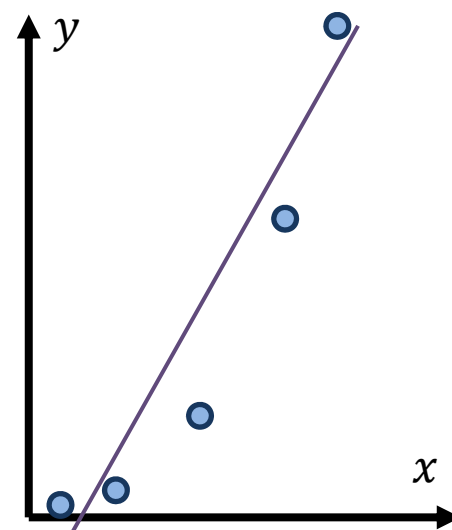
It finds  $\mathbf{w}$  that minimises sum of cross entropies per training pt

# Basis Expansion

Extending the utility of models via  
data transformation

# Basis expansion for linear regression

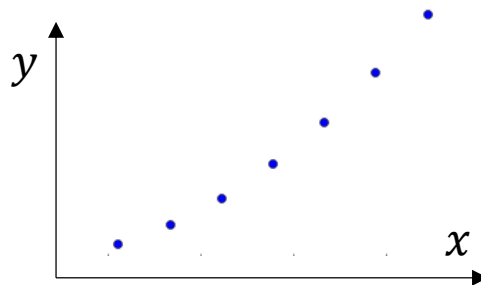
- Let's take a step back. Back to linear regression and least squares
- Real data is likely to be non-linear
- What if we still wanted to use a linear regression?
  - \* It's simple, easier to understand, computationally efficient, etc.
- How to marry non-linear data to a linear method?
- *if the mountain won't come to Muhammad then Muhammad must go to the mountain*



art: OpenClipartVectors at pixabay.com (CC0)

# Transform the data

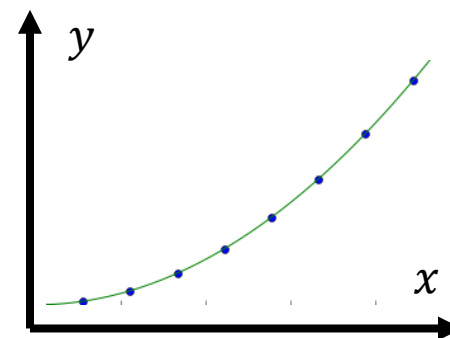
- The trick is to **transform the data**: Map data onto another features space, s.t. data is linear in that space
- Denote this transformation  $\varphi: \mathbb{R}^m \rightarrow \mathbb{R}^k$ . If  $\mathbf{x}$  is the original set of features,  $\varphi(\mathbf{x})$  denotes new feature set
- Example: 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

$$\begin{aligned}\varphi_1(x) &= x \\ \varphi_2(x) &= x^2\end{aligned}$$



- Next, apply linear regression to  $\varphi_1, \varphi_2$

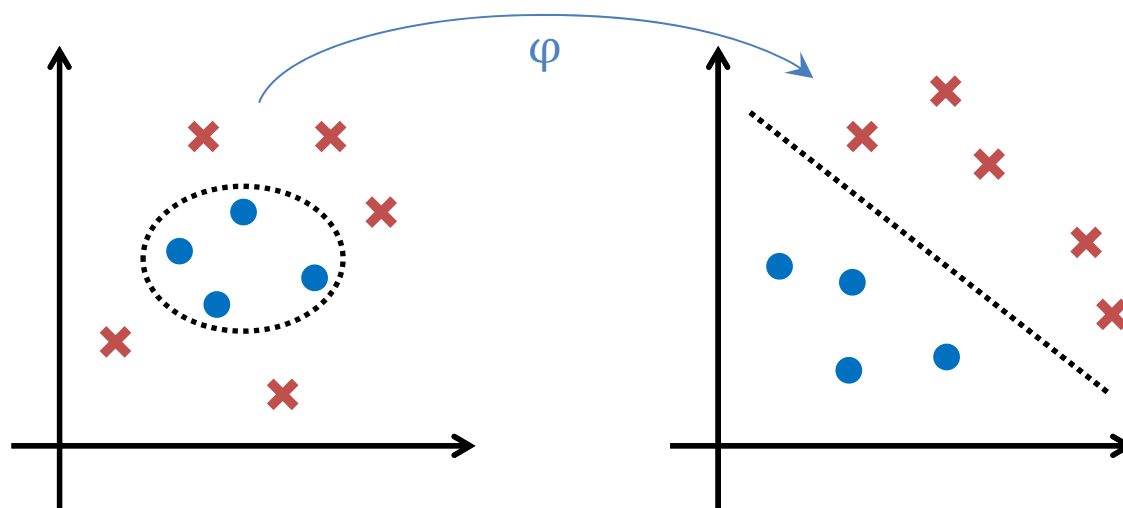
$$y = w_0 + w_1\varphi_1(x) + w_2\varphi_2(x) = w_0 + w_1x + w_2x^2$$

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  $\varphi$

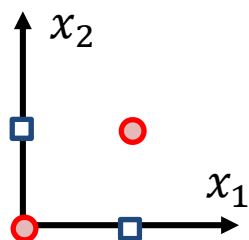


# Basis expansion for logistic regression

- Example binary classification problem: Dataset not **linearly separable**
- Define transformation as

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

- Choose  $\mathbf{z}_1 = [0,0]'$ ,  $\mathbf{z}_2 = [0,1]'$ ,  $\mathbf{z}_3 = [1,0]'$ ,  $\mathbf{z}_4 = [1,1]'$



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!

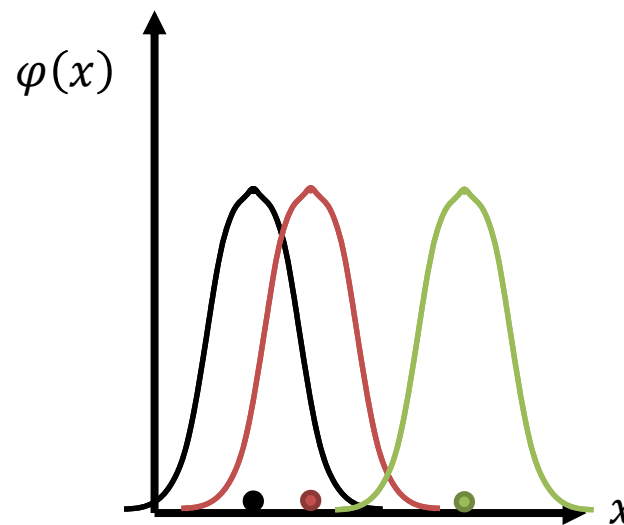
$x_1$	$x_2$	$y$
0	0	Class A
0	1	Class B
1	0	Class B
1	1	Class A

$\varphi_1$	$\varphi_2$	$\varphi_3$	$\varphi_4$
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

# 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  $\varphi(\mathbf{x}) = \psi(\|\mathbf{x} - \mathbf{z}\|)$ , where  $\mathbf{z}$  is a constant
- Examples:
  - $\varphi(\mathbf{x}) = \|\mathbf{x} - \mathbf{z}\|$
  - $\varphi(\mathbf{x}) = \exp\left(-\frac{1}{\sigma}\|\mathbf{x} - \mathbf{z}\|^2\right)$





# 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  $\mathbf{z}_i$
- Regarding  $\mathbf{z}_i$ , one can choose uniformly spaced points, or cluster training data and use cluster centroids
- Another popular idea is to use training data  $\mathbf{z}_i \equiv \mathbf{x}_i$ 
  - \* E.g.,  $\varphi_i(\mathbf{x}) = \psi(\|\mathbf{x} - \mathbf{x}_i\|)$
  - \* However, for large datasets, this results in a large number of features  $\rightarrow$  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  $\varphi$  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