Lecture 4. Logistic Regression. Basis Expansion

COMP90051 Statistical Machine Learning

Semester 2, 2017
Lecturer: Andrey Kan

THE UNIVERSITY OF MELBOURNE

This lecture

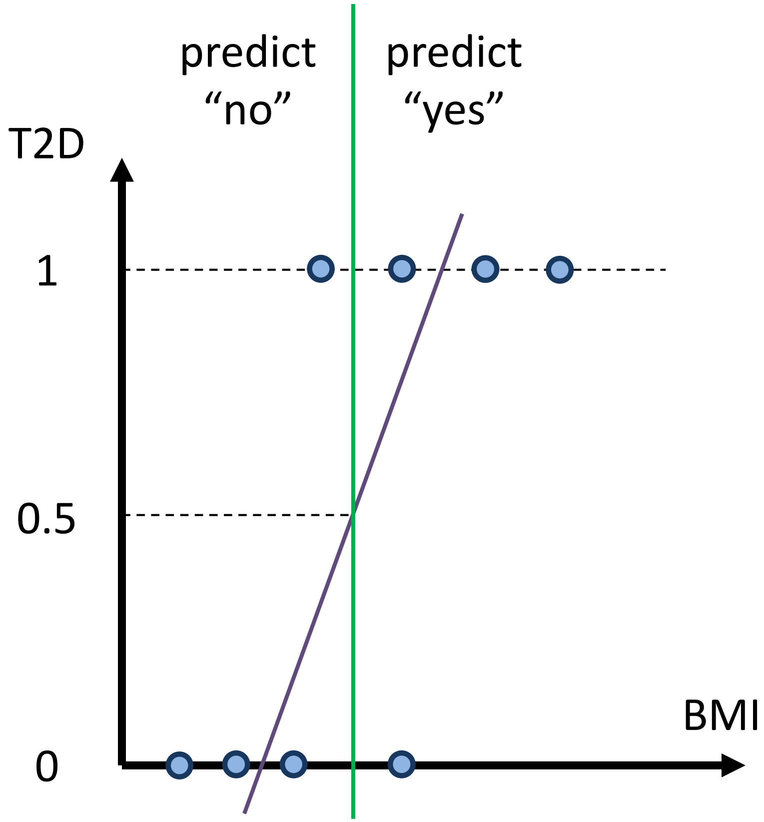
- Logistic regression
  - Binary classification problem
  - Logistic regression model
- Basis expansion
  - Examples for linear and logistic regression
  - Theoretical notes

Logistic Regression Model

A linear method for binary classification

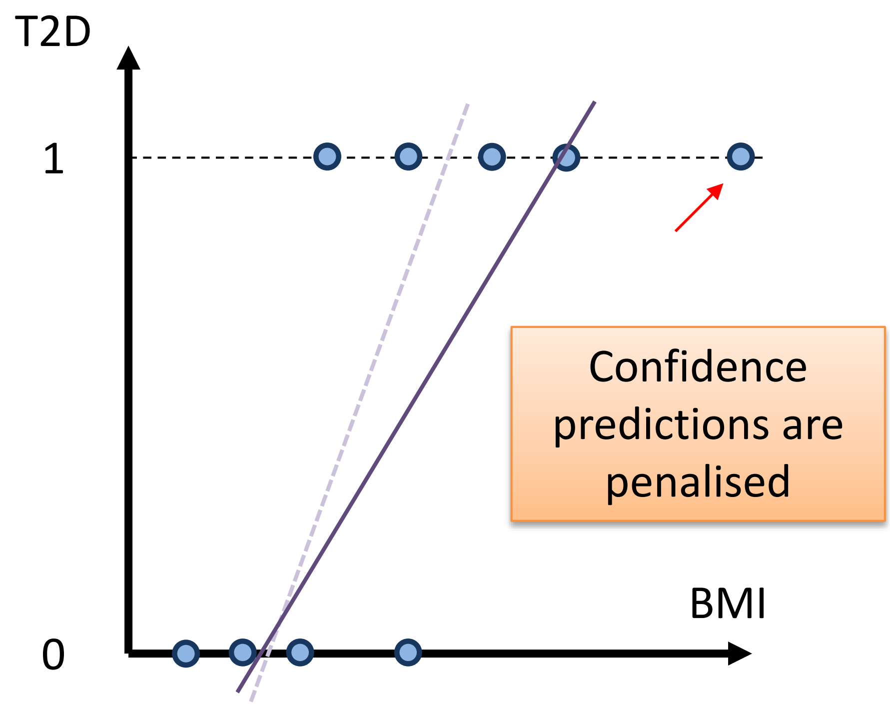
Binary classification problem

- Example: given body mass index (BMI) does a patient have type 2 diabetes (T2D)?
- This type of problems is called binary classification
- One can 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 $s < 0.5$



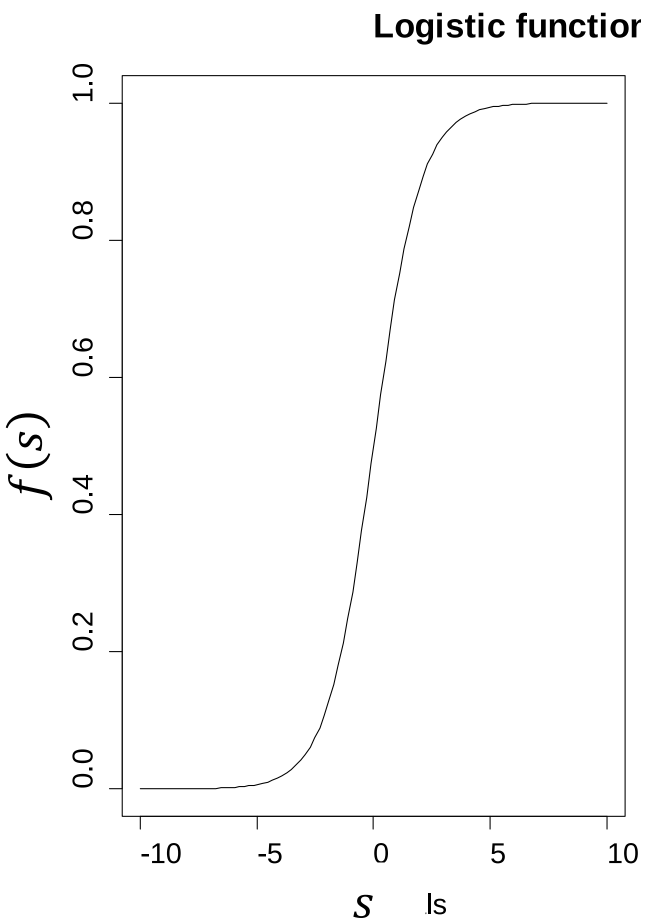
Approaches to classification

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



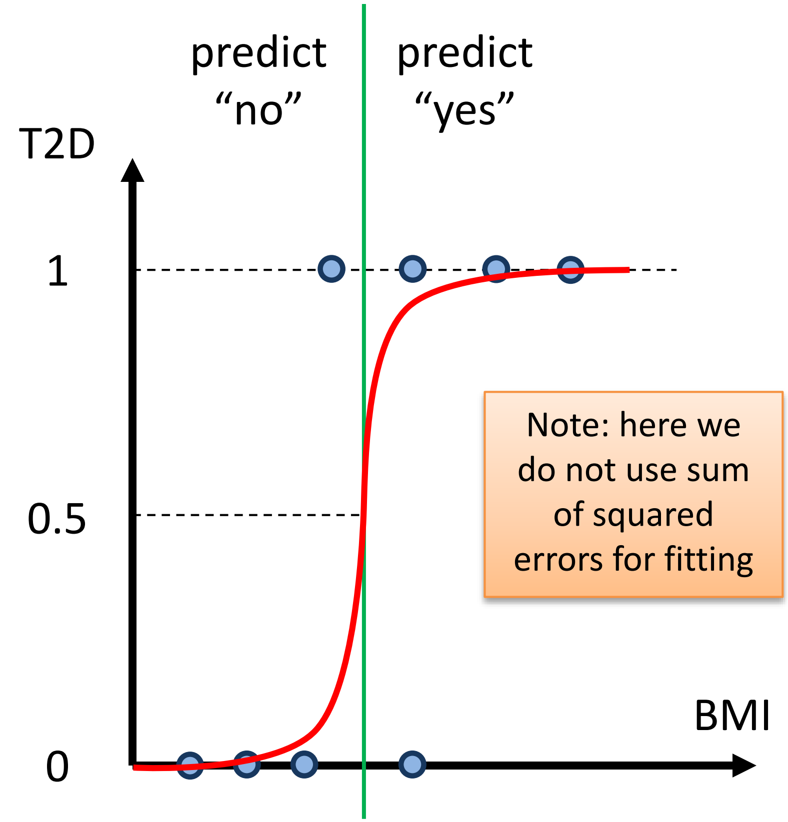
Logistic regression model

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

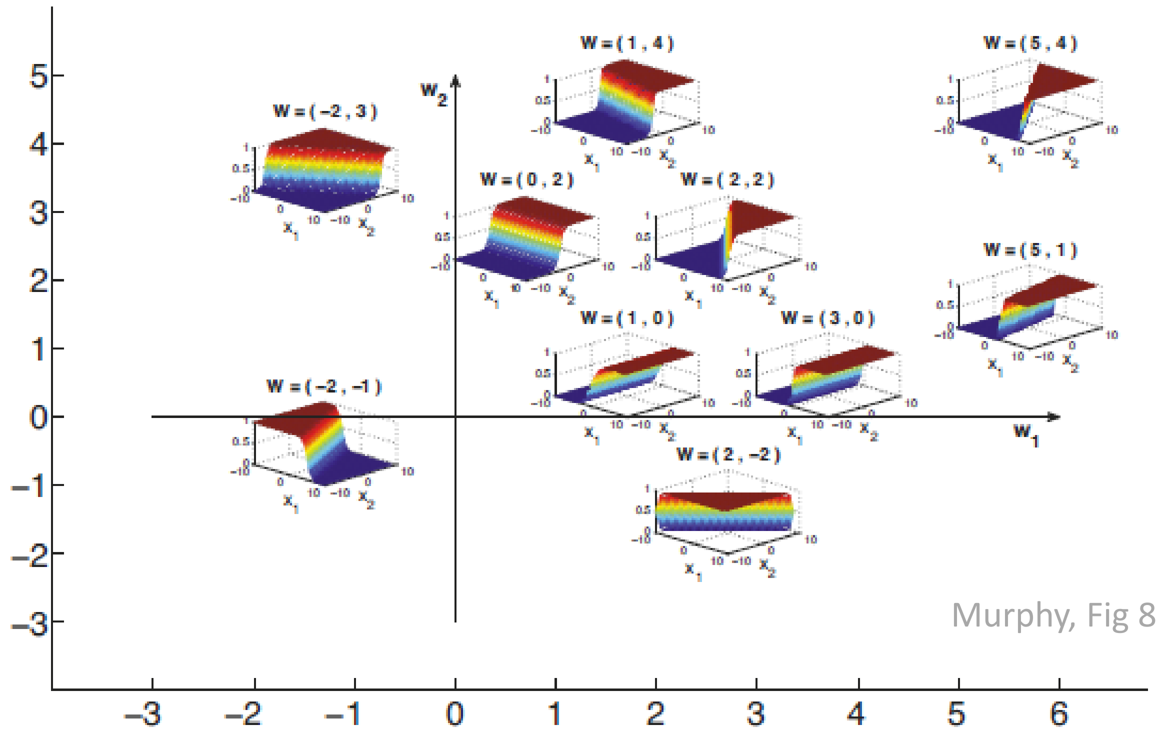
$$P(\mathcal{Y} = 1|\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{x}'\boldsymbol{w})}$$

- Equivalent to linear model for log-odds

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



Effect of parameter vector (2D problem)



Murphy, Fig 8.1, p246

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

Linear and logistic probabilistic models

- Linear regression assumes a Normal distribution with a fixed variance and mean given by linear model

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

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

$$p(y|\boldsymbol{x}) = Bernoulli\left(y|\theta(\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{x}'\boldsymbol{w})}\right)$$

- 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 maximising 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

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

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

- "Training" amounts to maximising this expression with respect to weights $\boldsymbol{w}$

Old good trick

- "Log trick": Instead of maximising the likelihood, maximise its logarithm

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

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

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

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

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:

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

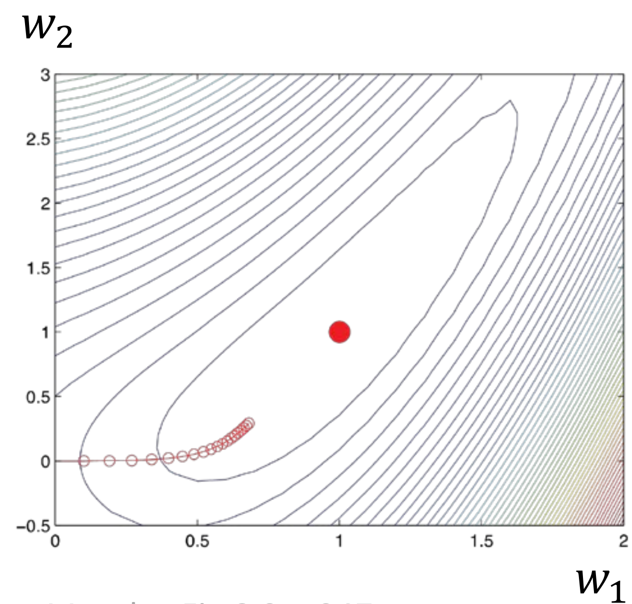
A is support of the distributions, e.g., $A = \{0,1\}$

Training as cross entropy minimisation

- Consider log-likelihood for a single data point
$$\log p(y_i|\boldsymbol{x_i}) = y_i \log\left(\theta(\boldsymbol{x}_i)\right) + (1 - y_i)\log\left(1 - \theta(\boldsymbol{x}_i)\right)$$
- This expression is the negative cross entropy
- Cross entropy $H\left(g_{ref}, g_{est}\right) = -\sum_a g_{ref}(a) \log g_{est}(a)$
- 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
$$g_{est}(1) = \theta(\boldsymbol{x}_i) \text{ and } g_{est}(0) = 1 - \theta(\boldsymbol{x}_i)$$

Notes on optimisation

- Training logistic regression amounts to finding $\boldsymbol{w}$ that maximise log-likelihood
  - Equivalently, finding $\boldsymbol{w}$ that minimise the sum of cross entropies for each training point
- The usual routine is to set derivatives of the objective function to zero and solve
- Bad news: There is no closed form solution, iterative methods are used instead (e.g., stochastic gradient descent)
- Good news: The problem is strictly convex (like a bowl) if there are no irrelevant features
- With irrelevant features, the problem is convex (like a ridge). Regularisation methods can be applied



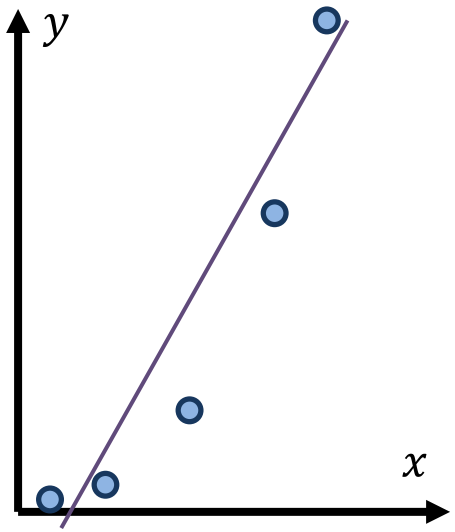
Murphy, Fig 8.3, p247

Basis Expansion

Extending the utility of models via data transformation

Basis expansion for linear regression

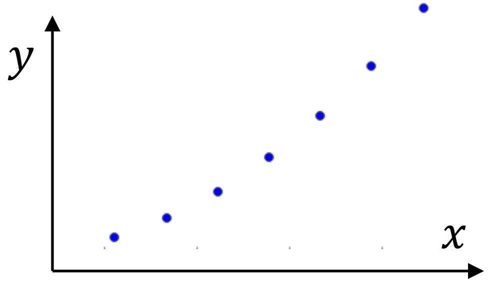
- Let's take a step back. Back to linear regression and least squares
- Real data is likely to be non-linear
- What if we still wanted to use a linear regression?
    * It's simple, easier to understand, computationally efficient, etc.
- How to marry non-linear data to a linear method?
- *if the mountain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



art: OpenClipartVectors at pixabay.com (CC0)

Transform the data

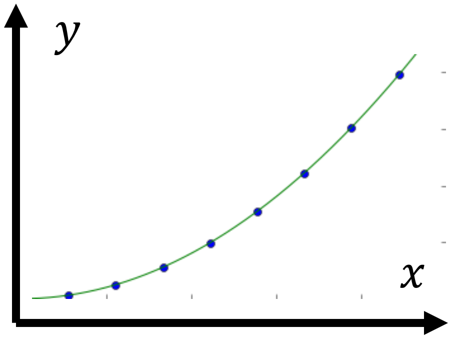
- The trick is to transform the data: Map the data onto another features space, such that the data is linear in that space
- Denote this transformation $\varphi: \mathbb{R}^m \rightarrow \mathbb{R}^k$. If $\boldsymbol{x}$ is the original set of features $\varphi(\boldsymbol{x})$ denotes the new set of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

$$\varphi_1(x) = x$$
$$\varphi_2(x) = x^2$$



- Next, apply linear regression to φ_1, φ_2

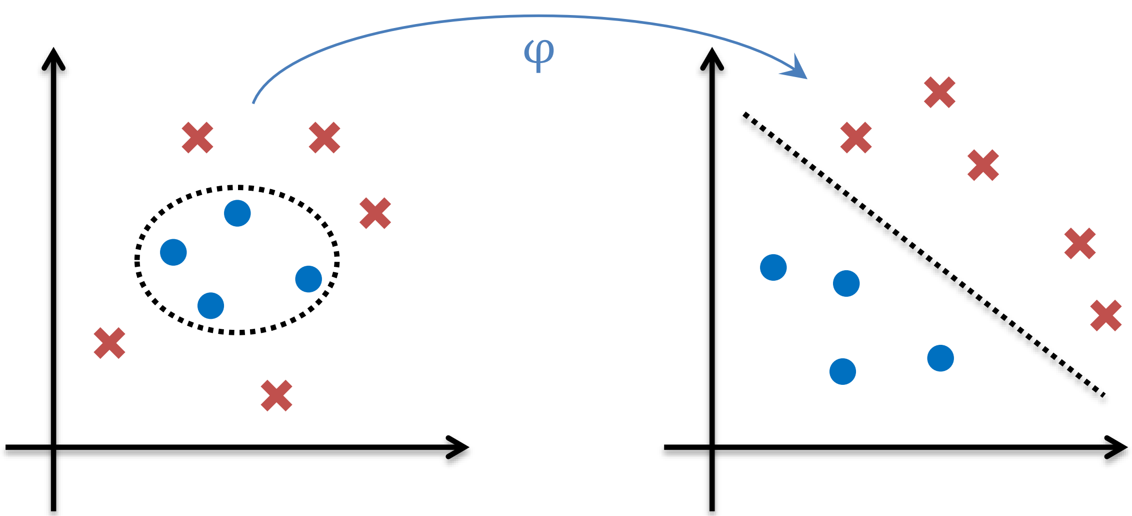
$$y = w_0 + w_1\varphi_1(x) + w_2\varphi_2(x) = w_0 + w_1 x + w_2 x^2$$

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



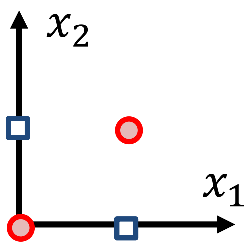
Basis expansion for logistic regression

- Consider the following binary classification problem (left). This dataset cannot be linearly separated

- Define transformation as

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

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



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

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

w_1	w_2	w_3	w_4
1	0	0	1

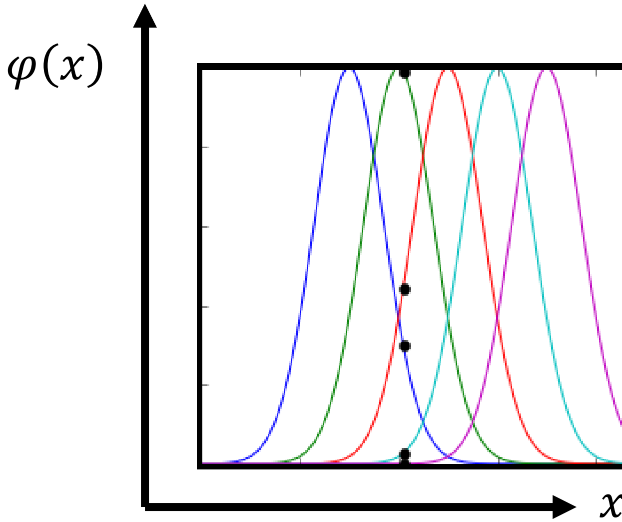
φ_1	φ_2	φ_3	φ_4
0	1	1	$\sqrt{2}$
1	0	$\sqrt{2}$	1
1	$\sqrt{2}$	0	1
$\sqrt{2}$	1	1	0

The transformed data is linearly separable!

$\boldsymbol{\varphi}'\boldsymbol{w}$	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 the 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 $\boldsymbol{z}_i$
- Regarding $\boldsymbol{z}_i$, one can choose uniformly space points, or cluster training data and use cluster centroids
- Another popular idea is to use training data $\boldsymbol{z}_i \equiv \boldsymbol{x}_i$
  - E.g., $\varphi_i(\boldsymbol{x}) = \psi(\|\boldsymbol{x} - \boldsymbol{x}_i\|)$
  - 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

This lecture

- Logistic regression
  - Binary classification problem
  - Logistic regression model
- Basis expansion
  - Examples for linear and logistic regression
  - Theoretical notes