

# Advanced Machine Learning



**QUEEN'S  
UNIVERSITY  
BELFAST**

Lesson 04: Classification

*k-nearest neighbour*

Vien Ngo  
EEECS, Queen's University Belfast  
Spring 2019

# **Classification**

# Classification

- **The simplest approach: k-nearest neighbour**
- Discriminate function
- Logistic regression for binary classification
- Multi-class classification

# Classification vs. Regression

**Regression**: assign each input  $x$  to

one (or more) **continuous/real**

**value**  $y$ .  $f : \mathbb{R}^d \mapsto \mathcal{R}$

# Classification vs. Regression

**Regression**: assign each input  $x$  to one (or more) **continuous/real value**  $y$ .  $f : \mathbb{R}^d \mapsto \mathcal{R}$

**Classification**: assign each input  $x$  to one (or more) **discrete** value  $y$ .  
 $f : \mathbb{R}^d \mapsto [0, 1, \dots, K]$  (assume  $K$  classes)

# Classification vs. Regression

**Regression**: assign each input  $x$  to one (or more) **continuous/real value**  $y$ .  $f : \mathbb{R}^d \mapsto \mathcal{R}$

**Classification**: assign each input  $x$  to one (or more) **discrete** value  $y$ .  
 $f : \mathbb{R}^d \mapsto [0, 1, \dots, K]$  (assume  $K$  classes)

Weather prediction

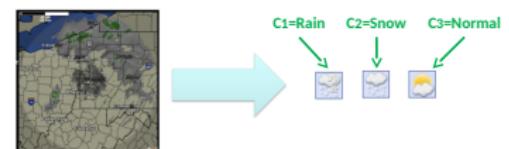


David Sonntag's lecture

$x = [\text{atmospheric pressure, humidity, wind speed, ...}]$

e.g.  $x = [20 \text{ Bars}, 0.85, 100\text{km/h}] \rightarrow y = 22.2^\circ\text{C}$

Weather prediction



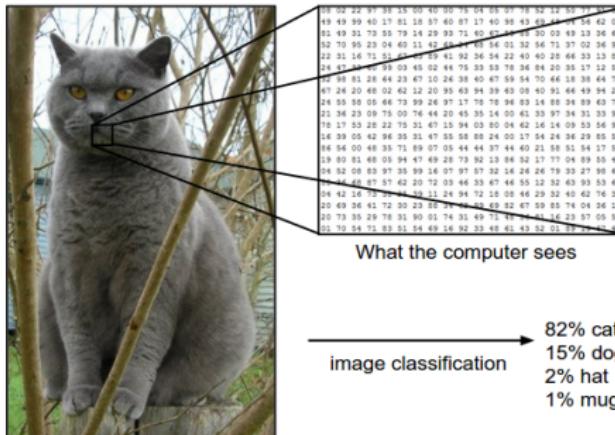
David Sonntag's lecture

$x = [\text{atmospheric pressure, humidity, wind speed, ...}]$

e.g.  $x = [20 \text{ Bars}, 0.85, 100\text{km/h}] \rightarrow \text{C1=Rain}$

# More examples of classification

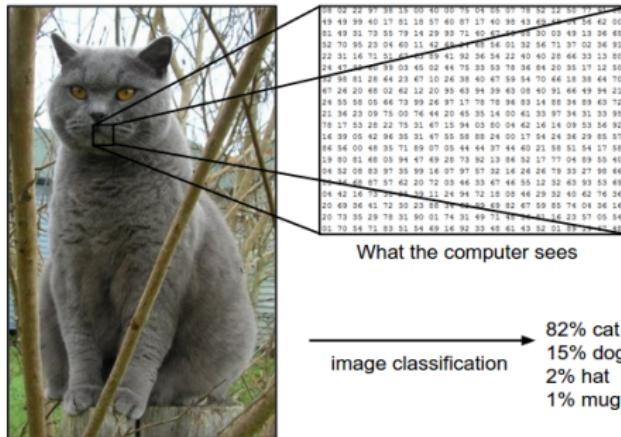
- image classification



Stanford's cs231n course

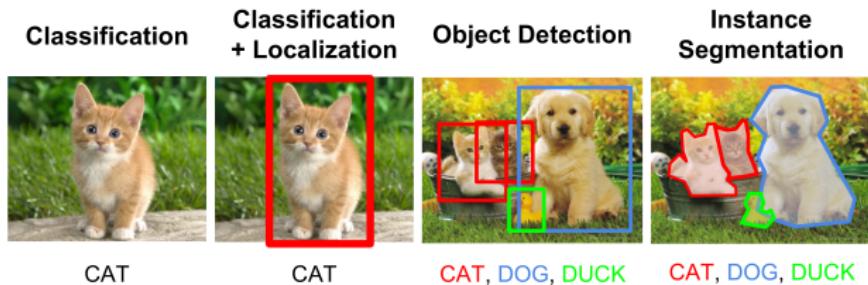
# More examples of classification

- image classification



Stanford's cs231n course

- classification + object detection



## Example: movie recommendation system

- A data-driven system to recommend a new movie: e.g. should I like *Gravity* if I know its rating and my own likes on some other movies?

Movie name	Mary's rating	John's rating	I like?
Lord of the Rings II	1	5	No
...	...	...	...
Star Wars I	4.5	4	Yes
Gravity	3	3	?

# Example: movie recommendation system

- A data-driven system to recommend a new movie: e.g. should I like *Gravity* if I know its rating and my own likes on some other movies?

Movie name	Mary's rating	John's rating	I like?
Lord of the Rings II	1	5	No
...	...	...	...
Star Wars I	4.5	4	Yes
Gravity	3	3	?

Le V. Quoc's tutorial

- **inputs**  $x_i$ : 2-dimensional ( $x_{i1}$  = Mary's rating,  $x_{i2}$  = John's rating)
  - $x_1 = [1, 5]$     $x_2 = [4.5, 4]$
- **outputs**  $y$ : Yes or No
  - $y_1 = \text{No}$     $y_2 = \text{Yes}$

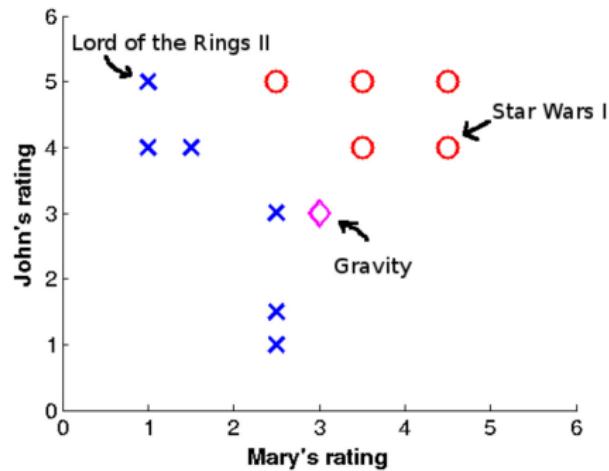
# Example: movie recommendation system

- A data-driven system to recommend a new movie: e.g. should I like *Gravity* if I know its rating and my own likes on some other movies?

Movie name	Mary's rating	John's rating	I like?
Lord of the Rings II	1	5	No
...	...	...	...
Star Wars I	4.5	4	Yes
Gravity	3	3	?

Le V. Quoc's tutorial

- **inputs**  $x_i$ : 2-dimensional ( $x_{i1}$  = Mary's rating,  $x_{i2}$  = John's rating)
  - $x_1 = [1, 5]$     $x_2 = [4.5, 4]$
- **outputs**  $y$ : Yes or No
  - $y_1 = \text{No}$     $y_2 = \text{Yes}$
- **predictions**: given a new  $x$ , predict the label of  $y$ 
  - $x = [3, 3]$ ,  $y = ?$  (I would like the movie Gravity or not?)

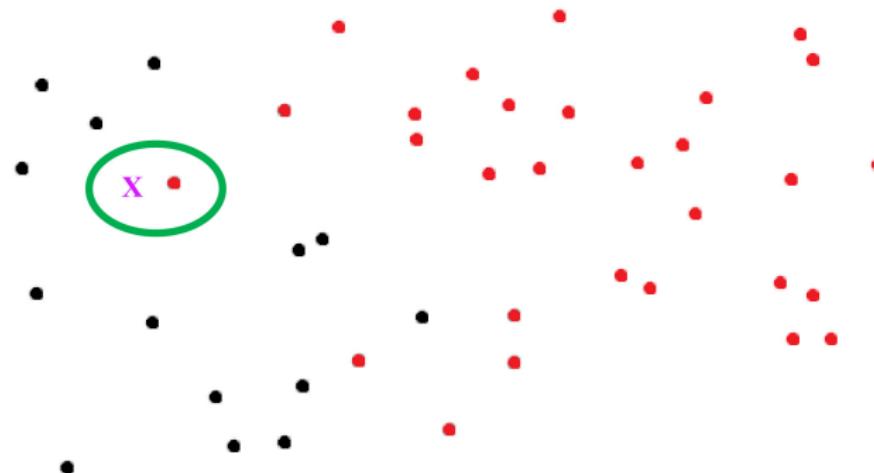


Le V. Quoc's tutorial

## **$k$ -nearest neighbour**

# Nearest neighbour method

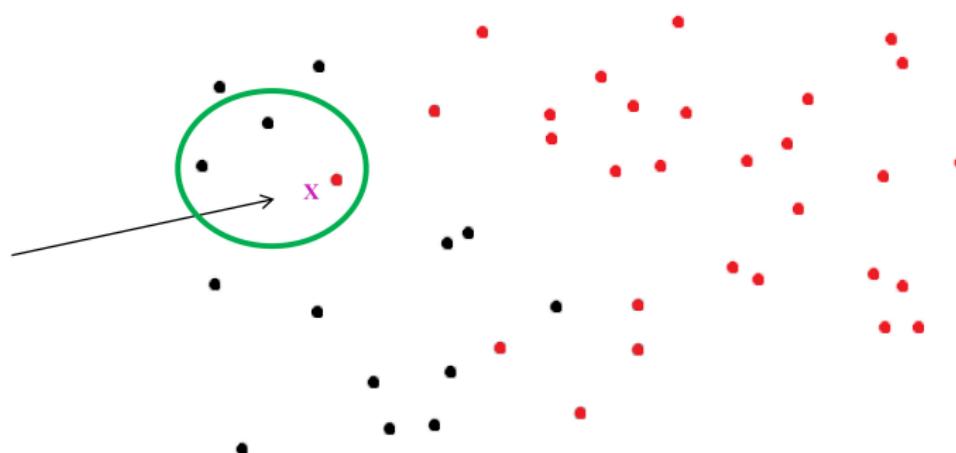
- nearest neighbour method = 1-nearest neighbour



MIT Course Number 6.0002

## $k$ -nearest neighbour method

- example:  $k = 3$  nearest neighbour



## *k*-nearest neighbour method: algorithm

- **Training:** just save all training examples (**EASY**: lazy learner)

## *k*-nearest neighbour method: algorithm

- **Training:** just save all training examples (**EASY**: lazy learner)
- **Predictions:** predict a label  $y$  for a new  $x$ 
  - Compute **distance** between  $x$  to all data points  $d(x, x_i)$

## *k*-nearest neighbour method: algorithm

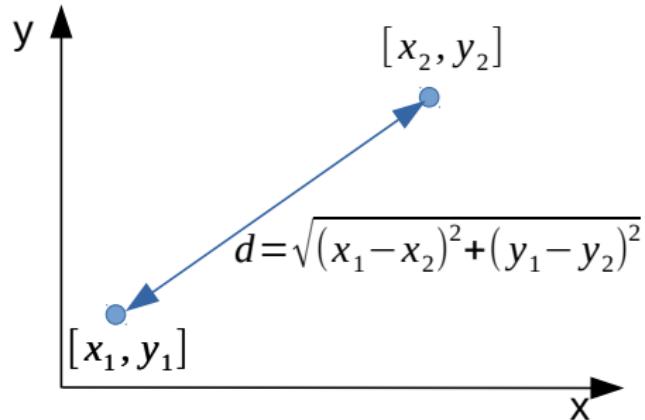
- **Training:** just save all training examples (**EASY**: lazy learner)
- **Predictions:** predict a label  $y$  for a new  $x$ 
  - Compute **distance** between  $x$  to all data points  $d(x, x_i)$
  - Select the ***k*-nearest neighbours** to  $x$  (in terms of  $d(x, x_i)$ )

## *k*-nearest neighbour method: algorithm

- **Training:** just save all training examples (**EASY**: lazy learner)
- **Predictions:** predict a label  $y$  for a new  $x$ 
  - Compute **distance** between  $x$  to all data points  $d(x, x_i)$
  - Select the ***k*-nearest neighbours** to  $x$  (in terms of  $d(x, x_i)$ )
  - Assign the label  $y$  by **the majority vote** in the set of  $k$ -nearest neighbours

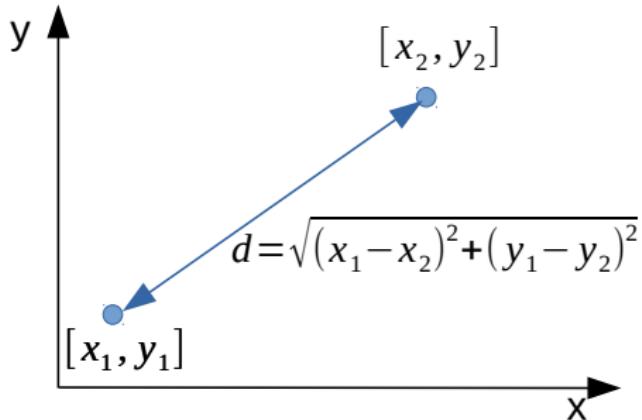
# distance measure

- a simple choice is **Euclidean distance**
  - Euclidean distance in 2D



# distance measure

- a simple choice is **Euclidean distance**
  - Euclidean distance in 2D



- Euclidean distance in  $\mathbb{R}^d$ , for example  $x \in \mathbb{R}^d$  and  $x' \in \mathbb{R}^d$   
where  $x = [x_1, x_2, \dots, x_d]^\top$ ,  $x' = [x'_1, x'_2, \dots, x'_d]^\top$

$$d(x, x') = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + \dots + (x_d - x'_d)^2}$$

## *k*-nearest neighbour: example

- a dataset for movie recommendation system

Movie name	Mom's rating	Dad's rating	I like?
The Godfather	1	5	Yes
Cassblanca	5	0.5	No
The Pianist	4	3	Yes
A Beautiful Mind	3	3	Yes
Gone with the Wind	3	1	No

- using nearest neighbour to predict whether if I like a new movie: *The Good, the Bad and the Ugly* if its ratings are 3.5 (from Mom) and 1.5 (from Dad) ?

## $k$ -nearest neighbour: example

- problem formulation:
  - the input  $x = [x_1, x_2]$  is:  $x_1$  is Mom's rating,  $x_2$  is Dad's rating.
  - output is  $y = \{\text{Yes}, \text{No}\}$
  - the dataset  $\mathcal{D}$  is  $\{x_i, y_i\}$  where

$$(x_{11} = 1, x_{12} = 5, y_1 = \text{Yes})$$

$$(x_{21} = 5, x_{22} = 0.5, y_2 = \text{No})$$

$$(x_{31} = 4, x_{32} = 3, y_3 = \text{Yes})$$

$$(x_{41} = 3, x_{42} = 3, y_4 = \text{Yes})$$

$$(x_{51} = 3, x_{52} = 1, y_5 = \text{No})$$

- predict  $y$  for  $x = [3.5, 1.5]$

# $k$ -nearest neighbour: example

- problem formulation:
  - the input  $x = [x_1, x_2]$  is:  $x_1$  is Mom's rating,  $x_2$  is Dad's rating.
  - output is  $y = \{\text{Yes}, \text{No}\}$
  - the dataset  $\mathcal{D}$  is  $\{x_i, y_i\}$  where

$$(x_{11} = 1, x_{12} = 5, y_1 = \text{Yes})$$

$$(x_{21} = 5, x_{22} = 0.5, y_2 = \text{No})$$

$$(x_{31} = 4, x_{32} = 3, y_3 = \text{Yes})$$

$$(x_{41} = 3, x_{42} = 3, y_4 = \text{Yes})$$

$$(x_{51} = 3, x_{52} = 1, y_5 = \text{No})$$

- predict  $y$  for  $x = [3.5, 1.5]$

- predict  $y$  for  $x = [3.5, 1.5]$

1. distances of  $x$  to all data points are

– to  $x_1$ :  $\sqrt{(3.5 - 1)^2 + (1.5 - 5)^2} = \sqrt{18.5}$

– to  $x_2$ :  $\sqrt{(3.5 - 5)^2 + (1.5 - 0.5)^2} = \sqrt{3.25}$

– to  $x_3$ :  $\sqrt{(3.5 - 4)^2 + (1.5 - 3)^2} = \sqrt{2.5}$

– to  $x_4$ :  $\sqrt{(3.5 - 3)^2 + (1.5 - 3)^2} = \sqrt{2.5}$

– to  $x_5$ :  $\sqrt{(3.5 - 3)^2 + (1.5 - 1)^2} = \sqrt{0.5}$

- predict  $y$  for  $x = [3.5, 1.5]$

1. distances of  $x$  to all data points are

- to  $x_1$ :  $\sqrt{(3.5 - 1)^2 + (1.5 - 5)^2} = \sqrt{18.5}$
- to  $x_2$ :  $\sqrt{(3.5 - 5)^2 + (1.5 - 0.5)^2} = \sqrt{3.25}$
- to  $x_3$ :  $\sqrt{(3.5 - 4)^2 + (1.5 - 3)^2} = \sqrt{2.5}$
- to  $x_4$ :  $\sqrt{(3.5 - 3)^2 + (1.5 - 3)^2} = \sqrt{2.5}$
- to  $x_5$ :  $\sqrt{(3.5 - 3)^2 + (1.5 - 1)^2} = \sqrt{0.5}$

- predict  $y$  for  $x = [3.5, 1.5]$ 
  1. distances of  $x$  to all data points are
    - to  $x_1$ :  $\sqrt{(3.5 - 1)^2 + (1.5 - 5)^2} = \sqrt{18.5}$
    - to  $x_2$ :  $\sqrt{(3.5 - 5)^2 + (1.5 - 0.5)^2} = \sqrt{3.25}$
    - to  $x_3$ :  $\sqrt{(3.5 - 4)^2 + (1.5 - 3)^2} = \sqrt{2.5}$
    - to  $x_4$ :  $\sqrt{(3.5 - 3)^2 + (1.5 - 3)^2} = \sqrt{2.5}$
    - to  $x_5$ :  $\sqrt{(3.5 - 3)^2 + (1.5 - 1)^2} = \sqrt{0.5}$
  2. assign label to  $x$ : nearest neighbour ( $k = 1$ ) is the nearest neighbor  $x_5$ , so the prediction is  $y = \text{No}$  for *The Good, the Bad and the Ugly*

- predict  $y$  for  $x = [3.5, 1.5]$ 
  1. distances of  $x$  to all data points are
    - to  $x_1$ :  $\sqrt{(3.5 - 1)^2 + (1.5 - 5)^2} = \sqrt{18.5}$
    - to  $x_2$ :  $\sqrt{(3.5 - 5)^2 + (1.5 - 0.5)^2} = \sqrt{3.25}$
    - to  $x_3$ :  $\sqrt{(3.5 - 4)^2 + (1.5 - 3)^2} = \sqrt{2.5}$
    - to  $x_4$ :  $\sqrt{(3.5 - 3)^2 + (1.5 - 3)^2} = \sqrt{2.5}$
    - to  $x_5$ :  $\sqrt{(3.5 - 3)^2 + (1.5 - 1)^2} = \sqrt{0.5}$
  2. assign label to  $x$ : nearest neighbour ( $k = 1$ ) is the nearest neighbor  $x_5$ , so the prediction is  $y = \text{No}$  for *The Good, the Bad and the Ugly*
- what happens if  $k = 3$ ?
  - the nearest neighbors are  $x_3, x_4, x_5$ , so the prediction is  $y = \text{Yes}$  (by majority vote: where  $y_3$  and  $y_4$  are Yes) for *The Good, the Bad and the Ugly*

## Exercise

- using 3-nearest neighbor: predict whether I would like or not a new movie *Thor: Ragnarok*, if its ratings are 2 (from Mom), and 4 (from Dad)

## Exercise

- using 3-nearest neighbor: predict whether I would like or not a new movie *Thor: Ragnarok*, if its ratings are 2 (from Mom), and 4 (from Dad)

**step 1:** distances of  $x = [2, 4]$  to all data points are

- to  $x_1$ :  $\sqrt{(2 - 1)^2 + (4 - 5)^2} = \sqrt{2}$
- to  $x_2$ :  $\sqrt{(2 - 5)^2 + (4 - 0.5)^2} = \sqrt{21.25}$
- to  $x_3$ :  $\sqrt{(2 - 4)^2 + (4 - 3)^2} = \sqrt{5}$
- to  $x_4$ :  $\sqrt{(2 - 3)^2 + (4 - 3)^2} = \sqrt{2}$
- to  $x_5$ :  $\sqrt{(2 - 3)^2 + (4 - 1)^2} = \sqrt{10}$

**step 2:** 3-nearest neighbours are  $x_1, x_3, x_4$  (where  $y_1 = Yes, y_3 = Yes, y_4 = Yes$ ), so  $y = Yes$  (I will like to see this movie *Thor: Ragnarok*)

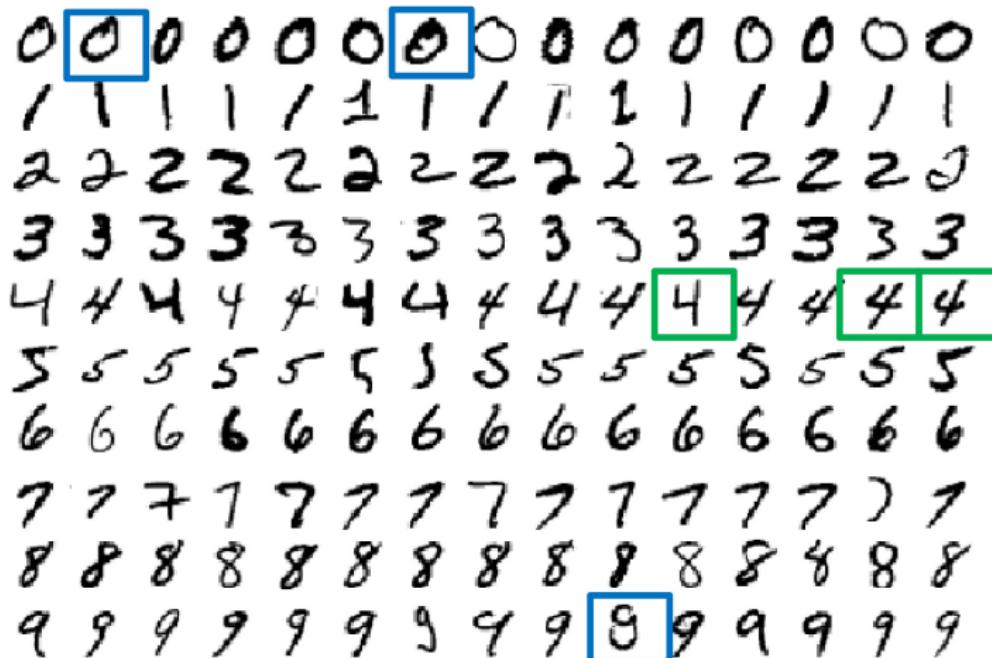
# Handwritten recognition example

Recognize number 40



## ***k*-nearest neighbour method**

## Handwritten recognition example



## *k*-nearest neighbour method: pros/cons

### Pros/Cons

- fast learning/training, no explicit training (**lazy learner**)

## *k*-nearest neighbour method: pros/cons

### Pros/Cons

- fast learning/training, no explicit training (**lazy learner**)
- Easy explanation/intuition

## *k*-nearest neighbour method: pros/cons

### Pros/Cons

- fast learning/training, no explicit training (**lazy learner**)
- Easy explanation/intuition
- Memory intensive and predictions can take a long time (curse of dimensionality)
- No explicit model built, **slow in prediction**
- Sensitive to  $k$ : use cross-validation to select  $k$  during training

## $k$ -nearest neighbour method: pros/cons

### Pros/Cons

- fast learning/training, no explicit training (**lazy learner**)
- Easy explanation/intuition
- Memory intensive and predictions can take a long time (curse of dimensionality)
- No explicit model built, **slow in prediction**
- Sensitive to  $k$ : use cross-validation to select  $k$  during training