Lecture 11 -- Linear and Logistic Regression

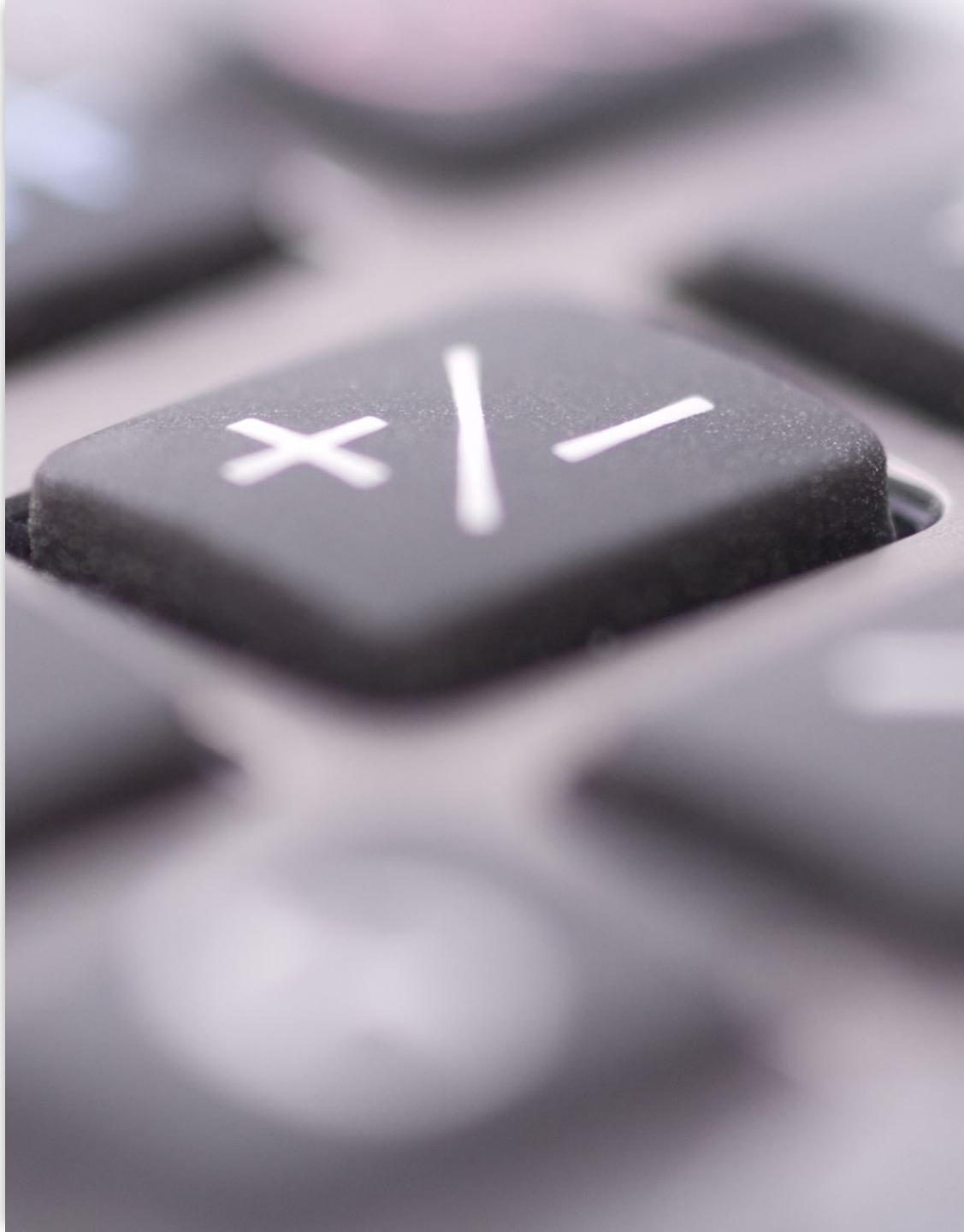
Prof. Xiaowei Huang

https://cgi.csc.liv.ac.uk/~xiaowei/

(Attendance Code: **825019**)

Up to now,

- Two Classical Machine Learning Algorithms
  - Decision tree learning
  - K-nearest neighbor
    - What is k-nearest-neighbor classification
    - How can we determine similarity/distance
    - Standardizing numeric features
    - Speeding up k-NN
      - edited nearest neighbour
      - k-d trees for nearest neighbour identification



Today's Topics

Indictive Bias

linear regression

mean square error

linear classification

0-1 loss

logistic regression

Sigmoid

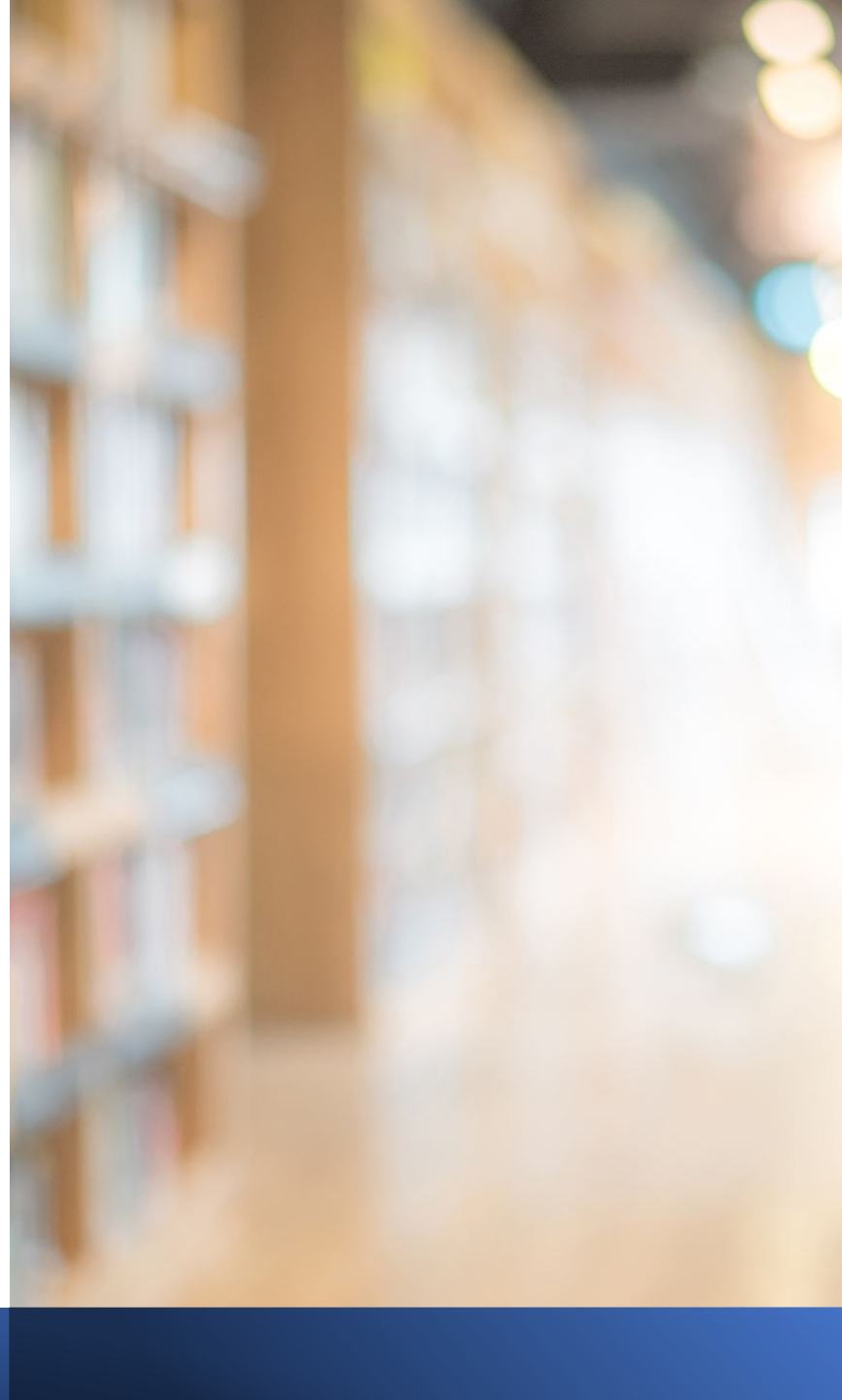
Probability as confidence

Log function

An example to learn how to design objective function.

Inductive bias

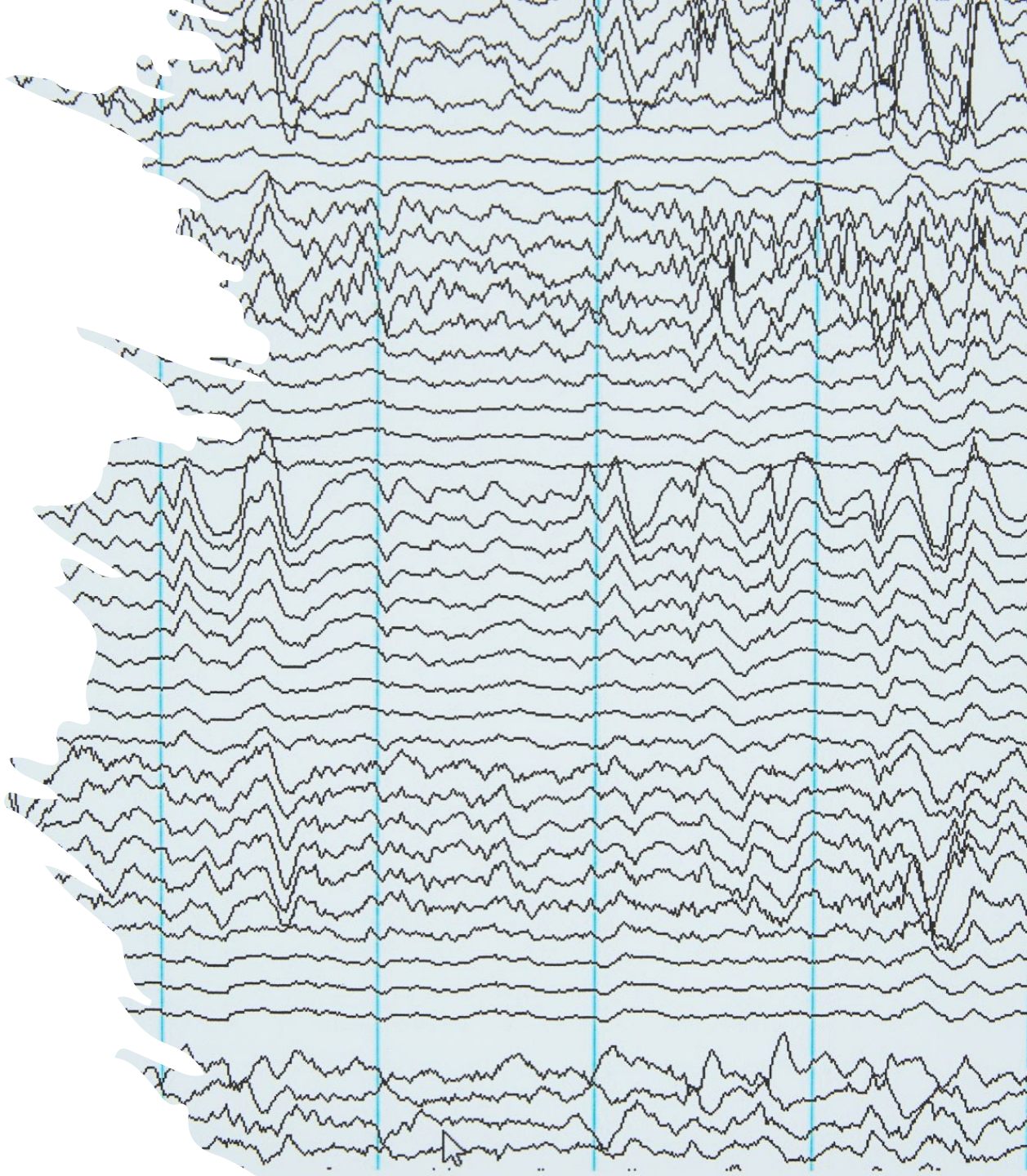
- *inductive bias* is the set of assumptions a learner uses to be able to predict y_i for a previously unseen instance x_i
- two components
  - *hypothesis space bias*: determines the models that can be represented
  - *preference bias*: specifies a preference ordering within the space of models
- in order to *generalize* (i.e. make predictions for previously unseen instances) a learning algorithm must have an inductive bias



Consider the inductive bias of DT and k-NN learners

learner	hypothesis space bias	preference bias
ID3 decision tree	trees with single-feature, axis-parallel splits	small trees identified by greedy search
k-NN	Voronoi decomposition determined by nearest neighbors	instances in neighborhood belong to same class

Linear regression



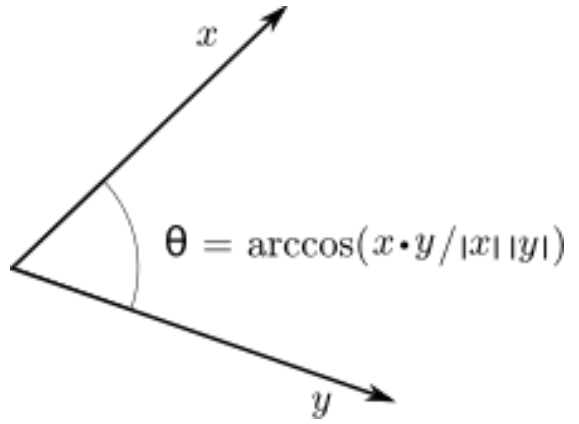
Recap: dot product in linear algebra

Dot product is a measure of how closely two vectors align, in terms of the directions they point.

$$f_w(x) = w^T x$$

$$w = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \quad x = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$$

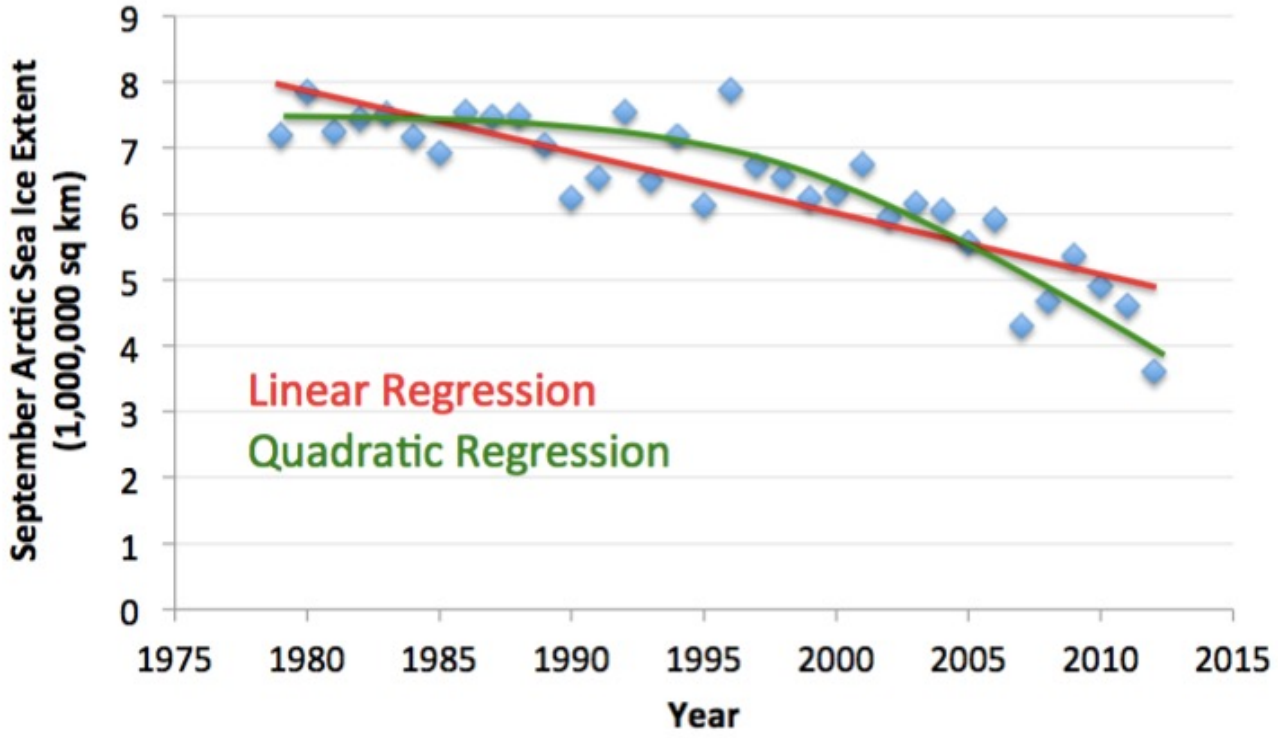
$$w^T x = 2 * 1 + 3 * 4 = 14$$



Geometric meaning: can be used to understand the angle between two vectors

Linear regression, illustrated

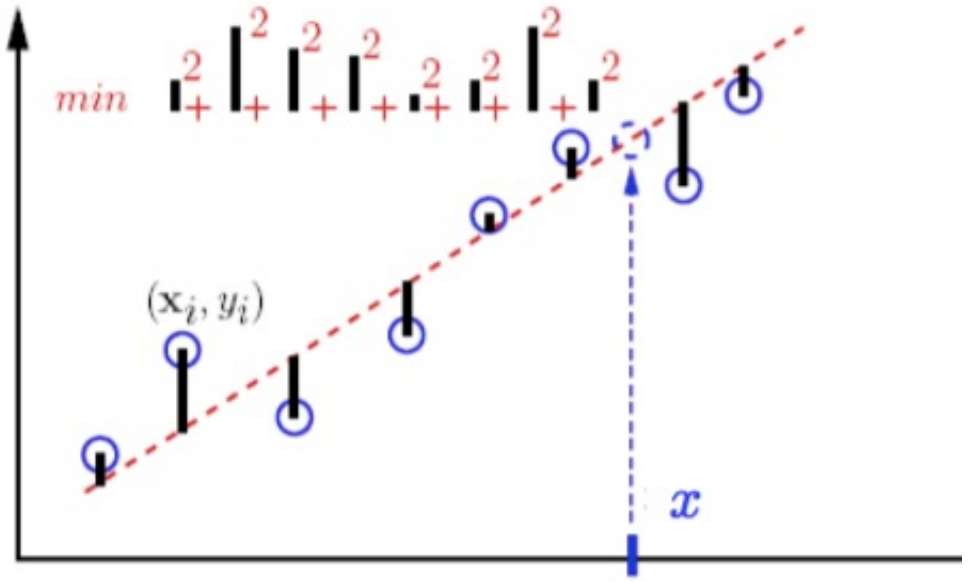
- Given training data $\{(x^{(i)}, y^{(i)}) : 1 \leq i \leq m\}$ i.i.d. from distribution D

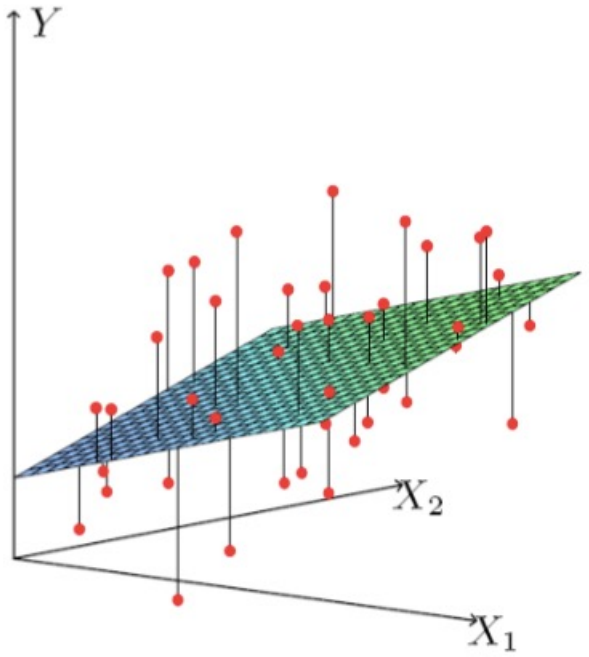


Use curves of a specific shape to fit the data

Linear regression, illustrated

- Given training data $\{(x^{(i)}, y^{(i)}) : 1 \leq i \leq m\}$ i.i.d. from distribution D
- Find $f_w(x) = w^T x$ that minimises $\hat{L}(f_w) = \frac{1}{m} \sum_{i=1}^{m} (w^T x^{(i)} - y^{(i)})^2$





Definition of linear regression

- Given training data $\{(x^{(i)}, y^{(i)}) : 1 \leq i \leq m\}$ i.i.d. from distribution D
- Find $f_w(x) = w^T x$ that minimises

$$\hat{L}(f_w) = \frac{1}{m} \sum_{i=1}^{m} (w^T x^{(i)} - y^{(i)})^2$$

Hypothesis Class H

L^2 loss, or mean square error

Definition of linear regression

- Given training data $\{(x^{(i)}, y^{(i)}) : 1 \leq i \leq m\}$ i.i.d. from distribution D
- Find $f_w(x) = w^T x$ that minimises

$$\hat{L}(f_w) = \frac{1}{m} \sum_{i=1}^{m} (w^T x^{(i)} - y^{(i)})^2$$

where

- $w^T x^{(i)} - y^{(i)}$ represents the error of instance $x^{(i)}$
- $\sum_{i=1}^{m} (w^T x^{(i)} - y^{(i)})^2$ represents the square error of all training instances

So, $\frac{1}{m} \sum_{i=1}^{m} (w^T x^{(i)} - y^{(i)})^2$ represents the mean square error of all training instances

Definition of linear regression

- Given training data $\{(x^{(i)}, y^{(i)}) : 1 \leq i \leq m\}$ i.i.d. from distribution D
- Find $f_w(x) = w^T x$ that minimises $\hat{L}(f_w) = \frac{1}{m}\sum_{i=1}^{m}(w^T x^{(i)} - y^{(i)})^2$

- Let X be a matrix whose i-th row is $(x^{(i)})^T$, y be the vector $(y^{(1)}, ..., y^{(m)})^T$

$$\hat{L}(f_w) = \frac{1}{m}\sum_{i=1}^{m}(w^T x^{(i)} - y^{(i)})^2 = \frac{1}{m}||Xw - y||_2^2$$

We will use example to explain this!

Solving this optimization problem will be introduced in later lectures.

Example

- Let X be a matrix whose i-th row is $(x^{(i)})^T$

$$x^{(1)} = \begin{bmatrix} 182 \\ 87 \\ 11.3 \end{bmatrix} \quad x^{(2)} = \begin{bmatrix} 189 \\ 92 \\ 12.3 \end{bmatrix} \quad x^{(3)} = \begin{bmatrix} 178 \\ 79 \\ 10.6 \end{bmatrix} \quad x^{(4)} = \begin{bmatrix} 183 \\ 90 \\ 12.7 \end{bmatrix} \quad y = \begin{bmatrix} 325 \\ 344 \\ 350 \\ 320 \end{bmatrix}$$

Football player example:
(height, weight, runningspeed)

Assume a weight vector $w = (1, -1, 20)$

Note: for linear regression, this is the parameter vector we want to learn. Here, we provide an example for illustration.

Question: how to compute loss?

Why care about loss? The smaller the loss, the better the weight vector.

Method 1: Compute loss individually

- Assume a function $f_w(x) = w^T x$ with weight vector $w = (1, -1, 20)$

$$w^T x^{(1)} = \begin{bmatrix} 1 & -1 & 20 \end{bmatrix} * \begin{bmatrix} 182 \\ 87 \\ 11.3 \end{bmatrix} = 321$$

$$w^T x^{(2)} = \begin{bmatrix} 1 & -1 & 20 \end{bmatrix} * \begin{bmatrix} 189 \\ 92 \\ 12.3 \end{bmatrix} = 343.0$$

$$w^T x^{(3)} = 311 \qquad w^T x^{(4)} = 347$$

And then, compute $\sum_{i=1}^{m} (w^T x^{(i)} - y^{(i)})^2$

where $y = \begin{bmatrix} 325 \\ 344 \\ 350 \\ 320 \end{bmatrix}$

Method 2, Step 1: Organise feature data into matrix

- Let X be a matrix whose i-th row is $\left(x^{(i)}\right)^T$

$$x^{(1)} = \begin{bmatrix} 182 \\ 87 \\ 11.3 \end{bmatrix} \quad x^{(2)} = \begin{bmatrix} 189 \\ 92 \\ 12.3 \end{bmatrix} \quad x^{(3)} = \begin{bmatrix} 178 \\ 79 \\ 10.6 \end{bmatrix} \quad x^{(4)} = \begin{bmatrix} 183 \\ 90 \\ 12.7 \end{bmatrix}$$

Football player example:
(height, weight, runningspeed)

$$X = \begin{bmatrix} 182 & 87 & 11.3 \\ 189 & 92 & 12.3 \\ 178 & 79 & 10.6 \\ 183 & 90 & 12.7 \end{bmatrix}$$

v1	v2	v3	y
182	87	11.3	325 (No)
189	92	12.3	344 (Yes)
178	79	10.6	350 (Yes)
183	90	12.7	320 (No)

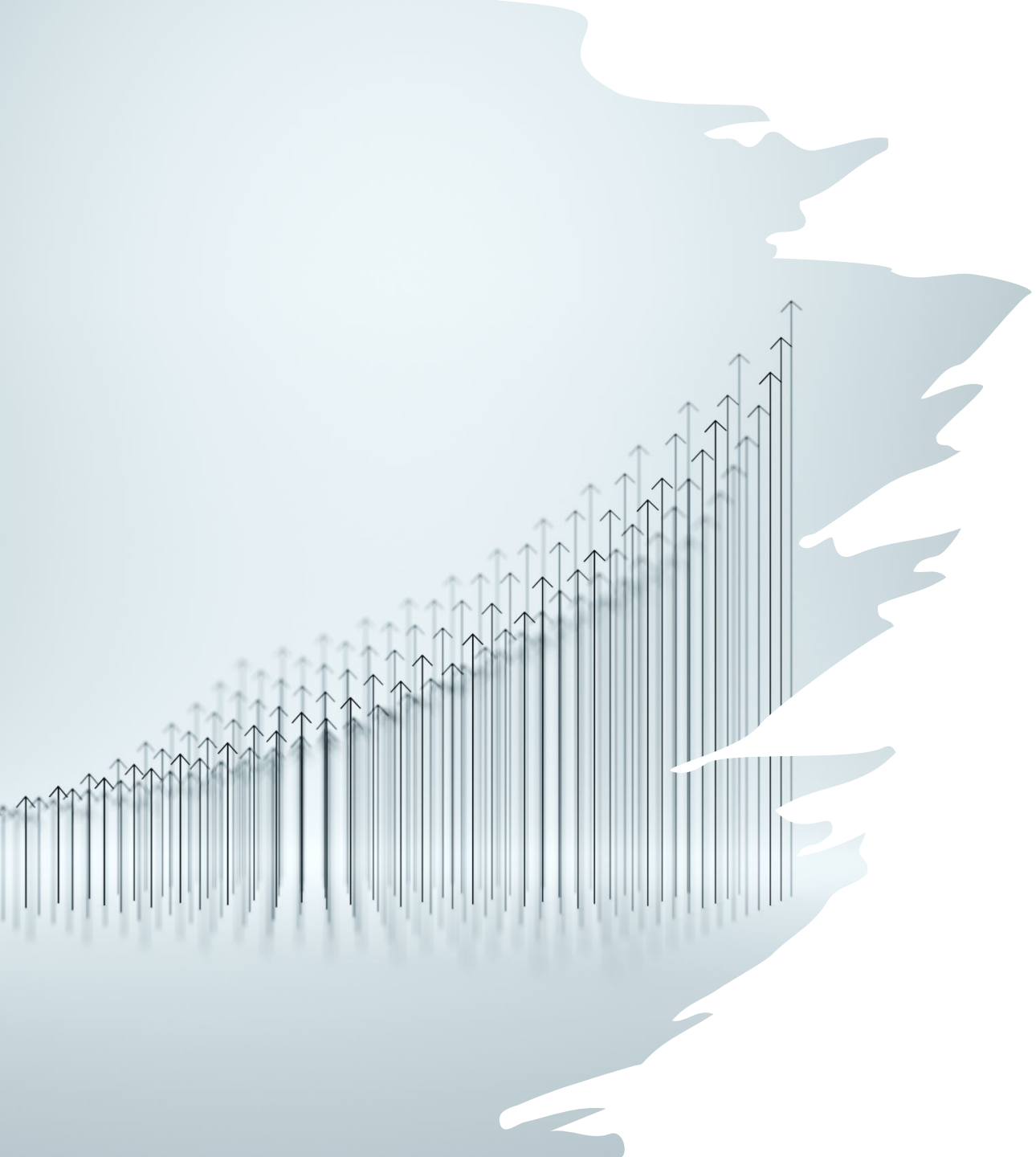
Method 2, Step 2: Matrix Computation

Compute $Xw = \begin{bmatrix} 321 \\ 343 \\ 311 \\ 347 \end{bmatrix}$

Then, by $y = \begin{bmatrix} 325 \\ 344 \\ 350 \\ 320 \end{bmatrix}$, we compute $||Xw - y||_2^2$

We can check that

$$\sum_{i=1}^{m} (w^T x^{(i)} - y^{(i)})^2 = ||Xw - y||_2^2$$



Variant: Linear regression with bias

Linear regression with bias

- Given training data $\{(x^{(i)}, y^{(i)}) : 1 \leq i \leq m\}$ i.i.d. from distribution D
- Find $f_w(x) = w^T x + b$ that minimises the loss

Bias Term

- Reduce to the case without bias:
  - Let $w' = [w; b], x' = [x; 1]$
  - Then $f_{w,b}(x) = w^T x + b = (w')^T (x')$

Intuitively, every instance is extended with one more feature whose value is always 1, and we already know the weight for this feature, i.e., b

Linear regression with bias

- Think about bias $b = -330$ for the football player example

Then, we have $X' = \begin{bmatrix} 182 & 87 & 11.3 & 1 \\ 189 & 92 & 12.3 & 1 \\ 178 & 79 & 10.6 & 1 \\ 183 & 90 & 12.7 & 1 \end{bmatrix}$ $w' = [1, -1, 20, -330]$

Finally, $X'w' = \begin{bmatrix} -9 \\ 13 \\ -19 \\ 17 \end{bmatrix}$

Can do a bit of exercise on this.

Variant: Linear regression with lasso penalty

Linear regression with lasso penalty

- Given training data $\{(x^{(i)}, y^{(i)}) : 1 \leq i \leq m\}$ i.i.d. from distribution D
- Find $f_w(x) = w^T x + b$ that minimises the loss

$$\hat{L}(f_w) = \frac{1}{m} \sum_{i=1}^{m} (w^T x^{(i)} - y^{(i)})^2 + \lambda |w|_1$$

lasso penalty: L^1 norm of the parameter, encourages sparsity

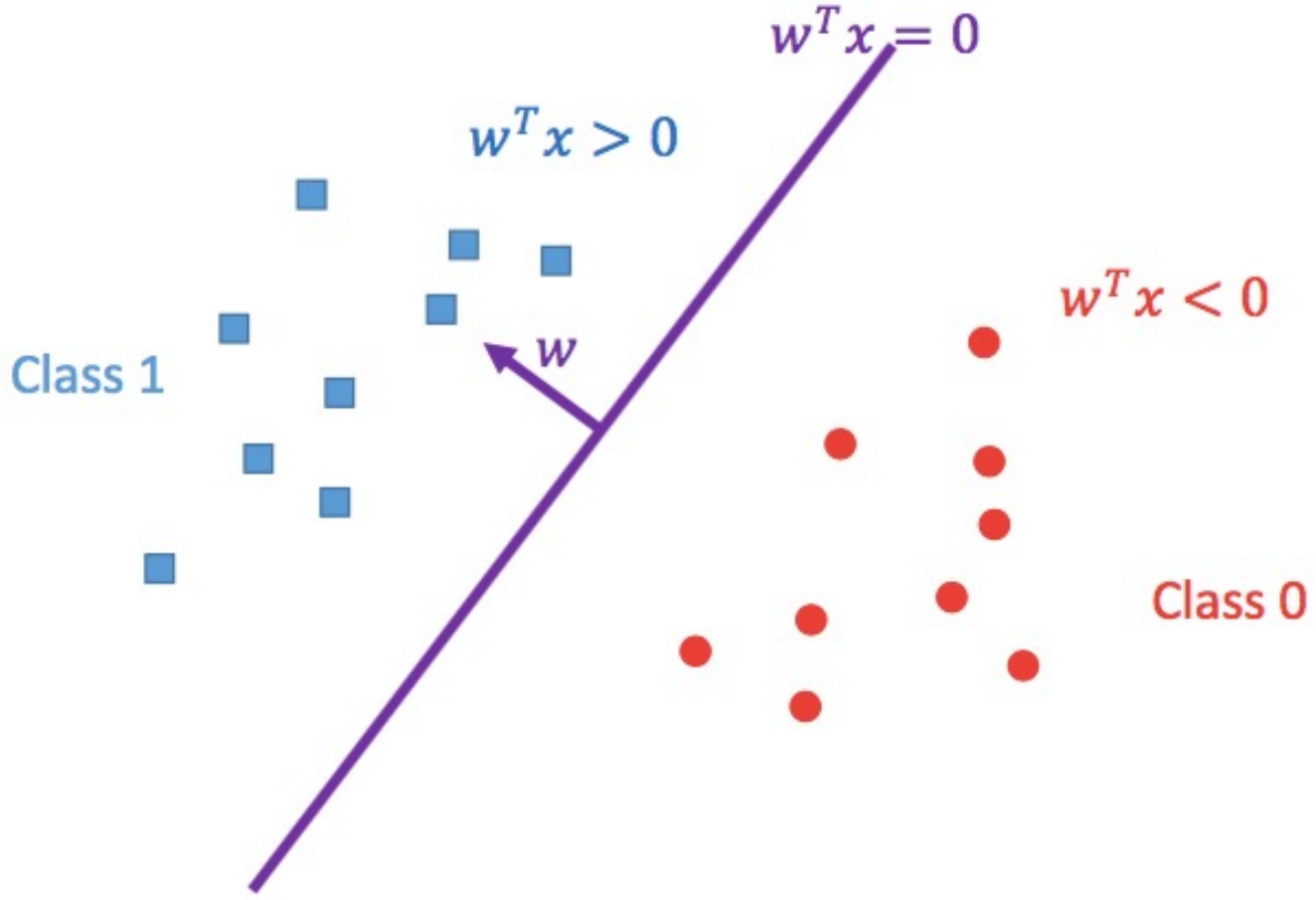
Variant: Evaluation Metrics

Evaluation Metrics

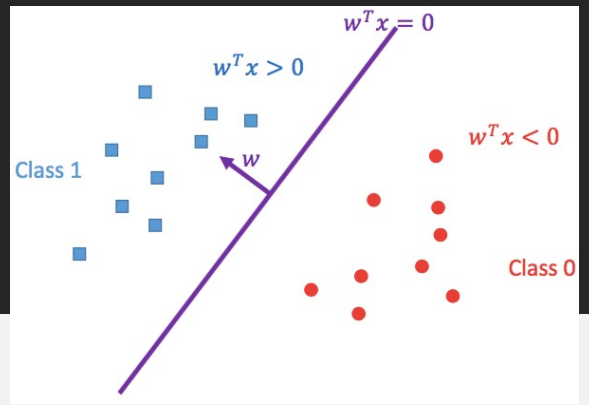
- Root mean squared error (RMSE)
- Mean absolute error (MAE) – average L^1 error
- R-square (R-squared)
- Historically all were computed on training data, and possibly adjusted after, but really should cross-validate

Linear classification

Linear classification: natural attempt



Linear classification: natural attempt



- Given training data $\{(x^{(i)}, y^{(i)}) : 1 \leq i \leq m\}$ i.i.d. from distribution D
- Hypothesis $f_w(x) = w^T x$
  - y = 1 if $w^T x > 0$
  - y = 0 if $w^T x < 0$

Piecewise Linear model $\mathcal{H}$

Or more formally, let $y = step(f_w(x)) = step(w^T x)$

where step(m) = 1, if m > 0 and
step(m) = 0, otherwise

Still, *w* is the vector of parameters to be trained.

But what is the optimization objective?

Linear classification: natural attempt

- Given training data $\{(x^{(i)}, y^{(i)}) : 1 \leq i \leq m\}$ i.i.d. from distribution D
- Find $f_w(x) = w^T x$ that minimises

$$\hat{L}(f_w) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{I}[step(w^T x^{(i)}) \neq y^{(i)}]$$

- Drawback: difficult to optimize
  - Non-differentiable
  - NP-hard in the worst case

0-1 loss

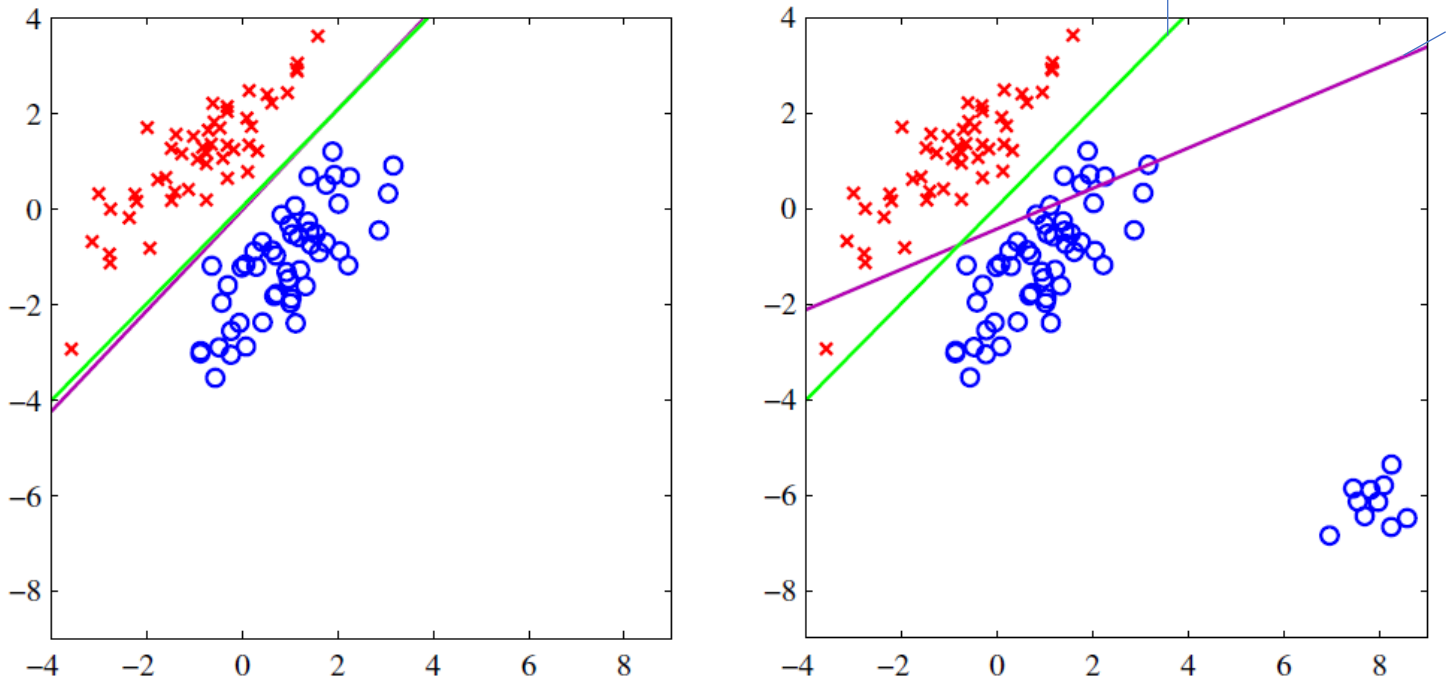
loss = 0, i.e., no loss, when the classification is the same as its label.

loss =1, otherwise.

Linear classification

Green: logistic regression (to be introduced later)

Magenta: linear classification



Drawback: not robust to “outliers”

So, linear classification is probably not the right scheme for classification

Figure 4.4 The left plot shows data from two classes, denoted by red crosses and blue circles, together with the decision boundary found by least squares (magenta curve) and also by the logistic regression model (green curve), which is discussed later in Section 4.3.2. The right-hand plot shows the corresponding results obtained when extra data points are added at the bottom left of the diagram, showing that least squares is highly sensitive to outliers, unlike logistic regression.

logistic regression

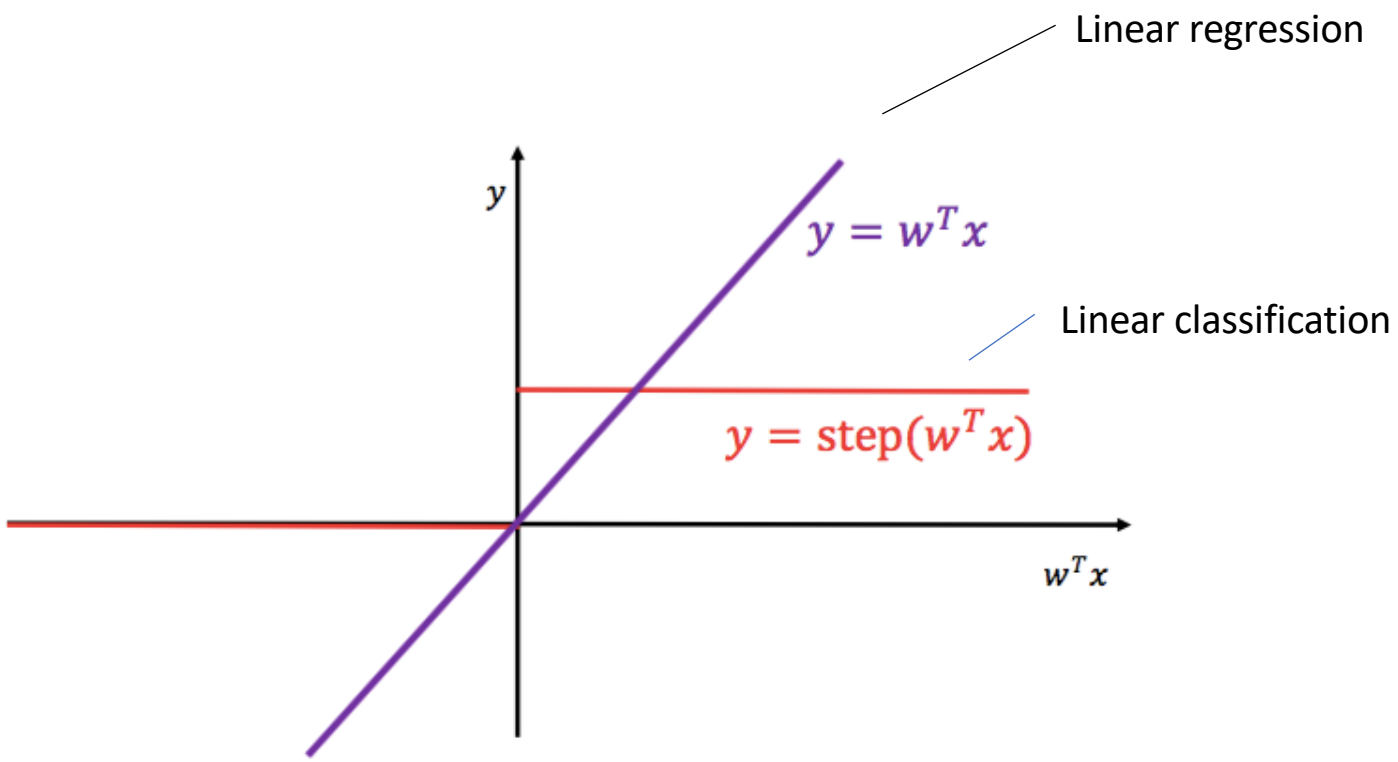
Why logistic regression?

- Goal: The entire procedure of pursuing logistic regression in the below slides are to find output probabilities.

Starting point

- It's tempting to use the linear regression output as probabilities.
- but it's a mistake because the output can be negative and greater than 1, whereas probability cannot.

Compare the two

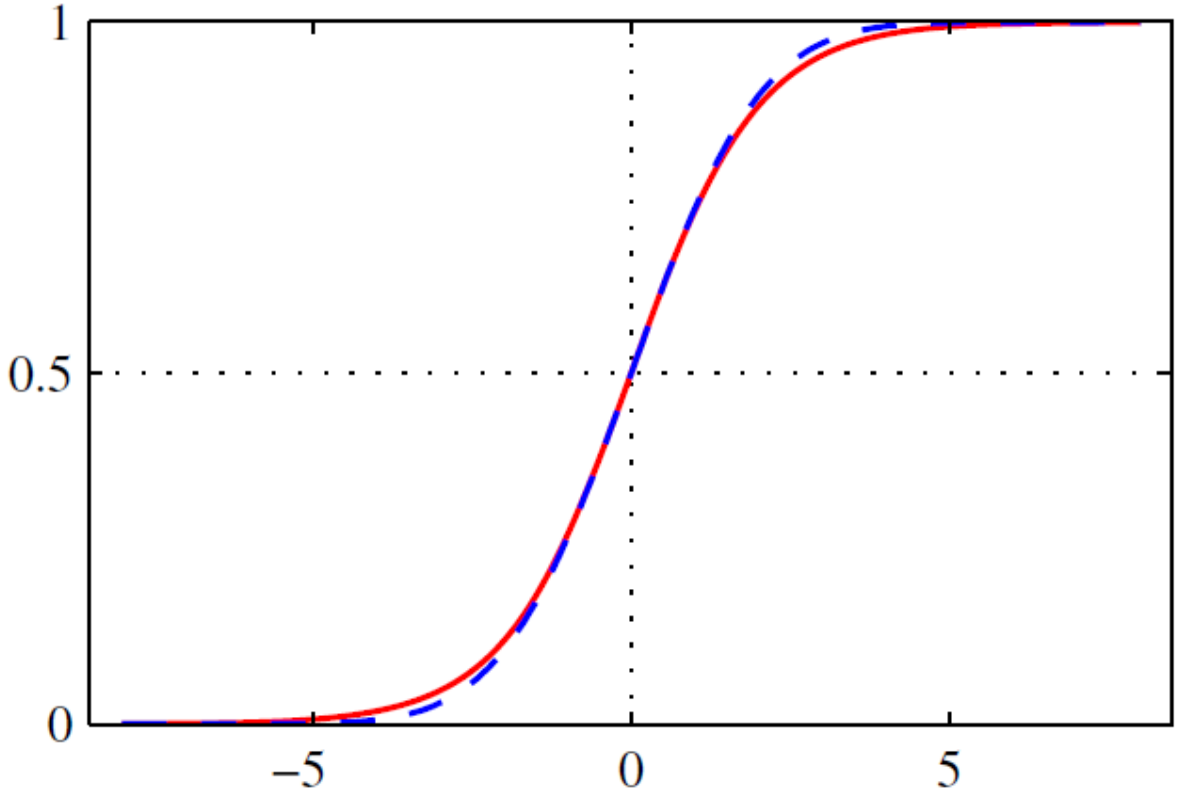


Between the two

- Prediction bounded in [0,1]
- Smooth

- Sigmoid:

$$\sigma(a) = \frac{1}{1+exp(-a)}$$



Linear regression: sigmoid prediction

- Squash the output of the linear function

$$Sigmoid(w^T x) = \sigma(w^T x) = \frac{1}{1+exp(-w^T x)}$$

New optimization objective

$$\hat{L}(f_w) = \frac{1}{m}\sum_{i=1}^{m}(\sigma(w^T x^{(i)}) - y^{(i)})^2$$

First step

Linear classification: logistic regression

$$\hat{L}(f_w) = \frac{1}{m}\sum_{i=1}^{m}(\sigma(w^T x^{(i)}) - y^{(i)})^2$$

- Is this the final?

We need a probability!

Second step

If $y^{(i)}$ is either 0 or 1, can we interpret $\sigma(w^T x^{(i)})$ as a probability value?

Linear classification: logistic regression

Second step

- A better approach: Interpret as a probability

Here we assume that y=0 or y=1

$$P_w(y = 1|x) = \sigma(w^T x)$$

$$P_w(y = 0 \mid x) = 1 - P_w(y = 1 \mid x) = 1 - \sigma(w^T x)$$

Conditional probability

Linear classification: logistic regression

- Find $f_w(x) = w^T x$ that minimises

Third step

$$\hat{L}(f_w) = -\frac{1}{m}\sum_{i=1}^{m} \log P_w(y^{(i)}|x^{(i)})$$

Why log function used? To avoid numerical instability.

unfold

$$\hat{L}(f_w) = -\frac{1}{m}\sum_{y^{(i)}=1} \log \sigma(w^T x^{(i)}) - \frac{1}{m}\sum_{y^{(i)}=0} \log[1 - \sigma(w^T x^{(i)})]$$

Increase this one to make it as close as possible to 1

Decrease this one to make it as close as possible to 0

Logistic regression:

MLE (maximum likelihood estimation) with sigmoid

Linear classification: logistic regression

- Given training data $\{(x^{(i)}, y^{(i)}) : 1 \leq i \leq m\}$ i.i.d. from distribution D
- Find w that minimises

$$\hat{L}(f_w) = \frac{1}{m} \sum_{y^{(i)}=1} \log \sigma(w^T x^{(i)}) - \frac{1}{m} \sum_{y^{(i)}=0} \log[1 - \sigma(w^T x^{(i)})]$$

Achieved

No close form solution;

Need to use gradient descent, which will be introduced in the next lecture

Why sigmoid function?

- Bounded

$$\sigma(a) = \frac{1}{1 + \exp(-a)} \in (0,1)$$

- Symmetric

$$1 - \sigma(a) = \frac{\exp(-a)}{1 + \exp(-a)} = \frac{1}{\exp(a) + 1} = \sigma(-a)$$

- Gradient

$$\sigma'(a) = \frac{\exp(-a)}{(1 + \exp(-a))^2} = \sigma(a)(1 - \sigma(a))$$

Exercises

Check the exercises of the lecture notes for answer.

- Given the dataset and consider the mean square root error, if we have the following two linear functions:
  - $f_w(x) = 2x_1 + 1x_2 + 20x_3 - 330$
  - $f_w(x) = 1x_1 - 2x_2 + 23x_3 - 332$

  please answer the following questions:
  - (1) which model is better for linear regression?
  - (2) which model is better for linear classification by considering 0-1 loss for $y^T=(0,1,1,0)$?
  - (3) which model is better for logistic regression for $y^T=(0,1,1,0)$?
  - (4) According to the logistic regression of the first model, what is the prediction result of the first model on a new input (181,92,12.4).

x1	x2	x3	y
182	87	11.3	325
189	92	12.3	344
178	79	10.6	350
183	90	12.7	320