Linear and Logistic Regression

Dr. Xiaowei Huang

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

Up to now,

- Two Classical Machine Learning Algorithms
  - Decision tree learning
  - K-nearest neighbor
- Model Evaluation Metrics
  - Learning curves
  - Training/Validation/Test datasets
  - Confusion matrices (accuracy, error, ROC curve, PR curve)

Confidence for decision tree (example)

- Random forest:
  - multiple decision trees are trained, by using different resamples of your data.
  - Probabilities can be calculated by the proportion of decision trees which vote for each class.

- For example, if 8 out of 10 decision trees vote to classify an instance as positive, we say that the confidence is 8/10.

Here, the confidences of all classes add up to 1

Confidence for k-NN classification (example)

- Classification steps are the same, recall $\hat{y} \leftarrow \underset{v \in \text{values}(Y)}{\text{argmax}} \sum_{i=1}^{k} \delta(v, y^{(i)})$
- Given a class $\hat{y}$, we compute

$$acc\_dist = \sum_{i=1}^{k} \delta(\hat{y}, y^{(i)}) \cdot distance(\hat{y}, y^{(i)})$$

Accumulated distance to the supportive instances

- apply sigmoid function on the reciprocal of the accumulated distance

$$confidence = \frac{1}{1 + e^{-\frac{1}{acc\_dist}}}$$

Here, the confidences of all classes may not add up to 1

Softmax?

Today’s Topics

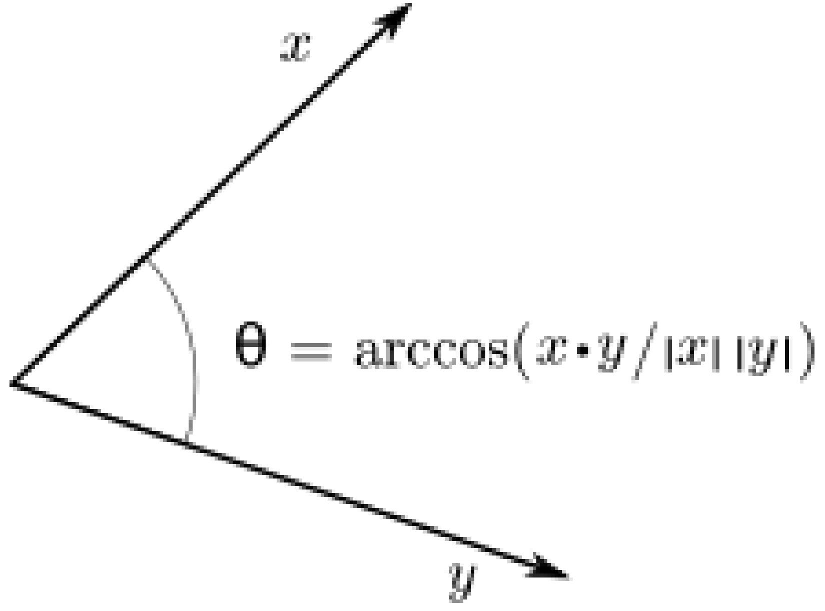
- linear regression
- linear classification
- logistic regression
- multiclass logistic regression (leave as extended materials)

Recap: dot product in linear algebra

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

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

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

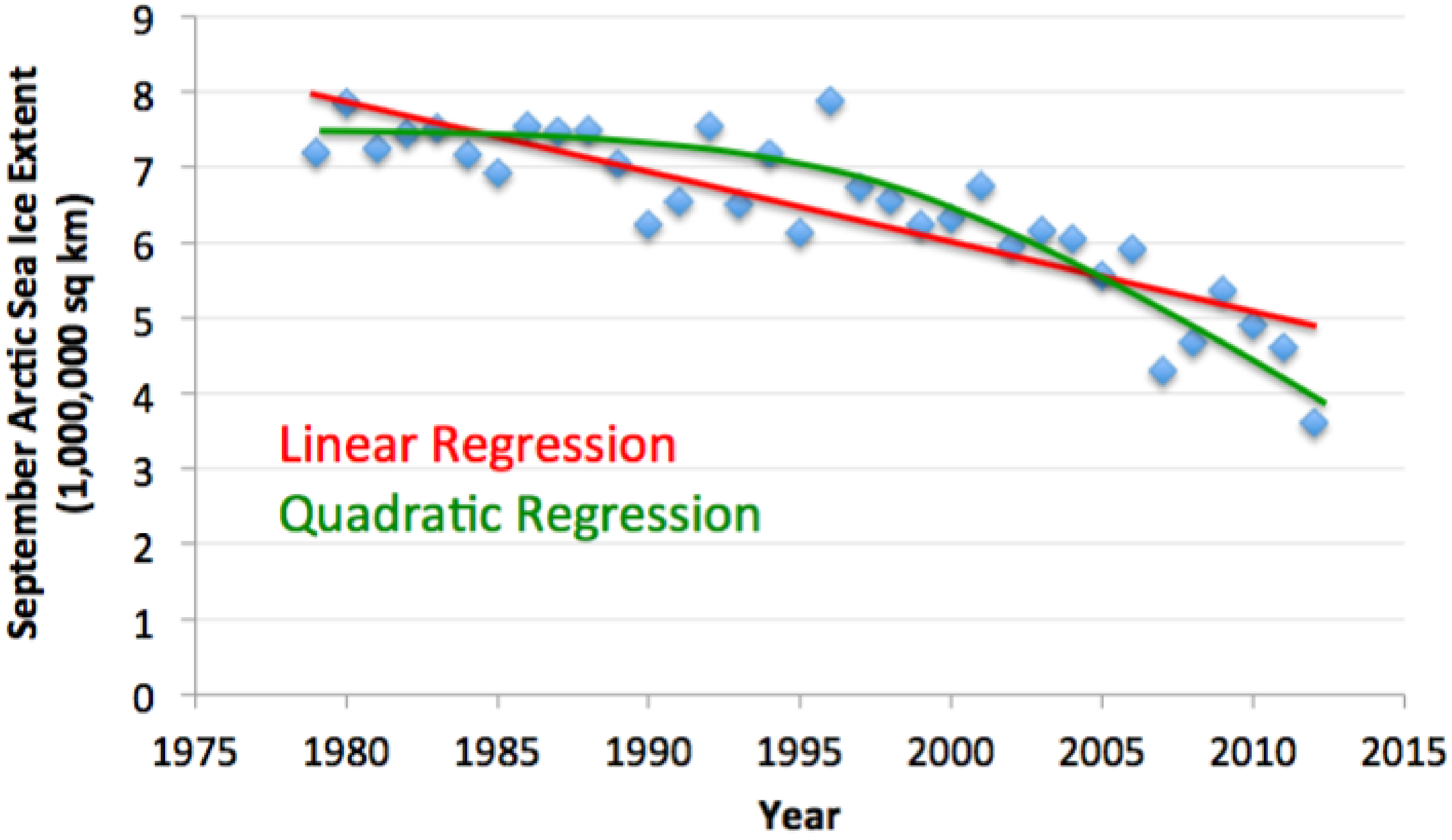


Geometric meaning: can be used to understand the angle between two vectors

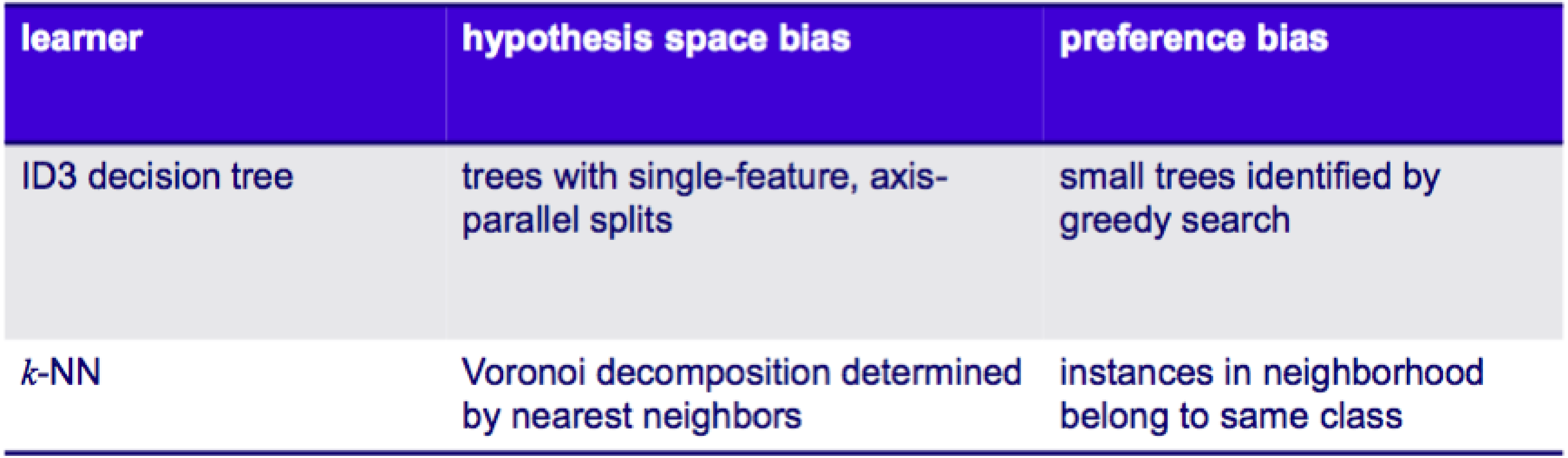
Linear regression

Linear regression

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



Recap: 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

Hypothesis Class H

- 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

L^2 loss, or mean square error

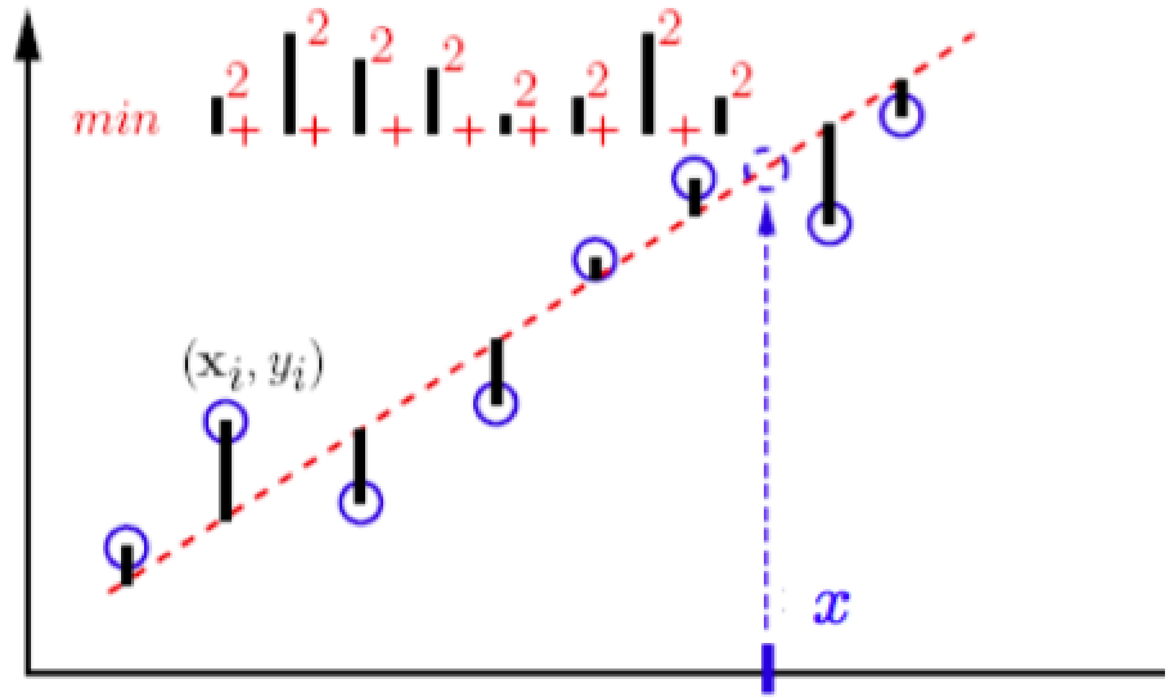
$$\hat{L}(f_w) = \frac{1}{m} \sum_{i=1}^{m} (w^T x^{(i)} - y^{(i)})^2$$

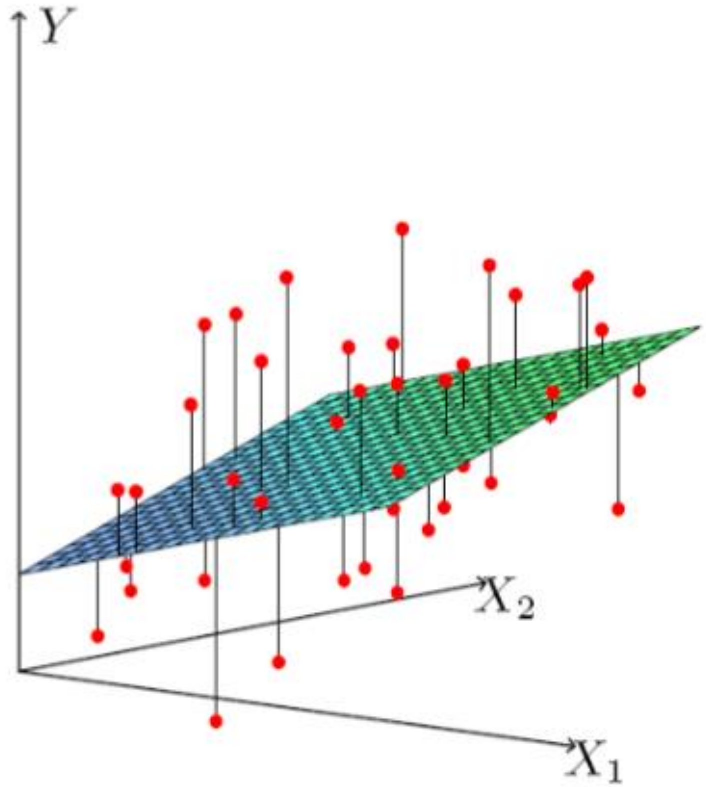
- where
  - $w^T x^{(i)} - y^{(i)}$ represents the error of instance $x^{(i)}$
  - $\sum_{i=1}^{m} (w^T x^{(i)} - y^{(i)})^2$ represents the square error of all training instances

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

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





Organise feature data into matrix

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

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	No
189	92	12.3	Yes
178	79	10.6	Yes
183	90	12.7	No

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

Transform input matrix with weight vector

- Assume a function $f_w(x) = w^T x$ with weight vector $w = (1, -1, 20)$
  - Intuitively,
    - by 20, running speed is more important than the other two features, and
    - by -1, weight is negatively correlated to y

This is the parameter vector we want to learn

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

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

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

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

Organise output into vector

- Let y be the vector $(y^{(1)}, \dots, y^{(m)})^T$

v1	v2	v3	y
182	87	11.3	325
189	92	12.3	344
178	79	10.6	350
183	90	12.7	320

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

Error representation

$$Xw = \begin{bmatrix} 321 \\ 343 \\ 311 \\ 347 \end{bmatrix} \qquad y = \begin{bmatrix} 325 \\ 344 \\ 350 \\ 320 \end{bmatrix}$$

- Square error of all instances

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

Linear regression : optimization

- 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 $(y^{(1)}, \ldots, y^{(m)})^T$

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

Now we knew where this comes from!

Solving this optimization problem will be introduced in later lectures.

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

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

Linear regression with lasso penalty

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

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

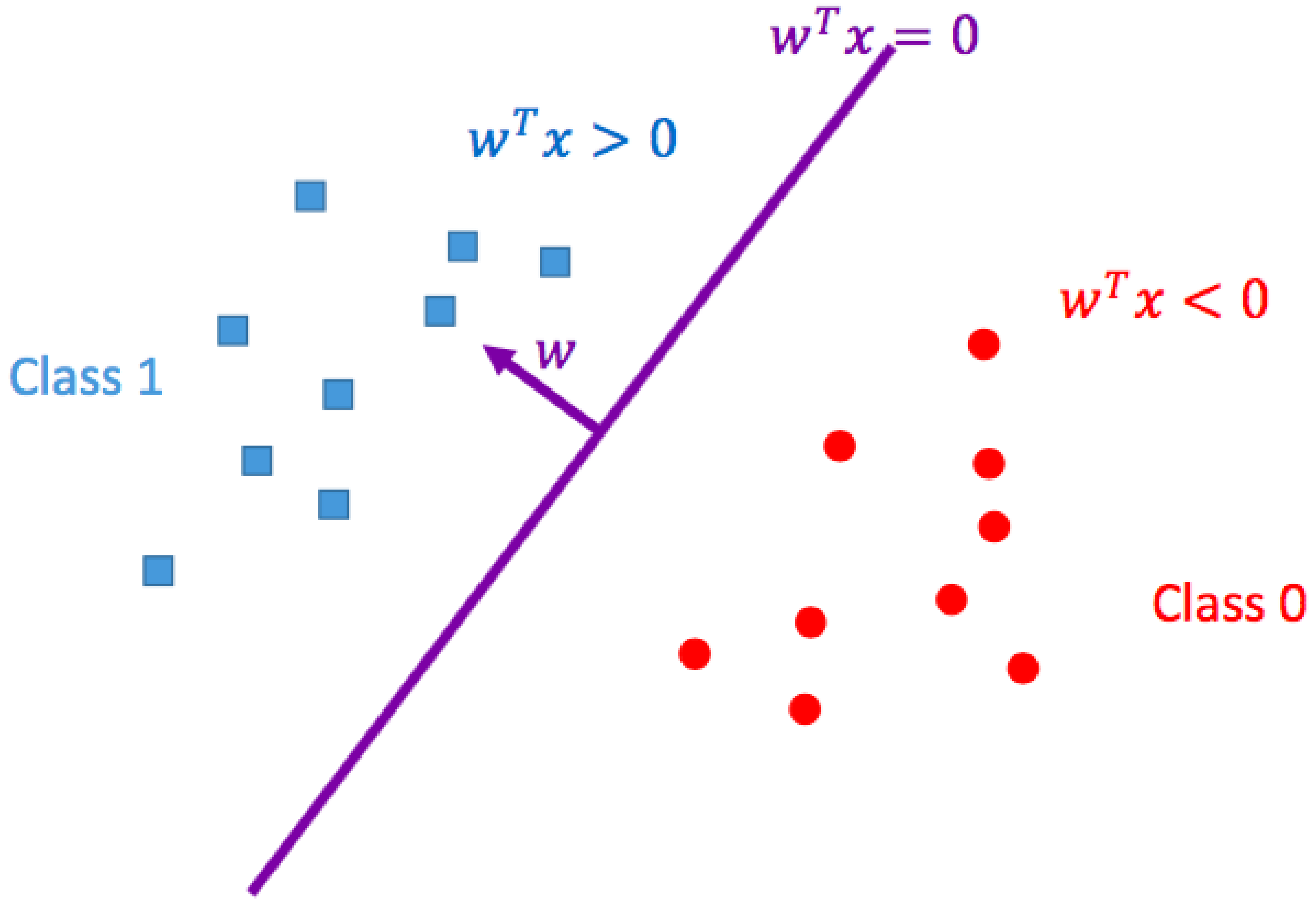
lasso penalty: L^1 norm of the parameter, encourages sparsity

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



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 $\boldsymbol{\mathcal{H}}$

- Prediction:

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

  - 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 optimisation objective?

Linear classification: simple approach

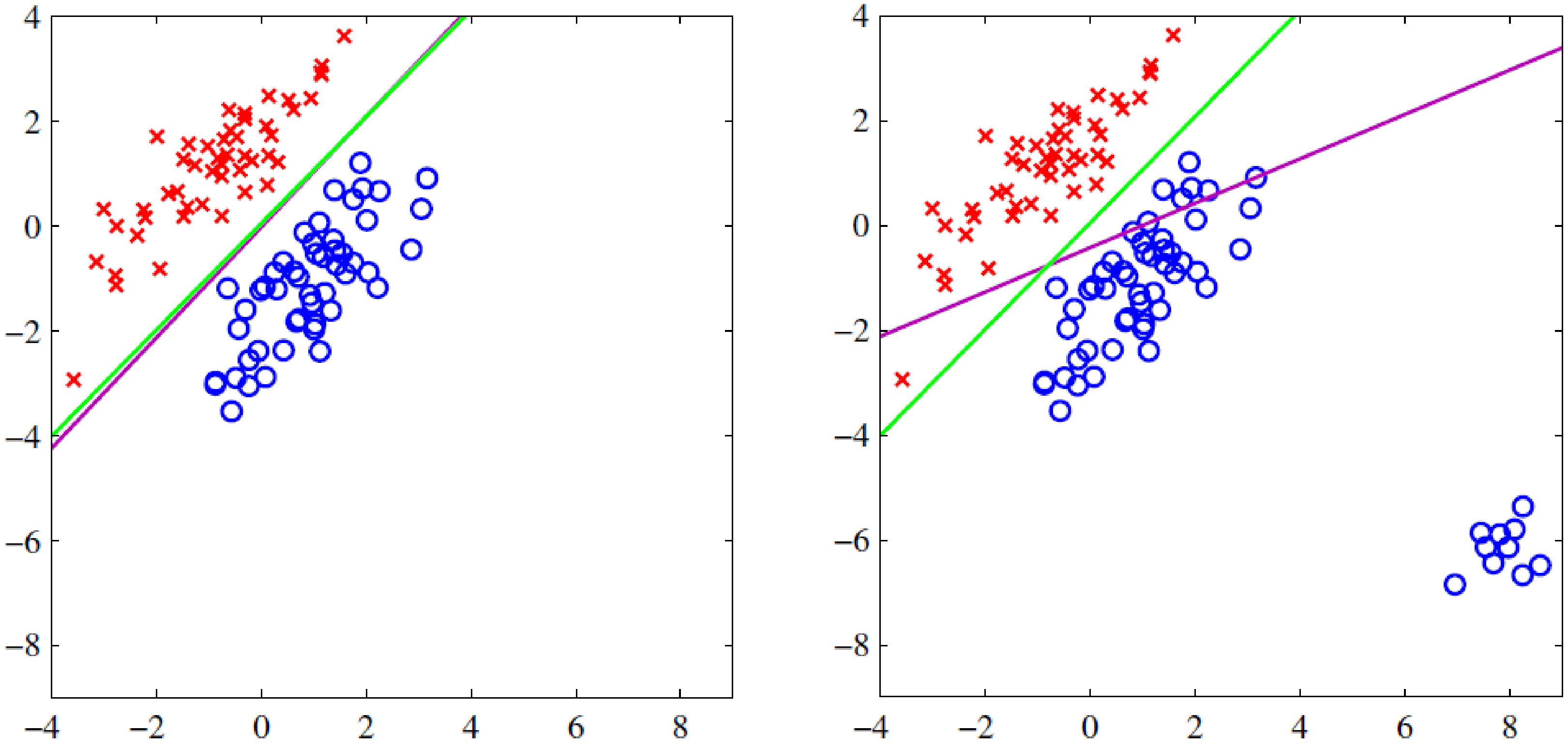


Figure 4.4 The left plot shows data from two classes, denoted by red crosses and blue circles, together with the decision boundary found by least squares (magenta curve) and also by the logistic regression model (green curve), which is discussed later in Section 4.3.2. The right-hand plot shows the corresponding 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.

Drawback: not robust to “outliers”

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

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

0-1 loss

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

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

loss =1, otherwise.

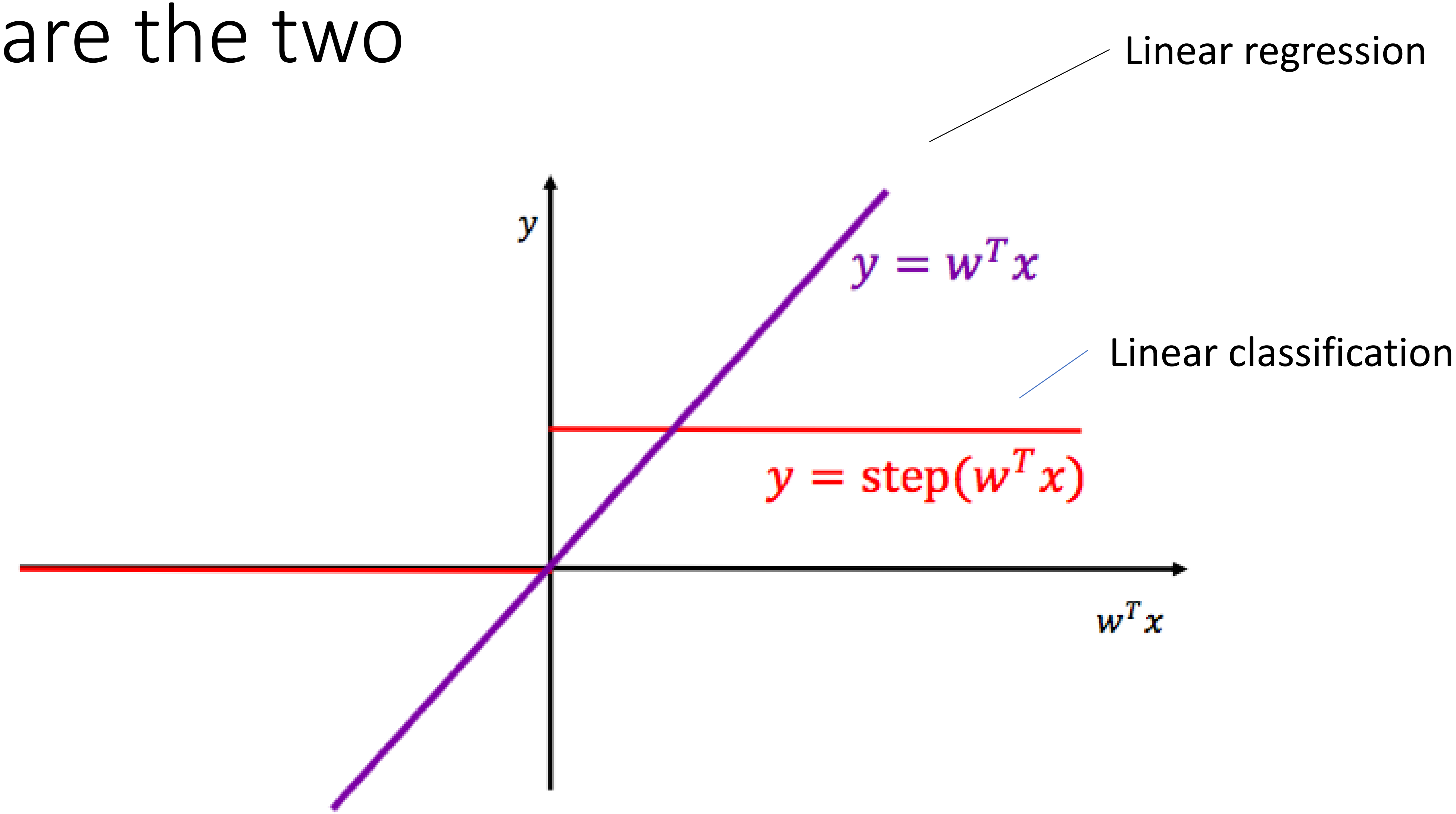
logistic regression

Why logistic regression?

- 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 can not.
- As regression might actually produce probabilities that could be less than 0, or even bigger than 1, logistic regression was introduced.

Logistic regression always outputs a value between 0 and 1

Compare the two

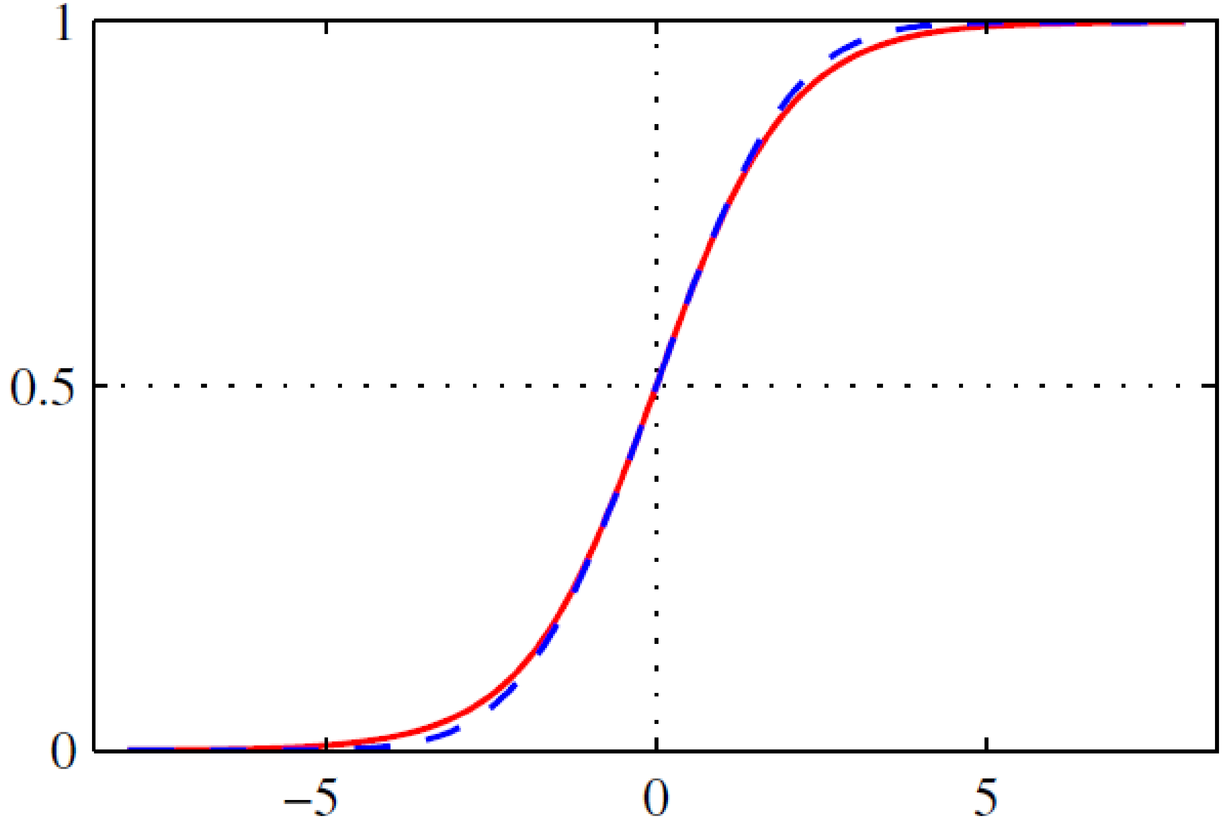


Between the two

- 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

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

- Find *w* that minimizes

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

Question: Do we need to squash y?

Linear classification: logistic regression

- Squash the output of the linear function

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

- A better approach: Interpret as a probability

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

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

Here we assume that y=0 or y=1

Linear classification: logistic regression

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

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

Why log function used? To avoid numerical unstability.

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

Logistic regression: MLE with sigmoid

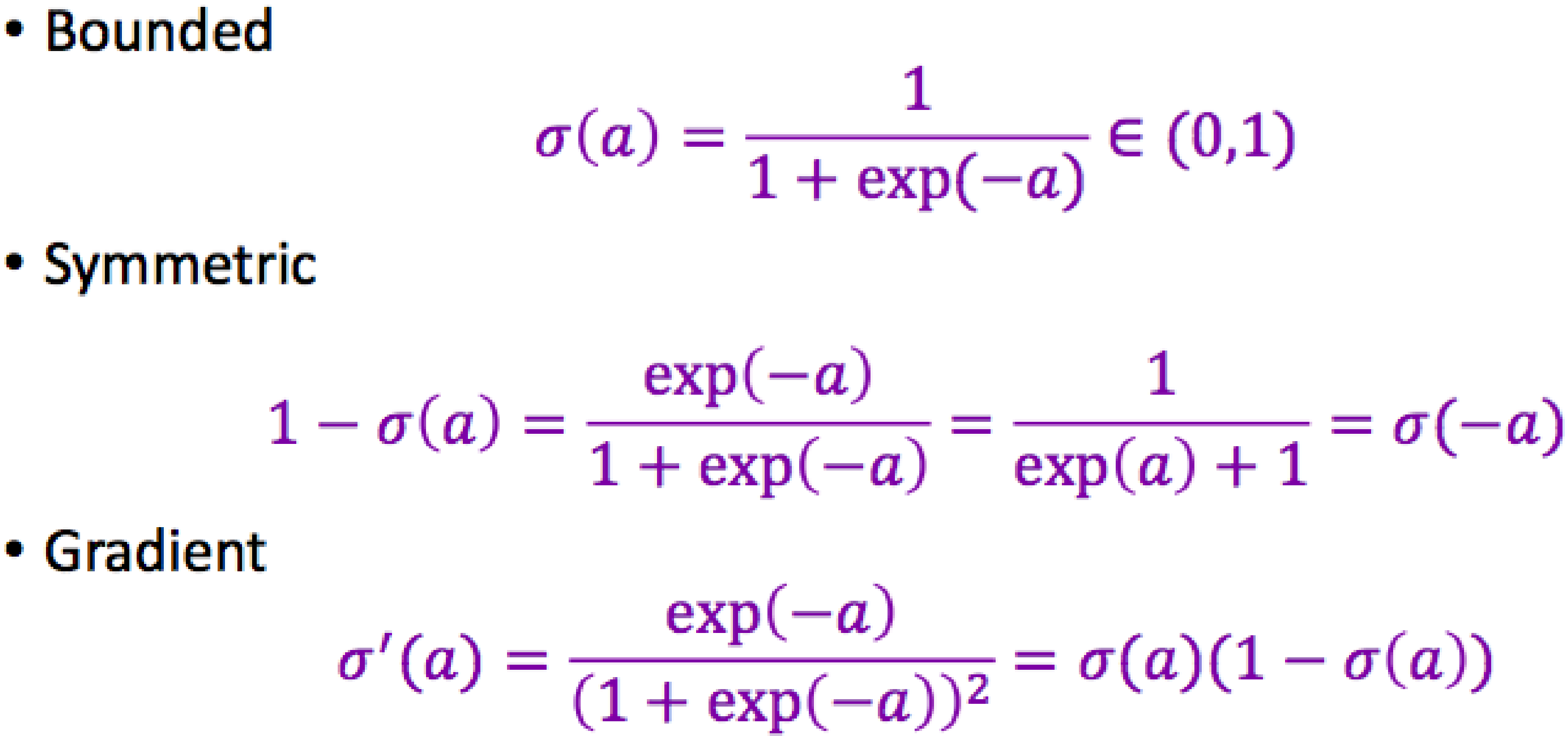
Linear classification: logistic regression

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

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

No close form solution;
Need to use gradient descent

Properties of 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

- 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=(No,Yes,Yes,No)?
  - (3) which model is better for logistic regression?
  - (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

Extended Materials

Review: binary logistic regression

- Sigmoid

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

- Interpret as conditional probability

$$p_w(y = 1|x) = \sigma(w^T x + b)$$

$$p_w(y = 0|x) = 1 - p_w(y = 1|x) = 1 - \sigma(w^T x + b)$$

- How to extend to multiclass?

Review: binary logistic regression

- Suppose we model the class-conditional densities $p(x|y=i)$ and class probabilities $p(y=i)$
- Conditional probability by Bayesian rule:

$$p(y=1|x) = \frac{p(x|y=1)p(y=1)}{p(x|y=1)p(y=1) + p(x|y=2)p(y=2)} = \frac{1}{1+\exp(-a)} = \sigma(a)$$

where we define

$$a \coloneqq \ln \frac{p(x|y=1)p(y=1)}{p(x|y=2)p(y=2)} = \ln \frac{p(y=1|x)}{p(y=2|x)}$$

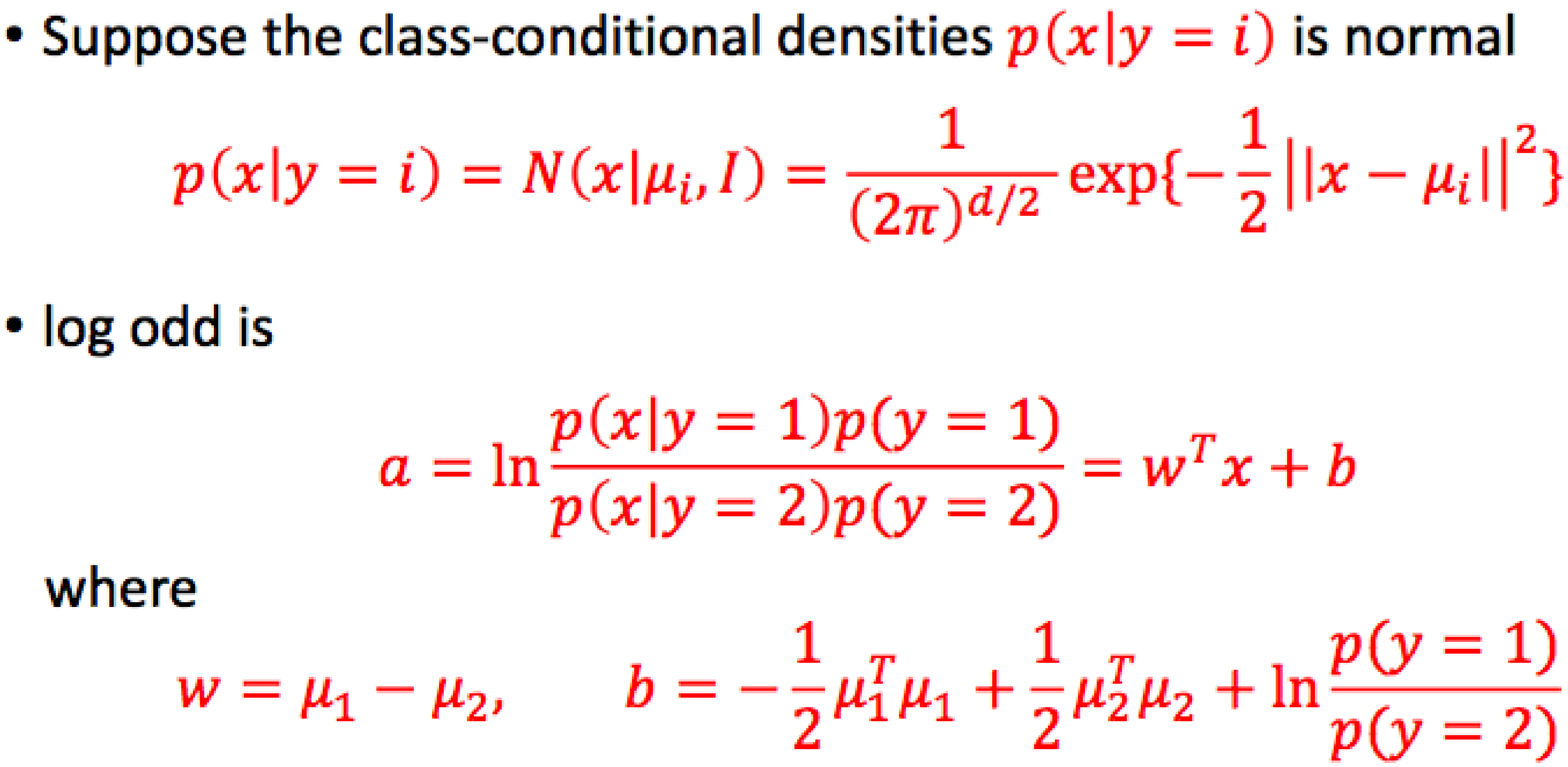
Review: binary logistic regression

- Suppose we model the class-conditional densities $p(x|y = i)$ and class probabilities $p(y = i)$
- $p(y = 1|x) = \sigma(a) = \sigma(w^T x + b)$ is equivalent to setting log odds to be linear:

$$a = \ln \frac{p(y = 1|x)}{p(y = 2|x)} = w^T x + b$$

- Why linear log odds?

Review: binary logistic regression



- Suppose the class-conditional densities $p(x|y=i)$ is normal

$$p(x|y=i) = N(x|\mu_i, I) = \frac{1}{(2\pi)^{d/2}} \exp\{-\frac{1}{2}||x-\mu_i||^2\}$$

- log odd is

$$a = \ln \frac{p(x|y=1)p(y=1)}{p(x|y=2)p(y=2)} = w^T x + b$$

where

$$w = \mu_1 - \mu_2, \qquad b = -\frac{1}{2}\mu_1^T\mu_1 + \frac{1}{2}\mu_2^T\mu_2 + \ln\frac{p(y=1)}{p(y=2)}$$

Multiclass logistic regression

- Suppose we model the class-conditional densities $p(x|y=i)$ and class probabilities $p(y=i)$
- Conditional probability by Bayesian rule:

$$p(y=i|x) = \frac{p(x|y=i)p(y=i)}{\sum_j p(x|y=j)p(y=j)} = \frac{\exp(a_i)}{\sum_j \exp(a_j)}$$

where we define

$$a_i \coloneqq \ln\left[p(x|y=i)p(y=i)\right]$$

Multiclass logistic regression

- Suppose the class-conditional densities $p(x|y=i)$ is normal

$$p(x|y=i) = N(x|\mu_i, I) = \frac{1}{(2\pi)^{d/2}} \exp\{-\frac{1}{2}||x-\mu_i||^2\}$$

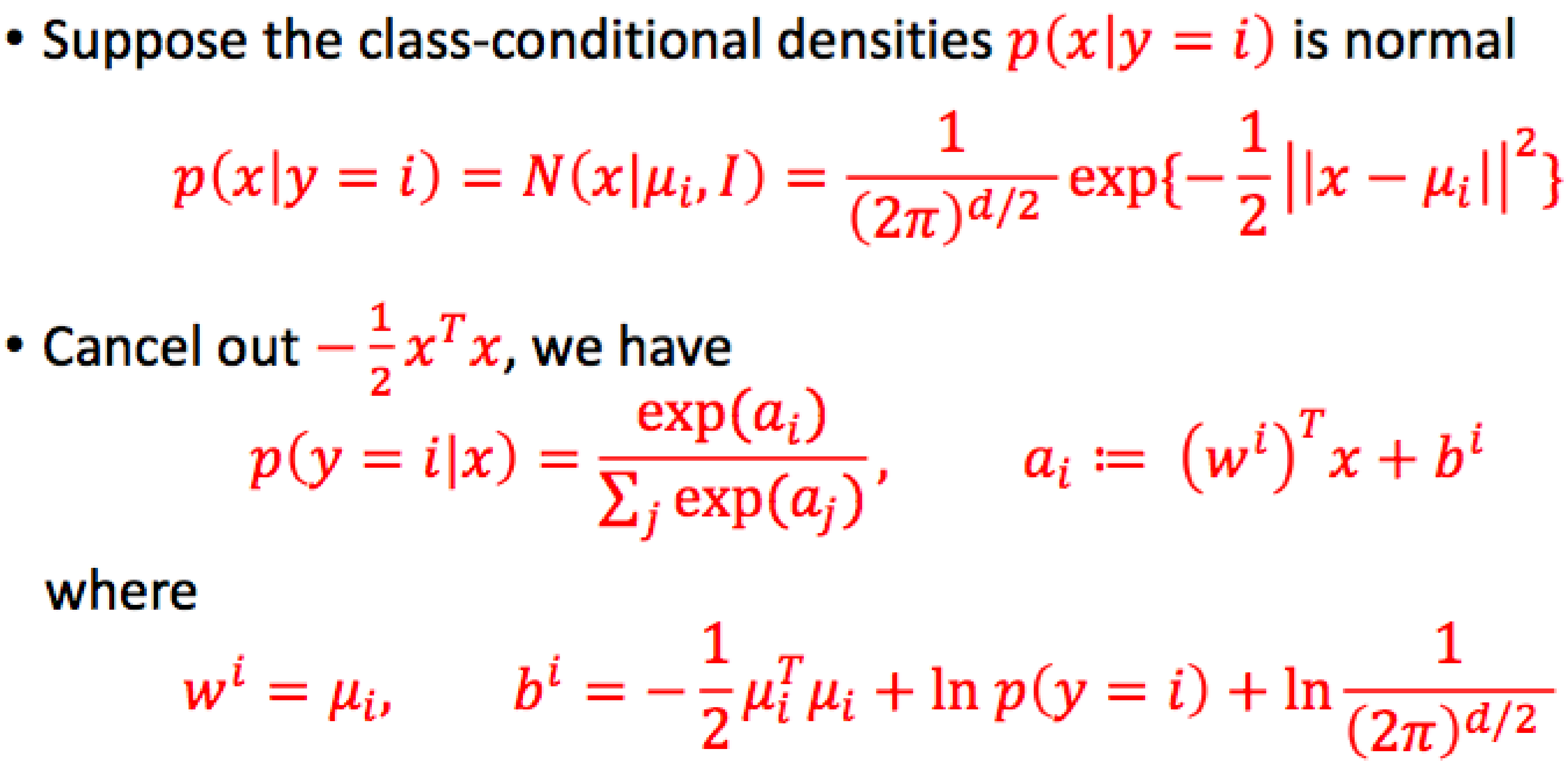
- Then

$$a_i := \ln\left[p(x|y=i)p(y=i)\right] = -\frac{1}{2}x^T x + \left(w^i\right)^T x + b^i$$

where

$$w^i = \mu_i, \qquad b^i = -\frac{1}{2}\mu_i^T\mu_i + \ln p(y=i) + \ln\frac{1}{(2\pi)^{d/2}}$$

Multiclass logistic regression



- Suppose the class-conditional densities $p(x|y=i)$ is normal

$$p(x|y=i) = N(x|\mu_i, I) = \frac{1}{(2\pi)^{d/2}} \exp\{-\frac{1}{2}||x-\mu_i||^2\}$$

- Cancel out $-\frac{1}{2}x^T x$, we have

$$p(y=i|x) = \frac{\exp(a_i)}{\sum_j \exp(a_j)}, \qquad a_i := (w^i)^T x + b^i$$

where

$$w^i = \mu_i, \qquad b^i = -\frac{1}{2}\mu_i^T\mu_i + \ln p(y=i) + \ln\frac{1}{(2\pi)^{d/2}}$$

Multiclass logistic regression: conclusion

- Suppose the class-conditional densities $p(x|y=i)$ is normal

$$p(x|y=i) = N(x|\mu_i, I) = \frac{1}{(2\pi)^{d/2}} \exp\{-\frac{1}{2}||x-\mu_i||^2\}$$

- Then

$$p(y=i|x) = \frac{\exp(\,(w^i)^T x + b^i)}{\sum_j \exp(\,(w^j)^T x + b^j)}$$

which is the hypothesis class for multiclass logistic regression

- It is softmax on linear transformation; it can be used to derive the negative log-likelihood loss (cross entropy)

Softmax

- A way to squash $a = (a_1, a_2, \dots, a_i, \dots)$ into probability vector p

$$\mathrm{softmax}(a) = \left(\frac{\exp(a_1)}{\sum_j \exp(a_j)}, \frac{\exp(a_2)}{\sum_j \exp(a_j)}, \dots, \frac{\exp(a_i)}{\sum_j \exp(a_j)}, \dots \right)$$

- Behave like max: when $a_i \gg a_j (\forall j \neq i)$, $p_i \cong 1, p_j \cong 0$

Cross entropy for conditional distribution

- Let $p_{\text{data}}(y|x)$ denote the empirical distribution of the data
- Negative log-likelihood

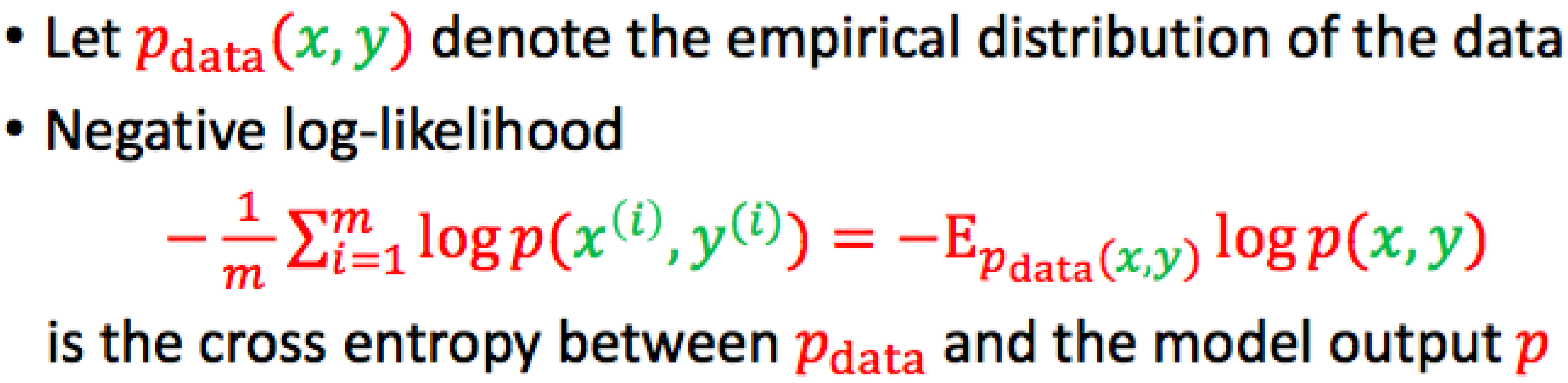
$$-\frac{1}{m}\sum_{i=1}^{m}\log p(y = y^{(i)}|x^{(i)}) = -\mathrm{E}_{p_{\text{data}}(y|x)}\log p(y|x)$$

  is the cross entropy between p_{data} and the model output p

- Information theory viewpoint: KL divergence

$$D(p_{\text{data}}||p) = \mathrm{E}_{p_{\text{data}}}[\log\frac{p_{\text{data}}}{p}] = \underbrace{\mathrm{E}_{p_{\text{data}}}[\log p_{\text{data}}]}_{\text{Entropy; constant}} \underbrace{-\,\mathrm{E}_{p_{\text{data}}}[\log p]}_{\text{Cross entropy}}$$

Cross entropy for full distribution



- Let $p_{\text{data}}(x, y)$ denote the empirical distribution of the data
- Negative log-likelihood

$$-\frac{1}{m}\sum_{i=1}^{m} \log p(x^{(i)}, y^{(i)}) = -\mathrm{E}_{p_{\text{data}}(x,y)} \log p(x, y)$$

is the cross entropy between p_{data} and the model output p