Recap

- So far
  - Two ML algorithms: Naive Bayes, decision tree
  - Feature-based ML: Features interpretable and based on »domain knowledge«
- Naive Bayes
  - Calculate $p(\text{CATEGORY}|\text{FEATURES})$, assign class with highest probability
  - Assume feature independence

UNIVERSITÄT
ZU KÖLN

Logistic Regression

Sprachverarbeitung (VL + Ü)

Nils Reiter

June 13, 2024

INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

Zahl

$$\left.\begin{array}{l}\text{Linear}\\ \text{Logistic}\end{array}\right\}\ \text{Regression}$$

Zahl => 0 - 1

$\Downarrow$

Klassifikation

Neural Networks

- Conceptually developed in the 20th century
- Mainstream ML method in NLP since 2010
- Building block of large language models
- But also a flexible ML algorithm by itself
- Building block of neural networks: Logistic regression

Regression

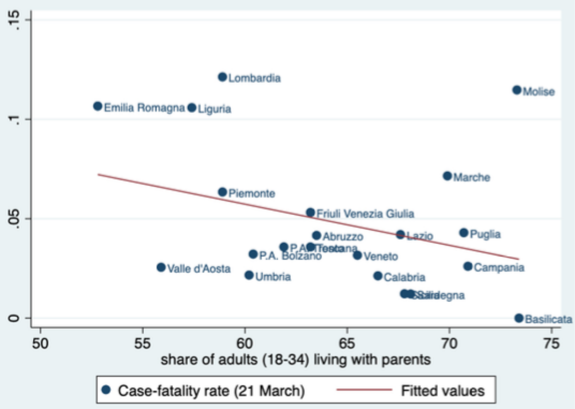
Linear regression

- Prediction of numeric values (e.g., COVID-19 cases)

Regression

Linear regression

- Prediction of numeric values (e.g., CC



Regression

Linear regression

- Prediction of numeric values (e.g., COVID-19 cases)
- »Linear« regression: Prediction of a linear relation (i.e., a line)
- Most real problems are not linear – in particular not COVID-19 cases ...

Regression

Linear regression

- Prediction of numeric values (e.g., COVID-19 cases)
- »Linear« regression: Prediction of a linear relation (i.e., a line)
- Most real problems are not linear – in particular not COVID-19 cases ...

Logistic Regression

- Classification algorithm: Instances are grouped into *previously known* classes
- Binary classification: Two classes (e.g., positive/negative)
- Extension of linear regression

Linear/logistic regression in parallel

Linear Regression

Task Setup

- Input (x): A (collection of) numeric feature values
- Output (y): A numeric value

Example

Given the length of a narrative text in words, predict the number of characters present in its plot

Linear Regression

The data set

Large

x	y (# characters)
10	3
105	5
150	8
210	12
250	7
295	13

Linear Regression

The data set

x	y (# characters)
10	3
105	5
150	8
210	12
250	7
295	13

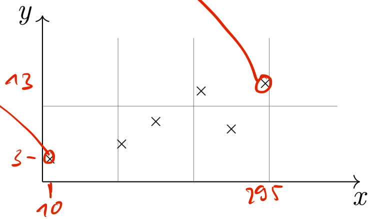


Figure: Data set, each $\times$ represents a text (x: text length, y: num. of characters)

Linear Regression

The data set

x	y (# characters)
10	3
105	5
150	8
210	12
250	7
295	13

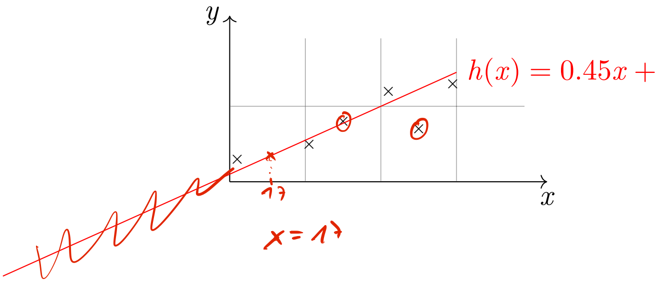
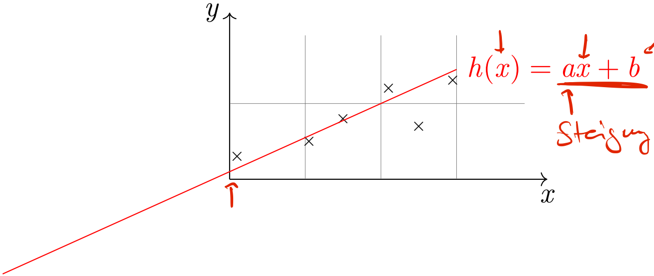


Figure: Data set, each $\times$ represents a text (x: text length, y: num. of characters)

Linear Regression



y-Achsen-abschnitt

Steigung

Prediction Model

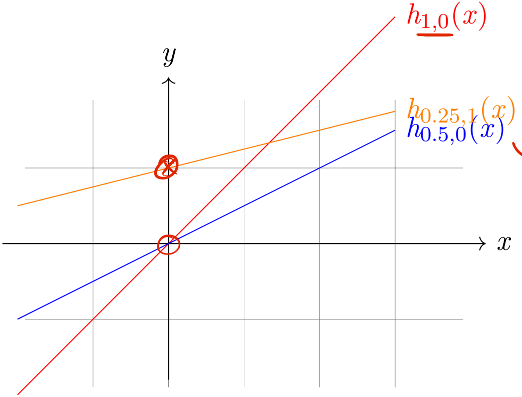
- Linear regression with one variable (= univariate linear regression)
- Data: (x, y)
- Prediction (hypothesis function): $y = h_{a,b}(x) = ax + b$
- How to set parameters a and b? $\rightarrow$ training algorithm

$h(x) = x$

Linear Regression

Prediction Model

- $h_{a,b}(x) = ax + b$ describes a set of functions
    - $h_{1,0}(x)$ is one concrete function



$h(x) = 0{,}25x + 1$

$h(x) = 0{,}5x + 0$

Linear vs. Logistic Regression

- Linear regression: Prediction of numerical data
- Logistic regression: Prediction of (binary) categorical data

Linear vs. Logistic Regression

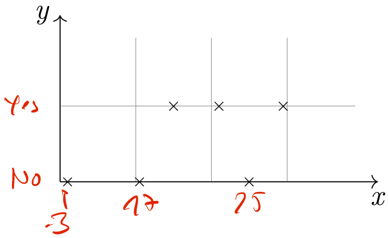
- Linear regression: Prediction of numerical data
- Logistic regression: Prediction of (binary) categorical data

Examples

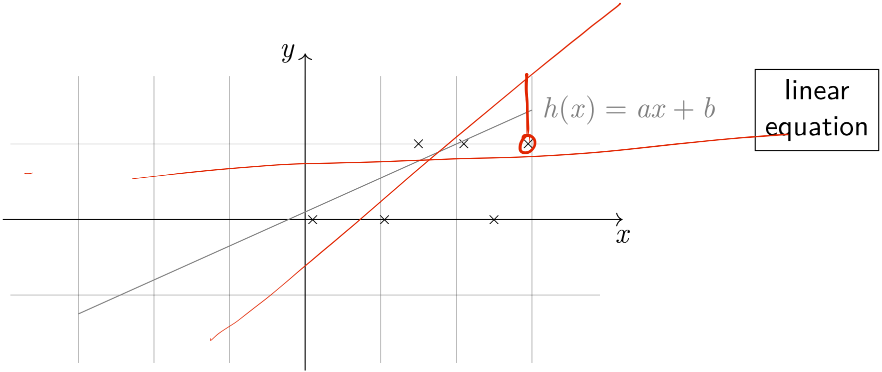
- Our interest
  - Literature quality
- Given the number of characters in a narrative text
- Will a book win the Nobel prize?
  - Two classes: Yes/No

Logistic Regression

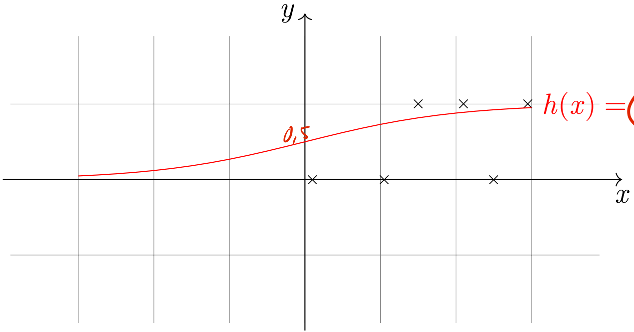
The data set



How to predict these values?



How to predict these values?

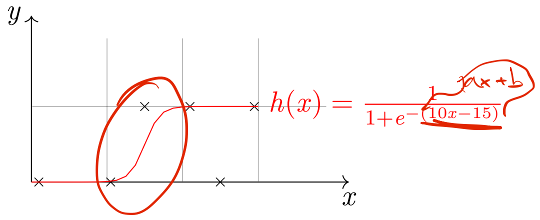


$$h(x) = \frac{1}{1+e^{-x}}$$

Konstante

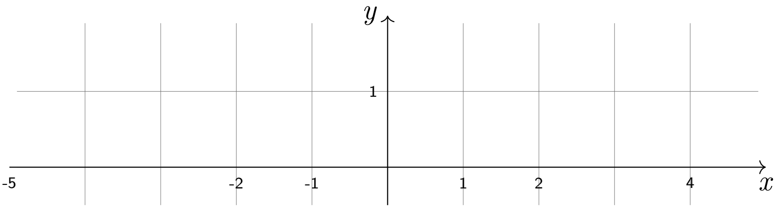
logistic function

Parameter Fitting



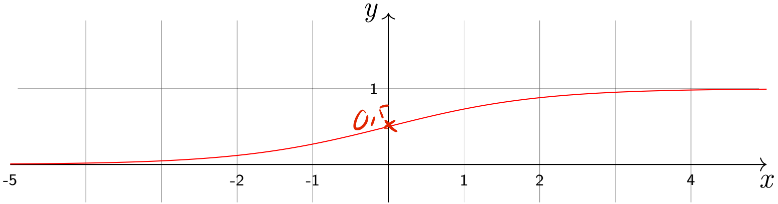
- Linear equations can be wrapped in a logistic one
- Same parameters to be tuned (a and b)
- $e = \sum_{n=0}^{\infty} \frac{1}{n!} = 2.71828$ (Euler's number)

The Logistic Function



$$y = \frac{1}{1+e^{-(ax+b)}} \qquad \text{(general form)}$$

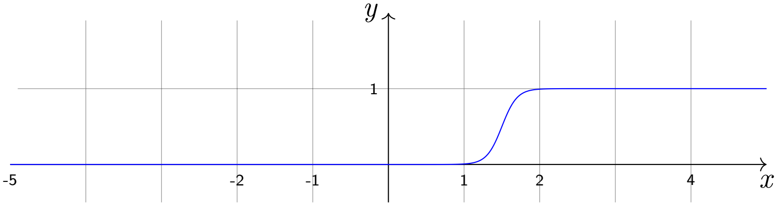
The Logistic Function



$$y = \frac{1}{1+e^{-(ax+b)}}$$ (general form)

$$y = \frac{1}{1+e^{-(1*x+0)}}$$

The Logistic Function

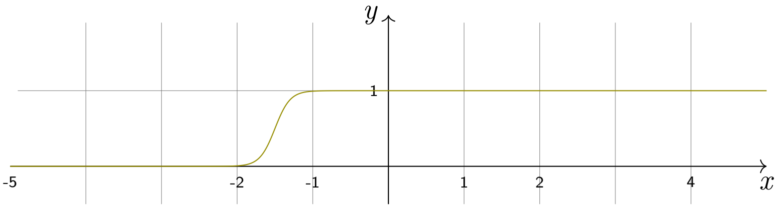


$$y = \frac{1}{1+e^{-(ax+b)}} \qquad \text{(general form)}$$

$$y = \frac{1}{1+e^{-(1*x+0)}} \qquad y = \frac{1}{1+e^{-(10*x-15)}}$$

a = 10
b = -15

The Logistic Function

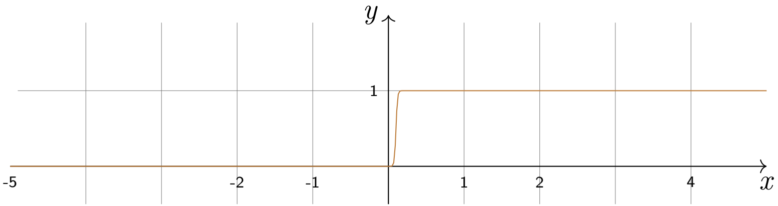


$$y = \frac{1}{1+e^{-(ax+b)}} \qquad \text{(general form)}$$

$$y = \frac{1}{1+e^{-(1*x+0)}} \qquad y = \frac{1}{1+e^{-(10*x-15)}}$$

$$y = \frac{1}{1+e^{-(10*x+15)}}$$

The Logistic Function

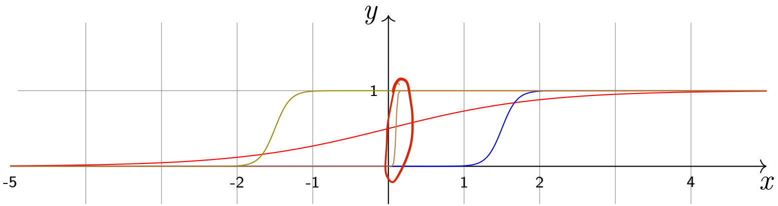


$$y = \frac{1}{1+e^{-(ax+b)}} \qquad \text{(general form)}$$

$$y = \frac{1}{1+e^{-(1*x+0)}} \qquad y = \frac{1}{1+e^{-(10*x-15)}}$$

$$y = \frac{1}{1+e^{-(10*x+15)}} \qquad y = \frac{1}{1+e^{-(100*x-10)}}$$

The Logistic Function



$$y = \frac{1}{1+e^{-(ax+b)}}$$ (general form)

$$y = \frac{1}{1+e^{-(1*x+0)}} \qquad y = \frac{1}{1+e^{-(10*x-15)}}$$

$$y = \frac{1}{1+e^{-(10*x+15)}} \qquad y = \frac{1}{1+e^{-(100*x-10)}}$$

Summary: Logistic Regression (with a single variable)

Logistic regression is half of the math of deep learning

Summary: Logistic Regression (with a single variable)

Logistic regression is half of the math of deep learning

- Logistic Regression: Predicting binary values
- Model
  - Logistic equations
  - $y = \frac{1}{1+e^{-(ax+b)}}$
- Learning algorithm: How to choose a and b?

Gradient Descent

Learning Regression Models

- How to select the parameters a, b such that the hypothesis function describes the data points as best as possible?
- Learning algorithm *Gradient Descent*

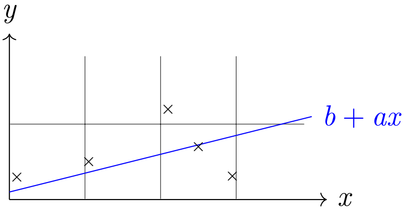
Learning Regression Models

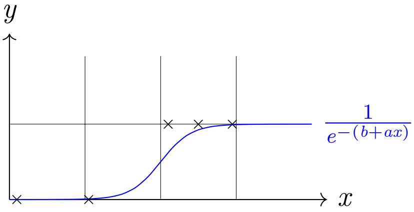
- How to select the parameters a, b such that the hypothesis function describes the data points as best as possible?
- Learning algorithm *Gradient Descent*

Gradient descent is half of the algorithms of deep learning

Loss: Intuition

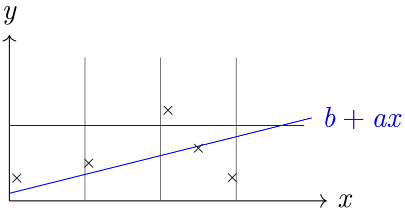
The *loss* measures the ›wrongness‹ of values for a and b.

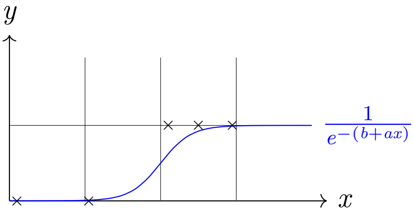




Loss: Intuition

The *loss* measures the ›wrongness‹ of values for a and b.

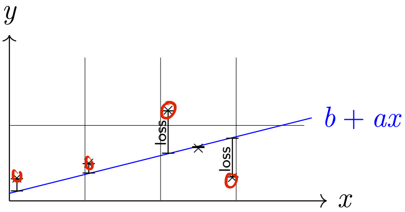


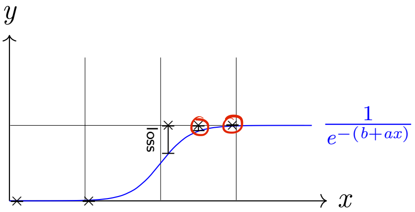


- How big is the gap between a hypothesis and the data?
- Is $(a, b) = (0.3, 0.5)$ or $(a, b) = (0.4, 0.4)$ better?

Loss: Intuition

The *loss* measures the ›wrongness‹ of values for a and b.





- How big is the gap between a hypothesis and the data?
- Is $(a, b) = (0.3, 0.5)$ or $(a, b) = (0.4, 0.4)$ better?

Loss function: Intuition

- Loss should be as small as possible
- Total loss can be calculated for given parameters $\vec{w} = (a, b)$ (and a full data set)
  $\Rightarrow$ I.e.: Loss can be expressed as a function of $\vec{w}$!

Loss function: Intuition

- Loss should be as small as possible
- Total loss can be calculated for given parameters $\vec{w} = (a, b)$ (and a full data set)

  $\Rightarrow$ I.e.: Loss can be expressed as a function of $\vec{w}$!
- Idea:
  - We change $\vec{w}$ until we find the minimum of the function
  - We use the derivative to find out if we are in a minimum
  - The derivative also tells us how to change the update parameters a and b

Loss Function: Intuition

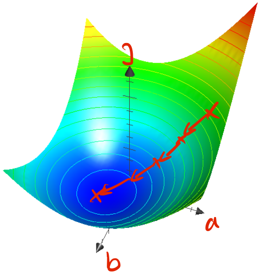
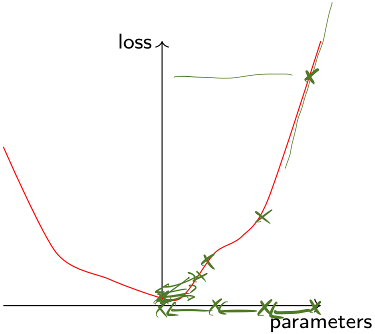
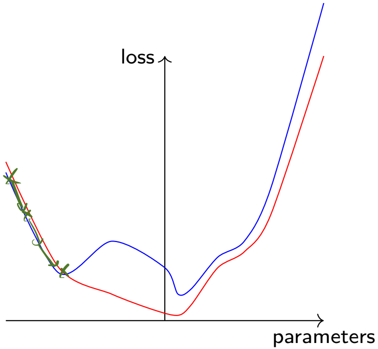


Figure: The loss function with two parameters

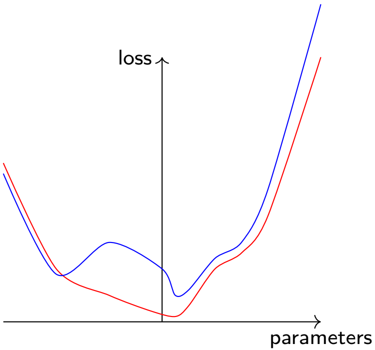
Loss function: Intuition



Loss function: Intuition



Loss function: Intuition



Function should be convex!
If not, we might get stuck in local minimum

$h_{a,b}(x) = ax + b$

Hypothesis vs. Loss Function

$h(x) = ax + b$

$J_D(a, b) = \ldots$

- Hypothesis function h
  - Calculates outcomes, given feature values x
- Loss function J
  - Calculates ›wrongness‹ of h, given parameter values $\vec{w}$ (and a data set)
  - In reality, $\vec{w}$ represents many more parameters (thousands)

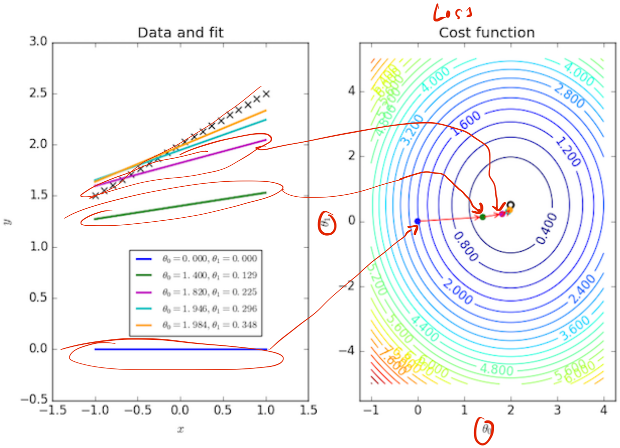


Figure: Visualizing gradient descent Source

Loss Function

Definition

Loss function depends on hypothesis function

Linear hypothesis function

- $h(x) = ax + b$
- Loss: Mean squared error

Loss Function

Definition

Loss function depends on hypothesis function

Linear hypothesis function

- $h(x) = ax + b$
- Loss: Mean squared error

Logistic hypothesis function

- $h(x) = \frac{1}{e^{-(b+ax)}}$
- Loss: (Binary) cross-entropy loss

Loss Function

Definition for Linear Regression

- The loss function is a function on parameter values a and b (for a given hypothesis function and data set)
  - Hypothesis function: $h_{\vec{w}} = w_1 x + w_0$

$\vec{w} = \langle a, b \rangle$

$\vec{w} = (a, b)$: parameters $h_{\vec{w}}$: hypothesis function m: number of items

$$J(\vec{w}) =$$

Loss Function

Definition for Linear Regression

- The loss function is a function on parameter values a and b
  (for a given hypothesis function and data set)
  - Hypothesis function: $h_{\vec{w}} = w_1 x + w_0$

$\vec{w} = (a, b)$: parameters $h_{\vec{w}}$: hypothesis function m: number of items

$$J(\vec{w}) =$$

$$\underbrace{\underset{17}{h_{\vec{w}}(x_i)} - \underset{16}{y_i}}_{1}$$

- Calculate the loss for item i

Loss Function

Definition for Linear Regression

- The loss function is a function on parameter values a and b
  (for a given hypothesis function and data set)
  - Hypothesis function: $h_{\vec{w}} = w_1 x + w_0$

$\vec{w} = (a, b)$: parameters $h_{\vec{w}}$: hypothesis function m: number of items

$$J(\vec{w}) = \qquad\qquad (h_{\vec{w}}(x_i) - y_i)^2$$

- Calculate the loss for item i
- Square the error

Loss Function

Definition for Linear Regression

- The loss function is a function on parameter values a and b
  (for a given hypothesis function and data set)
  - Hypothesis function: $h_{\vec{w}} = w_1 x + w_0$

$$\vec{w} = (a, b)\text{: parameters} \quad h_{\vec{w}}\text{: hypothesis function} \quad m\text{: number of items}$$

$$J(\vec{w}) = \qquad \sum_{i=1}^{m} (h_{\vec{w}}(x_i) - y_i)^2$$

- Calculate the loss for item i
- Square the error
- Sum them up

Loss Function

Definition for Linear Regression

- The loss function is a function on parameter values a and b
  (for a given hypothesis function and data set)
  - Hypothesis function: $h_{\vec{w}} = w_1 x + w_0$

$\vec{w} = (a, b)$: parameters $h_{\vec{w}}$: hypothesis function m: number of items

$$J(\vec{w}) = \underbrace{\frac{1}{m}\sum_{i=1}^{m}}_{\text{mean}} (\underbrace{h_{\vec{w}}(x_i) - y_i}_{\text{error}})^{2} \quad \text{squared}$$

- Calculate the loss for item i
- Square the error
- Sum them up
- Divide by the number of items
  - Known as: *Mean squared error*

Loss Function

Definition for Linear Regression

- The loss function is a function on parameter values a and b
  (for a given hypothesis function and data set)
  - Hypothesis function: $h_{\vec{w}} = w_1 x + w_0$

$\vec{w} = (a, b)$: parameters $h_{\vec{w}}$: hypothesis function m: number of items

$$J(\vec{w}) = \frac{1}{2} \frac{1}{m} \sum_{i=1}^{m} (h_{\vec{w}}(x_i) - y_i)^2$$

- Calculate the loss for item i
- Square the error
- Sum them up
- Divide by the number of items
  - Known as: *Mean squared error*
- Divide by two
  - out of convenience, because derivation

Loss function

Definition for Logistic Regression

- Two cases: $y_i = 0$ or $y_i = 1 - y_i$: real outcome for instance i

Loss function

Definition for Logistic Regression

- Two cases: $y_i = 0$ or $y_i = 1$ – y_i: real outcome for instance i

$$J(\vec{w}) = \qquad h_{\vec{w}}(x_i) + \qquad (1 - h_{\vec{w}}(x_i))$$

Loss function

Definition for Logistic Regression

- Two cases: $y_i = 0$ or $y_i = 1 - y_i$: real outcome for instance i

$$J(\vec{w}) = \qquad \log h_{\vec{w}}(x_i) + \qquad \log\left(1 - h_{\vec{w}}(x_i)\right)$$

y_i	$h_{\vec{w}}(x_i)$	$y_i \log h_{\vec{w}}(x_i) + (1 - y_i)\log(1 - h_{\vec{w}}(x_i))$
0	1	-23.2535
0	0	0

Loss function

Definition for Logistic Regression

- Two cases: $y_i = 0$ or $y_i = 1 - y_i$: real outcome for instance i

$$J(\vec{w}) = \boxed{\overset{1}{y_i}} \log h_{\vec{w}}(x_i) + \boxed{\overset{0}{(1 - y_i)}} \log\left(1 - h_{\vec{w}}(x_i)\right)$$

y_i	$h_{\vec{w}}(x_i)$	$y_i \log h_{\vec{w}}(x_i) + (1 - y_i) \log(1 - h_{\vec{w}}(x_i))$
0	1	-23.2535
0	0	0
1	1	0
1	0	-23.2535

Loss function

Definition for Logistic Regression

- Two cases: $y_i = 0$ or $y_i = 1$ – y_i: real outcome for instance i

$$J(\vec{w}) = -\frac{1}{m}\sum_{i=0}^{m} y_i \log h_{\vec{w}}(x_i) + (1 - y_i)\log\left(1 - h_{\vec{w}}(x_i)\right)$$

1

$1 - 0$ $1 - 1$ $\log 0$

y_i	$h_{\vec{w}}(x_i)$	$\cancel{y_i \log h_{\vec{w}}(x_i)} + (1 - y_i)\log(1 - h_{\vec{w}}(x_i))$
0	1	−23.2535
0	0	0
1	1	0
1	0	−23.2535
1	0.8	−0.3219281
1	0.2	−2.321928

Loss function

Definition for Logistic Regression

- Two cases: $y_i = 0$ or $y_i = 1$ – y_i: real outcome for instance i

$$J(\vec{w}) = -\frac{1}{m}\sum_{i=0}^{m} \underbrace{y_i \log h_{\vec{w}}(x_i)}_{0 \text{ iff } y_i=0} + \underbrace{(1-y_i)\log(1-h_{\vec{w}}(x_i))}_{0 \text{ iff } y_i=1}$$

y_i	$h_{\vec{w}}(x_i)$	$y_i \log h_{\vec{w}}(x_i) + (1-y_i)\log(1-h_{\vec{w}}(x_i))$
0	1	-23.2535
0	0	0
1	1	0
1	0	-23.2535
1	0.8	-0.3219281
1	0.2	-2.321928

Caveat: $\log 0$ is undefined
We may need to add something very small

Side note: Log Probabilities

- Relative order is stable: If $a > b$, then $\log a > \log b$
  - No information loss

Side note: Log Probabilities

- Relative order is stable: If $a > b$, then $\log a > \log b$
  - No information loss
- Multiplication turns to addition: $\log(a \cdot b) = \log a + \log b$
  - Addition is much faster than multiplication in a computer
  - Pays off because we're doing this *a lot*

More Dimensions

- Above: 1 dimension, 2 parameters
  - a: slope, b: y-intercept
  - Input feature x, a single value

More Dimensions

- Above: 1 dimension, 2 parameters
  - a: slope, b: y-intercept
  - Input feature x, a single value
- More dimensions
  - $\vec{w} = \langle w_0, w_1, \ldots, w_n \rangle$ (n dimensions)
  - Input vector $\vec{x}$ with $n - 1$ dimensions
  - Hypothesis function: $h_{\vec{w}}(x) = w_n x_n + w_{n-1} x_{n-1} + \ldots w_1 x_1 + w_0$
    - w_0: y-intercept, w_1 to w_n: slopes

More Dimensions

- Above: 1 dimension, 2 parameters
  - a: slope, b: y-intercept
  - Input feature x, a single value
- More dimensions
  - $\vec{w} = \langle w_0, w_1, \ldots, w_n \rangle$ (n dimensions)
  - Input vector $\vec{x}$ with $n-1$ dimensions
  - Hypothesis function: $h_{\vec{w}}(x) = w_n x_n + w_{n-1} x_{n-1} + \ldots w_1 x_1 + w_0$
    - w_0: y-intercept, w_1 to w_n: slopes
- Algorithms
  - Derivatives more complicated
  - Otherwise identical

Section 2

Summary

Summary

Regression

- Fitting parameters to a data distribution
  - Linear R: Numeric prediction algorithm
    - Prediction model: $h_{\vec{w}}(x) = ax + b$
  - Logistic R: Classification algorithm
    - Prediction model: $h_{\vec{w}}(x) = \frac{1}{e^{-(b+ax)}}$
- Learning algorithm: Gradient descent

Gradient Descent

- Initialise $\vec{w}$ with random values (e.g., 0)
- Repeat:
  - Find the direction to the minimum by taking the derivative
  - Change $\vec{w}$ accordingly, using a learning rate η
  - Stop when $\vec{w}$ don't change anymore