Machine Learning 2: Logistic Regression

VL Sprachliche Informationsverarbeitung

Nils Reiter
nils.reiter@uni-koeln.de

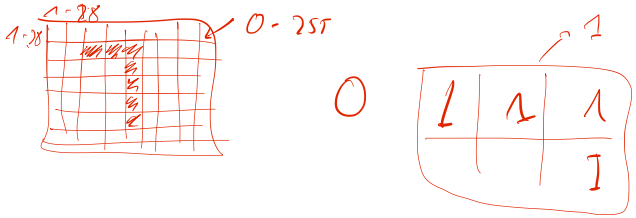
December 1, 2022
Winter term 2022/23

INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

Recap: Naive Bayes

- "Classical machine learning"
- Binary classification method
- Manually defined features (e.g., some tokens appearing in an e-mail)

Recap: Naive Bayes



- "Classical machine learning"
- Binary classification method
- Manually defined features (e.g., some tokens appearing in an e-mail)

Before moving on: Practical use

- MNIST data set: Recognize handwritten digits
- Simplified to distinguish 0 and 1
- Images represented as black/white in 28x28 pixels (= 784 numbers per image)
  - Each number represents the blackness of the pixel from 0 to 255
- Done in Python with scikit-learn

scikit-learn

demo

```python
import csv
from sklearn.naive_bayes import CategoricalNB
from sklearn.metrics import classification_report

def read_dataset(filename):
  x = []
  y = []
  with open(filename, newline='') as csvfile:
    next(csvfile)
    reader = csv.reader(csvfile, delimiter=',')
    for row in reader:
      y.append(int(row[0]))
      x.append([int(pixel) for pixel in row[1:]])
  return (x,y)

x_train, y_train = read_dataset("mnist01/train.csv")
x_test, y_test = read_dataset("mnist01/test.csv")

nb = CategoricalNB(min_categories=256)
nb.fit(x_train, y_train)

h_test = nb.predict(x_test)

print(classification_report(y_test, h_test))
```

Next: Neural Machine Learning

- Neural networks, a.k.a. “deep learning”
- State of the art in NLP (and many areas of artificial intelligence)

Section 1

Linear/Logistic Regression

Regression

vs. Classification

- Regression
  - Prediction of numeric values (e.g., future COVID-19 cases; number of nouns in a text, ...)
  - Based on some input features (e.g., "R-Wert", number of past cases, ...)

Regression

- Regression
  - Prediction of numeric values (e.g., future COVID-19 cases; number of nouns in a text, ...)
  - Based on some input features (e.g., "R-Wert", number of past cases, ...)
- Linear
  - The relation between input features and output values is linear
  - Math: $y = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n + b$

Regression

- Regression
  - Prediction of numeric values (e.g., future COVID-19 cases; number of nouns in a text, ...)
  - Based on some input features (e.g., "R-Wert", number of past cases, ...)
- Linear
  - The relation between input features and output values is linear
  - Math: $y = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n + b$
- Logistic
  - Relation between input and output follows a logistic equation σ:
    - $0 \leq \sigma(x) \leq 1$, for all values of x
    - They can be interpreted as probabilities

Linear Regression

Example

- Input
  - Number of words in a (narrative, prose) text
- Output
  - Number of literary characters in the text
    (in the sense of “Figur”, not in the sense of “Zeichen”)

Linear Regression

Example

- Input
  - Number of words in a (narrative, prose) text
- Output
  - Number of literary characters in the text
    (in the sense of "Figur", not in the sense of "Zeichen")
- Linear equation (with one input variable): $y = ax + b$
  - With x being the number of tokens and y the number of characters

Linear Regression

Example scenario

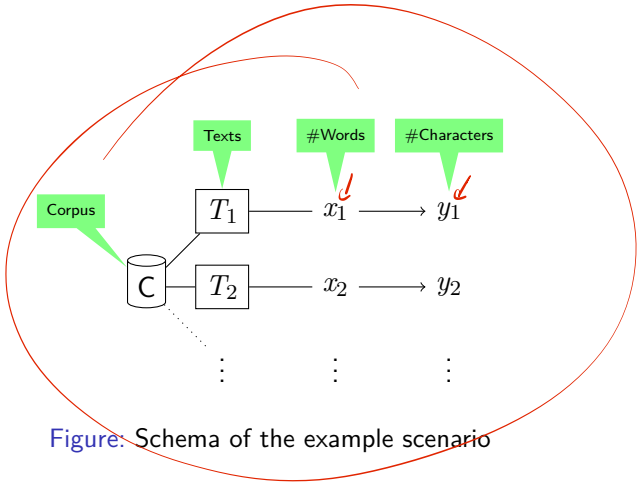


Figure: Schema of the example scenario

Linear Regression

The data set

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

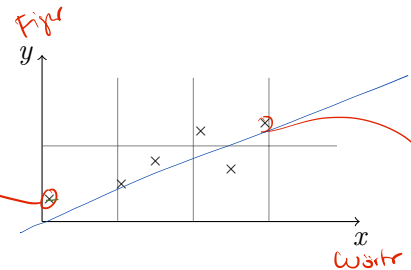


Figure: Data set, each × ~~represents a text~~ (x: text length, y: num. of characters)

Linear Regression

The data set

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

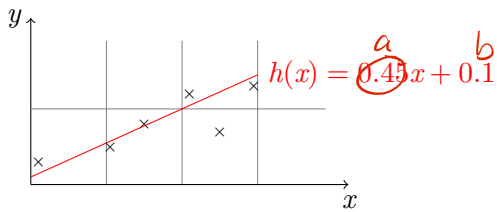
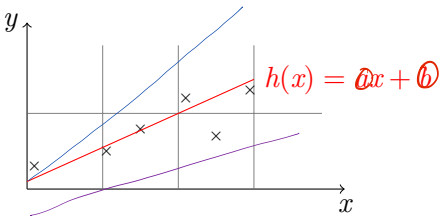


Figure: Data set, each $\times$ represents a text (x: text length, y: num. of characters)

Linear Regression

The Task



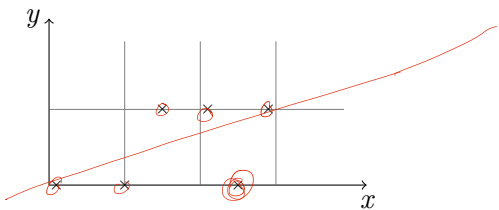
The Model

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

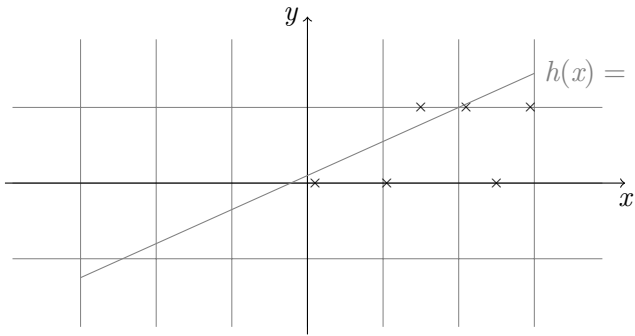
Doing Classification with Linear Regression

- Example task: Does a book will receive a nobel prize, given the number of characters in it?

# Characters	Win
1	No
10	No
15	Yes
21	Yes
25	No
29	Yes

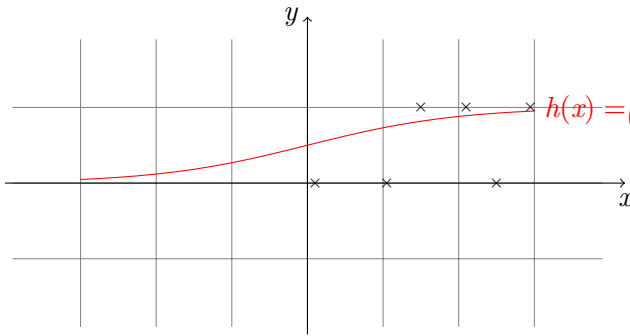


Fitting an Equation



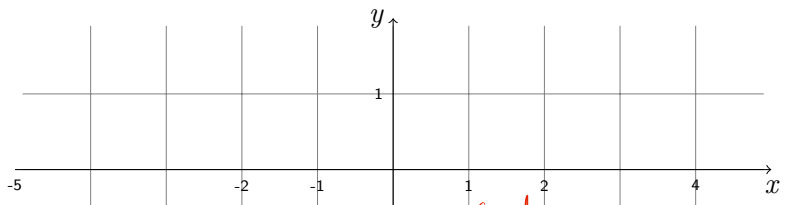
linear equation

Fitting an Equation



logistic function

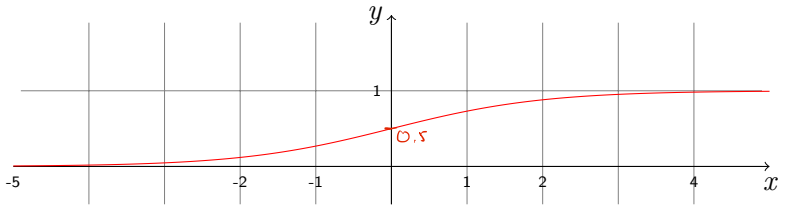
The Logistic Function



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

linear Ability

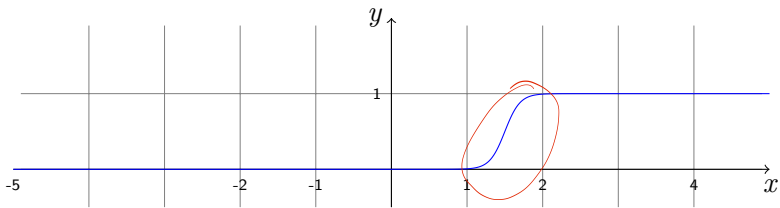
The Logistic Function



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

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

The Logistic Function

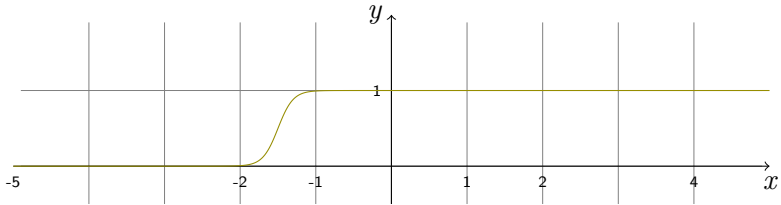


$$y = \frac{1}{1+e^{-(ax+b)}} \qquad \text{(general form)}$$

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

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

The Logistic Function



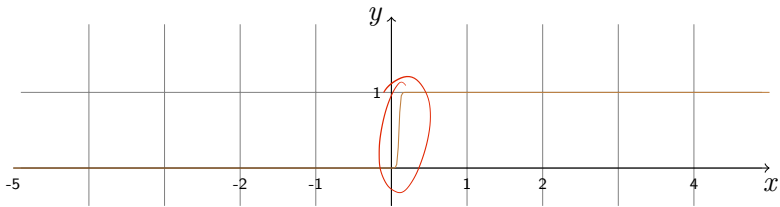
$$y = \frac{1}{1+e^{-(ax+b)}} \quad \text{(general form)}$$

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

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

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

The Logistic Function



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

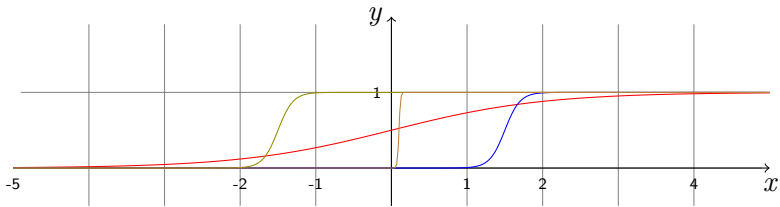
$$y = \frac{1}{1+e^{-(1*x+0)}}$$

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

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

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

The Logistic Function



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

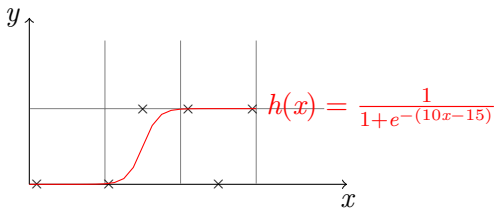
$$y = \frac{1}{1+e^{-(1*x+0)}}$$

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

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

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

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)

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

- Regression: Predicting probabilities ➔Binary classification
- 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*

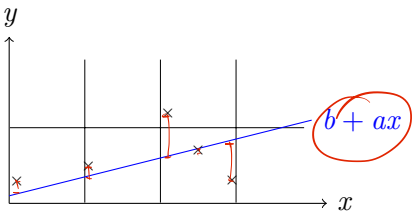
Learning Regression Models

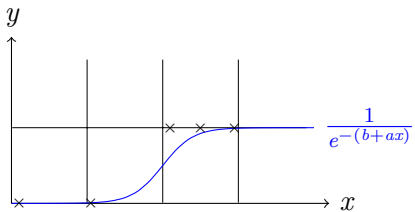
- How to select the parameters a, b such that the hypothesis function describes the data points as best as possible?
- Learning algorithm *Gradient Descent*

Gradient descent is half of the algorithms of deep learning

Loss: Intuition

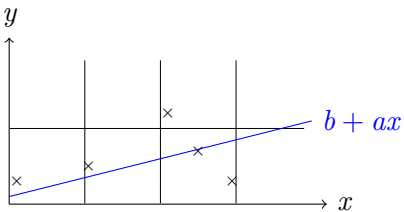
The *loss* measures the 'wrongness' of values for a and b.

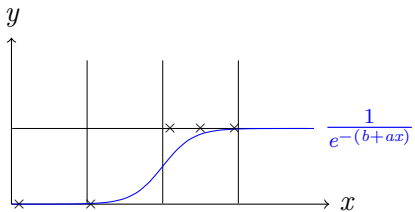




Loss: Intuition

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

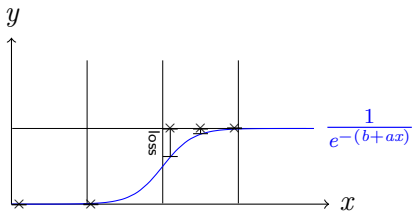




- How big is the gap between a hypothesis and the data?
- Is $(a, b) = (0.3, 0.5)$ or $(a, b) = (0.4, 0.4)$ better?

Loss: Intuition

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

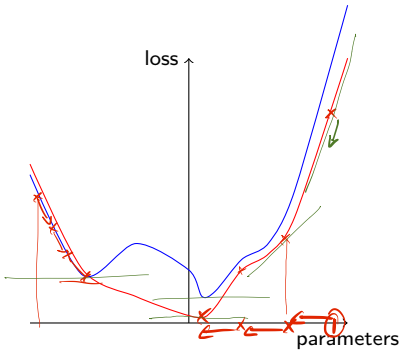


- How big is the gap between a hypothesis and the data?
- Is $(a, b) = (0.3, 0.5)$ or $(a, b) = (0.4, 0.4)$ better?

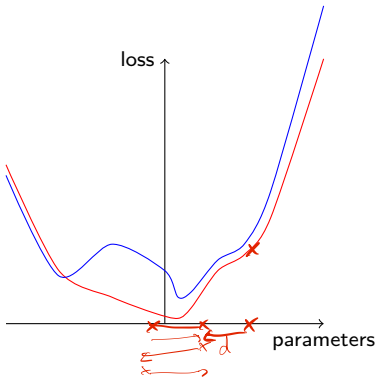
Loss function: Intuition

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

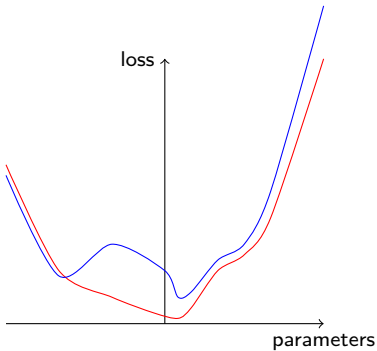


Loss function: Intuition



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

Loss function: Intuition



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

Hypothesis vs. Loss Function

- Hypothesis function h
  - Calculates outcomes, given feature values $\vec{x}$ – and parameter values $\vec{w} = (a, b)$
- 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)

(a, b)

Loss Function

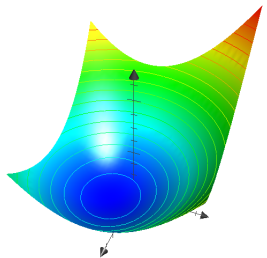


Figure: The loss function in our setting visualised

Loss Function

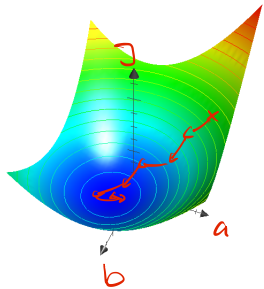


Figure: The loss function in our setting visualised

- Searching for the a, b settings with minimal loss
- $=$ Searching for the minimum!

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: $h_{\vec{w}} = w_1 x + w_0$

$\vec{w} = (a, b)$: 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

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

$h_{\vec{w}}(x_i) - y_i$ (Distanz; was herauskommen sollte; was herauskommt)

- 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

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

- 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: $h_{\vec{w}} = w_1 x + w_0$

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

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

- 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

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

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

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

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

$f(x) = x^2$

$f'(x) = 2x$

- 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

y	$h(x)$	Wie gut?	$h(x) - y$		$1-h(x)$
0	0.1	+	0.1	$1-h(x)$	0.9
0	0.9	−	0.9	$1-h(x)$	0.1
1	0.1	−	−0.9	$h(x)$	0.1
1	0.9	+	−0.1	$h(x)$	0.9

Loss function

Definition for Logistic Regression

- Two cases: $y_i = 0$ or $y_i = 1$ – y_i: real outcome for instance i
  - Caveat: $\log 0$ is undefined – add $\epsilon = 0.0000001$ if needed

Loss function

Definition for Logistic Regression

- Two cases: $y_i = 0$ or $y_i = 1$ – y_i: real outcome for instance i
  - Caveat: $\log 0$ is undefined – add $\epsilon = 0.0000001$ if needed

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

Loss function

Definition for Logistic Regression

- Two cases: $y_i = 0$ or $y_i = 1 - y_i$: real outcome for instance i
  - Caveat: $\log 0$ is undefined – add $\epsilon = 0.0000001$ if needed

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

y_i	$h_{\vec{w}}(x_i) + \epsilon$	$y_i \log h_{\vec{w}}(x_i) + (1 - y_i) \log(1 - h_{\vec{w}}(x_i))$
0	1.0000001	-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
  - Caveat: $\log 0$ is undefined – add $\epsilon = 0.0000001$ if needed

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

y_i	$h_{\vec{w}}(x_i) + \epsilon$	$y_i \log h_{\vec{w}}(x_i) + (1 - y_i) \log(1 - h_{\vec{w}}(x_i))$
0	1.0000001	-23.2535
0	0	0
1	1	0
1	0.0000001	-23.2535

Loss function

Definition for Logistic Regression

- Two cases: $y_i = 0$ or $y_i = 1 - y_i$: real outcome for instance i
  - Caveat: $\log 0$ is undefined – add $\epsilon = 0.0000001$ if needed

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

y_i	$h_{\vec{w}}(x_i) + \epsilon$	$y_i \log h_{\vec{w}}(x_i) + (1 - y_i)\log(1 - h_{\vec{w}}(x_i))$
0	1.0000001	−23.2535
0	0	0
1	1	0
1	0.0000001	−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
  - Caveat: $\log 0$ is undefined – add $\epsilon = 0.0000001$ if needed

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

y_i	$h_{\vec{w}}(x_i) + \epsilon$	$y_i \log h_{\vec{w}}(x_i) + (1-y_i)\log(1-h_{\vec{w}}(x_i))$
0	1.0000001	−23.2535
0	0	0
1	1	0
1	0.0000001	−23.2535
1	0.8	−0.3219281
1	0.2	−2.321928

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

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*

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 (because we interpret results as probabilities)
    - 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

Exercise 3

scikit-learn

- Take the Naive-Bayes-MNIST-Example and run it on your computer
- Adapt the code to use a logistic regression classifier instead
- Optional: Which classifier is better with a smaller training data set?

References I