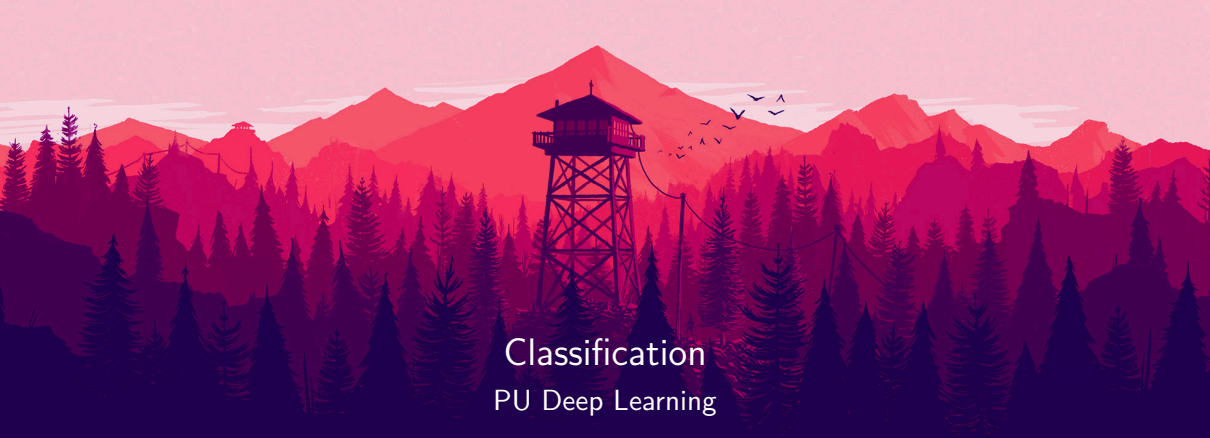


Classification

PU Deep Learning

Nils Reiter, nils.reiter@uni-koeln.de

May 11, 2021 (Summer term 2021)

Recap

- Input/Output
  - Stream-oriented
  - Open file, work with stream, close file
- Exceptions
  - Handle all kinds of runtime errors
  - `raise` to throw errors
  - `try: ... except:` to catch them
- Packaging
  - Use pip for installing python packages

Today

Classification

Logistic Regression

Gradient Descent

Scikit-Learn

Exercise

Section 1

Classification

Introduction

- Assigning *classes* to *objects/instances/items*
  - Words → parts of speech
  - Texts → genres
  - Sentences → polarity (positive/negative)

Introduction

- Assigning *classes* to *objects/instances/items*
  - Words → parts of speech
  - Texts → genres
  - Sentences → polarity (positive/negative)
- Many different models/algorithms available (all with variants):
  - Decision trees
  - Support vector machines
  - Naïve Bayes
  - Bayesian networks
  - ...

Introduction

- Assigning *classes* to *objects/instances/items*
  - Words → parts of speech
  - Texts → genres
  - Sentences → polarity (positive/negative)
- Many different models/algorithms available (all with variants):
  - Decision trees
  - Support vector machines
  - Naïve Bayes
  - Bayesian networks
  - ...
  - (Artificial) neural networks (a.k.a. »deep learning«)

Classification

Target classes

Classes: A finite set of categories

Examples

- Parts of speech: Noun, verb, adjective, ...
  - E.g., STTS tagset
- Genres: Abenteuerroman, Bildungsroman, Kriminalroman, ...
  - Many novels fall in multiple classes

Classification

Target classes

Classes: A finite set of categories

Examples

- Parts of speech: Noun, verb, adjective, ...
  - E.g., STTS tagset
- Genres: Abenteuerroman, Bildungsroman, Kriminalroman, ...
  - Many novels fall in multiple classes

Important first step: Clearly identify classes and problem properties

Two Parts

Prediction Model

How do we make predictions on data instances?
(e.g., how do we assign a part of speech tag for a word?)

Learning Algorithm

How do we create a prediction model, given annotated data?
(e.g. how do we create rules for assigning a part of speech tag for a word?)

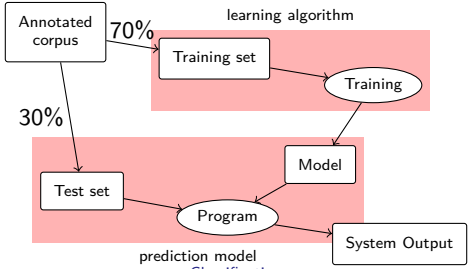
Two Parts

Prediction Model

How do we make predictions on data instances?
(e.g., how do we assign a part of speech tag for a word?)

Learning Algorithm

How do we create a prediction model, given annotated data?
(e.g. how do we create rules for assigning a part of speech tag for a word?)



Section 2

Logistic Regression

Introduction

- Linear regression: Prediction of numerical data
- Logistic regression: Prediction of nominal data
  - Here: Binary

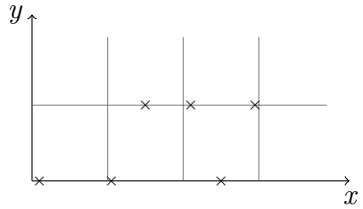
Introduction

- Linear regression: Prediction of numerical data
- Logistic regression: Prediction of nominal data
  - Here: Binary

Example (Nobel Prize Prediction)

Given the number of characters in a narrative text, will a book win the Nobel prize for literature?

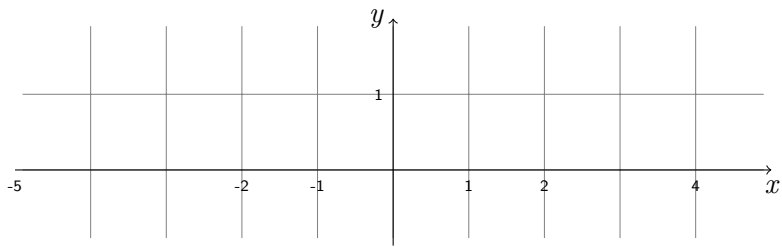
Data Representation



Which curve approximates the data points as well as possible?

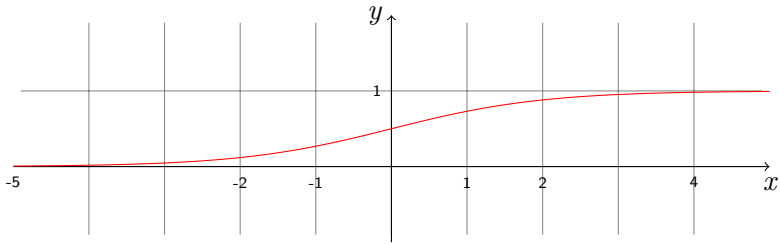
The Logistic Function

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = 2.71828$$



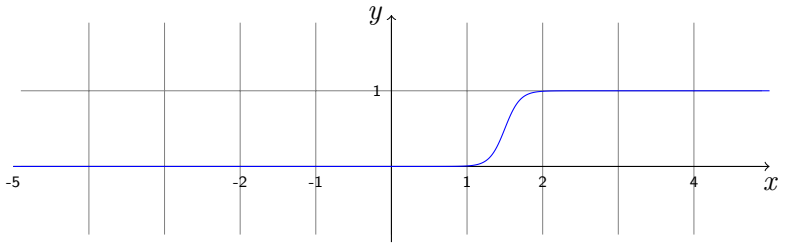
$$y = \frac{1}{1+e^{-x}} = \frac{1}{1+e^{-(ax+b)}}$$

The Logistic Function



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

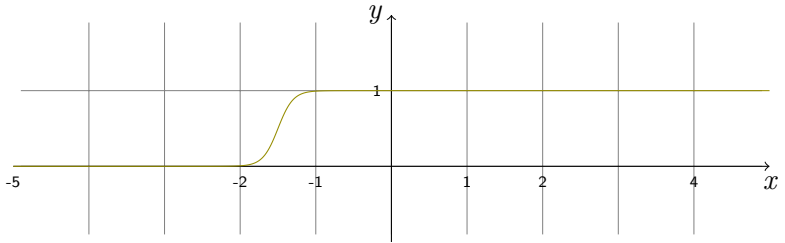
The Logistic Function



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

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

The Logistic Function

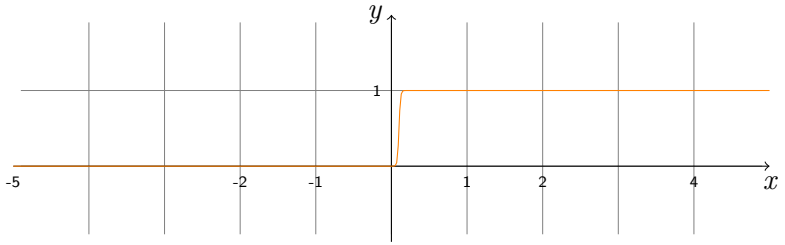


$$y = \frac{1}{1+e^{-x}} = \frac{1}{1+e^{-(ax+b)}} = \frac{1}{1+e^{-(1*x+0)}}$$

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

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

The Logistic Function



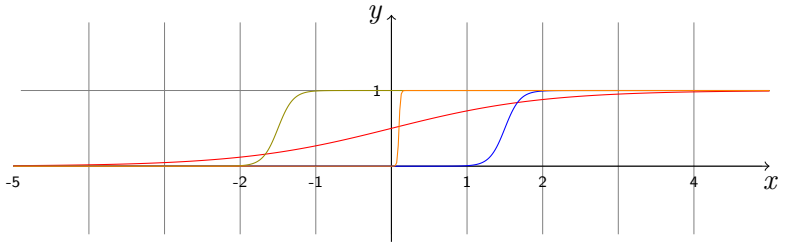
$$y = \frac{1}{1+e^{-x}} = \frac{1}{1+e^{-(ax+b)}} = \frac{1}{1+e^{-(1*x+0)}}$$

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

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

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

The Logistic Function



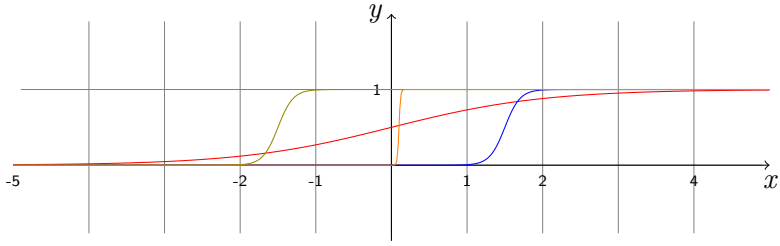
$$y = \frac{1}{1+e^{-x}} = \frac{1}{1+e^{-(ax+b)}} = \frac{1}{1+e^{-(1*x+0)}}$$

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

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

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

The Logistic Function



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

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

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

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

Learning algorithm: How to choose a and b?

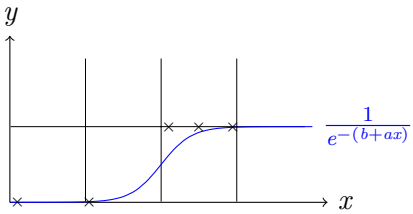
Section 3

Gradient Descent

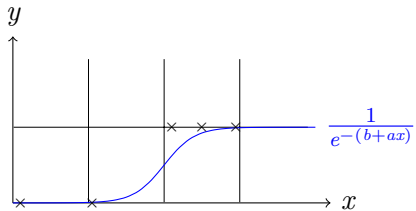
Learning Regression Models

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

Loss: Intuition

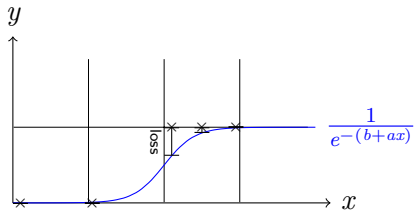


Loss: Intuition



- 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

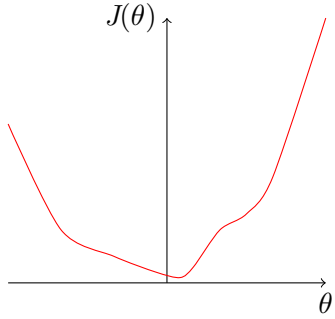


- How big is the gap between a hypothesis and the data?
- Is $(a, b) = (0.3, 0.5)$ or $(a, b) = (0.4, 0.4)$ better?

Loss function: Intuition

- Loss should be as small as possible
- Total loss can be calculated for given parameters $\theta = (a, b)$
- Idea:
  - We change θ 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 in which direction to go

Loss function: Intuition



Cost / Loss function

- Hypothesis function h
  Calculates outcomes, given feature values x
- Loss function J
  Calculates ›wrongness‹ of h, given parameter values θ (and a data set)
  - In reality, θ represents more than two parameters

Loss Function

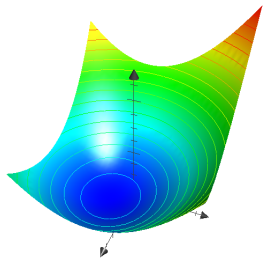


Figure: The loss function in our setting visualised

Loss Function

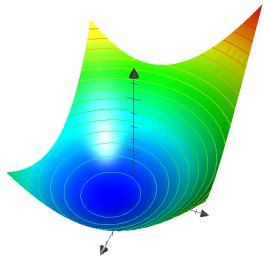


Figure: The loss function in our setting visualised

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

Loss Function

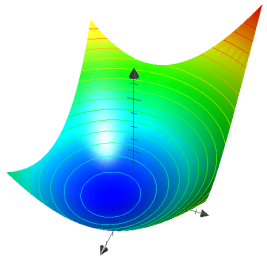


Figure: The loss function in our setting visualised

- Searching for the a, b settings with minimal loss
- $=$ Searching for the minimum!
- ...and we have efficient tools to search for the minimum of a curve (derivation etc.)

Loss function: Definition

Loss function depends on hypothesis function

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

Logistic
- $h(x) = \frac{1}{e^{-(b+ax)}}$
- Loss: (Binary) cross-entropy loss
  - Not the only choice

Loss function: Definition

$$J(a, b) =$$

Loss function: Definition

$$J(a, b) = -\frac{1}{m}\sum_{i=0}^{m}$$

Loss function: Definition

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

Loss function: Definition

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

Loss function: Definition

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

Log probabilities

- Relative order is stable: If $a > b$, $\log a > \log b$
  - No information loss
- Multiplication turns to addition: $\log(a \cdot b) = \log a + \log b$
  - If done many times, addition is much faster than multiplication

Summary

Gradient Descent

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

Section 4

Scikit-Learn

Introduction

- Generic machine learning library for Python
- Classification algorithms: Naive Bayes, support vector machines, ...
- Clustering algorithms: KMeans, agglomerative clustering, ...

Introduction

- Generic machine learning library for Python
- Classification algorithms: Naive Bayes, support vector machines, ...
- Clustering algorithms: KMeans, agglomerative clustering, ...
- Utility functions
  - Cross-validation, training and test splits
  - Evaluation (precision/recall/f-score)

Workflow

Preprocessing and Preparations

- Preprocessing
  - Read in data files
  - Remove columns that we cannot handle or columns that we feel are irrelevant
  - Convert features into numeric representations
  - Split into train and test data
  - Deal with missing values
  - Add additional features from other resources
  - ...

Workflow

Preprocessing and Preparations

- Preprocessing
  - Read in data files
  - Remove columns that we cannot handle or columns that we feel are irrelevant
  - Convert features into numeric representations
  - Split into train and test data
  - Deal with missing values
  - Add additional features from other resources
  - ...
- Preparations
  - Split into x, y
  - Split into train and test
- Machine learning $\rightarrow$ next slide

Workflow

Machine Learning

- Initialize object `cls = Classifier(...)`
- Call `cls.fit(x_train, y_train)` to train
- Call `y_pred = cls.predict(x_test)` to get predictions on the test set
- Call `*_score(y_test, y_pred)` to calculate evaluation scores

Workflow

Machine Learning

- Initialize object `cls = Classifier(...)`
- Call `cls.fit(x_train, y_train)` to train
- Call `y_pred = cls.predict(x_test)` to get predictions on the test set
- Call `*_score(y_test, y_pred)` to calculate evaluation scores
- »Initialize–fit–predict« pattern

Section 5

Exercise

Exercise 05

`https://github.com/idh-cologne-deep-learning/exercise-05`