Machine Learning

– winter term 2016/17 –

Course Mechanics

Prof. Adrian Ulges
Masters "Computer Science"
DCSM Department
University of Applied Sciences RheinMain

Outline

1. Projects!

2. 27.10.

3. Introduction

Food for Thought

Wikification

- Demo should enrich **some** kind of text with Wikipedia
- Live demo!?
- Remember: Choose Wikipedia Category and detect+link key terms
- Training data: Wikipedia (**text + link structure**)
- focus on certain categories?

Gaze Tracking

- steps: camera calibration, landmark detection, normalization, angle estimation
- hardware: webcam? Maybe limit head pose?
- Training phase for user?
- Live Demo!?

Food for Thought

Fish on Kaggle?

- in general, why not?
- Condition: Stick to your **own** approach
- Condition: Live Demo (?)

Outline

1. Projects!

2. 27.10.

3. Introduction

Organisatorisches

Prüfungsanmeldung

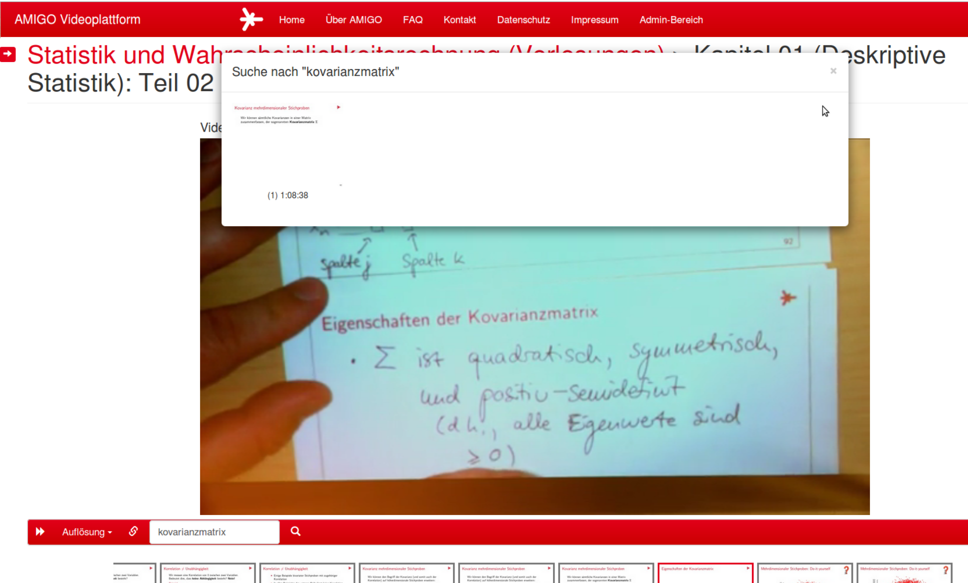
- seeehr analog → Zettel unterschreiben!
- heute (?)

Pause heute

- im zweiten Block (bitte dran erinnern / laut gähnen o.ä.)

http://video.cs.hs-rm.de (AMIGO Video-Plattform)

- Empfohlen: Firefox (mobiler Zugriff möglich).



AMIGO Videoplattform | Home | Über AMIGO | FAQ | Kontakt | Datenschutz | Impressum | Admin-Bereich

Statistik und Wahrscheinlichkeitsrechnung (Vorlesungen) › Kapitel 01 (Deskriptive Statistik): Teil 02

Vide

Suche nach "kovarianzmatrix" ×

Kovarianz mehrdimensionaler Stichproben

Wir können sämtliche Kovarianzen in einer Matrix zusammenfassen, der sogenannten Kovarianzmatrix Σ

(1) 1:08:38

Spalte j Spalte k

92

Eigenschaften der Kovarianzmatrix

- Σ ist quadratisch, symmetrisch, und positiv-Semidefinit (d.h., alle Eigenwerte sind ≥ 0)

Auflösung ▾ | kovarianzmatrix

Korrelation ≠ Unabhängigkeit | Korrelation ≠ Unabhängigkeit | Kovarianz mehrdimensionaler Stichproben | Kovarianz mehrdimensionaler Stichproben | Kovarianz mehrdimensionaler Stichproben | Eigenschaften der Kovarianzmatrix | Mehrdimensionaler Stichproben: Do-it-yourself | Mehrdimensionaler Stichproben: Do-it-yourself

Outline

1. Projects!

2. 27.10.

3. Introduction

Videos

- Der Plan ist, diese Vorlesung zu **filmen**
- Der Plan war, die Vorlesungen online zu stellen
- Wie stehen Sie dazu?
- **Problem**: Art 52a UrhG

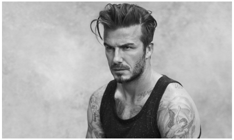
Course Language: English

But Why?

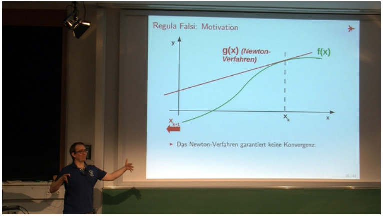
- ML terminology is mostly English anyway
- For the learning experience

But How?

- Lecturer speaks English
- Students speak whatever they like *(English appreciated)*
- Fachgespräch in German *(English terminology where it feels natural)*
- Coursework in German/English
- Presentations in X (?)
- Vocabulary issues ...? *(speak up!)*

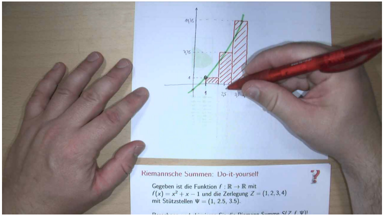


Course Process



Regula Falsi: Motivation

- Das Newton-Verfahren garantiert keine Konvergenz.



Riemannsche Summen: Do-it-yourself

Gegeben ist die Funktion $f : \mathbb{R} \to \mathbb{R}$ mit $f(x) = x^2 + x - 1$ und die Zerlegung $Z = (1, 2, 3, 4)$ mit Stützstellen $\Psi = (1, 2.5, 3.5)$.

Berechnen und skizzieren Sie die Riemann-Summe $S(Z, f, \Psi)$!

Part 01 (7 Weeks)

- **2 blocks** of Lecture-style presentation
- Some interactive parts ("Do-it-yourselfs"), lots of questions
- Writing on some slides → Do print slides beforehand!
- Weekly **coursework**

Part 02 (7 Weeks)

- **Project**: Solving a larger-scale machine learning problem

Time Table (tentative)

20.10.	Lecture 01 Introduction	Lecture 02 Decision trees	exercises
27.10.	Lecture 03 Feature Engineering	Lecture 04 N. Bayes + graph. models	exercises
03.11.	Lecture 05 Instance-based Learning	Lecture 06 Logistic Regression	exercises
10.11.	Lecture 07 Unsupervised Learning I	Lecture 08 Unsupervised Learning II	exercises
17.11.	Lecture 09 Neural Learning I	Lecture 10 Neural Learning II	exercises
24.11.	Lecture 11 Deep Learning	Lecture 12 Deep Learning @ DFKI	exercises
01.12.	Lecture 13 Recommender Systems	exercises	
08.12.	project		
15.12.	project		
22.12.(duh)	project		
12.01.	project		
19.01.	project		
26.01.	project		
02.02.	Project Presentations		

Learning Objectives

After this course, you should …

- … have an algorithmic understanding of the **most common machine learning techniques** (including 9 of the *Top 10 Algorithms in Data Mining*[1])
- … be able to **assess** the benefits and shortcomings of ML algorithms
- … be able to **apply** ML algorithms using state-of-the-art technology *(Python)*
- … have some understanding of the **“dark art” aspects** of ML
- … have gone through an **experimental development cycle** of an ML system.

[1] Wu et al., *Top 10 Algorithms in Data Mining, Knowl. Inf. Syst., 2008.*

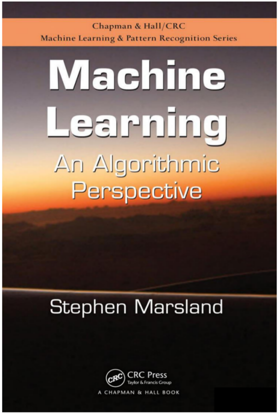
Ressources

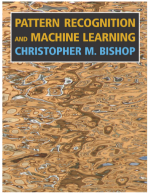
Primary Read

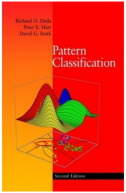
- S. Marsland, *Machine Learning: An Algorithmic Perspective* (Chapman & Hall/CRC).
- Several copies available in the library

Other Good Reads

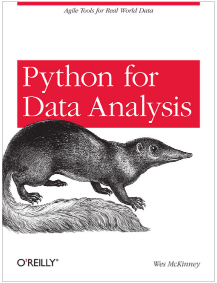
- Bishop: Pattern Recognition and Machine Learning (Springer)
- Duda Hart Stork: Pattern Classification (Wiley Interscience)
- Specific resources per chapter (later)
- Software: R, (matlab), numpy, scipy, matplotlib, scikit-learn







There’s more...

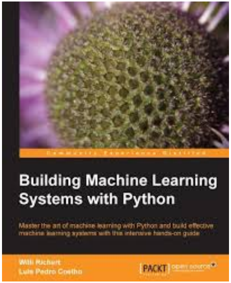


Agile Tools for Real World Data

Python for Data Analysis

O’REILLY®

Wes McKinney



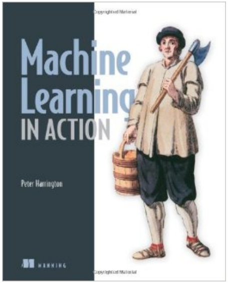
Community Experience Distilled

Building Machine Learning Systems with Python

Master the art of machine learning with Python and build effective machine learning systems with this intensive hands-on guide

Willi Richert
Luis Pedro Coelho

PACKT open source



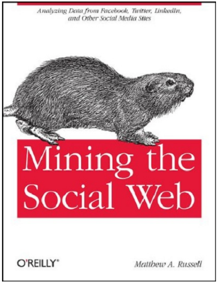
Copyrighted Material

Machine Learning IN ACTION

Peter Harrington

MANNING

Copyrighted Material

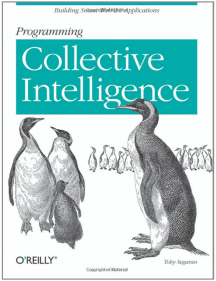


Analyzing Data from Facebook, Twitter, LinkedIn, and Other Social Media Sites

Mining the Social Web

O’REILLY®

Matthew A. Russell



Building Smart Web 2.0 Applications

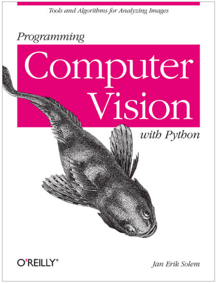
Programming

Collective Intelligence

O’REILLY®

Toby Segaran

Copyrighted Material



Tools and Algorithms for Analyzing Images

Programming

Computer Vision

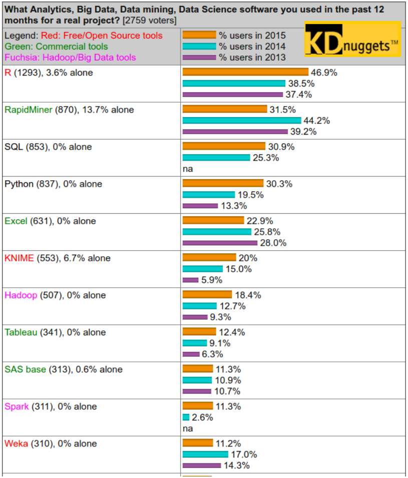
with Python

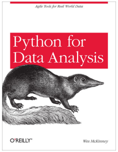
O’REILLY®

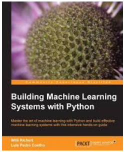
Jan Erik Solem

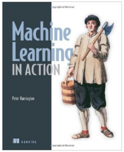
Python: Why?

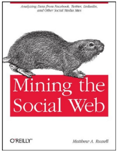
- OO scripting language with functional elements
- easy to learn *("pseudo-code that runs")*, powerful data structures + libraries
- widely used: most popular general-purpose scripting language *(rank 5 in the TIOBE Top Ten*[2]*)*
- very popular for data analysis

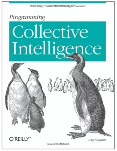


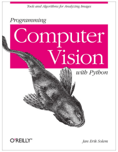












[2] `http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html`

The Python ML Ecosystem

Numpy

- working with vectors + matrices
- array operations for compact code
- numerical algorithms wrapped

Scikit-Learn (sklearn)

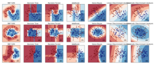
- machine learning algorithms!!!

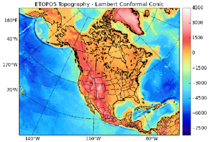
Tensorflow

- neural networks + deep learning
- scalable, portable, GPU support
- Alternatives: Caffe, Theano

Matplotlib

- visualization





Pandas

- 'excel perspective'
- feature engineering

Jupyter Notebook

- run experiments in a Wiki
- demo

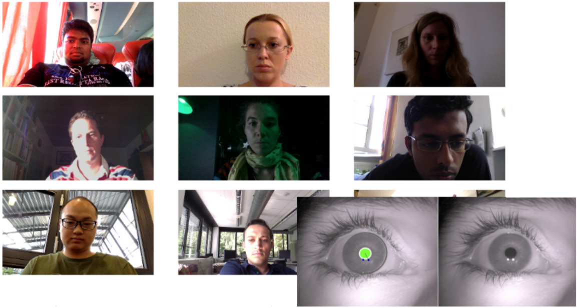
Projects

You Will...

- ... work in teams of 2 *(for the whole course, actually)*
- ... solve a machine learning challenge
- ... present your results in a talk *(last week)*
- ... honor the **ground rule** *(do not train on the testing data)*

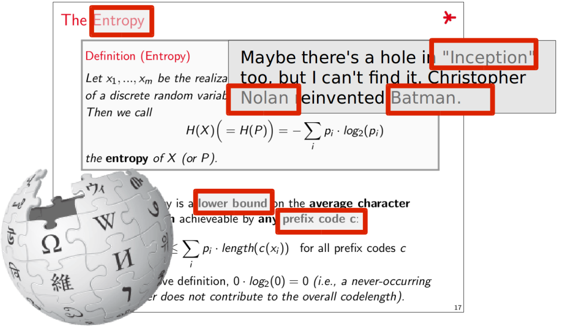
Challenge 1: Gaze Tracking

- Build a system that detects where on a computer screen the **user is looking**
- eye detection → pupil tracking → estimating gaze direction
- optionally: adapt to a specific user
- build a demo system
- Technology: Deep CNNs, Tensorflow



Challenge 2: Wikification

- Vision: Enrich any piece of text with **Wikipedia**
- Step 1: Sort text into the Wikipedia category tree
- Step 2: Detect keywords and link them to Wikipedia articles
- build a demo system
- Training data: Wikipedia (**text + link structure**)
- Technology: free of choice (NLTK recommended)



The Entropy

Definition (Entropy)

Let $x_1, \ldots, x_m$ be the realiza
of a discrete random variab
Then we call

$$H(X)\Big(=H(P)\Big) = -\sum_i p_i \cdot log_2(p_i)$$

the **entropy** *of X (or P).*

Maybe there's a hole in "Inception" too, but I can't find it. Christopher Nolan reinvented Batman.

y is a lower bound on the **average character**
h achieveable by **any** prefix code c:

$\leq \sum_i p_i \cdot length(c(x_i))$ for all prefix codes *c*

ve definition, $0 \cdot log_2(0) = 0$ (*i.e., a never-occurring*
r does not contribute to the overall codelength).

17

Grading

- Grading for this course will adhere to the rules of **Praktische Tätigkeit und Fachgespräch**
- Remember to sign in early on QIS!
- Attendance to “Praktikum” will be recorded *(do not miss more than 3!)*
- **Final grade**: 30% weekly coursework, 40% project, 30% Fachgespräch

1. Weekly Coursework

- Before the project phase, there will be weekly (mostly programming) exercises
- Grading based on brief “weeklies”
  - general progress
  - brief demo / results
  - code walk-through

Grading (cont'd)

2. Project

- Main deliverable: **project presentation** *(experiments+reading done, motivation for investigative path, scientific standard)*
- Do record results **while** working on the project!
- I will also check code *(should be able to install and run)*
- Please create a Gitlab repo

3. Fachgespräch

- Will use your project work as basis, but also include questions from lecture/exercises
- Prepare like for an **oral exam**!