Einführung in die mathematische Datenanalyse

Jan Heiland

FAU Erlangen-Nürnberg – Sommersemester 2022

Contents

Vorwort

Das ist ein Aufschrieb.

Korrekturen und Wünsche immer gerne als *issues* oder *pull requests* ans github-repo.

Chapter 1

Was ist Data Science?

Data Science umfasst unter anderem folgende Aufgaben:

1. Strukturieren/Aufbereiten (Umgehen mit falschen, korrumpierten, fehlenden, unformatierten Daten)
2. Data Exploration (Daten "verstehen")
3. Data Analysis (quantitative Analysen, Hypothesen aufstellen)
4. Data Visualization (Hypothesen graphisch kommunizieren)
5. Modelle erzeugen/validieren (Regeln/Muster erkennen, Vorhersagen treffen) – *das* ist Machine Learning aber es gibt auch viele andere Ansätze.
6. Daten Reduktion

1.1 Wie passiert die Datenanalyse?

Mit mathematischen Methoden aus den Bereichen der

- linearen Algebra (z.B. Matrizen, Basen, lineare Gleichungssysteme)
- Statistik (z.B. Mittelwerte, Korrelationen, Verteilungen)
- Analysis (Grenzwerte, Abschätzungen)
- …

Dabei hilft Software, z.B.,

- Excel
- **Python**
- Matlab
- R

bei der Berechnung, Automatisierung, Visualisierung.

Python ist ein de-facto Standard in Data Science und Machine Learning.

1.2 Was sind Daten?

Wie sehen Daten aus?

- Numerisch reell, z.B. Temperatur
- Numerisch diskret, z.B. Anzahl
- Ordinal: Element einer festen Menge mit expliziter Ordnung, z.B. {neuwertig, mit Gebrauchsspuren, defekt}
- Binär: Eine von zwei Möglichkeiten, z.B. Wahr/Falsch oder aktiv/inaktiv
- Kategoriell: Element einer festen Menge ohne klare Ordnung, z.B. {Säugetier, Vogel, Fisch}
- sonstige strukturierte Daten, z.B. geographische Daten, Graphen
- reiner Text, z.B. Freitext in Restaurantbewertung

Außerdem können wir noch allgemeine Eigenschaften (Qualitätsmerkmale) von Daten unterscheiden

- strukturiert
- lückenhaft
- fehlerbehaftet (*verrauscht*)
- interpretierbar
- geordnet (oder nicht zu ordnen)

1.3 Beispiele

1.3.1 Tabellendaten – Mietpreise

Hier wären die Aufgaben von Data Science:

- Daten "verstehen", Zusammenhänge zwischen Variablen aufdecken,
- visualisieren.
- Gegebenenfalls fehlende Einträge bei (z.B.) `kaltmiete` vorhersagen

Datenexploration und -analyse für einzelne Variablen 1/3 Wir betrachten eine *numerische* Variable in einem rechteckigen Datensatz, also eine *Spalte* (z.B. `kaltmiete`). Wir bezeichnen den i-ten Eintrag in dieser Spalte mit x_i, wobei $i = 1, \ldots, N$ (N Anzahl der Zeilen).

Folgende *Schätzer/Metriken* können dabei helfen, diese Spalte besser zu verstehen:

- Mittelwert $\overline{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$

	bundesland	stadt	baujahr	etage	hat_kueche	kaltmiete	wohnflaeche	zimmer
0	Nordrhein_Westfalen	Duisburg	1973	1	0	640	105	3
1	Baden_Württemberg	Ulm	1936	1	0	302	43	2
2	Nordrhein_Westfalen	Wuppertal	1986	3	0	150	67	2
3	Rheinland_Pfalz	Rhein_Lahn_Kreis	1970	3	0	315	47	3
4	Sachsen	Chemnitz	1900	2	0	315	63	2
5	Rheinland_Pfalz	Mainz	1989	1	0	1200	96	4
6	Bayern	Aschaffenburg_Kreis	1965	2	0	722	85	3
7	Berlin	Berlin	1952	0	1	706	63	2
8	Sachsen	Mittelsachsen_Kreis	1996	3	1	220	29	1
9	Brandenburg	Potsdam	1985	2	1	400	34	1

Figure 1.1: Abbildung: Tabelle von Wohnungsangeboten

	bundesland	stadt	baujahr	etage	hat_kueche	kaltmiete	wohnflaeche	zimmer
0	Nordrhein_Westfalen	Duisburg	1973	1	0	640	105	3
1	Baden_Württemberg	Ulm	1936	1	0	302	43	2
2	Nordrhein_Westfalen	Wuppertal	1986	3	0	150	67	2
3	Rheinland_Pfalz	Rhein_Lahn_Kreis	1970	3	0	315	47	3
4	Sachsen	Chemnitz	1900	2	0	315	63	2
5	Rheinland_Pfalz	Mainz	1989	1	0	1200	96	4
6	Bayern	Aschaffenburg_Kreis	1965	2	0	722	85	3
7	Berlin	Berlin	1952	0	1	706	63	2
8	Sachsen	Mittelsachsen_Kreis	1996	3	1	220	29	1
9	Brandenburg	Potsdam	1985	2	1	400	34	1

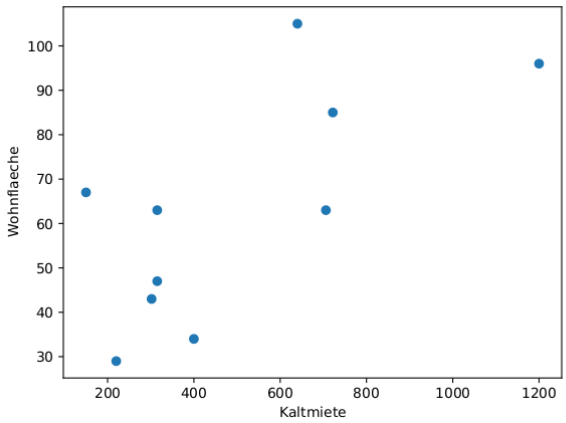
Figure 1.2: Abbildung: Eine Spalte der Tabelle

- gewichteter Mittelwert $\overline{x}_w = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{j=1}^{N} w_j}$, wobei w_i das Gewicht des i-ten Eintrages ist (z.B. eine andere Variable).
- Varianz: $s_x^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \overline{x})^2$
- Standardabweichung $s = \sqrt{s_x^2}$.
- Median $= \frac{315+400}{2} = 357.5$.

Datenexploration und -analyse für mehrere Variablen Wir betrachten zwei Spalten $x = (x_1, \dots, x_N)$ und $y = (y_1, \dots, y_N)$.

Das Verteilung von zwei Variablen läßt sich im sogenannte **Scatter Plot** visualisieren.

che	kaltmiete	wohnflaeche	zin
0	640	105	
0	302	43	
0	150	67	
0	315	47	
0	315	63	
0	1200	96	
0	722	85	
1	706	63	
1	220	29	
1	400	34	



Datenexploration und -analyse für mehrere Variablen Wir betrachten zwei Spalten $x = (x_1, \dots, x_N)$ und $y = (y_1, \dots, y_N)$.

- Kovarianz $s_{xy} = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})$
- Korrelation $\rho_{xy} = \frac{s_{xy}}{s_x \cdot s_y} \in [-1, 1]$.
- $\rho \approx 1$: Starke positive Korrelation, wenn x groß ist, ist y auch groß.
- $\rho \approx -1$: Starke negative Korrelation, wenn x groß ist, ist y klein
- $\rho \approx 0$: Wenig/keine Korrelation.

1.3.2 COVID-19 Daten

Vergleiche die Einführung in *Mathematik für Data Science 1* vom letzten Semester.

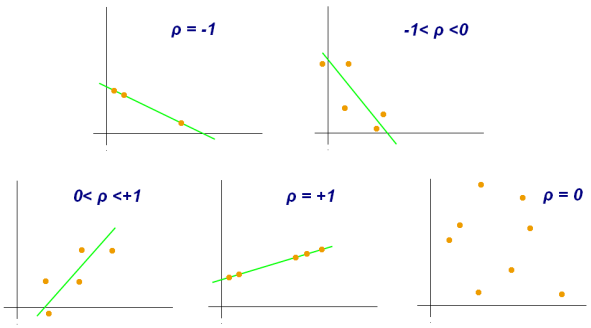


Figure 1.3: Von Kiatdd - Eigenes Werk, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=37108966

1.3.3 Netflix Prize

Hierbei geht es darum, ob aus bekannten Bewertungen von vielen verschiedenen Benutzern für viele verschiedene Filme abgeleitet werden kann, ob ein bestimmter Nutzer einen bestimmten Film mag (also positiv bewerten würde).

Vergleiche auch Wikipedia:Netflix_Prize

Das (Trainings-)Daten bestehen über `480189` Benutzer, die für `17770` Filme insgesamt `100480507` Bewertungen als ganze Zahlen zwischen `1` und `5` verteilten.

Ziel der Datenanalyse war es, für `2817131` "Paare" von Benutzern und Filmen, die Bewertung vorauszusagen. Neben der schieren Masse an Daten kamen noch Einschränkungen hinzu, die ein Mindestmaß an Qualität der Vorhersage sicherstellen sollten.

Das Problem ließe sich wie folgt darstellen.

Benutzer \ Film	`F1`	`F2`	...	`Fn`	...
`B1`	–	3	...	5	...
`B2`	3	4	...	2	...
`B3`	1	2	...	?	...
...	3	4	...	–	...

Gegeben viele (aber bei weitem nicht alle) Einträge in einer riesigen Tabelle. Können wir aus den Zusammenhängen bestimmte fehlende Einträge (z.B. wie findet Nutzer `B3` den Film `Fn`) herleiten?

Die besten Lösungen für dieses Problem basieren durchweg auf *Machine Learning* Ansätzen.

1.4 Python

Die Programmiersprache `python` wird uns durchs Semester begleiten. Einfach weil sie so wichtig ist für *Data Science* aber auch weil sie (meiner Meinung nach) einfach zu erlernen und zu benutzen ist.

1.5 Aufgaben

1.5.1 Python

Bringen sie ihr `python` zum Laufen, installieren sie `numpy`, `scipy` und

```
N = 20
xmax = 2
xmin = 0

xdata = np.linspace(xmin, xmax, N)
ydata = np.exp(xdata)

plt.figure(1)
plt.plot(xdata, ydata, '.')

plt.figure(2)
plt.semilogy(xdata, ydata, '.')
plt.show()
```

1.5.2 Einheitsmatrix

Schreiben sie ein script, dass die `5x5` Einheitsmatrix auf 3 verschiedene Arten erzeugt. (Eine Art könnte die eingebaute `numpy` Funktion `eye` sein).

```
import numpy as np

idfive = np.eye(5)
print(idfive)
```

Hinweis: schauen sie sich mal an wie `numpy`'s `arrays` funktionieren.

1.5.3 Matrizen Multiplikation und Potenz

Schreiben sie ein script, das die Übungsaufgabe aus der Vorlesung (potenzieren der Matrizen M_i, $i = 1, 2, 3, 4$) löst. Zum Beispiel mit

```
import numpy as np
mone = np.array([[0.9, 0.9], [0.9, 0.9]])

mone_ptwo = mone @ mone
print(mone_ptwo)

mone_pfour = mone_ptwo @ mone_ptwo
print(mone_pfour)
```

Oder so:

```
import numpy as np
mone = np.array([[0.9, 0.9], [0.9, 0.9]])
mone_p = np.eye(2)

for k in range(16):
```

```
    mone_p = mone_p @ mone
    if k == 1 or k == 3 or k == 15:
        print('k=', k+1)
        print(mone_p)
```

Achtung:

- bei Matrizen kann auch `*` benutzt werden – das ist aber nicht die richtige Matrizenmultiplikation (sondern die Multiplikation eintragsweise)
- Moegliche Realisierung der Matrizenmultiplikation
  - `np.dot(A, B)` – die klassische Methode
  - `A.dot(B)` – das selbe (manchmal besser, wenn `A` etwas allgemeiner ist (zum Beispiel eine `scipy.sparse` matrix)
  - `A @ B` – convenience Notation