

5. Programmieraufgabe Computerorientierte Mathematik I

Abgabe: 08.12.2023 über den Comajudge bis 17 Uhr

Bitte beachten Sie: Die Herausgabe oder der Austausch von Code (auch von Teilen) zu den Programmieraufgaben führt für *alle* Beteiligten zum *sofortigen Scheinverlust*. Die Programmieraufgaben müssen von allen Teilnehmenden alleine bearbeitet werden. Auch Programme aus dem Internet dürfen nicht einfach kopiert werden.

1 Problembeschreibung

Gegeben sei eine Menge von n Punkten (x_i, y_i) mit $i = 1, \dots, n$. Bei der linearen Regression ist eine Gerade $a \cdot x + b$ gesucht, die den quadratischen Abstand zur Punktmenge minimiert. Der quadratische Abstand ist definiert durch

$$\sum_{i=1}^n (a \cdot x_i + b - y_i)^2.$$

Eine optimale Lösung der linearen Regression kann folgendermaßen berechnet werden:

$$a = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \qquad b = \bar{y} - a \cdot \bar{x},$$

wobei

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^n x_i \qquad \bar{y} = \frac{1}{n} \cdot \sum_{i=1}^n y_i$$

die Mittelwert der x- bzw. y-Koordinaten sind.

In dieser Programmieraufgabe sollen Sie für eine gegebene Punktmenge und einer gegebenen Menge von Geraden, die Gerade aus der Menge finden, die den geringsten quadratischen Abstand hat und diesen Abstand mit dem quadratischen Abstand der optimalen Lösung vergleichen.

2 Aufgabenstellung und Anforderungen

1. Schreiben Sie eine Funktion

`get_linedistance(points, line),`

die den quadratischen Abstand zwischen einer Menge an Punkten und einer Geraden berechnet und ausgibt.

2. Schreiben Sie eine Funktion

`get_optimal_line(points),`

die die Parameter a und b einer optimalen Lösung der linearen Regression berechnet und als Tupel mit `return` ausgibt.

3. Schreiben Sie eine Funktion

`distance_to_opt(points, lines),`

die zunächst den kleinsten quadratischen Abstand von den gegebenen Geraden zu einer gegebenen Punktmenge bestimmt und anschließend die Differenz mit dem quadratischen Abstand der optimalen Lösung mit `return` ausgibt.

2.1 Eingaben

Als Eingabe werden die beiden Listen `points` und `lines` übergeben, welche jeweils aus Tupeln bestehen. Die Tupel beschreiben in `points` die Punkte in der Form $(x_i, y_i) \in \mathbb{Z}^2$ und in `lines` die Geraden in der Form $(a, b) \in \mathbb{Z}^2$. Die Eingabe `line` ist entsprechend ein Tupel $(a, b) \in \mathbb{Z}^2$.

2.2 Beispielaufufe

```
1 >>> points = [(-2,2),(1,3),(2,2),(6,-2)]
2 >>> lines = [(1,1),(-1,4)]
3 >>> d0 = get_linedistance(points, lines[0])
4 >>> d1 = get_linedistance(points, lines[1])
5 >>> print([d0, d1])
6 [92, 16]
7
8 >>> (a,b) = get_optimal_line(points)
9 >>> print(a)
10 -0.5419847328244275
11 >>> print(b)
12 2.198473282442748
13
14 >>> get_linedistance(points, (a,b))
15 5.129770992366414
16 >>> distance_to_opt(points, lines)
17 10.870229007633586
18
```

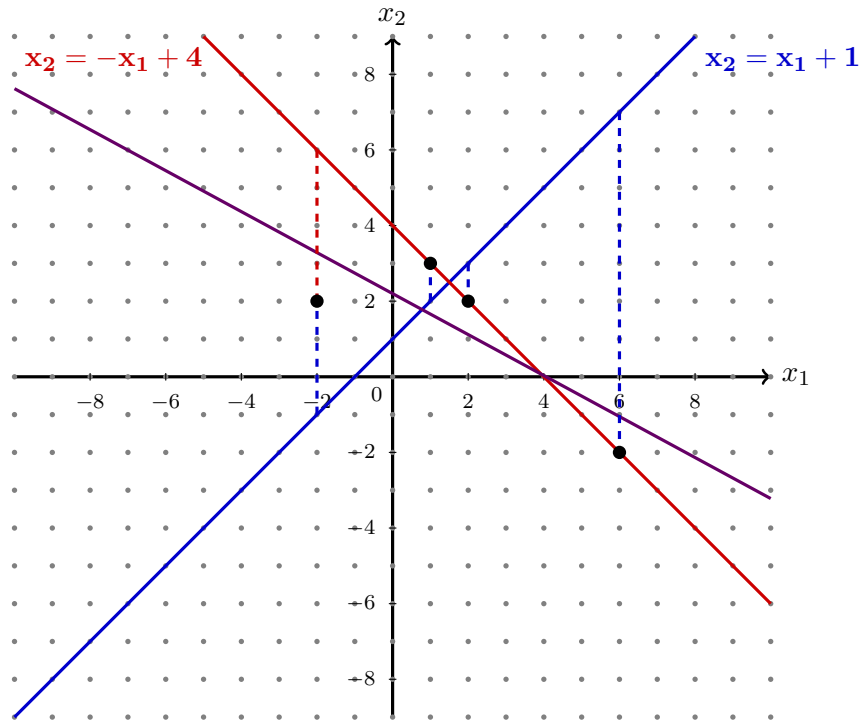


Abbildung 1: Visualisierung der Beispielaufrufe. Die blaue und die rote Gerade entsprechen den Geraden in `lines`.

3 Tipps und Anmerkungen

- In den Berechnungen treten Kommazahlen vom Datentyp `float` auf. Sie können diese ganz normal verwenden, ohne sich über Rundungsfehler Gedanken machen zu müssen.
- Alle angegebenen Funktionen werden vom Comajudge in ihrer beschriebenen Form verlangt und getestet.
- Zu Ihrer Erinnerung: Sie müssen eine der Programmieraufgaben PA04, PA05 oder PA06 bei einem Tutor oder einer Tutorin in den Rechnerbetreuungen vorstellen. Die vorläufige Deadline ist der 22.12.23.