Dokumentation zum Modul `least_squares.py`

Marisa Breßler und Anne Jeschke (PPI27)

14. Januar 2020

Inhaltsverzeichnis

1 Methoden

1.0.1 `read_input(filename, selection=None, number_of_columns=3)`

Diese Methode liest die Eingabe aus einer gegebenen Datei und übersetzt sie in einen mehrsimensionalen Array. Die Datei enthält für jede Messung eine Zeile und die Messwerte einer Messung sind in der Zeile durch Kommata getrennt. In unserem Fall sind es immer drei Messwerte pro Messung, dies ist jedoch variabel.

Input:

`filename` *(String)* Name der Eingabedatei

`selection=None` *(list of integers, optional)* Liste der Zeilen, die aus der Datei gelesen werden sollen. Falls alle Zeilen gelesen werden sollen, muss dies nicht gesetzt werden.

Returns:

(np.ndarray) Eingabedaten als mehrsimensionaler Array

1.0.2 `create_lgs(data, number_of_unknowns)`

Diese Methode erstellt ein lineares Gleichungssystem (LGS) der Form $Ax = b$ aus den Eingabedaten, d.h. eine Matrix A und einen Vektor b, für die einfache oder mehrfache lineare Regression. Die erste Spalte des Eingabearrays wird dabei zum Vektor b und eine variable Anzahl der restlichen Spalten plus eine Spalte Einsen zur Matrix A.

Input:

`data` *(np.ndarray)* Eingabedaten, wie oben beschrieben

`number_of_unknowns` *(int)* Anzahl der Spalten der Matrix A, bzw. Anzahl n der unbekannten x_1 bis x_n in der linearen Regression $p_0 = p_1 x_1 + ...p_{n-1} x_{n-1} + x_n$

Returns:

(np.ndarray) Matrix A des LGS

(np.ndarray) Vektor b des LGS

1.0.3 `create_lgs_p2(data)`

Diese Methode erstellt ein lineares Gleichungssystem (LGS) der Form $Ax = b$ aus den Eingabedaten, d.h. eine Matrix A und einen Vektor b, für die einfache lineare Regression der Form $p_0 = p_2 x_0 + x_1$. Die erste Spalte des Eingabearrays wird dabei zum Vektor b und die dritte Spalte plus eine Spalte Einsen zur Matrix A.

Input:

`data` *(np.ndarray)* Eingabedaten, wie oben beschrieben

Returns:

(np.ndarray) Matrix A des LGS

(np.ndarray) Vektor b des LGS

1.0.4 `get_qr(A)`

Diese Methode erstellt die QR-Zerlegung einer Matrix A mittels der scipy-Bibliotheksfunktion.

Input:

`data` *(np.ndarray)* Matrix A

Returns:

(np.ndarray) Orthogonale Matrix Q der QR-Zerlegung

(np.ndarray) Rechte obere Dreiecksmatrix R der QR-Zerlegung

1.0.5 `full_rank(A)`

Diese Methode testet, ob die Eingabematrix A einen vollen Spaltenrang hat.

Input:

`A` *(np.ndarray)* Matrix A

Returns:

(bool) True, wenn voller Spaltenrang. False sonst.

1.0.6 `solve_qr(A, b)`

Diese Methode löst das ggf. auch überbestimmte LGS $Ax = b$ mittels QR-Zerlegung.
Input:
`A` *(np.ndarray)* Matrix A
`b` *(np.ndarray)* Vektor b
Returns:
(np.ndarray) Lösungsvektor des LGS

1.0.7 `norm_of_residuum(A, b)`

Diese Methode ermittelt die euklidische Norm des Residuums $Ax-b$ der Lösung des LGS $Ax = b$.
Input:
`A` *(np.ndarray)* Matrix A
`b` *(np.ndarray)* Vektor b
Returns:
(float) euklidische Norm des Residuums.

1.0.8 `get_cond(A)`

Diese Methode ermittelt die Kondition der Matrix A unter der Spektralnorm.
Input:
`A` *(np.ndarray)* Matrix A
Returns:
(float) Kondition von A

1.0.9 `get_cond_transposed(A)`

Diese Methode ermittelt die Kondition der Matrix $A^T A$ unter der Spektralnorm.
Input:
`A` *(np.ndarray)* Matrix A
Returns:
(float) Kondition von $A^T A$

1.0.10 `plot_result(data_list, labels)`

Diese Methode gibt die Lösungen des Problems mittels einfacher linearen Regression der Form $p_0 = p_1 x_0 + x_1$ unter verschiedenen Modifikationen der Eingabedaten sowohl als Plot, als auch über die Konsole aus. Die ermittelten Funktionen werden ausgegeben und geplottet. Zu jedem LGS werden zusätzlich die Norm des Residuums und die Kondition von A und $A^T A$ ausgegeben.
Input:
`data_list` *(list of 2d-arrays)* Liste von verschiedenen modifizierten Versionen der Eingabedaten.

`labels` *(list of strings)* Liste von Beschreibungen der verschiedenen Modifikationen. Muss die selbe Länge haben, wie data_list.

Returns:

(None) –

1.0.11 `plot_result_p2(data)`

Diese Methode gibt die Lösungen des Problems mittels einfacher linearen Regression der Form $p_0 = p_1 x_0 + x_1$ und der Form $p_0 = p_2 x_0 + x_1$ sowohl als Plot, als auch über die Konsole aus. Die ermittelten Funktionen werden ausgegeben und geplottet. Zu jedem LGS werden zusätzlich die Norm des Residuums und die Kondition von A und $A^T A$ ausgegeben.

Input:

`data` *(2d-array)* Die Eingabedaten als 2 dimensionaler Array.

Returns:

(None) –

1.0.12 `plot_result_multilinear(data)`

Diese Methode gibt die Lösung des Problems mittels linearer Mehrfachregression der Form $p_0 = p_1 x_0 + p_2 x_1 + x_2$ sowohl als Plot, als auch über die Konsole aus. Die ermittelte Funktion wird ausgegeben und geplottet. Zu dem LGS werden zusätzlich die Norm des Residuums und die Kondition von A und $A^T A$ ausgegeben.

Input:

`data` *(2d-array)* Die Eingabedaten als 2 dimensionaler Array.

`labels` *(list of strings)* Liste von Beschreibungen der verschiedenen Modifikationen. Muss die selbe Länge haben, wie data_list.

Returns:

(None) –

2 Bedienung des Hauptprogrammes

In `least_squares.py` ist eine `main()` Funktion implementiert. Das Programm wird aufgerufen mittels `python3 least_squares.py eingabedatei.txt`. Die Datei enthält für jede Messung eine Zeile und die Messwerte einer Messung sind in der Zeile durch Kommata getrennt. Unsere Beispieldatei `pegel.txt` enthält zwölf Messungen mit jeweils drei Messwerten und hat die Form:

```
172, 93, 120
309, 193, 258
302, 187, 255
283, 174, 238
443, 291, 317
298, 184, 246
319, 205, 265
```

```
419, 260, 304
361, 212, 292
267, 169, 242
337, 216, 272
230, 144, 191
```

Daraufhin wird ein Plot erstellt der einfachen linearen Regression mit genau diesen Daten, einer Regression, die nur die ersten sechs Zeilen nutzt und einer, in die eine falsche Messung eingefügt wurde. Die geplotteten Funktionen werden zusätzlich ausgegeben. Zu jedem LGS werden zusätzlich die Norm des Residuums und die Kondition von A und $A^T A$ ausgegeben.
Daraufhin werden die Lösungen des Problems mittels einfacher linearen Regression unter der Nutzung der zweiten und im Vergleich auch unter der Nutztung der dritten Spalte der Daten sowohl als Plot, als auch über die Konsole ausgegeben.
Als letztes wird die Lösung des Problems mittels linearer Mehrfachregression der Form $p_0 = p_1 x_0 + p_2 x_1 + x_2$ sowohl als Plot, als auch über die Konsole ausgegeben.