Úloha B

Prevedenie úlohy LP do tvaru pre `scipy.optimize.linprog`

Metóda `linprog` z modulu `scipy.optimize` vyžaduje nasledujúci tvar úlohy LP:

$$
\begin{aligned}
&\text{min } c^T x \\
&\quad A_{ub} x \le b_{ub} \\
&\quad A_{eq} x \le b_{eq} \\
&\quad l \le x \le u \qquad\qquad l, u \in (\mathbb{R} \cup \{\texttt{None}\})^n
\end{aligned}
$$

Hodnota `None` vo vektoroch $l,\ u$ značí neohraničenosť v danom smere. Upravme teda úlohy vyjadrené vyššie do predpísaného tvaru.

Pre L^1 regresiu:

$$
\begin{aligned}
&\text{min } \left(\begin{array}{c|c} \mathbf{0}_{k+1}^T & \mathbf{1}_n^T \end{array} \right) \begin{pmatrix} \beta \\ \hline t \end{pmatrix} \\
&\left(\begin{array}{c|c} -\mathbf{A} & -\mathbb{I}_n \\ \hline \mathbf{A} & -\mathbb{I}_n \end{array} \right) \begin{pmatrix} \beta \\ \hline t \end{pmatrix} \le \begin{pmatrix} -y \\ \hline y \end{pmatrix} \\
&\texttt{None} \le \beta_i \le \texttt{None} \qquad\qquad i = 0, 1, \dots, k \\
&0 \le t_j \le \texttt{None} \qquad\qquad j = 1, \dots, n
\end{aligned}
$$

Pre L^∞ regresiu:

$$
\begin{aligned}
&\text{min } \left(\begin{array}{c|c} \mathbf{0}_{k+1}^T & 1 \end{array} \right) \begin{pmatrix} \beta \\ \hline \gamma \end{pmatrix} \\
&\left(\begin{array}{c|c} -\mathbf{A} & -\mathbf{1}_n \\ \hline \mathbf{A} & -\mathbf{1}_n \end{array} \right) \begin{pmatrix} \beta \\ \hline \gamma \end{pmatrix} \le \begin{pmatrix} -y \\ \hline y \end{pmatrix} \\
&\texttt{None} \le \beta_i \le \texttt{None} \qquad\qquad i = 0, 1, \dots, k \\
&0 \le \gamma \le \texttt{None}
\end{aligned}
$$

Úlohy v zdrojovom kóde sú implementované práve v tomto tvare.

Implementovanie regresných LP úloh

Na implementáciu formulovaných LP úloh využívame tri knžnice:

- `numpy` - tvorenie matíc a vektorov, načítanie dát
- `scipy.optimize` - implementovaný LP solver
- `matplotlib.pyplot` na vykresľovanie grafov.

Dáta relevantné pre túto úlohu sú uložené v súbore `data/A04plotregres.npz`. Jedná sa o 16 bodov v $\mathbb{R}^2$, kde prvá súradnica reprezentuje nezávislú premennú (vektor týchto súradníc označíme `x`) a druhá závislú premennú (označíme `y`).

Vytvorme si potrebné štruktúry pre využitie metódy `scipy.optimize.linprog` pre LP formuláciu s L^1 normou:

```
c = np.array([0,0] + [1] * len(x)) #objective function vector, two zeros
                                  stand for betas
A = np.matrix([[1] * len(x), x]).transpose()
I = np.identity(len(x)) # Identity matrix

A_ub = np.block([[-A,-I], [A,-I]]) #creating a block matrix
b_ub = np.concatenate([-y, y]) #right side vector
bounds = [(None, None), (None, None)] + [(0, None) for _ in range(len(x)
                                    )] # bounds for variables
```

Pomocou solvera získame vektor optimálnych β koeficientov:

$$\beta_0^{(1)} \approx -9.8378, \ \beta_1^{(1)} \approx 2.1297$$

Podobne implementujeme L^∞ formuláciu:

```
c_inf = np.array([0,0,1]) #objective function vector
A_inf = np.matrix([[1] * len(x), x]).transpose()
i_inf = np.array([[1] * len(x)]).transpose() # vector of ones

A_ub_inf = np.block([[-A_inf, -i_inf], [A_inf, -i_inf]]) # creating a
                                    block matrix
b_ub_inf = np.concatenate([-y, y])#right side vector
bounds = [(None, None), (None, None), (0, None)]
```

Znovu, pomocou solvera získame vektor optimálnych β koeficientov:

$$\beta_0^{(\infty)} \approx 15.4545, \ \beta_1^{(\infty)} \approx 1.7045$$

Pomocou získaných β koeficientov vykreslíme regresné priamky spolu s pôvodnými dátami.