

Úloha E

Spracovanie všeobecnej triedy pre L^1 a L^∞ regresiu

Vypracovali sme modul `Model` pre počítanie L^1 a L^∞ regresie z ľubovoľných číselných dát, ktorý využíva LP formulácie popísané vyššie. Konkrétne `L1Model` využíva formuláciu na minimalizovanie L^1 normy a `LInfModel` minimalizuje L^∞ normu. Príklad použitia tohto modelu sa nachádza v `model_demonstration.ipynb`. Následne opíšeme jednotlivé metódy jednotlivých modelov.

`Model(dependent_vect, independent_vect)`

Konstruktory triedy, spoločný pre oba modely, vytvorí inštanciu, ktorá si drží dáta a vie na nich vykonávať operácie popísané nižšie.

Argumenty:

- `dependent_vect`: `np.array` - vektor závislých premenných
- `independent_vect`: `np.array` - matica, ktorej stĺpce sú vektory nezávislých premenných

`Model.solve()`

Metóda, ktorá vyrieši regresnú LP úlohu na daných dátach. `L1Model.solve()` rieši minimalizáciu L^1 normy a `LInfModel.solve()`, rieši minimalizáciu L^∞ normy.

Vracia:

- `np.array` - vektor optimálnych β premenných

Po zavolaní tejto metódy si inštancia uloží vektor optimálnych β premenných do atribútu `self._beta`, potrebné pre metódy popísané nižšie.

`Model.r2()`

Vypočíta R^2 koeficient pre vypočítaný vektor β podľa predpisu:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad \hat{y} = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k, \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

Vracia:

- `float` - výsledný R^2 koeficient

`Model.visualize()`

Ak je počet nezávislých premenných 1 alebo 2, táto metóda vykreslí graf dát spolu s vypočítanou regresnou priamkou, resp. rovinou.

Vracia:

- `bool` - úspešnosť vizualizácie, kde `False` označuje, že nezávislých premenných je viac ako 2, čiže nie je možné vykresliť graf