OLS

Yury Hayeu
ČVUT–FIT
hayeuyur@fit.cvut.cz

27. dubna 2020

1 Úvod

Úkolem semestrální práce byla implementace modulu, který implementoval metodu nejmenších čtverců s použitím python balíčků **Numpy** a **Pandas**. Po implementaci by měla být provedena explorační analýza dat a predikce výstupních hodnot pro dataset obsahující informaci o bydlení v Bostonu [2].

Lineární regrese je metoda, účelem které je hledaní nejlepší aproximace koeficientů polynomu:

$$y = \alpha_0 + \alpha_1 * x_1 + ... + \alpha_n * x_n \tag{1}$$

Koeficientα_0 se nazývá bias. Koeficienty $\alpha_1, ..., \alpha_n$ jsou váhy. Vstupní data je matice velikosti (m, n).

2 Výpočet biasu

Implementace modulu redukuje výpočet biasu pomocí následující úvahy:

$$\begin{aligned} y = \alpha_0 + \alpha_1 * x_1 + ... + \alpha_n * x_n = \\ \alpha_0 * 1 + \alpha_1 * x_1 + ... + \alpha_n * x_n = \\ \alpha_0 * x_0 + \alpha_1 * x_1 + ... + \alpha_n * x_n \end{aligned}$$

Modul přidává vstupní matici zleva sloupec jedniček na začátku aproximace a predikce hodnot.

3 Metody

Modul implementuje následující metody:

1. Statistická lineární regrese je základní metoda, která spočítá váhy podle vzorce[3]:

$$\alpha = (X^T X)^{-1} X^T Y \tag{2}$$

X je vstupní data.
Y je cíl aproximace.
α je vektor vah.
X^T je transponovaná matice.
$(X^T X)^{-1}$ je inverzní matice.

2. Gradient descent(GD) je metoda, která iterativně spočítá váhy. Základní jednotkou GD je cost funkce, která spočítá chybu aproximace. Cílem semestrální práce bylo použití metody nejmenších čtverců, proto cost funkce je následující:

$$\epsilon = mean((X^{'} - Y)^2) \tag{3}$$

$X^{'}$ je predikovaná hodnota.
n - je počet řádků matic X a Y.

Potřebujeme zavést následující definice:

a) **Počet iteraci** je počet, kolikrát bude aproximovat funkce.

b) **Tolerance** je prahová hodnota, která ukazuje změnu v cost funkci. Pokud během dvě iterace se změna chyby bude menší než tolerance, pak se aproximace funkce zastaví.

c) **Learning step** (budeme značit β) je hodnota, která určuje rychlost aproximace.

Gradient funkce určuje směr (kladný nebo záporný) a změnu koeficientů aproximace funkce. Jedná se o derivaci výstupní funkce:

$$gradient = -2/n * X^T * (X^{'} - Y) \tag{4}$$

gradient je vektor koeficientů
$X^{'}$ je predikovaná hodnota.
Y je cíl aproximace.
n je počet záznamů.

Aproximace funkce je následující :

$$noveKoeficienty += learningStep * gradient \tag{5}$$

Algoritmus Gradient Descent:

a) Spočítej predikovanou hodnotu pro vstupní matici X.

b) Spočítej gradient podle vzorce (4).

c) Spočítej váhy podle vzorce (5).

d) Spočítej chybu aproximaci, pokud změna v chybě aproximace je menší než tolerance, ukonči aproximaci.

3. Stochastic gradient descent je modifikace GD pro větší rozsah dat. Modul implementuje základní verzí SGD, která aproximuje hodnoty funkce podle jednoho záznamů. [1]

4. Minibatch gradient descent je modifikace SGD, která aproximuje hodnoty funkce podle balíčku záznamů

4 Výsledky

Výsledkem semestrální práce je balíček OLS a jupyter notebook **EDA.ipynb**, kde se provádí předzpracovaní dat a vizualizace lineární regresi pro sloupce s vysokou korelaci.

5 Závěr

Balíček OLS je vhodný pro hledaní vah pří malém počtu záznamů, pří větším počtu záznamů by operace maticového násobení mohly používat zdroje GPU anebo nějaké jiné externí zdroje. Následujícím vylepšením by byla implementace různých druhů cost funkce a SGD metod.

Reference

[1] Stochastic gradient descent. online. [cit. 2020–27–04] `https://en.wikipedia.org/wiki/Stochastic_gradient_descent`.

[2] D. Harrison and D.L. Rubinfeld. Boston housing data. online, 1993. [cit. 2020–27–04] `https://archive.ics.uci.edu/ml/machine-learning-databases/housing/`.

[3] Purdue University. Statistics 512: Applied linear models. online. [cit. 2020–27–04] `https://www.stat.purdue.edu/~boli/stat512/lectures/topic3.pdf`.