Šesterokut šalabahteri

Donik Vršnak, Filip Novkoski

April 2021

1 Linearna regresija

Cilj linearne regresije je pronaći pravac koji prolazi kroz zadani skup podataka najbolje što može. Primjerice, mjerimo masu vode u nekoj boci u ovisnosti o volumenu vode koji smo ulili. Kao što znamo, $m = \rho V$. Ako nacrtamo tu ovisnost, to je pravac koji prolazi krozz 0 s nagibom ρ. No, naša vaga ima neku preciznost, kao i mjerenje volumena koje vršimo recimo menzurom, zbog meniskusa i ljudske greške, nećemo savršeno odrediti volumen. Stoga, naša mjerenja odstupaju od tog idelanog pravca. Prema tome, mi kroz naša mjerenja želimo provući pravac koji najbolje opisuje ta mjerenja, i usporediti s teorijskom vrijednosti koja nam je dobro poznata.

1.1 Jednadžbe i vektori

Prije nego što zavirimo u linrearnu regresiju i metodu najmanjih kvadrata, pogledajmo sustave jednadžbi. Krenimo sa sustavom od dvije jednadžbe, i dvije nepoznanice w_1 i w_2:

$$y_1 = w_1 x_1 + w_2 \tag{1}$$

$$y_2 = w_1 x_2 + w_2 \tag{2}$$

Ovdje su nam x_1,x_2,y_1,y_2 poznati realni brojevi, te mi želimo pronaći w_1 i w_2. Premetanjem jednadžbi to možemo lako učiniti. Grafički gledano, imamo dvije točke u ravnini (x_1, y_1) i (x_2, y_2) kroz koje želimo provući pravac. Taj pravac će imati nagib w_1 i odsječak na osi w_2. Npr. možemo uzeti točke $(2, 4)$ i $(5, 7)$. Tada su $w_1 = 1$ i $w_2 = 2$. No, što ako imamo slučaj sa 100 točaka, kako tada postupiti? Iz tog razloga, prelazimo na vektore, jer oni su upravo objekti s kojima lako možemo baratati i koji istovremeno mogu sadržavati veliki broj podataka. Stoga, pišemo prvo

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} w_1 x_1 + w_2 \\ w_1 x_2 + w_2 \end{pmatrix} \tag{3}$$

gdje smo u biti samo stavili zagrade oko gornjih jednadžbi. Kako dalje pojednostaviti, a da dobijemo neki jednostaviji oblik, npr. s desne strane da imamo još jedan stupac tj. vektor stupac. Pa, vidimo da s desne strane se u oba reda pojavljuju w_1 i w_2 u potpuno istom uzorku. Možemo to pisati kao

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} w_1 \cdot x_1 + w_2 \cdot 1 \\ w_1 \cdot x_2 + w_2 \cdot 1 \end{pmatrix} \tag{4}$$

Vidimo da w_1 uvijek množi jedan od x-eva, a w_2 uvijek množi 1. Iz toga "izlučujemo" (ovo nije trivijalna radnja, i ovdje ju uvodimo na neformalni način te nije bitno da znate sve detalje iza nje)

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \tag{5}$$

S desne strane imamo sad jednu tablicu, odnosno matricu, koja djeluje na jedan stupac, odnosno vektor. I to djelovanje odgovara upravo onom množenju koje je prikazano u prethodnoj jednadžbi. Množenje tablice i stupca daje stupac. I to na način da uzmemo prvi redak matrice, te prvi član retka pomnožimo s prvim članom stupca na koji djelujemo, tom iznosu dodajemo umnožak drugog člana retka i drugog člana stupca, pa dodajemo umnožak trećeg s trećim itd. Ta suma tada postaje prvi član našeg novog stupca. To ponavljamo s drugim retkom tablice, s istim djelovanjem na cijeli stupac.

Stupac tj. vektor s y označavamo kao $\boldsymbol{y}$, stupac s $w$ovima označavamo kao $\boldsymbol{w}$, a tablicu koja sadrži $x$eve označavamo kao X. Zašto stupce zovemo vektorima? Pa uzmimo recimo vektor u 2D. On ima neki smjer i iznos. Recimo da početak vektora stavimo u ishodište $(0, 0)$. Vrh strelice se tada nalazi u nekoj točki, recimo (y_1, y_2). Stoga, možemo jednoznačno poistovjetiti koordinate s vektorima, i suprotno, te naš stupac s y vrijednostima prozvati vektorom u 2D ravnini, kojem znamo iznos, smjer i orijentaciju.

Našu jednadžbu sad možemo pisati kao

$$\boldsymbol{y} = X\boldsymbol{w} \tag{6}$$

Ono što tražimo su naše nepoznanice w_1 i w_2. Kad bi se radilo o običnim brojevima umjesto o vektorima i matricama, jednostavno bismo podijelili obje strane s X. Iako, to ovdje ne možemo napraviti, postoji nešto što se zove inverz matrice. Matrici X možemo pridružiti njen inverz X^{-1}, te kada to dvoje pomnožimo nam daje jediničnu matricu (matrica koja ima 1 po dijagonali, od lijevo gore prema desno dolje, i 0 svugdje drugdje). Kada jedinična matrica pomnoži vektor, dobijemo isti taj vektor (ovo možete i sami provjeriti koristeći pravilo za množenje gore). Pa, ono što ćemo učiniti je da ćemo jednostavno pomnožiti obje strane jednadžbe s lijeve strane s tom inverznom matricom i dobiti

$$X^{-1}\boldsymbol{y} = \boldsymbol{w} \tag{7}$$

I eto, našli smo naše nepoznanice. Inverz matrice ćete rijetko računati ručno, za $2D$ postoji formula koja ju točno odreuje za sve matrice. Kao što smo vidjeli na predavanju, u Pythonu je račun inverza implementiran te je vrlo lako ju odrediti. No, ovaj račun je ispravan jedino kad naše točke (x_1, y_1) i (x_2, y_2) leže na istom pravcu. Recimo sad da imamo tri, četiri, ili sto točaka, što tada. Pa za početak, naši vektori postanju višlji, i naša matrica postaje i šira i višlja. S obzirom da točke ne moraju ležati na istom pravcu, mi tražimo pravac koji će ih najbolje opisivati, tj. imati će najmanje odstupanje od našeg skupa točaka. Tu uvodimo pseudoinverz matrice, koji je poput inverza, ali nije egzaktan, nego nam daje najbolji fit. U tom slučaju je

$$\boldsymbol{w} = X^{+}\boldsymbol{y} \tag{8}$$

Ono što pseudoinverz radi, je da nam daje rješenje odreeneog optimizacijskog problema. U tom problemu, želimo minimalizirati pogrešku. U biti, to je ideja iza modeliranja, da naš model ima što manju grešku, i da je funkcija greške što manja.

1.2 Model

$$\begin{aligned} \boldsymbol{y} &= X\boldsymbol{w} \\ \boldsymbol{y_i} &= \boldsymbol{x_i} \cdot \boldsymbol{w} \end{aligned} \tag{9}$$

- $\boldsymbol{y} = [y_1, y_2, \ldots, y_m]^T$ je vektor stupac ciljanih vrijednosti, broj redaka jednak je broju primjera za učenje
- X je matrica dizanja, proširena s jedinicama na početku, broj redaka jednak je broju primjera za učenje, a broj stupaca jednak je broju dimenzija podataka + 1 zbog proširenja s jedinicom ($n+1$ ako je $x \in R^n$, u primjeru na Slici 1 $n = 1$,a broj stupaca je 2)
- $w = [w_0, w_1 \ldots, w_n]^T$ je vektor stupac težina s $n + 1$ redaka
- Za pojedini primjer y_i izlaz modela računa se kao skalarni produkt $(\cdot)$ vektora primjera (i-ti redak u matrici dizajna) x_i i vektora težina w.

1.3 Funkcija gubitka

$$\begin{aligned} L &= \sum_{i=0}^{N} e_i \\ e_i &= (y_{mjereno,i} - \boldsymbol{x_i} \cdot \boldsymbol{w})^2 \end{aligned} \tag{10}$$

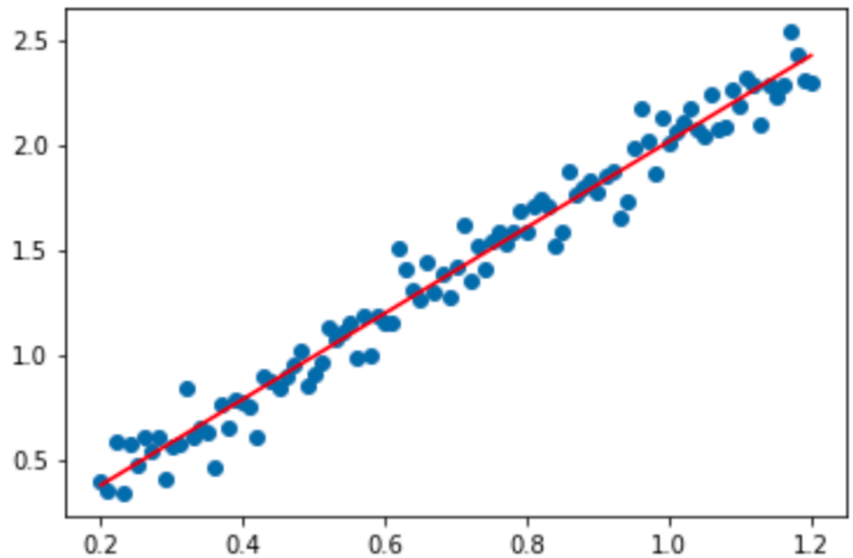

Figure 1: Slika koja prikazuje 1-d podatke X i mjerene vrijednosti $\boldsymbol{y}$. Crvena linija prikazuje pravac čiji su koeficijenti izračunati korištenjem linearne regresije.

- Funkcija gubitka L (u nekoj litraturi E) računa odstupanje modela od podataka
- Za linearnu regresiju koristimo kvadratnu funkciju pogreške e

1.4 Optimizacijski postupak

$$\begin{aligned} \boldsymbol{w}^* &= argmin_{\boldsymbol{w}} L \\ \boldsymbol{w}^* &= X^+ \boldsymbol{y} \\ X^+ &= (X^T X)^{-1} X^T \end{aligned} \tag{11}$$

- Optimizacijski postupak služi da nademo težine koje nam daju najmanju pogrešku L
- Rješenje se dobiva u obliku pseudo inverza matrice X pomnožene s desna s ciljanim vrijednostima y

2 Nelinearna regresija

- Model ostaje linearan: $\boldsymbol{y} = \Phi \boldsymbol{w}$, mjenjamo podatke $X \rightarrow \Phi$

- Uvodimo bazne funkcije $\Phi(x) = \{\phi_0(x), \phi_1(x), \ldots, \phi_m(x)\}$koje transformiraju podatke iz dimenzije n u dimenziju n (u slučaju kvadratne funkcije sa Slike 2, $n = 1m = 3$)

- Najčešće bazne funkcije koje se korsite su polinomijalne,
npr. za $\boldsymbol{x} = [x_1 \ x_2]$ polinomijalne bazne funkcije drugog reda su

$$
\begin{gathered}
\Phi([x_1 \ x_2]) = \{\phi_0([x_1 \ x_2]) = 1, \\
\phi_1([x_1 \ x_2]) = x_1, \ \phi_2([x_1 \ x_2]) = x_2, \\
\phi_1([x_1 \ x_2]) = x_1^2, \ \phi_2([x_1 \ x_2]) = x_2^2, \ \phi_1([x_1 \ x_2]) = x_1 x_2\}
\end{gathered} \tag{12}
$$

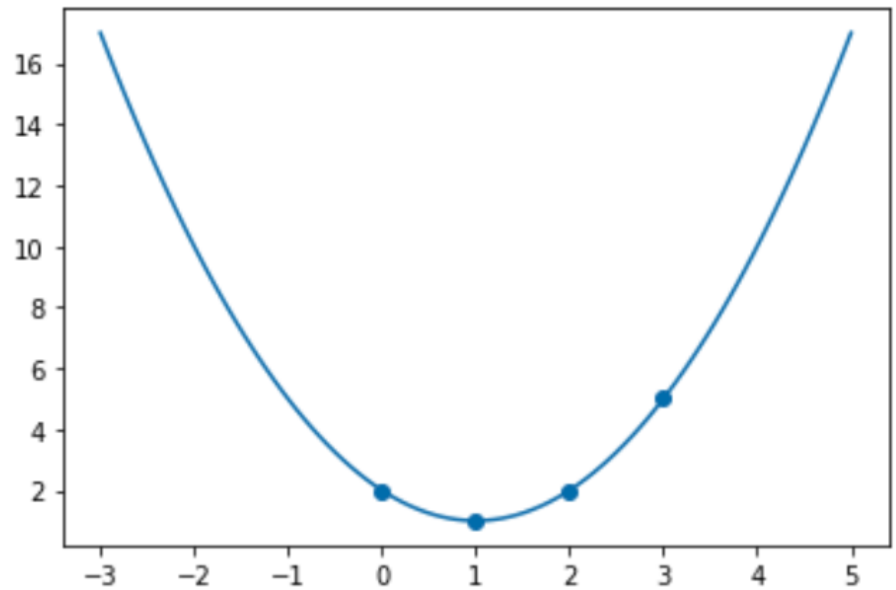

Figure 2: Slika prikazuje primjer nelinearne regresije na 1-d podacima. Korištene bazne funkcije za dobivanje modela parabole su
$\Phi(x) = \{1, x, x^2\}$