

Linearna regresija

Predstavljajmo si, izvajamo fizikalni poskus. Imamo vzmet, na kateri merimo raztezek v odvisnosti od sile, s katero napenjamo vzmet (na vzmet obešamo uteži z znano težo in merimo raztezek).

F [N]	1	2	3	4	5
x [cm]	10	20	35	55	80

Velja Hookov zakon, $F = kx$, torej sta x in F med seboj linearno odvisna. Poskušali bomo najti premico, ki se bo najboljše prilegala danim točkam v ravnini.

Če imamo 2 točki $T_1(x_1, y_1), T_2(x_2, y_2)$, lahko brez težav najdemo premico, ki gre točno skozi niju. Če pa je točk več, ni nujno, da obstaja premica, ki gre skozi vse točke. Iščemo taka a in b , da se bo premica kar najboljše prilegala danim točkam. Če gre premica skozi neko točko $T_i(x_i, y_i)$, potem velja:

$$0 = ax_i + b - y_i$$

Ker pa naša premica ne poteka direktno skozi vse točke pride do napake, ki jo bomo v točki T_i označili z ε_i .

$$\varepsilon_i = ax_i + b - y_i$$

Napaka je lahko pozitivna ali negativna, odvisno ali točka leži pod ali nad premico. Želimo zmanjšati velikost vseh napak, ne glede na to, ali so pozitivne ali negativne. (Lahko bi preprosto sešteli absolutne vrednosti napak, ampak pozneje funkcije ne bi mogli odvajati.) Zato seštejemo kvadrate vseh napak.

$$\varepsilon = \sum_{i=1}^N \varepsilon_i^2$$

¹Podatki pridobljeni iz: <http://www.clemson.edu/ces/phoenix/labs/124/shm/>

$$\varepsilon = \sum_{i=1}^N (ax_i + b - y_i)^2$$

Naš cilj je, da minimiziramo to napako. Napako bomo minimizirali s pomočjo gradientnega spusta.

Gradientni spust

Z gradientnim spustom iščemo minimum ali pa maksimum neke funkcije, to pomeni kje ima funkcija največjo oziroma najmanjšo vrednost. Gradientni spust je zelo primitivna metoda kjer se preprosto premikamo v smer kamor funkcija narašča oziroma pada.

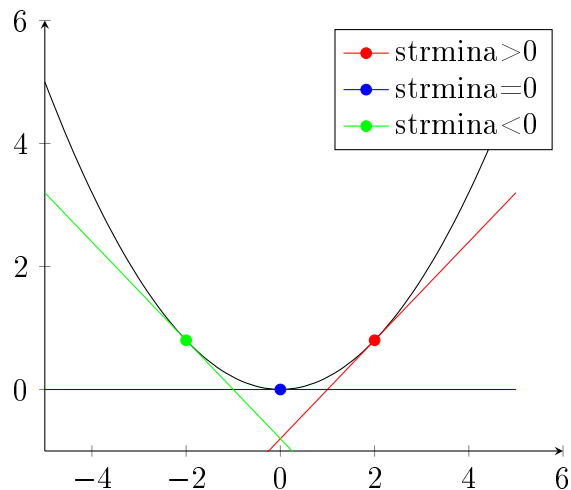
To si lahko predstavljamo s pomočjo slepega človeka, ki želi priti na vrh Golovca. Ko bo naš človek, poimenujmo ga Lojze, na neki točki hriba, ne bo videl v katero smer mora hoditi (ker je slep), čutil pa bo kako strm je hrib. Na začetku, ko je hrib bolj strm, bo Lojze naredil večji korak, ko pa se bo približeval vrhu in bo hrib postajal manj strm, pa bo delal manjše korake, ker se lahko drugače zgodi, da bo prestopil vrh in se znašel na drugi strani Golovca.

Z bolj matematičnimi besedami: gradient nam pove smer naraščanja funkcije. Preden nadaljujemo z gradientom, moramo definirati kaj nam predstavlja strmino funkcije (naraščanje in padanje). To bomo naredili z odvodi.

Odvod

Odvod neke funkcije $f(x)$ je strmina te funkcije v točki x . Odvod je pozitiven če funkcija narašča in negativen če funkcija pada. Večji kot je odvod, bolj strma je funkcija v dani točki. Odvod funkcije $f(x)$, ki ga označimo z $f'(x)$, nam torej pove, kakšen je koeficient tangente na graf funkcije $f(x)$ v točki x .

Slika 1: Odvod



Delni odvod

Funkcija lahko vzame več kot en parameter, kar pomeni da je odvisna od 2eh spremenljivk. Če bi tako funkcijo želeli narisati bi potrebovali 3D graf.

Slika 2: 3D funkcija

Tako funkcijo moramo odvajati po vsaki spremenljivki posebej, kar pomeni da potrebujemo funkcijo $f(x, y)$ odvajati po x in po y . S tem dobimo dva odvoda. Odvod po x nam pove kako funkcija narašča oziroma pada če spreminjamo x parameter, odvod po y pa nam seveda pove kako funkcija narašča oziroma pada po y parametru. Takemu odvodu rečemo delni odvod,

zapišemo pa ga kot $\frac{\partial f(x,y)}{\partial x}$ za delni odvod po x in $\frac{\partial f(x,y)}{\partial y}$ za delni odvod po y .

Gradient

Gradient neke funkcije je vektor, ki kaže v smer naraščanja te funkcije. Gradient funkcije $f(x)$ označimo z $\nabla f(x)$. Pri funkciji z enim parametrom ($f(x)$), ima gradient te funkcije samo eno komponento, to je odvod funkcije $f(x)$. To zapišemo kot:

$$\nabla f(x) = (f'(x))$$

Če je funkcija odvisna od dveh spremenljivk, je vsaka komponenta gradienta en delni odvod. To pomeni da nam prva komponenta gradienta pove kako se funkcija spreminja po x osi, druga komponenta gradienta pa nam pove kako se funkcija spreminja po y osi. Gradient funkcije $f(x, y)$ lahko torej zapišemo kot:

$$\nabla f(x, y) = \left(\frac{\partial f(x,y)}{\partial x}(x) \quad \frac{\partial f(x,y)}{\partial y}(y) \right)$$

V zgornjih dveh primerih smo zapisali gradient funkcije v določeni točki x oziroma (x, y) . Če bi drugi primer zapisali v splošnem bi to zgledalo kot:

$$\nabla f = \left(\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right)$$

To pomeni da lahko funkcijo odvisno od n parametrov, zapišemo kot:

$$\nabla f = \left(\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \frac{\partial f}{\partial x_2} \quad \cdots \quad \frac{\partial f}{\partial x_n} \right)$$

Prilagajanje premice

Sedaj ko razumemo kaj je gradient, se lahko vrnemo na naš začetni problem, to je najti tako premico, ki se najbolj prilega N točkam. Na tem primeru bomo razložili tudi gradientni spust.

Najprej se spomnimo kaj sploh želimo narediti. Naša funkcija je $\varepsilon_i = ax_i + b - y_i$, kjer ε_i predstavlja napako te funkcije v točki $T_i(x_i, y_i)$. Skupno napako smo definirali kot: $\varepsilon = \sum_{i=1}^N \varepsilon_i^2$. Sedaj želimo da je naša skupna napaka ε čim manjša. Da bomo to dosegli bomo najprej izračunali gradient naše napake.

Celo življenje smo za označevanje parametrov funkcij uporabljali $f(x)$, kar pomeni, da je funkcija f odvisna od spremenljivke oziroma parametra x . Seveda pa nismo omejeni samo na x . Naša funkcija je lahko odvisna od parametra a , kar pomeni da bi jo zapisali kot $f(a)$. Prav tako lahko namesto $f(x, y)$ napišemo $f(a, b)$. Naša napaka ε ni odvisna od spremenljivk x in y , ampak je odvisna od a in b . Spomnimo se, da imamo x in y že podan, saj je to točka, ki ji želimo določiti premico. Pri določanju premice pa spreminjamo a in b . Gradient napake je torej:

$$\nabla \varepsilon = \begin{pmatrix} \frac{\partial \varepsilon}{\partial a} & \frac{\partial \varepsilon}{\partial b} \end{pmatrix}$$

Gradient nam torej pove strmino in smer v katero funkcija narašča. Mi pa se potrebujemo premakniti v točno nasprotno smer, ker iščemo čim manjšo napako. To da se premaknemo v neko smer, pomeni da spremenimo naš a in b tako, da bo napaka manjša. Parameter a zmanjšamo za vrednost, prve komponenta našega gradienta, ker je ta komponenta delni odvod napake po a , kar pomeni da nam pove, kako se napaka spreminja, če spreminjamo a . Parameter b pa seveda zmanjšamo za vrednost druge komponente našega gradienta.

Pa razpišimo te spremembe. Spremembo a lahko napišemo kot:

$$\Delta a = -(\nabla \varepsilon)_1 \cdot \lambda$$

Minus uporabimo zato, ker se premikamo v smer kamor funkcija pada, λ pa je konstanta, ki nam predstavlja kako hitro spreminjamo naš parameter. λ določimo mi. O točnem pomenu λ se bomo pogovorili nekoliko kasneje.

Spremembo b pa lahko zapišemo kot:

$$\Delta b = -(\nabla \varepsilon)_2 \cdot \lambda$$

Sedaj rabimo izračunati delne odvode naše funkcije, da jih lahko vstavimo v enačbe za gradientni spust.

$$\frac{\partial \varepsilon}{\partial a} = \sum_{i=1}^N 2 \frac{\partial(ax_i + b - y_i)}{\partial a} = 2 \sum_{i=1}^N (ax_i + b - y_i)x_i$$

$$\frac{\partial \varepsilon}{\partial b} = \sum_{i=1}^N 2 \frac{\partial(ax_i + b - y_i)}{\partial b} = 2 \sum_{i=1}^N (ax_i + b - y_i) \cdot 1 = 2 \sum_{i=1}^N (ax_i + b - y_i)$$

To pomeni da lahko za dani a in b izračunamo za koliko moramo spremeniti ta dva parametra, da bo napaka manjša. Če to razpišemo:

$$\Delta a = - \sum_{i=1}^N (ax_i + b - y_i)x_i \cdot \lambda$$

$$\Delta b = - \sum_{i=1}^N (ax_i + b - y_i) \cdot \lambda$$

Pri obeh enačbah smo se lahko znebili 2, ker lahko povečamo našo konstanto λ in dobimo enak rezultat.

Sedaj lahko uporabimo ta princip, da se začnemo približevati bližje želeni vrednosti a in b , da bo napaka čim manjša. To pomeni da je naš novi a :

$$a = a + \Delta a$$

$$b = b + \Delta b$$

Sedaj smo za en korak bližje našim želenim vrednostim. Ta postopek moramo ponoviti še kaj nekajkrat, zato izkoristimo računalnik. Večkrat kot ponovimo postopek, bolj natančen bo naš rezultat.

Natančnost našega rezultata je odvisna tudi od λ (velikost premika) in začetnih vrednost spremenljivk a in b . Z manjšo velikostjo premika bo rezultat

bolj natančen, vendar bomo morali postopek večkrat ponoviti. Ker s postopkom iščemo samo lokalne minimume, nam začetna vrednost spremenljivk določi kateri minimum bomo našli.

Delni odvodi

Funkcija je odvisna od večih spremenljivk, zato jo moramo odvajati za vsako spremenljivko posebej. Ker nam odvod ene spremenljivke pove kako se funkcija obnaša samo po tej spremenljivki, temu rečemo delni odvod.

$$\frac{\partial \varepsilon}{\partial a} = \sum_{i=1}^N 2 \frac{\partial(ax_i - b - y_i)}{\partial a} = 2 \sum_{i=1}^N (ax_i - b - y_i)x_i$$

$$\frac{\partial \varepsilon}{\partial b} = \sum_{i=1}^N 2 \frac{\partial(ax_i - b - y_i)}{\partial b} = 2 \sum_{i=1}^N (ax_i - b - y_i)(-1) = -2 \sum_{i=1}^N (ax_i - b - y_i)$$

Rešitev najdemo tako, da ugotovimo, pri katerih a in b sta odvoda enaka 0. To lahko rešimo z uporabo matrik, vendar tega ne zamo, zato se bomo tega lotili s primitivno metodo gradientnega spusta.

Linerana regresija elipse

Imamo N točk krožnice nekega planeta $T_1(x_1, y_1), T_2(x_2, y_2) \dots, T_n(x_n, y_n)$. Podobno kot pri prejšnjem primeru, bomo tudi tokrat iskali funkcijo, ki se točkam najbolj prilega. Tokrat bomo namesto premice iskali elipso, saj se elipsa seveda bolj prilega krožnici planeta kot pa premica.

Elipsa je stožnica, zato zapišemo splošno enačbo za stožnice:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

Tako kot prej se nam bo pojavila napaka, ki jo označimo z ε .

$$\varepsilon_i = Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F$$

Za razliko od premice, je oblika stožnice odvisna od šestih parametrov namesto dveh. To so: A, B, C, D, E in F . Zato bomo torej spreminjali teh šest parametrov.

Podobno kot pri premici najprej definiramo skupno napako kot vsoto kvadratov vseh napak:

$$\varepsilon = \sum_{i=1}^N \varepsilon_i$$

$$\varepsilon = \sum_{i=1}^N (Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F)^2$$

Enako kot pri premici moramo našo napako delno odvajati po spremenljivkah od katerih je naša napaka odvisna, da bomo znali te spremenljivke spreminjati pravilno. To pomeni da potrebujemo izračunati delni odvod napake po A, B, C, D, E in F . Delni odvodi za enačbo stožnic so:

$$\frac{\partial \varepsilon}{\partial A} = \sum_{i=1}^N 2(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F)(x_i^2)$$

$$\frac{\partial \varepsilon}{\partial B} = \sum_{i=1}^N 2(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F)(x_iy_i)$$

$$\frac{\partial \varepsilon}{\partial C} = \sum_{i=1}^N 2(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F)(y_i^2)$$

$$\frac{\partial \varepsilon}{\partial D} = \sum_{i=1}^N 2(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F)(x_i)$$

$$\frac{\partial \varepsilon}{\partial E} = \sum_{i=1}^N 2(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F)(y_i)$$

$$\frac{\partial \varepsilon}{\partial F} = \sum_{i=1}^N 2(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F)$$

Da najdemo najboljše parametre, pri katerih bo funkcija imela najmanjšo napako, bomo ponovno uporabili gradientni spust. Za razliko od gradientnega spusti pri eni premici, bomo tokrat spreminjali šest parametrov. Naš gradient je torej:

$$\nabla \varepsilon = \left(\frac{\partial \varepsilon}{\partial A} \quad \frac{\partial \varepsilon}{\partial B} \quad \frac{\partial \varepsilon}{\partial C} \quad \frac{\partial \varepsilon}{\partial D} \quad \frac{\partial \varepsilon}{\partial E} \quad \frac{\partial \varepsilon}{\partial F} \right)$$

To pomeni da so naše spremembe parametrov:

$$\Delta A = -(\nabla \varepsilon)_1 \cdot \lambda$$

$$\Delta B = -(\nabla \varepsilon)_2 \cdot \lambda$$

$$\Delta C = -(\nabla \varepsilon)_3 \cdot \lambda$$

$$\Delta D = -(\nabla \varepsilon)_4 \cdot \lambda$$

$$\Delta E = -(\nabla \varepsilon)_5 \cdot \lambda$$

$$\Delta F = -(\nabla \varepsilon)_6 \cdot \lambda$$

Iz česar sledi, da so naši novi parametri podobno kot pri premici:

$$A = A + \Delta A$$

$$B = B + \Delta B$$

$$C = C + \Delta C$$

$$D = D + \Delta D$$

$$E = E + \Delta E$$

$$F = F + \Delta F$$

Pravilnost našega rezultata je seveda odvisna od tega koliko ponovitev bomo naredili, kolikšna bo naša λ in kakšne začetne vrednosti A, B, C, D, E in F smo si izbrali.

Logistična regresija

Glavna razlika med logistično in linearno regresijo je, da pri linearni regresiji določamo zvezno spremenljivko (y je odvisen od x), medtem ko nam pri logistični regresiji model vrne kakšna je verjetnost, da vhodni podatki sodijo v določeno kategorijo. Zato se logistična regresija uporablja za klasifikacijo (npr. prepoznavanje števk idr.).

Dodatno

Ekliptični koordinatni sistem

Eden izmed koordinatnih sistemov za določanje lege nebesnih teles. Lokacijo določimo s tremi podatki. Ker so koordinate definirane glede na ravnino zemljine orbite, je z koordinata za Zemljo vedno enaka 0.

l longituda, ekliptična dolžina, od 0° do 360° .

b latituda, ekliptična širina od -90° do 90° .

r razdalja

V kartezične koordinate podatke pretvorimo po formulah:

$$x = r \cos b \cos l$$

$$y = r \cos b \sin l$$

$$z = r \sin b$$

Ker tiri vseh planetov ležijo v (skoraj) isti ravnini, bomo z koordinato zanemarili.