UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Finančna matematika – 1. stopnja

Mitja Mandić

Iterativne numerične metode v posplošenih linearnih modelih

Delo diplomskega seminarja

Mentor: izred. prof. dr. Jaka Smrekar

Ljubljana, 2021

Kazalo

Iterativne numerične metode v posplošenih linearnih modelih

Povzetek

V povzetku na kratko opiši vsebinske rezultate dela. Sem ne sodi razlaga organizacije dela – v katerem poglavju/razdelku je kaj, pač pa le opis vsebine.

Iterative numerical methods in generalized linear models

Abstract

Math. Subj. Class. (2010): navedi vsaj eno klasifikacijsko oznako – dostopne so na www.ams.org/mathscinet/msc/msc2010.html

Ključne besede: navedi nekaj ključnih pojmov, ki nastopajo v delu

Keywords: angleški prevod ključnih besed

1. Uvod

Posplošeni linearni modeli so ključen del statistične analize. Pomagajo nam bolje razumeti relacije med rezultati meritev in s temi izsledki predvideti trende v prihodnosti. Modeli morajo biti kar se da enostavni, ampak hkrati zagotavljati določeno natančnost. Vsa teorija nam pa v praksi ne koristi, če konkretnih številk ne znamo izračunati.

V dobi ogromne množice podatkov je računska učinkovitost ključni del obdelave. Tu nam pomagajo numerične metode.

V delu bom najprej predstavil posplošene linearne modele in najpriljubljenejše numerične metode, ki se uporabljajo za njihovo obdelavo. Teorijo bom osvetlil tudi s praktičnimi primeri.

2. Posplošeni linearni modeli

2.1. **Sestavni deli posplošenega linearnega modela.** Vsak posplošeni linearni model sestavljajo trije deli: *slučajni del* je slučajna spremenljivka Y in njena porazdelitev, *sistematični del* predstavlja relacijo med pojasnjevalnimi spremenljivkami, *povezovalna funkcija* pa transformira Y, da se ta bolje prilega podatkom.

2.1.1. *Slučajni del. Slučajni del* privzame porazdelitev slučajnega vektorja Y, pri čemer privzemamo tudi neodvisnost komponent. Porazdelitev Y privzemamo odvisno od podatkov; mnogokrat je „binarna“, torej ima dve možni vrednosti - „uspeh“ ali „neuspeh“. Splošneje je lahko izid tudi število uspehov v fiksnem številu poskusov. V takih primerih privzamemo binomsko porazdelitev. Y nam lahko meri tudi številske podatke, naprimer koliko zabav je obiskal študent v preteklem mesecu. Seveda pa lahko Y predstavlja tudi zvezne podatke, v tem primeru lahko privzamemo normalno porazdelitev (ali pa kakšno drugo zvezno porazdelitev).

2.1.2. *Sistematični del. Sistematična komponenta* posplošenega linearnega modela poda relacije med pojasnjevalnimi spremenljivkami $x_{i,j}$. Te nastopajo linearno, torej je sistematični del enak

$$\beta_0 + x_{i,1}\beta_1 + x_{i,2}\beta_2 + \ldots + x_{i,p}\beta_p$$

2.1.3. *Povezovalna funkcija.* Tretji del posplošenega linearnega modela je *povezovalna funkcija*, ta nam poda funkcijo $g(\cdot)$ med slučajno komponento in sistematičnim delom. Če označimo $\mu = E(Y)$, je

$$g(\mu) = \beta_0 + x_{i,1}\beta_1 + x_{i,2}\beta_2 + \ldots + x_{i,p}\beta_p$$

Najenostavnejša taka funkcija je kar identiteta, torej $g(\mu) = \mu$. Ta nam torej da linearno povezavo med pojasnjevalnimi spremenljivkami in pričakovano vrednostjo naših slučajnih spremenljivki. To je ena od oblik regresije za zvezne podatke. Mnogokrat pa linearna relacija ni primerna - fiksna sprememba pojasnjevalnih spremenljivk ima lahko večji vpliv, če je pričakovana vrednost bližje 0, kot če je bližje 1. Recimo, da je π verjetnost, da bo oseba kupila nov avto, ko je njen dohodek enak x. Sprememba v dohodku za 10.000€ ima manjši vpliv, če je dohodek 1.000.000€, kot če je 50.000€. Takrat je smiselno uporabiti kakšno drugo povezovalno funkcijo, ki dopušča tudi nelinearne kombinacije pojasnjevalnih spremenljivk. Naprimer, $g(\mu) = \log(\mu)$ modelira pričakovano vrednost logaritma. Smiselno jo je uporabiti, če pričakovana vrednost ne more zavzeti negativnih vrednosti. Takemu modelu rečemo *log-linearen* model. Spet druga povezovalna funkcija je $\text{logit}(\mu) = \log(\frac{\mu}{1-\mu})$, ki nam modelira

logaritem deležev - smiselno jo je uporabiti, ko μ ne zavzame vrednosti izven $(0, 1)$, torej ko imamo opravka z verjetnostmi. Takemu modelu rečemo logistični model.

2.2. **Točkovno ocenjevanje.** Preden se natančneje posvetimo posplošenim linearnim modelom, si oglejmo dve najbolj znani metodi za ocenjevanje parametrov. Najprej si definirajmo nekaj pojmov, ki jih bomo uporabljali v nadaljnjih poglavjih.

Cenilka za realnoštevilsko karatkreristiko c proučevane porazdelitve je funkcija vzorca $T = T(X_1, \ldots, X_n)$, s katero ocenjujemo c. Ta cenilka je *nepristranska*, če za porazdelitev vzorca F velja $E(T(X_1, \ldots, X_n)) = c(F)$. Imejmo sedaj zaporedje cenilk za vzorce velikosti $n = 1, 2, \ldots$. To zaporedje je *dosledno*, če v verjetnosti konvergira h konstanti $c(F)$.

Če povzamem z drugimi besedami; nepristranska cenilka nam v povprečju vrne pravi rezultat, dosledna cenilka pa z večjim vzorcem vrne rezultat vedno bližje ocenjevani karakteristiki.

2.2.1. *Metoda momentov.* Metodo momentov Čebišev je leta 1887 predstavil v svojem dokazu centralnega limitnega izreka. V splošnem ni tako uporabna kot spodaj opisana metoda največjega verjetja, je pa precej enostavna za računanje brez računalnika. Po besedah profesorja Smrekarja iz predavanj Verjetnosti in statistike „metoda momentov sloni na filozofiji: vse kar se da izraziti z momenti, izrazimo z momenti.“

V splošnem z metodo momentov postopamo takole: če je ocenjevano karakteristiko proučevane slučajne spremenljivke $c(X)$ mogoče izraziti kot funkcijo momentov,t.j. če v danem modelu ti momenti obstajajo,

$$c(X) = g(m_1(X), m_2(X), \ldots, m_r(X)),$$

za neko funkcijo g, potem $c(X)$ ocenjujemo s cenilko $g(\hat{m}_1, \ldots, \hat{m}_r)$. Če je g zvezna, dobimo dosledno cenilko.

2.2.2. *Metoda največjega verjetja.* Imejmo parametrični model s prostorom parametrov Θ in pripadajoč vektorski parameter $\theta = (\theta_1, \ldots, \theta_r)$. Privzemimo, da imajo vse proučevane porazdelitve gostote oziroma verjetnostne funkcije oblike

$$f(x; \theta) = f(x; \theta_1, \ldots, \theta_r).$$

Funkcijo verjetja za vzorec velikosti n definiramo kot funkcijo paramtetra θ, in sicer

$$\ell(X_1, \ldots, X_n; \underbrace{\theta_1, \ldots, \theta_r}_{\theta}) = f(X_1, \theta) \cdots f(X_n, \theta).$$

Kot funkcija vektorja x pa je ℓ gostota slučajnega vektorja $X = (X_1, \ldots, X_n)$.

Najti želimo tak parameter, v katerem bo funkcija verjetja zavzela svoj maksimum, torej

$$\ell(\hat{\theta}) = \max_{\theta \in \Theta} \ell(\theta).$$

Opazimo, da si računanje lahko precej poenostavimo, če obe strani zgornje enačbe logaritmiramo

$$\log(\ell(\theta)) = L(\theta) = \sum_{i=1}^{n} \log f(x_i, \theta), \tag{1}$$

in funkciji L rečemo logaritemska funkcija verjetja, njene stacionarne točke pa bodo kandidati za cenilko največjega verjetja. Ker je logaritem naraščajoča funkcija, bodo

ekstremi L in ℓ sovpadali. Rešiti moramo torej sistem enačb

$$\frac{\partial}{\partial_j}(L(\theta)) = 0 \ , j = 1, \ldots, r, \tag{2}$$

ki mu rečemo tudi sistem *enačb verjetja*, odvod logaritemske funkcije verjetja pa v statistiki pogosto poimenujejo *zbirna funkcija*. Ko rešimo enačbe verjetja, najdemo ekstrem funkcije verjetja in dobimo *cenilko največjega verjetja*, v angleščini pogosto označeno MLE (okrajšava za *maximum likelihood estimator*).

Tako dobljene cenilke niso nujno nepristranske, so pa dosledne, če je rešitev (2) enolična. V splošnem take enačbe niso rešljive eksplicitno, zato se poslužujemo različnih numeričnih metod za njihovo reševanje. Nekatere so predstavljene v drugem delu naloge.

2.3. **Linearna regresija.** Linearna regresija je najenostavnejši primer posplošenega linearnega modela. Enostavno jo lahko zapišemo kot: $Y = X\beta + \varepsilon$ kjer je Y proučevan slučajni vektor, X je matrika pojasnjevalnih slučajnih spremenljivk, β je vektor koeficientov, ki jih želimo oceniti, ε pa slučajna spremenljivka, ki predstavlja napako - pri računanju, meritvah .... Privzemimo, da je $E(\varepsilon) = 0$. Iz tega sledi $\mu = E(Y) = X\beta$. Model torej pričakovano vrednost slučajne spremenljivke predstavi kot linearno funkcijo pojasnjevalnih spremenljivk. Parametre β ocenimo z metodo najmanjših kvadratov in ob predpostavki polnega ranga za matriko X dobimo $\hat{\beta} = (X^\top X)^{-1} X^\top Y$.

2.4. **Poissonova regresija.**

2.5. **Logistična regresija.** Logistična regresija se uporablja za določanje deležev oziroma računanje verjetnosti. V poštev pride, ko imamo odgovore tipa uspeh-neuspeh oziroma govorimo o prisotnosti ali odsotnosti neke lastnosti. Spomnimo se Binomske porazdelitve $Y_i \sim B(n_i, p_i)$. Ta pravi, da je

$$P(Y_i = y_i) = \binom{n_i}{y_i} p_i^{y_i} (1 - p_i)^{n_i - y_i}$$

Pričakovana vrednost in varianca sta odvisni le od p_i, in sta enaki $E(Y_i) = n_i p_i$ in $Var(Y_i) = n_i p_i (1 - p_i)$. Poglejmo si sedaj podrobneje *logit* transformacijo. Če se spomnemo, želimo določiti verjetnost nekega dogodka pri danih podatkih. Ob uporabi identitente transformacije se nam kaj hitro lahko zgodi, da za posamezne verjetnosti dobimo vrednosti izven intervala $[0, 1]$. Ta problem bomo rešili v dveh korakih. Najprej uvedimo

$$\text{delež}_i = \frac{p_i}{1 - p_i}$$

kjer se premaknemo iz verjetnosti v *deleže* – verjetnost dogodka proti verjetnosti, da se ne bo zgodil. Če je p_i enak $\frac{1}{2}$, bo delež enak 1. Vidimo, da so deleži vedno pozitivni in niso omejeni navzgor. V naslednjem koraku pa poglejmo logaritem deležev ali logit verjetnosti

$$\eta_i = \text{logit}(p_i) = \log \frac{p_i}{1 - p_i}$$

s tem pa si odstranimo tudi omejitev navzdol. Opazimo še, da če je $p_i = \frac{1}{2}$, je delež enak 1 in je logaritem 0. Kot funkcija p, je logit strogo naraščajoča, torej imamo

inverz. Običajno ga imenujemo *antilogit*, izrazimo ga z:

$$p_i = \text{logit}^{-1}(\eta_i) = \frac{\exp \eta_i}{1 + \exp \eta_i} = \frac{\exp x_i^\top \beta}{1 + \exp x_i^\top \beta}$$

Vse skupaj nam da *logistični model*, ki za slučajni del vzame binomsko porazdelitev. Kot vidimo, zveza med prediktorji in verjetnostjo ni linearna, zato je težko oceniti, kako bo sprememba parametrov vplivala na verjetnost. Na to vprašanje lahko približno odgovorimo tako, da odvajamo po spremenljivki x_j (kar ima seveda smisel le za zvezne pojasnjevalne spremenljivke) in dobimo $\partial/\partial x_j = \beta_j p_i(1 - p_i)$. Vidimo, da na spremembo j-tega prediktorja vpliva tako verjetnost kot tudi parameter β. Običajno se za analizo vzame vzorčno povprečje verjetnosti opazovane vrednosti.

2.5.1. *Ocenjevanje parametrov.* Imamo binomske slučajne spremenljivke in imamo povezovalno funkcijo, $logitp_i = X\beta$, kjer so β neznani parametri. V naslednjem razdelku si bomo ogledali kako zanje izpeljemo enačbe verjetja, ki jih nato uporabimo v numeričnih algoritmih. Kot v vsakem posplošenem linearnem modelu tudi v tem predpostavimo neodvisnost komponent slučajnega vektorja Y zato

$$\begin{aligned} P(Y = \vec{y}) &= \prod_{i=1}^{n} P(Y_i = y_i) \\ &= \prod_{i=1}^{n} \binom{n_i}{y_i} p_i^{y_i} (1 - p_i)^{n_i - y_i} \end{aligned}$$

Naprej si oglejmo logaritemsko funkcijo verjetja. V nadaljnem računanju bom izpuščal binomski simbol na začetku - je samo konstanta, ki na končen rezultat nima vpliva. Po prejšnjih oznakah je torej

$$\begin{aligned} \ell(p_i) &= \log\{\prod_{i=1}^{n} p_i^{y_i} (1 - p_i)^{n_i - y_i}\} \\ &= \sum_{i=1}^{n} \{y_i \log p_i + (n_i - y_i) \log(1 - p_i)\} \\ &= \sum_{i=1}^{n} \{n_i \log 1 - p_i + y_i \log \left(\frac{p_i}{1 - p_i}\right)\} \end{aligned} \tag{3}$$

Po predpostavki logističnega modela je

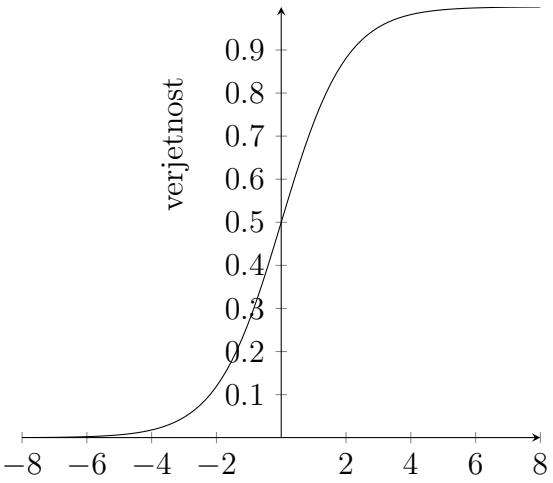
$$\text{logit}(p_i) = \log \left(\frac{p_i}{1 - p_i}\right) = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ir}\beta_r = x_i^\top \beta,$$

iz česar lahko izrazimo verjetnost p_i

$$p_i = \frac{\exp x_i^\top \beta}{1 + \exp x_i^\top \beta} \text{ ter} \tag{4}$$

$$1 - p_i = \frac{1}{1 + \exp x_i^\top \beta} \tag{5}$$

Zgornji funkciji rečemo *sigmoida*. Iz njenega grafa je morda še bolj očitno, zakaj jo je smiselno uporabiti za modeliranje verjetnosti



Slika 1. Graf sigmoide

Če izpeljane izraze za verjetnost upoštevamo v logaritemski funkciji verjetja dobimo

$$\begin{aligned}\ell(\beta) &= \sum_{i=1}^{n} \left(n_i \log \frac{1}{1+\exp x_i^\top \beta} + y_i \log \left(\frac{\frac{\exp x_i^\top \beta}{1+\exp x_i^\top \beta}}{\frac{1}{1+\exp x_i^\top \beta}} \right) \right) \\ &= \sum_{i=1}^{n} \Big(y_i (x_i^\top \beta) - n_i \log(1+\exp x_i^\top \beta) \Big) \qquad (6)\end{aligned}$$

Od tod vidimo, da je naša funkcija verjetja zares odvisna le od parametrov β, vse ostalo nam je poznano. Da torej poiščemo maksimum in s tem cenilko največjega verjetja, funkcijo odvajamo in zbirno funkcijo enačimo z 0

$$\dot{\ell}(\beta) = \begin{bmatrix} \frac{\partial \ell(\beta)}{\partial \beta_0} \\ \frac{\partial \ell(\beta)}{\partial \beta_1} \\ \vdots \\ \frac{\partial \ell(\beta)}{\partial \beta_p} \end{bmatrix}$$

Pomembno je opaziti, da parametri β vedno nastopajo ob pojasnjevalnih spremenljivkah linearno – zato bodo komponente zbirne funkcije simetrične. J-ta komponenta bo tako enaka

$$\frac{\partial \ell(\beta)}{\partial \beta_j} = \sum_{i=1}^{n} \left(x_{ij}(y_i - n_i p_i(\beta))\right), \quad j = 0, 1, \ldots r, \quad kjersmoupoštevali \tag{7}$$

$$\begin{aligned} \frac{\partial}{\partial \beta_j}(x_i^\top \beta) &= \frac{\partial}{\partial \beta_j}\left(\beta_0 + x_{i1}\beta_1 + \ldots x_{ir}\beta_r\right) \\ &= x_{ij}\text{ter} \end{aligned} \tag{8}$$

$$\begin{aligned} \frac{\partial}{\partial \beta_j} \log(1 + \exp(x_i^\top \beta)) &= \frac{\frac{\partial}{\partial \beta_j} \exp(x_i^\top \beta)}{1 + \exp(x_i^\top \beta)} \\ &= \frac{\exp(x_i^\top \beta)}{1 + \exp(x_i^\top \beta)} \frac{\partial}{\partial \beta_j}(x_i^\top \beta) \\ &= p_i(\beta) x_{ij} \end{aligned} \tag{9}$$

Enačbe, ki jih s tem postopkom dobimo, v splošnem niso eksplicitno rešljive. Za reševanje se uporablja numerične metode, ki pa slonijo na Newtonovi iteraciji. Zanjo pa je potrebno izračunati še drugi odvod, zato to storimo tu. Zopet odvajamo po komponentah, tako kot zgoraj. Najprej izračunajmo

$$\begin{aligned} \frac{\partial p_i(\beta)}{\partial \beta_k} &= \frac{\partial}{\partial \beta_k} \frac{\exp x_i^\top \beta}{1 + \exp x_i^\top \beta} \\ &= x_{ik} p_i(\beta)(1 - p_i(\beta)) \end{aligned}$$

Vse sedaj skupaj sestavimo v

$$\frac{\partial^2}{\partial \beta_j \partial \beta_k} = -\sum_{i}^{n} \left(x_{ij} x_{ik} n_i p_i(\beta)(1 - p_i(\beta))\right), \quad j, k = 0, 1, \ldots, r \tag{10}$$

Spomnimo se, da delamo z binomskimi slučajnimi spremenljivkami in torej velja $var(Y_i) = v_i(\beta) = n_i p_i(1 - p_i)$, kar vključimo v zgornjo enačbo in končno dobimo

$$\ddot{\ell}(\beta) = -\sum_{i=1}^{n} \left(x_{ij} x_{ik} v_i(\beta)\right). \tag{11}$$

Zapišimo zgoraj izpeljane zveze v berljivejšo matrično notacijo.

$$\log\left(\frac{p}{1-p}\right) = \mathbf{X}\beta$$

Vektorsko definiramu tudi

$$\exp \mathbf{X}\beta = \begin{bmatrix} \exp x_1^\top \beta \\ \vdots \\ \exp x_n^\top \beta \end{bmatrix},$$

spomnimo se enačbe (3) in iz nje izpeljimo

$$\begin{aligned} \ell(\beta) &= \sum_{i=1}^{n} \{n_i \log 1 - p_i + y_i \log\left(\frac{p_i}{1 - p_i}\right)\} \\ &= y^\top \mathbf{X}\beta - n^\top \log(1 + \exp \mathbf{X}\beta), \end{aligned} \tag{12}$$

in še odvoda zgornje funkcjie, ki pa ga lahko zapišemo kot

$$\dot{\ell}(\beta) = \mathbf{X}^\top (y - m \circ p(\beta)), \tag{13}$$

kjer je s $\circ$ označeno Hadamardovo množenje po elementih. S pričakovano vrednostjo vektorja označimo vektor pričakovanih vrednosti komponent in torej lahko zapišemo

$$\mathrm{E}(Y) = m \circ p(\beta) \equiv \mu(\beta), \tag{14}$$

in lahko končno vse povzamemo v

$$\dot{\ell}(\beta) = \mathbf{X}^\top (y - m \circ p(\beta)) = X^\top (y - \mu(\beta)) \tag{15}$$

Ostane nam le še dvojni odvod. Najprej si oglejmo

$$v(\beta) = \begin{bmatrix} v_1(\beta) & & & \\ & v_2(\beta) & & \\ & & \ddots & \\ & & & v_n(\beta) \end{bmatrix},$$

iz tega potem takoj sledi, da je

$$\ddot{\ell}(\beta) = -\mathbf{X}^\top v(\beta) \mathbf{X}, \tag{16}$$

torej element v j-ti vrstici in k-tem stolpcu je $\sum_{i=1}^{n} x_{ij} x_{ik} v_i(\beta)$.

2.6. **Probit regresija.** Probit regresija se uporablja v podobne namene kot logistična, torej za določanje verjetnosti in razvrščanje. Razvili so jo v tridesetih letih dvajsetega stoletja, ime pa je skovanka – pride iz anglešlih besed ***prob****ability un****it***. V glavnem se od logistične regresije razlikuje v sistematičnem delu. Verjetnost pozitivnega izida torej po modelu predpostavljamo

$$p_i(\beta) = \Phi(\beta_0 + x_{i1}\beta_1 + \ldots + x_{ir}\beta_r), \tag{17}$$

kjer Φ predstavlja kumulativno porazdelitveno funkcijo standardne normalne slučajne spremenljivke. Ta seveda ni linearna, ne znamo je niti zapisati z elementarnimi funkcijami. Podobno kot v logističnem modelu, se bomo ocenjevanja parametrov lotili po metodi največjega verjetja.

2.6.1. *Ocenjevanje parametrov probit regresije.*

3. Numerične metode

V sledečih razdelkih si bomo od bliže pogledali nekaj numeričnih metod, uporabljenih v kasnejših zgledih. Te metode slonijo na stoletja starih idejah, ki smo jih spoznali tekom študija, uporabljajo pa se tudi v številnih praktičnih aplikacijah.

3.1. **Newton – Raphsonova metoda.** Newton – Raphson (oziroma le Newtonova) metoda je bila v osnovi razvita za iskanje ničel funkcije. V najosnovnejši (ter najpogostejši) verziji za iskanje ničle funkcije ene spremenljivke začnemo v neki točki, naslendnjo pa izberemo v presčišču tangente, izračunane v tej točki, z x-osjo. Postopek tako iterativno nadaljujemo. Ideja je torej sila preprosta, za izpeljavo pa tudi ni potrebno preveč truda. Predpostavimo odvedljivost funkcije na nekem intervalu in recimo, da imamo trenuten približek x_n. Razvijmo sedaj funkcijo v Taylorjev polinom prve stopnje okoli x_n :

$$f(x) \approx f(x_n) + f'(x_n)(x - x_n)$$

Presečišče najdemo, če zgornjo enačbo enačimo z 0 in dobimo znano formulo

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Metoda bo skonvergirala za začetne približke dovolj blizu ničli in v neki okolici ničle konvergirala s kvadratično hitrostjo. Na težave naletimo v več primerih. Najprej, blizu stacionarne točke metoda odpove, saj bi delili z 0 (oziroma vrednostmi blizu ničle, kar je numerično nestabilno). Problem lahko predstavlja tudi računanje odvoda, ki zna biti zahtevno, ter dejstvo, da za slabe začetne približke ničle morda ne bomo našli. S temi težavami se bomo soočili v nadaljevanju. Imamo torej algoritem, ki najde ničlo, v luči iskanja cenilke največjega verjetja pa bi želeli algoritem, ki poišče maksimum oziroma minimum funkcije. Recimo, da imamo neko logaritemsko funkcijo verjetja L, in trenutni približek θ_n. Razvijmo funkcijo okoli približka v Taylorjev polinom druge stopnje:

$$L(\theta) \approx L(\theta_n) + dL(\theta_n)(\theta - \theta_n) + \frac{1}{2}(\theta - \theta_n)^\top d^2L(\theta_n)(\theta - \theta_n) \tag{18}$$

Maksimizirati želimo desno stran (18). To storimo tako, da gradient L enačimo z nič:

$$\nabla L(\theta_n) + \nabla^2 L(\theta_n)(\theta - \theta_n) = 0$$

in izrazimo naslednji približek

$$\theta_{n+1} = \theta_n - \nabla^2 L(\theta_n)^{-1} \nabla L(\theta_n).$$

S tem postopkom imamo lahko dva problema. Prvič, lahko je zahtevno računati in invertirati drugi odvod (Hesian) funkcije, morda za kakšen θ_n sploh ne obstaja. Drugič, proč od $\hat{\theta}$ lahko Newtonova metoda napreduje navzgor ali navzdol – oboje je enako verjetno. Z drugimi besedami, Newtonova metoda ni naraščajoč algoritem in torej ne da $L(\theta_n) < L(\theta_{n+1})$. Mi pa bi želeli algoritem, ki bo konvergiral globalno (in ne le na nekem intervalu okoli rešitve). Težavo z računanjem inverza rešimo tako, da namesto invertiranja problem prevedemo na reševanje sistema enačb za premik:

$$\begin{aligned} x_{n+1} &= x_n + p_n \\ \nabla^2 L(\theta_n) p_n &= -\nabla L(\theta_n) \end{aligned} \tag{19}$$

Zadnji vrstici v (19) rečemo tudi *Newtonova enačba*. Radi bi še dosegli, da bi se Newtonov algoritem premikal v eno smer, torej naraščal ali padal. S tem bi vedeli, kaj se bo zgodilo v iteraciji in lažje predvideli morebitne nevšečnosti. Newtonova metoda za iskanje minimuma (maksimum) funkcije je optimizacijski problem drugega reda in realna funkcija ima globalni minimum (maksimum) tam, kjer je njen drugi odvod pozitiven, oziroma v primeru funkcij več spremenljivk, kjer je njen Hesian pozitivno definiten (in je tam gradient enak nič). Če bi torej imeli strogo pozivino definitno matriko, bi bil ta optimizacijski problem konveksen in kot tak rešljiv globalno (veljati morajo še pogoji Karush-Kuhn-Tuckerja, vendar je to skoraj vedno res). Imejmo torej v točki x^* pozitivno definitno Hessejevo matriko H. Zapišimo Taylorjev polinom druge stopnje okoli te točke

$$f(x^* + s) = f(x_x^*) + \nabla f(x^*)s + \frac{1}{2}s^\top H(x^*)s.$$

Če velja še pogoj prvega reda, torej $\nabla f(x^*) = 0$, imamo

$$f(x^* + s) = f(x^*) + \frac{1}{2}s^\top H(x^*)s,$$

kar pomeni, da se vrednost funkcije vedno poveča, če se premaknemo iz stacionarne točke x^* (drugi člen je vedno pozitiven zaradi pozitivne definitnosti) Tako vidimo, da imamo strogo padajoč algoritem.

3.2. **Fisher's scoring.** Fisher's scoring algoritem je variacija v prejšnjem razdelku opisanega Newton – Raphsonovega algoritma, ki se v statistiki uporablja za numerično reševanje enačb največjega verjetja. Poimenovana je po Ronaldu Fisherju, enem najpomembenjših angleških statistikov dvajsetega stoletja. Uvedimo najprej nekaj terminologije in oznak. Kot v prejšnjih poglavijh, imamo veliko opravka logaritemsko funckijo verjetja (angl. *log-likelihood function*), ki jo tu označimo z $L = \log\left(f(x_1,\theta)\dots f(x_n,\theta)\right)$, ki jo obravnavamo kot funkcijo parametra θ pri fiksnem vzorcu. Po drugi strani pa je to tudi transformacija porazdelitvene funkcije slučajnega vektorja $X = (X_1, X_2, \dots, X_n)$, kjer privzamemo neodvisnost komponent (kar pa seveda velja v vseh modelih, ki jih obravnavamo v tej nalogi). Zbirna funkcija (angl. *score function*) je gradient logaritemske funkcije največjega verjetja po ocenjevanem parametru. Informacijska (oziroma Fisherjeva Informacijska) matrika (angl. *(Fisher) information matrix*) je definirana kot

$$I(\theta) = \mathrm{E}[\nabla L(\theta)\nabla L(\theta)^\top].$$

Fisher scoring algoritem sedaj je po zgornjih oznakah

$$\theta n + 1 = \theta_n - I(\theta)^{-1}\nabla L(\theta) \tag{20}$$

3.2.1. *Fisherjeva informacijska matrika in informacijska enakost.* V sledečem razdelku bomo utemeljili uporabo Fisher scoring algoritma in prikazali njegovo glavno prednost pred običajno Newtonovo iteracijo, opisano v poglavju 3.1. Ob določenih predpostavkah o gladkosti logaritemske funkcije verjetja (ki za eksponentno družino vedno držijo), bomo sedaj pokazali, da je pričakovana vrednost zbirne funkcije enaka nič. Tu smo $\ell(\theta)$ označili z $\log f_\theta(x)$.

$$\begin{aligned}\mathrm{E}[\nabla L(\theta)] &= \int f_\theta(x)\nabla_\theta \log(f_\theta(x))\,dx = \int f_\theta(x)\frac{\nabla f_\theta(x)}{f_\theta(x)}\,dx \\ &= \int \nabla f_\theta(x)\,dx \overset{*}{=} \nabla \int f_\theta(x)\,dx = \nabla 1 = 0,\end{aligned} \tag{21}$$

kjer smo v enakosti $(*)$ zamenjali integracijo in odvajanje, kar po teoriji mere smemo - gostota porazdelitev eksponentne družine je zvezno odvedljiva. Velja tudi

$$\nabla^2 \log f_\theta(x) = \frac{\nabla^2 f_\theta(x)}{f_\theta(x)} - \frac{\nabla f_\theta(x)\nabla f_\theta(x)^\top}{f_\theta(x)^2} = \frac{\nabla^2 f_\theta(x)}{f_\theta(x)} - \dot{\ell}(\theta)\dot{\ell}(\theta)^\top,$$

kjer smo v zadnji enakosti upoštevali

$$\nabla L(\theta) = \nabla \log(f_\theta(x)) = \frac{\nabla f_\theta(x)}{f_\theta(x)}$$

Povzemimo vse, kar smo dokazali v prejšnjih enačbah in imamo

$$\begin{aligned}I(\theta) &= \mathrm{E}[\nabla L(\theta)(\nabla L(\theta))^\top] = -\int f_\theta(x)\nabla^2 \log f_\theta(x)\,dx + \int \nabla^2 f_\theta(x)\,dx \\ &= -\mathrm{E}[\nabla^2 \log f_\theta(x)] + \nabla^2 \int \nabla^2 \log f_\theta(x)\,dx = -\mathrm{E}[\nabla^2 \log f_\theta(x)]\end{aligned} \tag{22}$$

Enačbi (22) rečemo tudi *informacijska enakost*. To nam bo koristilo pri dokazovanju enakosti med Newton – Raphson algoritmom in Fisher scoringom za logistični model. Kot smo videli v prejšnjih poglavjih, želimo imeti pozitivno semi-definitno matriko in s tem monoton algoritem. Izkaže se, da je informacijska matrika ravno variančno-kovariančna matrika zbirne funkcije. Po enačbi (21) se spomnimo, da je pričakovana

vrednost zbirne funckije enaka 0. Potem takoj sledi

$$\begin{aligned} I(\theta) &= \mathrm{E}[\nabla L(\theta) \nabla L(\theta)^\top] \\ &= \mathrm{E}[(\nabla L(\theta) - \mathrm{E}[\nabla L(\theta)]) (\nabla L(\theta) - \mathrm{E}[\nabla L(\theta)])^\top] \\ &= \mathrm{Var}[\nabla L(\theta)]. \end{aligned} \tag{23}$$

Variančno-kovariančne matrike pa so pozitivno semi-definitne.

3.2.2. *Fisher's scoring v logističnem modelu.* Poglejmo za trenutek nazaj v poglavje 2.5.1, natančneje k enačbam (12),(15) in (16). Iz prejšnjega razdelka vemo tudi, da velja

$$I(\theta) = \mathrm{E}[-\nabla^2 L(\theta)] \overset{(16)}{=} X^\top v(\theta) X = -\nabla^2 L(\theta),$$

kjer smo seveda uporabili tudi prej dokazano informacijsko enakost. Tako vidimo, da Fisher's scoring in Newton–Raphsonova metoda v primeru logistične regresije res sovpadata, saj je matrika drugih odvodov ravno enaka informacijski matriki. Če zapišemo sedaj vse skupaj

$$\begin{aligned} \hat{\theta}_{i+1} &= \hat{\theta}_i - (\nabla^2 L(\hat{\theta}_i))^{-1} \nabla L(\hat{\theta}_i) \\ &= \hat{\theta}_i + (\mathbf{X}^\top v(\hat{\theta}_i) \mathbf{X})^{-1} \mathbf{X}^\top (y - \mu(\hat{\theta}_i)) \end{aligned} \tag{24}$$

4. Primeri

4.1. **Ocenjevanje parametrov v logističnem modelu.** V praktično usmerjenem delu te naloge smo v Pythonu implementirali zgoraj opisani postopek Fisher scoring algoritma za binomsko porazdeljene slučajne spremenljivke. Za delo v Pythonu smo uporabili več knjižnic `NumPy` za računanje z matrikami in vektorji, reševanje sistemov enačb ter invertiranje, knjižnico `pandas` za uvoz podatkov in njihovo začetno urejanje. Na koncu smo si s paketom `Pyplot` iz knjižnice `Matplotlib` rezultate izrisali. V implementaciji smo popolnoma sledili zgoraj izpeljanim enačbam, zato jih tu nebomo ponovno navajali.

4.2. **Ocenjevanje parametrov v probit modelu.**

Slovar strokovnih izrazov

Literatura

[1] Alan Agresti. *An introduction to categorical data analysis*. John Wiley & Sons, 2007.

[2] Erik B. Erhard. Logistic regression and nr. https://statacumen.com/teach/SC1/SC1_11_LogisticRegression.pdf.

[3] Kenneth Lange. *Numerical analysis for statisticians*. Springer Science & Business Media, 2010.

[4] Germán Rodríguez. Lecture notes on generalized linear models. https://data.princeton.edu/wws509/notes/, 2007.