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

1. Uvod 4
2. Eksponentna družina 4
3. Posplošeni linearni modeli 6
3.1. Sestavni deli posplošenega linearnega modela 6
3.2. Točkovno ocenjevanje 7
3.3. Linearna regresija 8
3.4. Logistična regresija 8
3.5. Obstoj rešitve enačb verjetja v logističnem modelu 12
3.6. Kanonični modeli v splošnem 15
3.7. Probit regresija 16
4. Numerične metode 19
4.1. Newton – Raphsonova metoda 19
4.2. Fisher's scoring 23
5. Primeri 24
5.1. Ocenjevanje parametrov v logističnem modelu 24
5.2. Ocenjevanje parametrov v probit modelu 28
5.3. Primerjava logit in probit modela 29
Slovar strokovnih izrazov 29
Literatura 29

Iterativne numerične metode v posplošenih linearnih modelih

Povzetek

V delu obravnavamo numerične metode, ki se uporabljajo pri računanju cenilk največjega verjetja v posplošenih linearnih modelih. Za uvod si postavimo teoretične temelje z eksponentno družino, nato pa natančneje spoznamo logistični in probit model, za katera tudi izpeljemo enačbe verjetja. Dobljene enačbe komentiramo tudi v splošnem in komentiramo, zakaj je smiselno uporabljatii t.i. kanonične modele. V delu, namenjem numeričnim metodam, najprej navedemo in dokažemo nekaj dejstev o Newtonovi metodi za računanje ničel funkcij, ki jo nato prilagodimo za iskanje ekstremov funkcij verjetja. Vse zaključke preizkusimo in komentiramo tudi na praktičnih primerih.

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

V sodobnem svetu neomejenih podatkov, je njihovo obvladanje in koristno uporabljanje ključnega pomena. Velikokrat relacije med njimi niso vidne na prvi pogled, a zato niso nič manj pomembne. Posplošeni linearni modeli so orodje, s katerimi te povezave modeliramo in jih poskušamo razumeti. So ena osnovnih metod statističnga raziskovanja. Vendar pa najenostavnejši modeli niso nujno najboljši in z zahtevnostjo naraste tudi problematika računanja. Tu pa nam na pomoč priskočijo numerične metode, ki pa morajo biti dovolj robustne in hkrati čim enostavnejše za računanje.

V delu si bomo najprej ogledali linearni model, natančneje pa spoznali dva glavna modela za obdelavo podatkov, kjer so odgovori 0 ali 1 - binarnih podatkov, to sta *logistični* in *probit model*, uporabna za računanje verjetnosti dogodgkov. Za njihovo uporabo bomo razvili numerične algoritme in jih na koncu preizkusili na podatkih o odpovedih tesnil vesoljskih poletov pred dobo *Challengerja.*

2. Eksponentna družina

Za uvod v nalogo si najprej definirajmo osnovo, na kateri bo kasneje temeljil eden glavnih zaključkov naloge. Predvsem nam bodo zaključki poglavja pomagali pri posploševanju rezultatov. Slučajna spremenljivka Y torej pripada *enoparametrični eksponentni družini z disperzijskim parametrom,* če je njegova gostota glede na neko $\sigma-$končno mero oblike

$$f_Y(y;\theta,\phi) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right), \tag{1}$$

za neke funkcije $a(\cdot), b(\cdot)$ in $c(\cdot)$. Parametru θ pravimo *kanonični* oziroma *naravni* parameter, ϕ pa imenujemo *disperzijski parameter*.

Koristno je pogledati logaritem zgornje enačbe

$$\log f_Y(y;\theta,\phi) = \frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi) \tag{2}$$

$$\frac{\partial}{\partial\theta}\log f_Y(y;\theta,\phi) = V_\theta(y) = \frac{y - b'(\theta)}{a(\phi)} \tag{3}$$

$$\frac{\partial^2}{\partial\theta^2}\log f_Y(y;\theta,\phi) = -\frac{b''(\theta)}{a(\phi)}, \tag{4}$$

kjer funkcijo V imenujemo *funkcija zbira*, oziroma v angleščini *score function.*

Dokažimo sedaj nekaj koristnih zvez, ki jih bomo uporabili v kasnejših izpeljavah.

Trditev 2.1. *Naj bo Y slučajna spremenljivka, katere gostota pripada eksponentni družini. Potem za pričakovano vrednost in varianco veljata sledeči zvezi:*

$$\mathbb{E}(Y) = b'(\theta), \quad Var(Y) = b''(\theta)a(\phi)$$

Dokaz. Za dokaz prve enakosti si poglejmo

$$\mathbb{E}(V(Y)) = \int f_Y(y)\frac{\partial}{\partial\theta}\log f_Y(y;\theta,\phi)\,dy = \int f_Y(y)\frac{1}{f_Y(y)}\frac{\partial f_Y(y)}{\partial\theta}\,dy$$
$$= \int \frac{\partial f_Y(y)}{\partial\theta}\,dy = \frac{\partial}{\partial\theta}\int f_Y(y)\,dy = \frac{\partial}{\partial\theta}1 = 0,$$

saj je $f_Y(y)$ gostota. V zgornji in sledečih zvezah bomo namesto $f_Y(y;\theta,\phi)$, kjer to ne bo vodilo v dodatne težave, pisali kar $f_Y(y)$ Od tu sledi

$$\begin{aligned}\mathbb{E}(V(y)) &= \mathbb{E}(\frac{Y-b'(\theta)}{a(\phi)}) = 0\\ \mathbb{E}(Y) &= b'(\theta)\end{aligned}$$

Za drugo pa si oglejmo

$$\begin{aligned}\mathbb{E}(\frac{\partial}{\partial\theta}V(Y)+V(Y)^2) &= \int f_Y(y)\left(\frac{\partial}{\partial\theta}\left(\frac{1}{f_Y(y)}\frac{\partial f_Y(y)}{\partial\theta}\right)+\left(\frac{\partial}{\partial\theta}\log f_Y(y)\right)^2\right)dy\\ &= \int f_Y(y)\left(-\frac{1}{f_Y(y)^2}\left(\frac{\partial f_Y(y)}{\partial\theta}\right)^2+\frac{1}{f_Y(y)}\frac{\partial^2 f_Y(y)}{\partial\theta^2}+\left(\frac{1}{f_Y(y)}\frac{\partial f_Y(y)}{\partial\theta}\right)^2\right)dy\\ &= \int\frac{\partial^2}{\partial\theta^2}f_Y(y)\,dy = \frac{\partial^2}{\partial\theta^2}\int f_Y(y)\,dy = 0\end{aligned}$$

Spet uporabimo prej izpeljane zveze in dobimo

$$\frac{\partial^2}{\partial\theta^2}\log f_Y(y) = \frac{\partial}{\partial\theta}\left(\frac{y-b'(\theta)}{a(\phi)}\right) = -\frac{b''(\theta)}{a(\phi)},\quad \mathbb{E}\left(-\frac{b''(\theta)}{a(\phi)}\right) = -\frac{b''(\theta)}{a(\phi)}$$

$$\mathbb{E}\left(V(Y)^2\right) = \mathbb{E}\left(\left(\frac{Y-b'(\theta)}{a(\phi)}\right)^2\right) = \frac{1}{a(\phi)^2}\mathbb{E}((Y-\mathbb{E}(Y))^2) = \frac{1}{a(\phi)^2}Var(Y),$$

in po zgoraj dokazani enakosti za funkcijo zbira torej velja

$$\begin{aligned}-\mathbb{E}\left(\frac{\partial^2}{\partial\theta^2}\log f_Y(y)\right) &= \mathbb{E}\left(V(Y)^2\right)\\ \frac{b''(\theta)}{a(\phi)} &= \frac{1}{a(\phi)^2}Var(Y)\\ \mathrm{Var}(Y) &= a(\phi)b''(\theta)\end{aligned}$$

□

Zgornja trditev nam torej pove, da lahko pričakovano vrednost in varianco porazdelitve iz eksponentne družine, z nekaj odvajanja, preberemo kar iz gostote - izognemo se integriraciji, iz zadnje zveze pa vidimo zakaj se parametru ϕ reče ravno disperzijski parameter.

Pričakovano vrednost kvadrata funkcije zbira v splošnem imenujemo *Fisherjeva informacija*, $\mathrm{FI}(\theta) = \mathbb{E}((V(Y))^2)$, izpeljano zvezo, ki poveže funkcijo zbira in njene odvode pa *informacijska enakost.* Uporabnost teh zvez bo postala jasna v sledečih poglavjih.

Oglejmo si sedaj nekaj primerov porazdelitev eksponentne družine:

- **Normalna porazdelitev.** Normalno porazdeljena slučajna spremenljivka $Y \sim N(\mu,\sigma^2)$ ima gostoto $f_Y(y) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$. Če zgornjo enačbo logaritmiramo dobimo

$$\begin{aligned}\log f_Y(y;\mu,\sigma) &= \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)-\frac{(y-\mu)^2}{2\sigma^2} = -\frac{1}{2}\log(2\sigma^2\pi)-\frac{y^2-2\mu y+\mu^2}{2\sigma^2}\\ &= \frac{y\mu-{}^{\mu^2}\!/_2}{\sigma^2}-\frac{1}{2}\left(\frac{y^2}{\sigma^2}+\log(2\pi\sigma^2)\right)\end{aligned}$$

Od tu preberemo zgoraj definirane vrednosti

$$\theta = \mu,\ \phi = \sigma^2,\ a(\sigma^2) = \sigma^2,\ b(\mu) = \frac{\mu^2}{2}$$

in iz trditve 2.1 sledijo zaključki

$$\mathbb{E}(Y) = b'(\mu) = \mu \text{ in } \mathrm{Var}(Y) = a(\sigma^2)b''(\mu) = \sigma^2.$$

- **Binomska porazdelitev.** Imejmo binomsko porazdeljeno slučajno spremenljivko $Y \sim \mathrm{Bin}(n, p)$. Izrazimo

$$\mathrm{P}(Y = y) = \binom{n}{y} p^y (1-p)^{n-y} = \exp\left(y \log(\frac{p}{1-p}) + n\log(1-p) + \log\binom{n}{k}\right),$$

od koder direktno sledi

$$\theta = \log\frac{p}{1-p},\ b(\theta) = n\log(1+e^\theta),\ a(\phi) = 1,$$

in opazimo da je tokrat naravni parameter $\log\frac{p}{1-p}$, kar imenujemo tudi *logit* verjetnosti.

3. Posplošeni linearni modeli

3.1. **Sestavni deli posplošenega linearnega modela.** Vsak posplošeni linearni model sestavljajo trije deli: *slučajni del* je slučajni vektor Y in njegova porazdelitev, *sistematični del* predstavlja relacijo med pojasnjevalnimi spremenljivkami, *povezovalna funkcija* pa transformira $\mathbb{E}(Y)$, da se ta bolje prilega podatkom. V nalogi bomo proučevali vektor $Y = (Y_1, Y_2, \ldots, Y_n)$, kjer so komponente neodvisne slučajne spremenljivke iz enoparametrične eksponentne družine z disperzijskim parametrom.

3.1.1. *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 števne 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).

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

$$\beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + \ldots + x_{ip}\beta_p$$

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

$$g(\mu) = \beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + \ldots + x_{ip}\beta_p$$

Najenostavnejša taka funkcija je kar identiteta, torej $g(\mu) = \mu$. Ta nam torej da linearno povezavo med pojasnjevalnimi spremenljivkami in pričakovano vrednostjo naših slučajnih spremenljivk. 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.

3.2. **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 $\mathbb{E}(T(X_1, \ldots, X_n)) = c(F)$. 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.

3.2.1. *Metoda momentov.* Metodo momentov je Čebišev 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. Če malce karikiramo, lahko idejo metode momentov povzamemo v „vse kar se da izraziti z momenti, ocenimo s cenilkami momentov."

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 $g(\hat{m}_1, \ldots, \hat{m}_r)$,kjer je $\hat{m}_k = \frac{1}{n}\sum_{i=1}^{n} X_i^k$. Če je g zvezna, dobimo dosledno cenilko.

3.2.2. *Metoda največjega verjetja.* Imejmo parametrični model s prostorom parametrov $\Theta \subseteq \mathbb{R}^r$ 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

$$F(x_1, \ldots, x_n; \underbrace{\theta_1, \ldots, \theta_r}_{\theta}) = f_1(x_1, \theta) \cdots f_n(x_n, \theta).$$

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

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

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

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

$$\log(F(\theta)) = L(\theta) = \sum_{i=1}^{n} \log f_i(x_i, \theta). \tag{5}$$

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 F sovpadali. Rešiti moramo torej sistem enačb

$$\frac{\partial}{\partial \theta_j}(L(\theta)) = 0 \ , j = 1, \ldots, r, \tag{6}$$

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 (6) enolična. V splošnem take enačbe niso rešljive eksplicitno, zato se poslužujemo različnih numeričnih metod za njihovo reševanje. Nekatere so predstavljene v drugem delu naloge.

3.3. **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 dimenzije n, $X \in \mathbb{R}^{n\times(p+1)}$ je matrika pojasnjevalnih slučajnih spremenljivk, β je vektor koeficientov dimenzije $p+1$, ki jih želimo oceniti, ε pa slučajna spremenljivka, ki predstavlja napako - pri računanju, meritvah $\ldots$. 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 - iščemo tak $\hat{\beta}$, ki bo zadoščal

$$\|y - X\hat{\beta}\|^2 = \min_{\beta} \|y - X\beta\|^2.$$

Želimo torej element slike matrike X, ki bo v drugi normi najbližje vektorju y, in izkaže se, da je to ravno pravokotna projekcija tega vektorja na $im(X)$. To lahko zapišemo Kot

$$\langle y - X\hat{\beta}, Xh\rangle = 0 \ \forall h \in \mathbb{R}^{p+1},$$

kar pa velja natanko tedaj kot

$$X^\top(y - X\hat{\beta}) = 0$$
$$(X^\top X)\hat{\beta} = X^\top y.$$

Če je $rang(X) = p + 1$ in je posledično matrika $X^\top X$ obrnljiva, dobimo enolično rešitev po metodi najmanjših kvadratov oblike

$$\hat{\beta} = (X^\top X)^{-1} X^\top y.$$

3.4. **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. Kot smo že omenili, bomo proučevali vektor, katerega komponente so iz eksponentne družine z disperzijskim parametrom, kamor seveda spada tudi binomska porazdelitev: $Y_i \sim B(n_i, p_i)$. 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

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

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. Označimo z $\eta_i = \exp x_i^\top \beta$.Običajno ga imenujemo *antilogit*, izrazimo ga z:

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

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.

3.4.1. *Ocenjevanje parametrov.* Imamo binomske slučajne spremenljivke in imamo povezovalno funkcijo, $\text{logit}(p_i) = X\beta$, 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

$$
\begin{aligned}
L(p_i) &\propto \log\{\prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{n_i-y_i}\} \\
&\propto \sum_{i=1}^{n}\{y_i \log p_i + (n_i - y_i)\log(1-p_i)\} \\
&\propto \sum_{i=1}^{n}\{n_i \log(1-p_i) + y_i \log\left(\frac{p_i}{1-p_i}\right)\} \qquad (7)
\end{aligned}
$$

Po predpostavki logističnega modela je

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = x_{i0}\beta_0 + x_{i1}\beta_1 + \ldots + x_{ir}\beta_r = x_i^\top \beta,$$

kjer je $x_{i0} = 1,\ i = 1, \ldots, n$

Od tod lahko izrazimo verjetnosti p_i

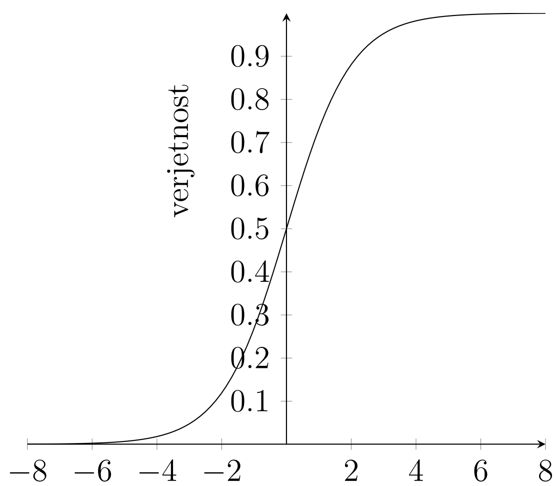
$$p_i = \frac{\exp x_i^\top \beta}{1 + \exp x_i^\top \beta} \text{ ter} \qquad (8)$$

$$1 - p_i = \frac{1}{1 + \exp x_i^\top \beta}. \qquad (9)$$

Spodnji funkciji rečemo *sigmoida*, definiramo jo kot

$$f(x) = \frac{e^x}{1 + e^x}.$$

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

$$
\begin{aligned}
L(\beta) &\propto \sum_{i=1}^{n} \left(n_i \log \frac{1}{1+\exp x_i^\top \beta} + y_i \log \left(\frac{\frac{\exp x_i^\top \beta}{1+\exp x_i^\top \beta}}{\frac{1}{1+\exp x_i^\top \beta}} \right) \right) \\
&\propto \sum_{i=1}^{n} \left(y_i (x_i^\top \beta) - n_i \log(1+\exp x_i^\top \beta) \right) \qquad (10)
\end{aligned}
$$

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

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

Pomembno je opaziti, da parametri β vedno nastopajo ob pojasnjevalnih spremenljivkah linearno, zato bodo vse komponente enake oblike. J-ta komponenta bo tako enaka

$$
\frac{\partial L(\beta)}{\partial \beta_j} = \sum_{i=1}^{n} \left(x_{ij}(y_i - n_i p_i(\beta)) \right), \quad j = 0, 1, \ldots r, \quad \text{kjer smo upoštevali} \qquad (11)
$$

$$
\begin{aligned}
\frac{\partial}{\partial \beta_j}(x_i^\top \beta) &= \frac{\partial}{\partial \beta_j} \left(\beta_0 + x_{i1}\beta_1 + \ldots x_{ir}\beta_r \right) \\
&= x_{ij}, \qquad (12)
\end{aligned}
$$

ter

$$
\begin{aligned}
\frac{\partial}{\partial \beta_j} \log(1+\exp(x_i^\top \beta)) &= \frac{\frac{\partial}{\partial \beta_j} \exp(x_i^\top \beta)}{1+\exp(x_i^\top \beta)} \\
&= \frac{\exp(x_i^\top \beta)}{1+\exp(x_i^\top \beta)} \frac{\partial}{\partial \beta_j}(x_i^\top \beta) \\
&= p_i(\beta) x_{ij} \qquad (13)
\end{aligned}
$$

Enačbe, ki jih s tem postopkom dobimo, v splošnem niso eksplicitno rešljive. Za reševanje se uporablja numerične metode, ki slonijo na Newtonovi iteraciji. Kot bomo kasneje pokazali, je zanjo 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

$$
\frac{\partial^2}{\partial \beta_j \partial \beta_k} L(\beta) = -\sum_{i}^{n} \left(x_{ij} x_{ik} n_i p_i(\beta)(1-p_i(\beta)) \right), \quad j, k = 0, 1, \ldots, r \qquad (14)
$$

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

$$\ddot{\ell}(\beta) = -\sum_{i=1}^{n} \left(x_{ij} x_{ik} v_i(\beta) \right). \tag{15}$$

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 (7) in iz nje izpeljimo

$$\begin{aligned} L(\beta) &= \sum_{i=1}^{n} \{ n_i \log 1 - p_i + y_i \log \left(\frac{p_i}{1-p_i} \right) \} \\ &= y^\top \mathbf{X}\beta - n^\top \log(1 + \exp \mathbf{X}\beta), \end{aligned} \tag{16}$$

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

$$\dot{L}(\beta) = \mathbf{X}^\top (y - m \circ p(\beta)), \tag{17}$$

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

$$\mathrm{E}(Y) = m \circ p(\beta) \equiv \mu(\beta), \tag{18}$$

in lahko končno vse povzamemo v

$$\dot{L}(\beta) = \mathbf{X}^\top (y - m \circ p(\beta)) = X^\top (y - \mu(\beta)) \tag{19}$$

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

$$\ddot{L}(\beta) = -\mathbf{X}^\top v(\beta) \mathbf{X}, \tag{20}$$

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

3.5. **Obstoj rešitve enačb verjetja v logističnem modelu.** *To poglavje je še nepopolno, moram razmisliti kaj vključiti.*

V prejšnjem odseku smo izpeljali enačbe verjetja za logistično regresijo in videli, da v splošnem niso analitično rešljive. Porodi pa se vprašanje, kdaj rešitev pravzaprav sploh obstaja? Izkaže se, da je obstoj in enoličnost rešitve v logističnem modelu moč dokazati iz podatkov. Sledeče poglavje temelji na članku [2].

Zopet bomo vpeljali nekaj novih oznak. Imejmo n neodvisnih opazovanj vektorja dimenzije p in določimo (x, H), kjer je $x^\top = (x_0, \ldots, x_p), x_0 \equiv 1$, H pa je spremenljivka, ki zavzame vrednosti $H_1, \ldots, H_g$ in pokaže kateri skupini pripada določeno opazovanje. V našem primeru je $g = 2$, H_1 ustreza Bernoullijevi enki, H_2 pa Bernoullijevi ničli.

3.5.1. *Funkcija verjetja.* Za potrebe tega poglavja vpeljimo novo notacijo za funkcijo verjetja

$$
\begin{aligned}
\mathbf{pr}(H_s|x) &= \exp(\beta_s^\top x)\mathbf{pr}(H_g|x), \;\; s = 1, \ldots, g-1 \\
\mathbf{pr}(H_g|x) &= {}^1\!/\!\textstyle\sum_{s=1}^{g} \exp(\beta_s^\top x) \\
\beta_s^\top &= (\beta_{s0}, \ldots, \beta_{sp}), \;\; s = 1, \ldots, g-1, \;\; \beta_g^\top = 0
\end{aligned} \tag{21}
$$

Za ilustracijo si zopet poglejmo primer $g = 2$, enačbe verjetja potem izgledajo

$$
\begin{aligned}
\mathbf{pr}(H_2|x) &= \frac{1}{\exp(\beta_1^\top x) + \exp(\beta_2^\top x)} = \frac{1}{1 + \exp(\beta^\top x)} \\
\mathbf{pr}(H_1|x) &= \frac{\exp(\beta_1^\top x)}{1 + \exp(\beta_1^\top x)},
\end{aligned} \tag{22}
$$

kar se seveda sklada z enačbami iz prejšnjega poglavja.

Potrebujemo še način za razvrščanje vektorjev v skupine. Vektor x pripada skupini H_s natanko tedaj, ko velja

$$(\beta_s - \beta_t)^\top x \geq 0, \;\; t = 1, \ldots, g.$$

Predpostavimo še, da je matrika opazovanj X, dimenzije $n \times (p+1)$, polnega ranga. Označimo z E_s množico identifikatorjev vrstic matrike X, ki pripadajo skupini H_s. Logaritemsko funkcijo verjetja v splošnem zapišemo kot

$$\log L(X, \beta) = \sum_{j=1}^{g} \sum_{i \in E_j} \log \left(\frac{1}{\sum_{t=1}^{g} \exp(\beta_t - \beta_j)^\top x_i} \right)$$

V nadaljevanju bomo ločeno obravnavali možnosti, kako so lahko podatki razporejeni glede na to, kateri skupini pripadajo. Podrobneje si bomo pogledali in narisali primere, ko sta skupini dve - tedaj imamo tri možnosti:

- podatki so popolnoma ločeni - popolna separacija
- podatki so popolnoma ločeni, vendar nekateri ležijo ravno na meji - nepopolna separacija
- podatki se prekrivajo - prekrivanje

Na podlagi tega lahko določimo, ali rešitev enačb verjetja obstaja in je enolična.

3.5.2. *Popolna separacija.* Popolna separacija v podatkih je prisotna, če obstaja tak vektor β, da za vse $i \in E_j$ in $j, t = 1, \ldots, g, \;\; j \neq t$ velja

$$(\beta_j - \beta_k)^\top x_i > 0.$$

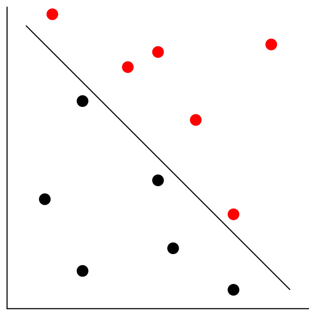
Torej obstaja vektor β, ki nam podatke popolnoma loči na skupine. V primeru dveh skupin se pogoj prevede na $\beta^\top x_i > 0, i \in E_1$ in $\beta^\top x_i < 0, i \in E_2$.

Izrek 3.1. *Če je v podatkih prisotna popolna separacija, cenilka največjega verjetja $\hat{\beta}$ ne obstaja in velja*

$$\max_{\beta} L(X, \beta) = 1.$$

Izkaže se, da svoj maksimum funkcija verjetja doseže, ko parameter pošljemo v neskončnost - torej končna rešitev in s tem cenilka največjega verjetja ne obstaja. Grafično si v primeru $g = 2$ popolno separacijo predstavljamo takole

kjer ena barva predstavlja podatke v prvi skupini, druga pa v drugi. Podatke lahko s premico razdelimo na dva dela, v katerem so samo tisti, ki ustrezajo bodisi Bernoullijevi enici bodisi Bernoullijevi ničli.



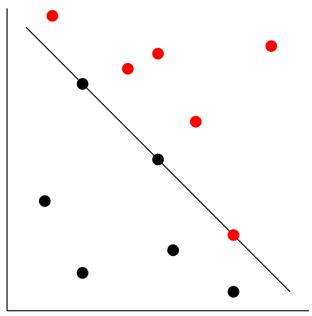
Slika 2. Popolna separacija

3.5.3. *Nepopolna separacija.* Nepopolna separacija v podatkih je prisotna, če obstaja tak vektor β, da za vse $i \in E_j$ in $j, t = 1, \ldots, g, \ j \neq t$ velja

$$(\beta_j - \beta_k)^\top x_i \geq 0,$$

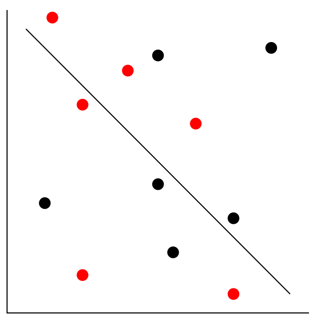
pri čemer velja enakost za vsaj eno trojico (i, j, t).

Kot je omenjeno že zgoraj, v tem primeru podatke lahko popolnoma ločimo, a nekatri ležijo popolnoma na meji. Lahko pa se zgodi, da je podatke možno ločiti na več načinov. V spodnjih slikah sta obravnavana oba primera - ko imamo eno samo ločnico in ko jih je možno najti več.



Slika 3. Nepopolna separacija, prvi primer

3.5.4. *Prekrivanje.* Če je v podatkih prisotno prekrivanje, torej ne padejo v nobeno od prejšnjih dveh kategorij, je sistem enačb verjetja rešljiv enolično.



Slika 4. Prekrivanje

Vidimo, da v tem primeru podatkov ni možno razdeliti na dve ločeni skupini

3.5.5. *Ugotavljanje separacije.*

3.6. **Kanonični modeli v splošnem.** Kot bomo spoznali v sledečem razdelku, spada logistična regresija med tako imenovane modele s "kanonično„ povezovalno funkcijo. Za vpeljavo tega ter prenekaterih ostalih pojmov pa potrebujemo nekaj dodatne teorije.

3.6.1. *Pomembnost kanoničnih povezvalnih funkcij.* Kot smo omenili že v uvodu povezovalna funkcija opisuje relacijo med pričakovano vrednostjo opazovane spremenljivke in desno stranjo našega modela, torej sistematično komponento modela. Eksponentno družino torej v splošnem sestavljajo porazdelitve, z gostotami oblike

$$f_Y(y;\theta,\phi) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right).$$

Enačbo logaritmiramo in dobimo

$$L(y;\theta) = \frac{y\theta - b(\theta)}{a(\phi)} + c(y;\phi),$$

njen odvod, torej funkcija zbira pa je

$$\frac{\partial}{\partial\theta}L(y;\theta) = \frac{y - b'(\theta)}{a(\phi)}.$$

O kanonični povezovalni funkciji govorimo, če velja $\theta = \eta$, torej je naravni parameter eksponentne družine ravno enak funkciji pričakovane vrednosti v modelu. Da uporabimo logit verjetnosti v logističnem modelu torej ni naključje - videli smo, da je $\textbf{logit}(p_i)$ enak parametru θ. Spodaj je navedenih še nekaj ostalih kanoničnih povezovalnih funkcij, njihovo uporabnost bomo spoznali v naslednjem razdelku.

Porazdelitev	$f(\mu)$	Uporaba
Normalna	$id(\mu)$	Linearni odgovori
Poissonova	$\log\mu$	Število pojavitev
Binomska	$\text{logit}\mu$	Binarni podatki
Gamma	$-\frac{1}{\mu}$	

Kot smo že v zgledu z logistično regresijo videli, potrebujemo odvode logaritemske funkcije verjetja po parametrih β. Uporabiti moramo torej verižno pravilo

$$\frac{\partial L}{\partial\beta_j} = \left(\frac{\partial L}{\partial\theta}\right)\left(\frac{\partial\theta}{\partial\mu}\right)\left(\frac{\partial\mu}{\partial\eta}\right)\left(\frac{\partial\eta}{\partial\beta_j}\right).$$

Lotimo se ga po korakih. Prvi člen smo že zgoraj izračunali kot $\frac{y-b'(\theta)}{a(\phi)}$. Z uporabo $(b')^{-1}(\mu) = \theta$ in pravila za odvajanje inverzne funkcije dobimo $\frac{\partial\theta}{\partial\mu} = \frac{1}{b''(\theta)} = \frac{a(\phi)}{\text{Var}(Y)}$, zadnji člen pa bo kar vedno enak x_{ij}. Tretji člen je odvisen od povezovalne funkcije in se mu bomo posvetili nekoliko kasneje. Sestavimo vse skupaj in dobimo

$$\frac{\partial L}{\partial\beta_j} = \frac{y-\mu}{var(Y)}\frac{\partial\mu}{\partial\eta}x_{ij}.$$

Opazimo: če imamo opravka s kanonično povezovalno funkcijo je $\eta = \theta$! Torej namesto odvajanja po prvem parametru, lahko μ odvajamo po θ in dobimo $\frac{\partial\mu}{\partial\theta} = b''(\theta)$ in se odvod še dodatno poenostavi v

$$\frac{\partial L}{\partial\beta_j} = \frac{y-\mu}{var(Y)}b''(\theta)x_{ij} = \frac{y-\mu}{a(\phi)}x_{ij}. \tag{23}$$

Za numerične metode bomo potrebovali še druge odvode, kjer pa nam pomaga informacijska enakost iz dokaza trditve 2.1.

Najprej izračunajmo pričakovano vrednost odvoda funkcije zbira

$$\begin{aligned} -\mathbb{E}(\frac{\partial^2 L}{\partial\beta_j\partial\beta_k}) &= \mathbb{E}((\frac{\partial L}{\partial\beta_j})(\frac{\partial L}{\partial\beta_k})) \\ &= \mathbb{E}(\frac{y-\mu}{var(Y)^2})(\frac{\partial\mu}{\partial\eta})^2 x_{ij}x_{ik} \\ &= \frac{1}{var(Y)}(\frac{\partial\eta}{\partial\mu})^2 x_{ij}x_{ik} \\ &= \frac{b''(\theta)}{a(\phi)}x_{ij}x_{ik}. \end{aligned}$$

Po drugi strani pa je običajen drugi odvod enak

$$\begin{aligned} \frac{\partial^2 L}{\partial\beta_j\partial\beta_k} &= \frac{\partial}{\beta_k}\{\left(\frac{\partial L}{\partial\theta}\right)\left(\frac{\partial\theta}{\partial\beta_j}\right)\} \\ &= \frac{\partial L}{\partial\theta}\left(\frac{\partial^2\theta}{\partial\beta_j\partial\beta_k}\right) + \left(\frac{\partial\theta}{\partial\beta_j}\right)\left(\frac{\partial^2 L}{\partial\theta^2}\frac{\partial\theta}{\partial\beta_k}\right) \\ &= 0 + \frac{\partial^2 L}{\partial\theta^2}x_{ij}x_{ik}, \end{aligned}$$

prej pa smo že dokazali da je

$$\frac{\partial^2 L}{\partial\theta^2} = -\frac{b''(\theta)}{a(\phi)}.$$

Sledi

$$\mathbb{E}\left(\frac{\partial^2 L}{\partial\theta^2}\right) = \frac{\partial^2 L}{\partial\theta^2}. \tag{24}$$

Uporabnost zgornjega rezultata pa nam bo dalo poglavje o numeričnih metodah.

3.6.2. *Poljubna povezovalna funkcija.* Za poljubno povezovalno funkcijo smo v zgornjem razdelku pokazali

$$\begin{aligned} \frac{\partial}{\partial\beta_j}L &= \frac{y-\mu}{\mathrm{Var}(Y)}\left(\frac{\partial\mu}{\partial\eta}\right)x_{ij} \\ -\mathbb{E}(\frac{\partial^2}{\partial\beta_j\partial\beta_k}) &= \frac{1}{\mathrm{Var}(Y)}\left(\frac{\partial\mu}{\partial\eta}\right)^2 x_{ij}x_{ik} \end{aligned}$$

3.7. **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 angleških besed ***prob****ability* *un**it***. V glavnem se od logistične regresije razlikuje v sistematičnem delu. Verjetnost pozitivnega izida torej po modelu predpostavljamo

$$p_i(\beta) = \Phi(\beta_0 + x_{i1}\beta_1 + \ldots + x_{ir}\beta_r), \tag{25}$$

kjer Φ predstavlja kumulativno porazdelitveno funkcijo standardne normalne slučajne spremenljivke. Ta seveda ni linearna (v nasprotju s prejšnjimi modeli), podana je kot

$$\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{\frac{-t^2}{2}}\,dt$$

Očitno v tem primeru ne delamo s kanonično povezovalno funkcijo, kot smo to počeli v prejšnjem poglavju.

3.7.1. *Ocenjevanje parametrov probit regresije.* Podobno kot v logističnem modelu, se bomo ocenjevanja parametrov lotili po metodi največjega verjetja.

Za sistematični del modela privzemimo binomsko porazdeljene slučajne spremenljivke s parametroma $\mathrm{Bin}(m_i, p_i(\beta))$, verjetnost pozitivnega izida pa izrazimo z

$$P(Y_i = y_i) = \binom{m_i}{y_i} p_i(\beta)^{y_i}(1 - p_i(\beta))^{m_i - y_i} = \binom{m_i}{y_i} (\Phi(x_i^\top \beta)^{y_i})(1 - \Phi(x_i^\top \beta))^{m_i - y_i}$$

Funkcijo verjetja, tako kot zgoraj izrazimo z gostotami posameznih komponent

$$F(\beta) = \prod_{i=1}^{n} \binom{m_i}{y_i} \Phi(x_i^\top \beta)^{y_i}(1 - \Phi(x_i^\top \beta))^{m_i - y_i},$$

kjer binomski simbol izpustimo zaradi enostavnejšega pisanja. Zgornjo enačbo logaritmiramo in dobimo

$$\log(F(\beta)) = L(\beta) = \sum_{i=1}^{n} \Big(y_i \log \Phi(x_i^\top \beta) + (m_i - y_i) \log(1 - \Phi(x_i^\top \beta)) \Big) \tag{26}$$

Enačbo odvajamo po parametru β, vendar se nam v tem primeru ne poenostavi kot z logistično funkcijo. Označimo s $\phi = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$, gostoto standardne normalne porazdelitve.

$$\begin{aligned}
\frac{\partial}{\partial \beta_j} L(\beta) &= \frac{\partial}{\partial \beta_j} \sum_{i=1}^{n} \Big(y_i \log \Phi(x_i^\top \beta) + (m_i - y_i) \log(1 - \Phi(x_i^\top \beta)) \Big) = \\
&= \sum_{i=1}^{n} \left(\frac{\phi(x_i^\top \beta)}{\Phi(x_{i^\top \beta})} x_{ij} - (m_i - y_i) \frac{\phi(x_i^\top \beta)}{1 - \Phi(x_{i^\top \beta})} x_{ij} \right) = \\
&= \sum_{i=1}^{n} \phi(x_i^\top \beta) x_{ij} \left(\frac{y_i}{\Phi(x_i^\top \beta)} - \frac{m_i - y_i}{1 - \Phi(x_i^\top \beta)} \right) = \\
&= \sum_{i=1}^{n} \phi(x_i^\top \beta) x_{ij} \left(\frac{y_i - m_i \Phi(x_i^\top \beta)}{\Phi(x_i^\top \beta)(1 - \Phi(x_i^\top \beta))} \right)
\end{aligned}$$

Sistem enačb verjetja se torej glasi

$$\sum_{i=1}^{n} \phi(x_i^\top \beta) x_{ij} \left(\frac{y_i - m_i \Phi(x_i^\top \beta)}{\Phi(x_i^\top \beta)(1 - \Phi(x_i^\top \beta))} \right) = 0, \quad j = 0, 1, \ldots, r \tag{27}$$

Te enačbe očitno niso rešljive analitično in se bomo zopet morali poslužiti numeričnih metod. Kot smo videli že pri izpeljavi enačb za logistično regresijo, bomo za to potrebovali še druge odvode.

$$\begin{aligned}
\frac{\partial^2}{\partial \beta_j \partial \beta_k} L(\beta) = \sum_{i=1}^{n} x_{ij} \Bigg(& \frac{\partial}{\partial \beta_k} [\phi(x_i^\top \beta)] \frac{y_i - m_i \Phi(x_i^\top \beta)}{\Phi(x_i^\top \beta)(1 - \Phi(x_i^\top \beta))} \\
& + \frac{\partial}{\partial \beta_k} \left(\frac{y_i - m_i \Phi(x_i^\top \beta)}{\Phi(x_i^\top \beta)(1 - \Phi(x_i^\top \beta))} \right) \phi(x_i^\top \beta) \Bigg)
\end{aligned}$$

Izračunajmo najprej prvi člen

$$\begin{aligned}\frac{\partial}{\partial \beta_k}\phi(x_i^\top\beta) &= \frac{\partial}{\partial \beta_k}\left(\frac{1}{\sqrt{2\pi}}e^{-\frac{(x_i^\top\beta)^2}{2}}\right)\\ &= \frac{-x_i^\top\beta}{\sqrt{2\pi}}e^{-\frac{(x_i^\top\beta)^2}{2}}x_{ik}\\ &= -x_i^\top\beta x_{ik}\phi(x_i^\top\beta).\end{aligned}$$

Drugi člen povzroča nekaj več preglavic.

(28)

$$\frac{\partial}{\partial\beta_k}\left(\frac{y_i - m_i\Phi(x_i^\top\beta)}{\Phi(x_i^\top\beta)(1-\Phi(x_i^\top\beta))}\right) = \frac{-m_i\phi(x_i^\top\beta)x_{ik}\Phi(x_i^\top\beta)(1-\Phi(x_i^\top\beta)) - (y_i - m_i)^{\partial}/_{\partial\beta_k(\Phi(x_i^\top\beta)(1-\Phi(x_i^\top\beta))}}{(\Phi(x_i^\top\beta)(1-\Phi(x_i^\top\beta)))^2}$$

Posebej izračunajmo še

$$\begin{aligned}\frac{\partial}{\partial\beta_k}(\Phi(x_i^\top\beta)(1-\Phi(x_i^\top\beta))) &= \phi_{x_i^\top\beta}x_{ik}(1-\Phi(x_i^\top\beta)) - \Phi(x_i^\top\beta)\phi(x_i^\top\beta)x_{ik}\\ &= \phi(x_i^\top\beta)x_{ik}(1-2\Phi(x_i^\top\beta))\end{aligned}$$

in vključimo to v enačbo (28)

$$\frac{-m_i\phi(x_i^\top\beta)\Phi(x_i^\top\beta)(1-\Phi(x_i^\top\beta))x_{ik} - (y_i - m_i\Phi(x_i^\top\beta))\phi(x_i^\top\beta)x_{ik}(1-2\Phi(x_i^\top\beta))}{(\Phi(x_i^\top\beta)(1-\Phi(x_i^\top\beta)))^2}$$

$$\phi(x_i^\top\beta)x_{ik}\frac{2y_i\Phi(x_i^\top\beta) - m_i\Phi(x_i^\top\beta)^2 - y_i}{(\Phi(x_i^\top\beta)(1-\Phi(x_i^\top\beta)))^2}$$

Vse skupaj povzemimo v

$$\begin{aligned}\frac{\partial^2}{\partial\beta_j\partial\beta_k}L(\beta) &= \sum_{i=1}^n x_{ij}\left(-x_i^\top\beta x_{ik}\phi(x_i^\top\beta)\frac{y_i - m_i}{\Phi(x_i^\top\beta)(1-\Phi(x_i^\top\beta))} + \phi(x_i^\top\beta)^2x_{ik}\frac{2y_i - m_i\Phi(x_i^\top\beta)^2 - y_i}{(\Phi(1-\Phi(x_i^\top\beta)))^2}\right)\\ &= \sum_{i=1}^n x_{ij}\frac{\phi(x_i^\top\beta)}{\Phi(x_i^\top\beta)(1-\Phi(x_i^\top\beta))}\left(\phi(x_i^\top\beta)\frac{2y_i - m_i\Phi(x_i^\top\beta)^2 - y_i}{(\Phi(1-\Phi(x_i^\top\beta)))} - (x_i^\top\beta)(y_i - m_i\Phi(x_i^\top\beta))\right)\end{aligned}$$

Zopet je koristno enačbe zapisati v matrični obliki. Za funkcijo zbira definirajmo vektor faktorjev

$$s = \begin{bmatrix}\phi(x_1^\top\beta)\frac{y_1 - m_1\Phi(x_1^\top\beta)}{\Phi(x_1^\top\beta)(1-\Phi(x_1^\top\beta))}\\ \vdots \\ \phi(x_n^\top\beta)\frac{y_n - m_n\Phi(x_n^\top\beta)}{\Phi(x_n^\top\beta)(1-\Phi(x_n^\top\beta))}\end{bmatrix}$$

in tako funkcijo zbira poenostavimo v

$$\frac{\partial}{\partial\beta}L(\beta) = \mathbf{X}^\top s.$$

Na podoben način se lotimo tudi Hessejeve matrike. Definiramo diagonalno matriko, kjer so na diagonali členi

$$h_i = \frac{\phi(x_i^\top\beta)}{\Phi(x_i^\top\beta)(1-\Phi(x_i^\top\beta))}\left(\phi(x_i^\top\beta)\frac{2y_i - m_i\Phi(x_i^\top\beta)^2 - y_i}{(\Phi(1-\Phi(x_i^\top\beta)))} - (x_i^\top\beta)(y_i - m_i\Phi(x_i^\top\beta))\right),$$

torej $\mathbf{H} = diag(h_1, \ldots, h_n)$ in Hessejevo matriko zapišemo v preglednejši obliki

$$\frac{\partial^2}{\partial\beta^2}L(\beta) = \mathbf{X}^\top\mathbf{H}\mathbf{X}$$

Končno, Newtonova metoda za iskanje ničel funkcije zbira z uporabo vseh zgornjih oznak

$$\beta_{i+1} = \beta_i - (\mathbf{X}^\top \mathbf{H} \mathbf{X})^{-1} \mathbf{X}^\top s. \tag{29}$$

4. 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.

4.1. **Newton – Raphsonova metoda.** Newton – Raphson (oziroma le Newtonova) metoda je bila v osnovi razvita za iskanje ničel funkcije. Spada v razred *navadnih iteracij*, torej metod za iterativno reševanje enačb $f(x) = 0$, ki jih prevedemo na $g(x) = x$, izberemo začetni približek x_0 in ponavljamo

$$x_{r+1} = g(x_r).$$

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:

$$L(\theta) \approx L(\theta_n) + \frac{\partial}{\partial \theta} L(\theta_n)(\theta - \theta_n) + \frac{1}{2}(\theta - \theta_n)^\top \frac{\partial^2}{\partial \theta^2} L(\theta_n)(\theta - \theta_n) \tag{30}$$

Maksimizirati želimo desno stran (30). To storimo tako, da gradient L enačimo z nič:

$$\frac{\partial}{\partial \theta} L(\theta_n) + \frac{\partial^2}{\partial \theta^2} L(\theta_n)(\theta - \theta_n) = 0$$

in izrazimo naslednji približek

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

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:

$$\begin{aligned} x_{n+1} &= x_n + p_n \\ \nabla^2 L(\theta_n) p_n &= -\nabla L(\theta_n) \end{aligned} \tag{31}$$

Zadnji vrstici v (31) 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.

4.1.1. *Potencialne težave Newtonove metode.* Newtonova metoda ima mnogo pozitivnih plati, vendar pa je iz določenih vidikov precej občutljiva. Morda najbolj očiten problem je slaba izbira začetne točke iteracije. Če je ta stacionarna točka obravnavane funkcije, nam metoda narekuje deljenje z 0, kar pa seveda nima smisla. Očiten primer bi bil iskanje ničle funkcije $f(x) = 1 + x^2$ z začetnim približkom $x_0 = 0$. Na pamet takoj vidimo, da so ničle v 1 in -1, če pa bi upoštevali iteracijo pa dobimo

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0 - \frac{1}{0}.$$

Enaka težava seveda nastopi, če v sledečih korakih dobimo stacionarno točko oziroma se ji približujemo in tako delimo z vedno manjšimi števili, kar pa vodi v vselej slabše približke.

Sicer redkeje, ampak lahko se zgodi, da se približki „zaciklajo“. Primer take funkcije je $f(x) = x^3 - 2x + 2$, če za začetni približek vzamemo 0. Tako v zaporednih korakih najprej dobimo $x_1 = 1$ in nato $x_2 = 0$, kar pa je seveda naša začetna točka. Obstajajo okolice teh dveh točk, ki vedno konvergirajo v ta dvojni cikel in ne h iskani ničli. V splošnem zna biti obnašanje takega zaporedja precej zapleteno, imenuje se Newtonov fraktal in se vanj tu ne bomo spuščali.

Naslednja težava pa lahko nastopi, če se odvod naše funkcije lokalno ne obnaša dovolj „lepo.“ Prvič, odvod v ničli morda ne obstaja. Enostaven primer tega je

$f(x) = \sqrt[3]{x}$. Izračunamo lahko

$$x_{n+1} = x_n - \frac{x_n^{1/3}}{\frac{1}{3}x_n^{1-1/3}} = -2x_n,$$

in vidimo, da za vsak začetni približek različen od nič metoda divergira. Splošneje, podoben rezultat dobimo za vsako funkcijo oblike $f(x) = |x|^\alpha, 0 < \alpha < \frac{1}{2}$, za $\alpha = \frac{1}{2}$ pa metoda sicer ne divergira, vendar se kot v prejšnjem primeru zacikla in ne pridemo do rešitve.

V zgoraj naštetih primerih torej Newtonova metoda ne konvergira h iskani ničli. Smiselno pa sledi vprašanje, katerim pogojem mora biti zadoščeno, da pa vednarle dobimo pravilno rešitev. To nam poda sledeči izrek:

Izrek 4.1. *Naj iteracijska funkcija g na intervalu $I = [\alpha - \delta, \alpha + \delta]$ zadošča Lipschitzovemu pogoju*

$$|g(x) - g(y)| \leq m|x - y|$$

za poljubna $x, y \in I$ in konstanto $0 \leq m \leq 1$. Potem za vsak $x_0 \in I$ zaporedje $x_{r+1} = g(x_r), r \geq 1$ konvergira k α. Poleg tega veljata tudi oceni

$$|x_r - \alpha| \leq m^r|x_0 - \alpha| \tag{32}$$

in

$$|x_{r+1} - \alpha| \leq \frac{m}{1-m}|x_r - x_{r-1}| \tag{33}$$

Dokaz. Označimo z $\varepsilon_r = x_r - \alpha$ napako približka x_r. Velja

$$|\varepsilon_r| = |x_r - \alpha| = |g(x_{r-1}) - g(\alpha)| \leq m|x_{r-1} - \alpha| = m\varepsilon_{r-1}.$$

Ta postopek nadaljujemo in sledi

$$|\varepsilon_r| \leq m|\varepsilon_{r-1}| \leq m^2|\varepsilon_{r-2}| \leq \ldots \leq m^r|\varepsilon_0|$$

Od tu vidimo (32), iz katere sledi da zaporedje x_r konvergira proti α.

Za drugo neenakost pa si oglejmo

$$|x_{r+1} - \alpha| \leq |x_{r+1} - x_{r+2}| + |x_{r+2} - x_{r+3}| + \ldots$$

Upoštevamo še

$$|x_{r+k} - x_{r+k+1}| = |g(x_{r+k-1}) - g(x_{r+k})| \leq m|x_{r+k-1} - x_{r+k}| \leq \ldots \leq m^k v|x_{r-1} - x_r|$$

in končno dobimo

$$|x_{r+1} - \alpha| \leq (m + m^2 + \cdots)|x_{r-1} - x_r| = \frac{m}{1-m}|x_{r-1} - x_r|$$

□

Dodatno informacijo o območju konvergence navadne iteracije nam dajo tudi vrednosti odvoda iteracijske funkcije, o čemer govori naslednji izrek.

Izrek 4.2. *Naj bo iteracijska funkcija zvezno odvedljiva v negibni točki α in naj velja $|g'(\alpha)| < 1$. Potem obstaja okolica I negibne točke, da za vsak začetni približek $x_0 \in I$ iteracija konvergira k α.*

Dokaz. Odvod je po predpostavki na neki okolici α strogo manjši od 1 in zaradi zveznosti obstajata $\delta > 0$ in konstanta $m < 1$, da je $|g'(x)| < m < 1 zax \in I$, kjer z I označimo $[\alpha - \delta, \alpha + \delta]$. Potem po Lagrangeovem izreku velja $|g(x) - g(y)| \leq |g'(\xi)||x - y|$, za poljubna $x, y \in I$. Ker je odvod na tem intervalu manjši od 1 sledi da je funkcija g Lipschitzova in zato po (32) konvergira. □

4.1.2. *Asimptotsko obnašanje in konvergenca.* Kot je pri numeričnih metodah to običajno, nas zanima njihovo obnašanje po več ponovitvah iteriacije. Pomembno je, kako hitro pridemo do rešitve saj želimo računanje čim manjkrat ponoviti in dobiti najboljši možen rezultat.

Definirajmo si *red konvergence.* Idejno je to število točnih decimalnih mest, ki jih pridobimo z vsakim korakom iteracije.

Definicija 4.3. Naj zaporedje (x_r) konvergira k α. Red konvergence je enak p, če obstajata taki števili C_1, C_2, da velja

$$C_1|x_r - \alpha|^p \leq |x_{r+1} - \alpha| \leq C_2|x_r - \alpha|^p.$$

Ekvivalentno: red konvergence je p, če obstaja $C > 0$ tak, da

$$\lim_{r\to\infty} \frac{|x_{r+1} - \alpha|}{|x_r - \alpha|^p} = C.$$

Red konvergence navadne iteracije je običajno precej enostavno določiti. Metodo nam daje naslenji izrek

Izrek 4.4. *Naj bo iteracijska funkcija g v okolici svoje fiksne točke p-krat zvezno odvedljiva in $|g'(\alpha)| \leq 1, g^{(k)}(\alpha) = 0$ za $k = 1, \ldots, p-1$ in $g^p(\alpha) \neq 0$. Potem ima iterativna metoda lokalno red konvergence p.*

Dokaz. Razvijmo g v Taylorjevo vrsto okrog α

$$x_{r+1} = g(x_r) = \alpha + \frac{1}{p!}(x_r - \alpha)^p g^{(p)}(\xi),$$

kjer ξ leži med x_r in α. Ocenimo odvod navzgor in navzdol ter s tem dobimo konstanti C_1, C_2 iz 4.3. □

Pa poskusimo sedaj določiti red konvergence Newtonove metode. Označimo

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Odvajamo in dobimo

$$g'(x) = \frac{f(x)f''(x)}{f'^2(x)}.$$

Vidimo, da je potrebno ločiti dva primera:

- Če je $g'(\alpha) = 0$, torej je α enostavna ničla je konvergenca vsaj kvadratična. Z nadaljnjim računom dobimo

  $$g''(\alpha) = \frac{f''(\alpha)}{f'(\alpha)},$$

  in vidimo, da je pri $f''(\alpha) \neq 0$ konvergenca kvadratična. Sicer postopek nadaljujemo, dokler ne najdemo prvega odvoda z vrednostjo v α 0.
- Če je α m-kratna ničla pa se da pokazati

  $$\lim_{x\to\alpha} g'(x) = 1 - \frac{1}{m},$$

  od koder sledi linearna konvergenca.

4.1.3. *Newton–Raphsonova metoda v višjih dimenzijah.* Newtonovo metodo se da enostavno posplošiti za iskanje ničel vektorskih funkcij. Recimo, da imamo funkcijo $F : \mathbb{R}^{\ltimes} \to \mathbb{R}^{\ltimes}$ in iščemo tak vektor $x^* \in \mathbb{R}^{\ltimes}$, za katerega bo $F(x^*) = (f_1(x_1^*), \ldots, f_n(x_n^*)) = (0, \ldots, 0)$.

Podobno kot zgoraj tvorimo zaporedje

$$x^{(r+1)} = x^{(r)} - JF(x^{(r)})^{-1} F(x^{(r)}),\ r = 0, 1, \ldots$$

kjer smo z JF označili Jacobijevo matriko. V praksi njenega odvoda ne računamo, temveč uvedemo premike in rešujemo sistem enačb

$$\begin{aligned} JF(x^{(r)})h &= -F(x^{(r)}), \\ x^{(r+1)} &= x^{(r)} + h,\ r = 0, 1, \ldots \end{aligned}$$

od koder izračunamo vektor premikov $h \in \mathbb{R}^{\ltimes}$ in nato posodobimo prejšnji približek.

Izpeljava je podobna tisti v eni dimenziji, preko razvoja v Taylorjevo vrsto. Recimo, da so vse komponente funkcije F dvakrat zvezno odvedljive v okolici rešitve. Potem razvijemo

$$f_i(x+h) = f_i(x) + \sum_{k=1}^{n} \frac{\partial f_i(x)}{\partial x_k} * h_k + \ldots,\ i = 1, \ldots, n$$

Želimo, da bo $f_i(x+h) = 0$ za vsak i, zanemarimo člene od kvadratnega dalje in rešujemo sistem za premike

$$\begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \ldots & \frac{\partial f_1(x)}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n(x)}{\partial x_1} & \ldots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix} \begin{bmatrix} h_1 \\ \vdots \\ h_n \end{bmatrix} = - \begin{bmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{bmatrix}.$$

V okolici rešitve enačbe ima metoda kvadratično konvergenco, težava pa je v iskanju začetnega približka.

4.2. **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.

Ponovimo najprej nekaj terminologije. Funkcija zbira je gradient logaritemske funkcije verjetja po ocenjevanem parametru. Informacijska (oziroma Fisherjeva Informacijska) matrika (angl. *(Fisher) information matrix*) je definirana kot

$$\mathrm{FI}(\theta) = \mathbb{E}\left(\left(\frac{\partial}{\partial \theta} L(\theta) \right) \left(\frac{\partial}{\partial \theta} L(\theta) \right)^{\top} \right)$$

Fisher scoring algoritem je po zgornjih oznakah

$$\theta n + 1 = \theta_n - FI(\theta)^{-1} \nabla L(\theta) \tag{34}$$

4.2.1. *Ujemanje Newton-Raphson in Fisher's scoring za kanonične povezovalne funkcije.* Spomnimo se zaključkov poglavja 3.6.1. Tam smo dokazali, da za modele s kanonično povezovalno funkcijo velja $\mathbb{E}(\frac{\partial^2}{\partial \theta^2} L) = \frac{\partial^2}{\partial \theta^2}$, po informacijski enakosti pa velja

$$\mathrm{FI}(\theta) = \mathbb{E}\left((\nabla L)(\nabla L)^{\top} \right) = \mathbb{E}(\frac{\partial^2}{\partial \theta^2} L) = \frac{\partial^2}{\partial \theta^2} L \to \mathrm{FI}(\theta) = \frac{\partial^2}{\partial \theta^2} L,$$

torej Fisherjeva informacija je za kanonične modele enaka Hessejevi matriki logaritemske funkcije verjetja! Poleg tega pa velja še

$$\begin{aligned}
\mathrm{FI}(\theta) &= \mathrm{E}[(\nabla L(\theta))(\nabla L(\theta))^\top] \\
&= \mathrm{E}[(\nabla L(\theta) - \mathrm{E}[\nabla L(\theta)])\,(\nabla L(\theta) - \mathrm{E}[\nabla L(\theta)])^\top] \\
&= \mathrm{Var}[\nabla L(\theta)],
\end{aligned} \tag{35}$$

variančno kovariančne matrike pa so pozitivno semidefinitne, kar pomeni da imamo konstanten algoritem.

Povzemimo; za kanonične povezovalne funkcije smo dokazali enakost med Fisherjevo informacijo in Hessejevo matriko. S tem se v enem koraku izognemo računanju matrike drugih odvodov in pridobimo pozitivno semidefinitno matriko v imenovalcu. Koristi uporabe kanoničnih povezovalnih funkcij so toraj očitne.

4.2.2. *Fisher's scoring v logističnem modelu.* Poglejmo za trenutek nazaj v poglavje 3.4.1, natančneje k enačbam (16),(19) in (20). Iz prejšnjega razdelka vemo tudi, da velja

$$\mathrm{FI}(\theta) = \mathbb{E}[-\nabla^2 L(\theta)] \overset{(20)}{=} X^\top v(\theta) X = -\nabla^2 L(\theta),$$

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

$$\begin{aligned}
\hat{\theta}_{i+1} &= \hat{\theta}_i - (\nabla^2 L(\hat{\theta}_i))^{-1} \nabla L(\hat{\theta}_i) \\
&= \hat{\theta}_i + (\mathbf{X}^\top v(\hat{\theta}_i)\mathbf{X})^{-1} \mathbf{X}^\top (y - \mu(\hat{\theta}_i))
\end{aligned} \tag{36}$$

5. Primeri

5.1. **Ocenjevanje parametrov v logističnem modelu.** V praktično usmerjenem delu 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.

5.1.1. *Algoritem za logistični model in rezultati.* Povežimo vso izpeljano teorijo v algoritem za ocenjevanje parametrov modela oblike

$$\mathrm{logit}(p_i) = X\beta,$$

torej kanoničnega logističnega modela.

```
function LogitModel(iteracije, X, Y, β_zacetni, ε)
  p = exp(X^⊤ β_zacetni) / (1+exp(X^⊤ β_zacetni))
  V = p(1 − p)
  Score = X^⊤(Y − p)
  Info = X^⊤ V X
  Reši sistem na h: Info ∗ h = Score
  β_star = β_zacetni
  β_nov = β_star + h
  while i ≤ iteracije do
    if β_nov − β_star ≥ ε then
      p = exp(X^⊤ β_nov) / (1+exp(X^⊤ β_nov))
      V = p(1 − p)
      Score = X^⊤(Y − p)
      Info = X^⊤ V X
      Razreši na h:
      Info ∗ h = Score
      β_star = β_nov
      β_nov = β_star + h
    else
      Dosegli smo želeno natančnost v zadostnem številu korakov
      return β_nov
    end if
  end while
```

Algoritma pa ne bomo samo navedli, preizkusili ga bomo na konkretnih podatkih. Zanimalo nas bo, kako sta temperatura in pritisk vplivala na odpoved tesnil na vesoljskih misijah preden je v veljavo stopil *Challenger*. Najprej bomo pogledali le enodimenzionalne ocene, potem pa vse skupaj združili.

Spodaj je tabela; tej podatki so shranjeni v matriki X, le da je tam dodan prvi stolpec enic - za izračun β_0, ki nastopa brez pojasnjevalne spremenljivke. *POLET* označuje zaporedno številko poleta, *TEMPERATURA* in *PRITISK* sta temperatura in pritisk v okolici, *TESNILO* pa je binarna spremenljivka - 1 označuje, da je tesnilo popustilo, 0 pa da je pogoje vzdržalo.

POLET	TEMPERATURA	PRITISK	TESNILO
1	66	50	0
2	70	50	1
3	69	50	0
4	68	50	0
5	67	50	0
6	72	50	0
7	73	100	0
8	70	100	0
9	57	200	1
10	63	200	1
11	70	200	1
12	78	200	0
13	67	200	0
14	53	200	1
15	67	200	0
16	75	200	0
17	70	200	0
18	81	200	0
19	76	200	0
20	79	200	0
21	75	200	1
22	76	200	0
23	58	200	1

TABELA 1. Podatki uporabljeni v analizi

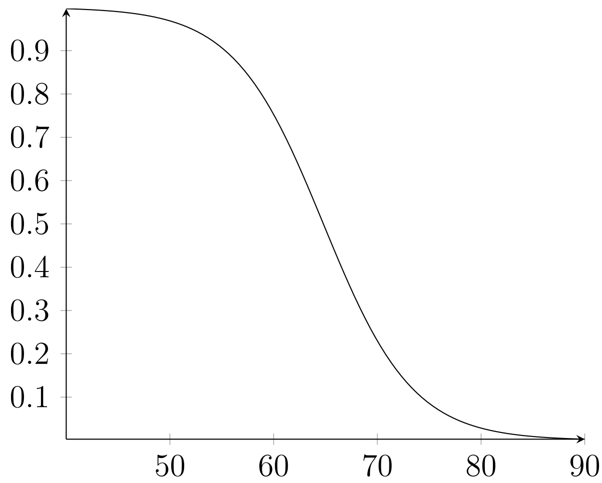
Poženimo algoritem najprej le na podatkih o temperaturi. Za začetno vrednost približka β vzemimo 0, natančnost ϵ pa si predipšemo na 0,001. Ker imamo le eno pojasnjevalno spremenljivko uporabljamo torej model oblike

$$\text{logit}(p_i) = \beta_0 + x_i\beta_1$$

Za izračun približka znotraj 1 tisočinke, z začetno vrednostjo $\beta = (0, 0)$, smo potrebovali le nekaj korakov. Vstavimo v invertirano logit transformacijo in dobimo

$$p = \frac{e^{15.04290 - 0.23216x}}{1 + e^{15.04290 - 0.23216x}},$$

torej $\beta_0 = 15.04290$ in $\beta_1 = -0.23216$. Če graf narišemo, dobimo

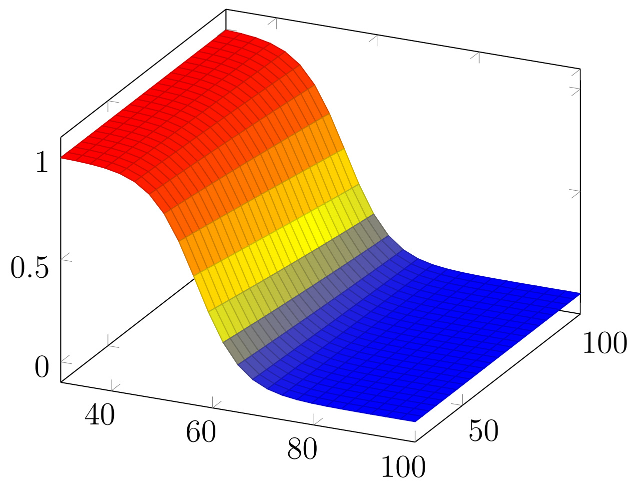


SLIKA 5. Izračunane verjetnosti z eno pojasnjevalno spremenljivko

Vidimo da je sigmoida v tem primeru obrnjena drugače kot na sliki 1 - tak rezultat nam da negativen predznak parametra β_1. Sklepamo lahko torej da pri višjih temperaturah tesnila redkeje odpovejo. Za računanje smo si predpisali natančnost $\epsilon = 0,001$, razlika med dvema zaporednima približkoma ne sme presegati te vrednosti. Z uporabo našega algoritma (ter vgrajenih funkcij za invertiranje matrik in reševanje sistemov enačb) smo do rešitve prišli v vsega štirih iteracijah.

Enako rešimo še primer z dvema pojasnjevalnima spremenljivkama, pritisku in temperaturi. Rezultat je ploskev, saj nas zanima odpoved pri vsakem možnem paru temperature in pritiska. Izračunani parametri znašajo $\beta_0 = 13.29236$, $\beta_1 = -0.22867$, $\beta_2 = -0.01040$. Prikazan je graf funkcije

$$\frac{e^{13.29236-0.22867*x-0.01040*y}}{1+e^{(13.29236-0.22867*x-0.01040*y)}}.$$



SLIKA 6. Izračunane verjetnosti z dvema pojasnjevalnima spremenljivkama

Zopet smo algoritem pognali na enakih začetnih vrednostih približka in zahtevane natančnosti, rešitev pa dobili po štirih iteracijah.

5.2. **Ocenjevanje parametrov v probit modelu.** Sedaj ponovimo postopek še s probit modelom. Zaradi neoptimizirane numerične metode pričakujemo večje število iteracij za doseženo želeno natančnost.

5.2.1. *Algoritem za probit model.* Najprej postavimo algoritem. Ta se idejno sicer ne bo bistveno razlikoval od tistega uporabljenega v prejšnjem odseku, vendar pa je očitno precej zapletenejši za računanje. Posebej smo označili *koeficiente* pri Hessejevi matriki in funkciji zbira - to so le deli dejanske formule, izračunani posebej za večjo preglednost.

---

function ProbitModel(iteracije, X, Y, $\beta_{zacetni}$, ϵ)

$p = \Phi(X^\top \beta_{zacetni})$
$Score_{koef} = \frac{Y-p}{p*(1-p)} * \phi(X^\top \beta_{zacetni})$
$Score = X^\top Score_{koef}$
$Hess_{koef} = \frac{\phi(X^\top \beta_{zacetni})}{p*(1-p)} * \Big(\phi(X^\top \beta_{zacetni})\frac{2*p*Y-p^2-Y}{p*(1-p)} - X(Y-p)\Big)$
$H = X^\top Hess_{koef} X$
Reši sistem na h: $H * h = Score$
$\beta_{star} = \beta_{zacetni}$
$\beta_{nov} = \beta_{star} - h$
while $i \leq$ iteracije **do**
  if $\beta_{nov} - \beta_{star} \geq \epsilon$ **then**
    $p = \Phi(X^\top \beta_{nov})$
    $Score_{koef} = \frac{Y-p}{p*(1-p)} * \phi(X^\top \beta_{nov})$
    $Score = X^\top Score_{koef}$
    $Hess_{koef} = \frac{\phi(X^\top \beta_{nov})}{p*(1-p)} * \Big(\phi(X^\top \beta_{nov})\frac{2*p*Y-p^2-Y}{p*(1-p)} - X(Y-p)\Big)$
    $H = X^\top Hess_{koef} X$
    Razreši na h:
    $H * h = Score$
    $\beta_{star} = \beta_{nov}$
    $\beta_{nov} = \beta_{star} - h$
  else
    Dosegli smo želeno natančnost v zadostnem številu korakov
    return β_{nov}
  end if
end while

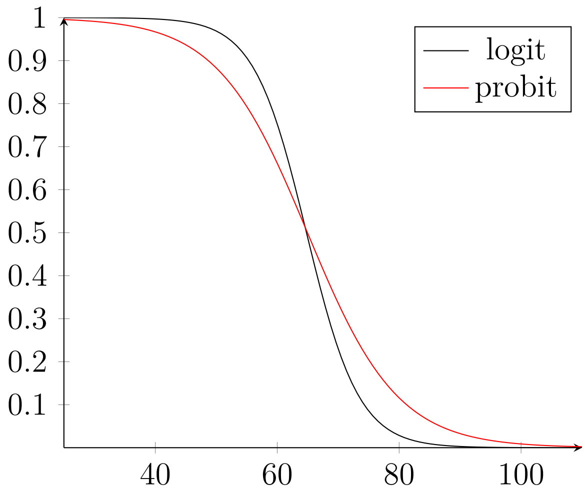
Ponovimo postopek in za oceno le z eno pojasnjevalno spremenljivko, po vsega petih iteracijah dobimo $\beta_0 = 8.77495,\ \beta_1 = -0.13510$ in funkcijo

$$\frac{e^{8.77495--0.13510x}}{1+e^{8.77495-0.13510x}}.$$

Za vključeni obe spremenljivki pa $\beta_0 = 8.08004,\ \beta_1 = -0.13774,\ \beta_2 = -0.006014$

Torej je predvidevanje, da bomo potrebovali bistveno več iteracij napačna, saj je bila potrebna le ena več kot pri logistični regresiji. Ugibamo, da do tega pride zaradi majhne količine testnih podatkov in bi se pri večjih podatkovnih bazah razlike primerno povečale.

5.3. **Primerjava logit in probit modela.** Zanimivejše kot le naštevanje parametrov, pa je primerjanje dobljenih rezultatov.



Slika 7. Izračunane verjetnosti z eno pojasnjevalno spremenljivko

Z rdečo je na zgornjem grafu narisana sigmoida, dobljena iz probit modela, s črno pa iz logističnega. Kljub temu, da se število iteracij za njun izračun ne razlikuje bistvemo, bi v tem primeru raje izbrali logistični model. Razlog tiči v hitrosti padanja, ki je višja pri slednjem, kar nam koristi pri nadaljnji analizi.

Slovar strokovnih izrazov

generalized linear model posplošeni linearni model
score function funkcija zbira
Fisher information matrix Fisherjeva informacijska matrika
link function povezovalna funkcija
maximum likelihood estimator (MLE) cenilka največjega verjetja
exponential family eksponentna družina

Literatura

[1] Alan Agresti. *An introduction to categorical data analysis*. John Wiley & Sons, 2007.
[2] Adelin Albert and John A Anderson. On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, 71(1):1–10, 1984.
[3] Erik B. Erhard. Logistic regression and Newton-Raphson. https://statacumen.com/teach/SC1/SC1_11_LogisticRegression.pdf.
[4] Kenneth Lange. *Numerical analysis for statisticians*. Springer Science & Business Media, 2010.
[5] P. McCullagh and J.A. Nelder. *Generalized linear models*. Springer US, 1989.
[6] Bor Plestenjak. *Numerične metode*. https://www.fmf.uni-lj.si/~kozak/PedagoskoDelo/Gradiva/NumericneMetodeI_praktiki/Skripta/BorPlestenjakKnjigaNM.pdf, 2010.
[7] Germán Rodríguez. Lecture notes on Generalized Linear Models. https://data.princeton.edu/wws509/notes/, 2007.