

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 | 9 |
| 3.5. Obstoje rešitve enačb verjetja v logističnem modelu | 12 |
| 3.6. Kanonični modeli v splošnem | 14 |
| 3.7. Probit regresija | 17 |
| 4. Numerične metode | 19 |
| 4.1. Newton – Raphsonova metoda | 19 |
| 4.2. Fisherjeva zbirna metoda | 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 nalogi smo se spustili z namenom razumeti postopek ocenjevanja parametrov v posplošenih linearnih modelih. Za uvod si postavimo teoretične temelje z eksponentno družino in izpeljemo nekaj lastnosti. Nato definiramo posplošene linearne modele in si ogledamo nekaj najpomembnejših primerov ter predstavimo metode za ocenjevanje parametrov, s poudarkom na metodi največjega verjetja. Izpeljemo enačbe verjetja za logistični model, rezultate nato komentiramo v luči eksponentne družine in jih posplošimo na vse porazdelitve, ki ji pripadajo. Izpeljemo tudi enačbe verjetja v probit modelu in opazimo prednosti uporabe kanoničnih povezovalnih funkcij. V drugem delu naloge se posvetimo numeričnim metodam. Izpeljemo Newtonovo metodo in komentiramo težave, ki lahko nastopijo z njeno uporabo. Izpeljemo tudi Fisherjevo zbirno metodo in dokažemo, da se ob uporabi modela s kanonično povezovalno funkcijo ujema z Newtonovo metodo.

Izpeljano teorijo v zadnjem delu povežemo v praktičnem primeru. Primerjamo rezultate dobljene s probit in logističnim modelom in komentiramo morebitne razlike.

Iterative numerical methods in generalized linear models

ABSTRACT

We have conducted the following research in order to understand the process of parameter estimation in generalized linear models. In the beginning we lay theoretical foundations with exponential family and derive some of its properties. Then we define generalized linear models, inspect some more important cases and define multiple methods for parameter estimation, taking a closer look at the maximum likelihood method. We go on to derive maximum likelihood equations in the logistic model and generalize the result for the exponential family. As an alternative, we derive the same equations also for the probit model and comment on the advantages of using canonical link functions. The second part focuses on numerical methods. We derive the Newton method and comment on its possible issues. We also define the Fisher's scoring algorithm and prove the equivalence of the methods for canonical distribution models.

Theory is then put to work in the last part of the research. We compare probit and logit models and comment on the differences between the two.

Math. Subj. Class. (2010): 62J12

Ključne besede: eksponentna družina, kanonični parameter, cenilka največjega verjetja, logistični model, Fisherjeva zbirna metoda

Keywords: exponential family, canonical parameter, maximum likelihood estimator, logistic model, Fisher's scoring algorithm

1. UVOD

Kaj natanko se zgodi v računalniškem ozadju, ko v programskem paketu požemo ukaz za oceno parametrov po metodi največjega verjetja? Kateri algoritmi se uporabljajo in ali lahko kdaj naletimo na težave? V sodobnem svetu neomejene količine podatkov so stabilni in učinkoviti postopki za njihovo obdelavo izjemnega pomena. Potrebno je najti ravnovesje med kompleksnostjo algoritma in računsko zahtevnostjo. S tem in podobnimi vprašanji se srečamo v analizi numeričnih metod za posplošene linearne modele. V delu najprej natančno postavimo teoretične temelje, začevši z eksponentno družino in njenimi lastnostmi in zgledi. Nato predstavimo linearno regresijo kot alternativno metodo za računanje ocen parametrov, podrobneje pa izpeljemo enačbe verjetja v logističnem in probit modelu.

V sledečih poglavjih izpeljemo numerične metode, po korakih izpeljemo Newtonovo metodo in komentiramo njene prednosti in slabosti ter jo nadgradimo v Fisherjevo zbirno metodo, ki jo tudi implementiramo in rezultate predstavimo v zadnjem poglavju.

2. EKSPONENTNA DRUŽINA

Za uvod v nalogo 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 njena gostota glede na neko σ -končno mero oblike

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

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

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

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

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

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:*

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

Dokaz. Za dokaz prve enakosti si pogledjmo

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

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

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

Za drugo pa si oglejmo

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

Spet uporabimo prej izpeljane zveze in dobimo

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

in po zgoraj dokazani enakosti za funkcijo zbira torej velja

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

□

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

Pričakovano vrednost kvadrata funkcije zbira v splošnem imenujemo *Fisherjeva informacija*, $\text{FI}(\theta) = \mathbb{E}((V(Y))^2)$, 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

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

in iz trditve 2.1 sledijo zaključki

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

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

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

od koder direktno sledi

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

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 $Y = (Y_1, Y_2, \dots, Y_n)$, 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 (koliko zabav je obiskal študent v preteklem mesecu), ali pa zvezne podatke. V slednjem primeru običajno privzamemo normalno porazdelitev (ali 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

$$\eta_i = \beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + \dots + x_{ir}\beta_r,$$

oziroma

$$\eta = X\beta$$

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

$$g(\mu) = \beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + \dots + x_{ir}\beta_r$$

Najenostavnejša taka funkcija je kar identiteta, torej $g(\mu) = \mu$. Ta nam 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 karaktéristiko c proučevane porazdelitve je funkcija vzorca $T = T(X_1, \dots, X_n)$, s katero ocenjujemo c . Ta cenilka je *nepriistranska*, če za porazdelitev vzorca F velja $\mathbb{E}(T(X_1, \dots, X_n)) = c(F)$. Imejmo sedaj zaporedje cenilk za vzorce velikosti $n = 1, 2, \dots$. To zaporedje je *dosledno*, če v verjetnosti konvergira h konstanti $c(F)$.

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

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 karaktéristiko proučevane slučajne spremenljivke $c(X)$ mogoče izraziti kot funkcijo momentov, t.j. če v danem modelu ti momenti obstajajo,

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

za neko funkcijo g , potem $c(X)$ ocenjujemo s cenilko $g(\hat{m}_1, \dots, \hat{m}_r)$, 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 $\theta = (\theta_1, \dots, \theta_r)$. Privzemimo, da imajo vse proučevane porazdelitve gostote oziroma verjetnostne funkcije oblike

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

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

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

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

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

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

Računanje si lahko precej poenostavimo, če obe strani zgornje enačbe logaritmiramo

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

Zgornji funkciji rečemo logaritemska funkcija verjetja, njene stacionarne točke pa bodo kandidati za cenilko največjega verjetja. Ker je logaritem naraščajoča funkcija, bodo ekstrema F in $\log(F)$ sovpadali. Rešiti moramo torej sistem enačb

$$(6) \quad \frac{\partial}{\partial \theta_j}(\ell(\theta)) = 0, j = 1, \dots, r,$$

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 sistema (6) enolična. V splošnem te 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 posplošnega linearnega modela. Enostavno jo lahko zapišemo kot: $Y = X\beta + \varepsilon$ kjer je Y proučevan slučajni vektor dimenzije n , $X \in \mathbb{R}^{n \times (r+1)}$ je matrika pojasnjevalnih slučajnih spremenljivk, β je vektor koeficientov dimenzije $r+1$, 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 - 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

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

kar pa velja natanko tedaj ko je $X^T(y - X\hat{\beta}) = 0$ oziroma

$$(X^T X)\hat{\beta} = X^T y.$$

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

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

3.4. Logistična regresija. Logistična regresija se uporablja za določanje deležev oziroma računanje verjetnosti. V poštrev 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: $Y_i \sim B(m_i, p_i)$. Ta pravi, da je

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

Pričakovana vrednost in varianca sta odvisni le od p_i , in sta enaki $E(Y_i) = m_i p_i$ in $\text{Var}(Y_i) = m_i p_i (1 - p_i)$. 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 *obete* – verjetnost dogodka proti verjetnosti, da se ne bo zgodil. Če je p_i enak $\frac{1}{2}$, bo enak 1. Vidimo, da so obeti vedno pozitivni in niso omejeni navzgor. V naslednjem koraku pa pogledimo njihov logaritem 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:

$$(7) \quad p_i = \text{logit}^{-1}(\eta_i) = \frac{\exp \eta_i}{1 + \exp \eta_i},$$

kjer je

$$\eta_i = \exp x_i^\top \beta.$$

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 $\frac{\partial}{\partial x_j} = \beta_j p_i (1 - p_i)$. 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 algoritmi. Kot v vsakem posplošenem linearnem modelu tudi v tem predpostavimo neodvisnost komponent slučajnega vektorja Y zato

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

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

$$\begin{aligned}
 \log(P(Y)) = \ell(\beta) &= \log\left\{\prod_{i=1}^n p_i^{y_i} (1 - p_i)^{m_i - y_i}\right\} \\
 &= \sum_{i=1}^n \{y_i \log p_i + (m_i - y_i) \log(1 - p_i)\} \\
 (8) \qquad &= \sum_{i=1}^n \{m_i \log(1 - p_i) + y_i \log\left(\frac{p_i}{1 - p_i}\right)\}
 \end{aligned}$$

Po predpostavki logističnega modela je

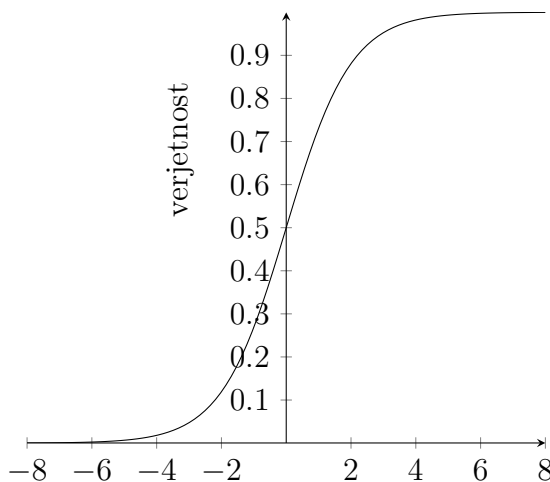
$$\text{logit}(p_i) = \log\left(\frac{p_i}{1 - p_i}\right) = x_{i0}\beta_0 + x_{i1}\beta_1 + \dots + x_{ir}\beta_r = x_i^\top \beta,$$

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

Sigmoida je omejena, povsod definirana odvedljiva realna funkcija, ki ima natanko en prevoj in povsod pozitiven prvi odvod. Prikladno jo je definirati 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 7 upoštevamo v logaritemski funkciji verjetja dobimo

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

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

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

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

$$(10) \quad \frac{\partial}{\partial \beta_j} \ell(\beta) = \sum_{i=1}^n (x_{ij}(y_i - m_i p_i(\beta))) = \sum_{i=1}^n (x_{ij}(y_i - \mu_i)), \quad j = 0, 1, \dots, r,$$

kjer smo upoštevali

$$\frac{\partial}{\partial \beta_j} (x_i^\top \beta) = \frac{\partial}{\partial \beta_j} (\beta_0 + x_{i1}\beta_1 + \dots x_{ir}\beta_r) = x_{ij}$$

ter

$$(11) \quad \begin{aligned} \frac{\partial}{\partial \beta_j} \log(1 + \exp(x_i^\top \beta)) &= \frac{\frac{\partial}{\partial \beta_j} \exp(x_i^\top \beta)}{1 + \exp(x_i^\top \beta)} \\ &= \frac{\exp(x_i^\top \beta)}{1 + \exp(x_i^\top \beta)} \frac{\partial}{\partial \beta_j} (x_i^\top \beta) \\ &= p_i(\beta) x_{ij} \end{aligned}$$

in

$$m_i p_i(\beta) = \mu_i(\beta).$$

Funkcijo zbira kot produkt matrik zapišemo

$$(12) \quad \frac{\partial}{\partial \beta} \ell(\beta) = X^\top (y - \mu(\beta))$$

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

$$(13) \quad \frac{\partial^2}{\partial \beta_j \partial \beta_k} \ell(\beta) = - \sum_{i=1}^n (x_{ij} x_{ik} m_i p_i(\beta) (1 - p_i(\beta)))$$

$$(14) \quad = - \sum_{i=1}^n (x_{ij} x_{ik} v_i(\beta)), \quad j, k = 0, 1, \dots, r$$

kjer smo v drugi vrstici upoštevali $\text{Var}(Y_i) = v_i(\beta) = m_i p_i (1 - p_i)$.

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

$$(15) \quad \frac{\partial}{\partial \beta} \ell(\beta) = -X^\top v(\beta) X,$$

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. V prejšnjem odseku smo izpeljali enačbe verjetja za logistično regresijo in pripomnili, 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. Sledeči podrazdelek temelji na članku [2].

Zopet bomo vpeljali nekaj novih oznak. Imejmo n neodvisnih opazovanj vektorja dimenzije p in določimo (x, H) , kjer je $x^\top = (x_0, \dots, x_r)$, $x_0 \equiv 1$, H pa je spremenljivka, ki zavzame vrednosti H_1, \dots, 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.

Potrebujemo način za razvrščanje vektorjev v skupine. Vektor x pripada skupini H_1 natanko tedaj, ko velja

$$\langle x, \beta \rangle \geq 0$$

in skupini H_2 , če velja

$$\langle x, \beta \rangle \leq 0.$$

Označimo še $E_1 = \{i \mid y_i = 1\}$ in $E_2 = \{i \mid y_i = 0\}$, torej indekse vrstic kjer je element vektorja Y enak nič oziroma 1.

Logaritemsko funkcijo verjetja v splošnem zapišemo kot

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

v našem primeru pa se poenostavi na

$$\ell(X, \beta) = \sum_{i \in E_1} \log \left(\frac{1}{1 + \exp(-\langle x_i, \beta \rangle)} \right) + \sum_{i \in E_2} \log \left(\frac{1}{\exp(\langle x_i, \beta \rangle) + 1} \right)$$

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 ali je enolična.

3.5.1. *Popolna separacija.* Popolna separacija v podatkih je prisotna, če

$$\exists \beta : \forall i \in E_1 : \langle x_i, \beta \rangle > 0, \forall i \in E_2 : \langle x_i, \beta \rangle < 0$$

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

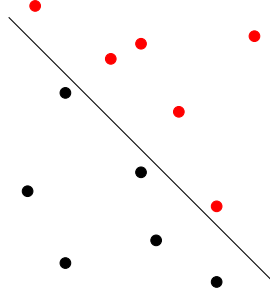
$$\sup_{\beta} \ell(X, \beta) = 0.$$

Dokaz. Če tak β obstaja, potem je

$$\log F(X, \beta) = \sum_{i \in E_1} \log \left(\frac{1}{1 + \exp(-\underbrace{\langle x_i, \beta \rangle}_{>0})} \right) + \sum_{i \in E_2} \log \left(\frac{1}{\exp(\underbrace{\langle x_i, \beta \rangle}_{<0}) + 1} \right).$$

V log-verjetje vstavljamo $\lambda \cdot \beta$ in ko pošljemo λ proti neskončnosti, gre zgornji izraz proti 0. \square

Grafično si popolno separacijo predstavljamo takole



SLIKA 2. Popolna separacija

kjer so s črno predstavljeni podatki v eni skupini, z rdečo pa v drugi.

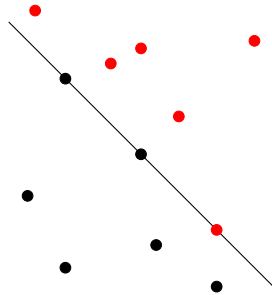
3.5.2. *Nepopolna separacija.* Nepopolna separacija v podatkih je prisotna, če obstaja tak vektor β , da velja

$$(16) \quad \forall i \in E_1 : \langle x_i, \beta \rangle \geq 0, \forall i \in E_2 : \langle x_i, \beta \rangle \leq 0,$$

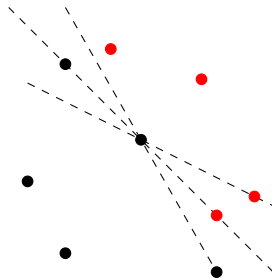
pri čemer velja enakost za vsaj en i .

Kot je omenjeno že zgoraj, v tem primeru podatke lahko popolnoma ločimo, a nekatere 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.

Poglejmo si primer, ko je $g = 2$. Označimo s $Q(\beta)$ tiste podatke, ki zadoščajo enakosti v (16) in recimo, da je takih $r \neq 0$. Naj bo X^q matrika teh opažanj dimenzij $r \times (p + 1)$. Velja $X^q \beta = 0$, torej so vrstice v X^q linearno odvisne in pripadajo podprostoru dimenzije $d \leq p$. Velja tudi $d = \text{rang}(X^q)$. Če je $\text{rang}(X^q) = p$ je torej ločnica med podatki enolična, kot v sliki 3, sicer pa je takih ločnic lahko več, kot vidimo v sliki 4.

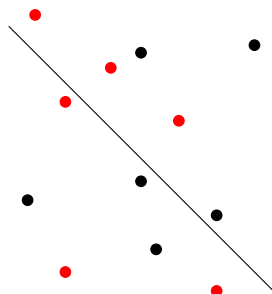


SLIKA 3. Nepopolna separacija, prvi primer



SLIKA 4. Nepopolna separacija, drugi primer

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



SLIKA 5. Prekrivanje

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

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 povezovalnih funkcij. Kot smo omenili že v uvodu povezovalna funkcija opisuje relacijo med pričakovano vrednostjo opazovane spremenljivke in sistematično komponento modela. EkspONENTNO družino in nekaj osnovnih zvez smo predstavili že v poglavju 2. Povezovalna funkcija posplošenega linearnega modela je kanonična, če je enaka naravnemu parametru eksponentne družine. V logistični regresiji je povezovalna funkcija logit, v poglavju 2 pa smo pokazali, da je za binomsko porazdelitev to ravno naravni parameter. Logit je torej kanonična povezovalna funkcija v logističnem modelu.

V spodnji tabeli je navedenih še nekaj drugih primerov takih funkcij.

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

V sledečem razdelku smo zaradi poenostavljenja uporabili hibridne oznake. Logaritemsko funkcijo verjetja in njene odvode upoštevamo kot funkcijo le enega opazovanja ($y_i = y$), odvode po komponentah β pa še vedno pišemo z indeksi. Rezultate za celoten vzorec tako dobimo s seštevanjem po $i = 1, \dots, n$.

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

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

Lotimo se ga po korakih. Prvi faktor 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 faktor 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

$$\frac{\partial \ell}{\partial \beta_j} = \frac{y - \mu}{\text{Var}(Y)} \frac{\partial \mu}{\partial \eta} x_{ij}.$$

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

$$(17) \quad \frac{\partial \ell}{\partial \beta_j} = \frac{y - \mu}{\text{Var}(Y)} b''(\theta) x_{ij} = \frac{y - \mu}{a(\phi)} x_{ij}.$$

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

$$\begin{aligned} -\mathbb{E} \left(\frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k} \right) &= \mathbb{E} \left(\left(\frac{\partial \ell}{\partial \beta_j} \right) \left(\frac{\partial \ell}{\partial \beta_k} \right) \right) \\ &= \mathbb{E} \left(\left(\frac{y - \mu}{\text{Var}(Y)} \right)^2 \left(\frac{\partial \mu}{\partial \eta} \right)^2 x_{ij} x_{ik} \right) \\ &= \frac{1}{\text{Var}(Y)} \left(\frac{\partial \mu}{\partial \eta} \right)^2 x_{ij} x_{ik} \\ &= \frac{b''(\theta)}{a(\phi)} x_{ij} x_{ik}, \end{aligned}$$

kjer smo, tako kot v prejšnji enačbi, v prehodu v zadnjo vrstico upoštevali $\eta = \theta$.

Po drugi strani pa je običajen drugi odvod enak

$$\begin{aligned} \frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k} &= \frac{\partial}{\partial \beta_k} \left\{ \left(\frac{\partial \ell}{\partial \theta} \right) \left(\frac{\partial \theta}{\partial \beta_j} \right) \right\} \\ (18) \quad &= \frac{\partial \ell}{\partial \theta} \left(\frac{\partial^2 \theta}{\partial \beta_j \partial \beta_k} \right) + \left(\frac{\partial \theta}{\partial \beta_j} \right) \left(\frac{\partial^2 \ell}{\partial \theta^2} \right) \left(\frac{\partial \theta}{\partial \beta_k} \right). \end{aligned}$$

Lotimo se ga po korakih. Izračunali smo že, da je $\frac{\partial \ell}{\partial \theta} = \frac{y-b'(\theta)}{a(\phi)}$. Drugi odvod je torej

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

Za $\frac{\partial \theta}{\partial \beta_j}$ zopet v poštev pride verižno pravilo

$$\frac{\partial \theta}{\partial \beta_j} = \frac{\partial \theta}{\partial \mu} \frac{\partial \mu}{\partial \eta} \frac{\partial \eta}{\partial \beta_j} = \frac{1}{b''(\theta)} \frac{\partial \mu}{\partial \eta} x_{ij},$$

v enaki obliki tudi za $\frac{\partial \theta}{\partial \beta_k}$, le da x_{ij} zamenjamo z x_{ik} . Drugi odvod funkcije zbira je z upoštevanjem zgornjega

$$\frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k} = \frac{y-b'(\theta)}{a(\phi)} \frac{\partial}{\partial \beta_k} \left(x_{ij} \frac{1}{b''(\theta)} \frac{\partial \mu}{\partial \eta} \right) + \frac{1}{b''(\theta)^2} \left(\frac{\partial \mu}{\partial \eta} \right)^2 \left(-\frac{b''(\theta)}{a(\phi)} \right) x_{ij} x_{ik}$$

Preostane računanje drugega faktorja prvega člena.

$$\begin{aligned} \frac{\partial}{\partial \beta_k} \left(x_{ij} \frac{1}{b''(\theta)} \frac{\partial \mu}{\partial \eta} \right) &= x_{ij} \left(\frac{-b'''(\theta)}{b''(\theta)^2} \frac{\partial \mu}{\partial \eta} \frac{\partial \theta}{\partial \beta_k} + \frac{1}{b''(\theta)} \frac{\partial^2 \mu}{\partial \eta^2} x_{ik} \right) \\ &= x_{ij} \left(\frac{-b'''(\theta)}{b''(\theta)^2} \left(\frac{\partial \mu}{\partial \eta} \right)^2 \frac{x_{ik}}{b''(\theta)} + \frac{1}{b''(\theta)} \frac{\partial^2 \mu}{\partial \eta^2} x_{ik} \right). \end{aligned}$$

Splošna formula Hessejeve matrike je torej

$$(19) \quad \frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k} = \frac{y-b'(\theta)}{a(\phi)} x_{ij} x_{ik} \left(\frac{-b'''(\theta)}{b''(\theta)^3} \left(\frac{\partial \mu}{\partial \eta} \right)^2 + \frac{1}{b''(\theta)} \frac{\partial^2 \mu}{\partial \eta^2} \right) - \frac{1}{b''(\theta)^2} \left(\frac{\partial \mu}{\partial \eta} \right)^2 \frac{b''(\theta)}{a(\phi)} x_{ij} x_{ik}$$

Nadaljnje računanje bo odvisno od povezovalne funkcije. Če od tu naprej predpostavimo kanonično (in s tem $\theta = \eta$) se zgornji izraz nadaljuje v

$$\begin{aligned} \frac{x_{ij} x_{ik}}{b''(\theta)} \left(-b'''(\theta) + \frac{\partial}{\partial \eta} \left(\frac{\partial \mu}{\partial \eta} \right) \right) &= \frac{x_{ij} x_{ik}}{b''(\theta)} \left(b'''(\theta) + \frac{\partial}{\partial \eta} \left(\frac{\partial \mu}{\partial \theta} \right) \right) \\ &= \frac{x_{ij} x_{ik}}{b''(\theta)} \left(-b'''(\theta) + \frac{\partial}{\partial \eta} (b''(\theta)) \right). \end{aligned}$$

V drugem členu oklepaja upoštevamo kanonično povezovalno funkcijo

$$\frac{\partial}{\partial \eta} (b''(\theta)) = \frac{\partial}{\partial \theta} (b''(\theta)) = b'''(\theta).$$

Sledi

$$\frac{x_{ij} x_{ik}}{b''(\theta)} \left(-b'''(\theta) + \frac{\partial}{\partial \eta} (b''(\theta)) \right) = \frac{x_{ij} x_{ik}}{b''(\theta)} (-b'''(\theta) + b'''(\theta)) = 0,$$

torej za kanonične povezovalne funkcije prvi člen enačbe 18 odpade in ostane le

$$(20) \quad \frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k} = \frac{1}{b''(\theta)^2} \left(\frac{\partial \mu}{\partial \eta} \right)^2 \left(-\frac{b''(\theta)}{a(\phi)} \right) x_{ij} x_{ik}$$

$$(21) \quad = -x_{ij} x_{ik} \frac{\partial^2 \ell}{\partial \theta^2}.$$

Sledi

$$(22) \quad \mathbb{E} \left(\frac{\partial^2 \ell}{\partial \theta^2} \right) = \frac{\partial^2 \ell}{\partial \theta^2}.$$

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

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 *probability unit*. V glavnem se od logistične regresije razlikuje v sistematičnem delu. Verjetnost pozitivnega izida torej po modelu predpostavljamo

$$(23) \quad p_i(\beta) = \Phi(\beta_0 + x_{i1}\beta_1 + \dots + x_{ir}\beta_r),$$

kjer Φ predstavlja kumulativno porazdelitveno funkcijo standardne normalne slučajne spremenljivke. Ta seveda ni linearna, 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 $\text{Bin}(m_i, p_i(\beta))$, 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

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

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

$$(24) \quad \log(L(\beta)) = \ell(\beta) = \sum_{i=1}^n (y_i \log \Phi(x_i^\top \beta) + (m_i - y_i) \log(1 - \Phi(x_i^\top \beta)))$$

Izpeljava enačb verjetja v probit modelu je idejno zelo podobna tisti v razdelku 3.4.1, direktno pa lahko uporabimo tudi zaključke prejšnjega razdelka. Funkcija zbira je

$$\frac{\partial \ell_i}{\partial \beta_j} = \frac{y_i - \mu_i}{\text{Var}(Y)} \frac{\partial \mu_i}{\partial \eta_i} x_{ij} = \frac{y_i - m_i \Phi(x_i^\top \beta)}{\Phi(x_i^\top \beta)(1 - \Phi(x_i^\top \beta))} \phi(x_i^\top \beta) x_{ij},$$

rešujemo pa sistem

$$(25) \quad \sum_{i=1}^n \frac{\partial \ell_i}{\partial \beta_j} = 0, \quad j = 0, 1, \dots, r$$

Te enačbe v splošnem niso rešljive analitično in zopet se bomo morali poslužiti numeričnih metod, za katere potrebujemo še druge odvode.

Po oznakah iz poglavja 2 je za binomsko porazdelitev s parametroma n in p

$$b(\theta) = n \log(1 + e^\theta), \quad \theta = \log\left(\frac{p}{1-p}\right).$$

Potrebovali bomo prve tri odvode funkcije b

$$\begin{aligned} b'(\theta) &= m_i \Phi(x_i^\top \beta) \\ b''(\theta) &= m_i \Phi(x_i^\top \beta) (1 - \Phi(x_i^\top \beta)) \\ b'''(\theta) &= m_i \Phi(x_i^\top \beta) (1 - \Phi(x_i^\top \beta)) (1 - 2\Phi(x_i^\top \beta)), \end{aligned}$$

in pa odvode $\frac{\partial \mu}{\partial \eta}$

$$\begin{aligned} \mu_i &= m_i \Phi(\eta) \\ \frac{\partial \mu_i}{\partial \eta} &= m_i \phi(\eta) \\ \frac{\partial^2 \mu_i}{\partial \eta^2} &= -m_i \phi(\eta) \eta \end{aligned}$$

Uporabimo prej izpeljano zvezo 19 za Hessejevo matriko in dobimo

$$\begin{aligned} \frac{\partial^2}{\partial \beta_j \partial \beta_k} \ell(\beta) &= \sum_{i=1}^n x_{ij} \frac{(y_i - m_i p_i)(-\phi(x_i^\top \beta) x_i^\top \beta) - m_i \phi(x_i^\top \beta)^2}{p_i(1-p_i)} x_{ik} - \\ &\quad - x_{ij} \frac{\phi(x_i^\top \beta)^2 (1 - 2p_i)}{(p_i(1-p_i))^2} x_{ik} \end{aligned}$$

kjer upoštevamo $p_i = \Phi(x_i^\top \beta)$.

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

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

Na podoben način se lotimo tudi Hessejeve matrike. Definiramo diagonalno matriko, kjer so na diagonalni členi (zopet upoštevajoč zgornjo oznako p_i)

$$h_i = \frac{(y_i - m_i p_i)(-\phi(x_i^\top \beta) x_i^\top \beta) - m_i \phi(x_i^\top \beta)^2}{p_i(1-p_i)} - \frac{\phi(x_i^\top \beta)^2 (1 - 2p_i)}{(p_i(1-p_i))^2}$$

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

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

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

$$(26) \quad \beta_{i+1} = \beta_i - (X^\top \mathbf{H} X)^{-1} X^\top s.$$

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 – Raphsonova (oziroma le Newtonova) metoda je bila v osnovi razvita za iskanje ničel funkcije. Spada v razred *navadnih iteracij*, torej metod za iterativno reševanje enačb $f(x) = 0$.

V najosnovnejši (ter najpogostejši) verziji za iskanje ničle funkcije ene spremenljivke začnemo v neki točki, naslendnjo pa izberemo v presečišču tangente, izračunane v tej točki, z x-osjo. Postopek tako iterativno nadaljujemo. Ideja je torej sila preprosta. 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 konvergira za začetne približke dovolj blizu ničli in v neki okolici ničle konvergirala s kvadratično hitrostjo. Na težave lahko 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, ter dejstvo, da za slabe začetne približke ničle morda ne bomo našli. S temi težavami se bomo soočili v nadaljevanju.

4.1.1. Newton–Raphsonova metoda v višjih dimenzijah. Newtonovo metodo se da enostavno splošiti za iskanje ničel vektorskih funkcij. Recimo, da imamo funkcijo $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ in iščemo tak vektor $x^* \in \mathbb{R}^n$, za katerega bo $F(x^*) = (f_1(x^*), \dots, f_n(x^*)) = (0, \dots, 0)$. Podobno kot zgoraj tvorimo zaporedje

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

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

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

od koder izračunamo vektor premikov $h \in \mathbb{R}^n$ 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

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

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

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

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

Imamo torej algoritem za iskanje ničel, v luči iskanja cenilke največjega verjetja pa bi želeli algoritem, ki poišče ekstrem funkcije. V našem primeru iščemo maksimum logaritemske funkcije verjetja ℓ . Tega se lotimo tako, da njen gradient enačimo z nič

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

in izrazimo naslednji približek

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

S tem postopkom imamo lahko dva problema. Prvič, drugi odvod (Hessian) funkcije je lahko zahtevno računati in invertirati, za kakšen θ_n morda sploh ne obstaja. Drugič, Newtonova metoda ni naraščajoč algoritem in ne velja nujno $\ell(\theta_n) < \ell(\theta_{n+1})$. Mi pa bi želeli algoritem, ki bo konvergirala 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:

$$(27) \quad \begin{aligned} x_{n+1} &= x_n + p_n \\ \frac{\partial^2}{\partial \theta^2} \ell(\theta_n) p_n &= - \frac{\partial}{\partial \theta} \ell(\theta_n) \end{aligned}$$

Zadnji vrstici v (27) 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 (maksimuma) 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 Hessian pozitivno definiten (in je tam gradient enak nič). Če bi torej imeli strogo pozitivno 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

$$f(x^* + s) = f(x^*) + \frac{\partial}{\partial \theta} f(x^*) s + \frac{1}{2} s^\top \frac{\partial^2}{\partial \theta^2} f(x^*) s.$$

Če velja še pogoj prvega reda, torej $\frac{\partial}{\partial \theta} f(x^*) = 0$, imamo

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

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.

4.1.2. *Potencialne težave Newtonove metode.* Newtonova metoda ima mnogo pozitivnih plati, vendar pa je iz določenih vidikov precej občutljiva. Morda najbolj očitni 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. Očitni 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, bi dobili

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

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 tudi 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 k iskani ničli. V splošnem zna biti obnašanje takega zaporedja precej zapleteno, imenuje se Newtonov fraktal in vanj se tu ne bomo spuščali.

Naslednja težava 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

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

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 k iskani ničli. Smiselno sledi vprašanje, katerim pogojem mora biti zadoščeno, da vednarle dobimo pravilno rešitev. To nam poda sledeči izrek, ki zagotavlja konvergenco, če je iteracijska funkcija v okolici negibne točke skrčitev.

Izrek 4.1. *Naj za iteracijsko funkcijo velja $g(\alpha) = \alpha$ in naj na intervalu $I = [\alpha - \delta, \alpha + \delta]$ zadošča Lipschitzovemu pogoju*

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

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

$$(28) \quad |x_r - \alpha| \leq m^r |x_0 - \alpha|$$

in

$$(29) \quad |x_{r+1} - \alpha| \leq \frac{m}{1-m} |x_r - x_{r-1}|$$

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

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

koder vidimo oceno (28).

Za drugo neenakost pa si oglejmo

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

Upoštevamo še

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

in končno dobimo

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

□

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 $|g'(x)| < m < 1$ za $x \in I$, 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 (28) iteracija konvergira. □

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

Definirajmo *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 tako število $C > 0$, da

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

Red konvergence navadne iteracije je običajno precej enostavno določiti. Metodo nam daje 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 $k = 1, \dots, p-1$ 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

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

Vidimo, da je potrebno ločiti dva primera:

- Če je α enostavna ničla funkcije f , je $g'(\alpha) = 0$ in 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 $\alpha \neq 0$.

- Če je α m -kratna ničla pa velja

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

od koder sledi linearna konvergenca. Dokažimo drugo alinejo.

Dokaz. Ker je α m -kratna ničla je prvih $m - 1$ odvodov funkcije f enakih 0. Opazimo, da moramo $(2m - 2)$ -krat uporabiti l'Hospitalovo pravilo, da dobimo neničeln ulomek

$$\lim_{x \rightarrow \alpha} g'(x) = \lim_{x \rightarrow \alpha} \frac{f(x)f''(x)}{(f'(x))^2} = \lim_{x \rightarrow \alpha} \frac{\sum_{k=0}^{2m-2} \binom{2m-2}{k} f^{(k)}(x) f^{(2m-2-k+2)}(x)}{\sum_{k=0}^{2m-2} \binom{2m-2}{k} f^{(2m-2-k+1)}(x) f^{(k+1)}(x)},$$

v imenovalcu ostane le m -ti člen, v števcu pa $(m - 1)$ -ti. Izračunamo

$$\frac{\binom{2m-2}{m} (f^{(m)}(x))^2}{\binom{2m-2}{m-1} (f^{(m)}(x))^2} = \frac{\frac{(2m-2)!}{(m-2)!m!}}{\frac{(2m-2)!}{(m-1)!(m-1)!}} = \frac{(m-1)!(m-1)!}{(m-2)!m!} = \frac{m-1}{m}$$

kar je ravno želeni rezultat. □

4.2. Fisherjeva zbirna metoda. Fisherjeva zbirna metoda je variacija v prejšnjem razdelku opisanega Newton – Raphsonovega algoritma, ki se v statistiki uporablja za numerično reševanje enačb verjetja. Poimenovana je po Ronaldu Fisherju, enemu najpomembnejših 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

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

Fisherjeva zbirna metoda je po zgornjih oznakah

$$(30) \quad \theta_{n+1} = \theta_n - \text{FI}(\theta)^{-1} \left(\frac{\partial}{\partial \theta} \ell(\theta) \right)$$

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

$$\text{FI}(\theta) = \mathbb{E} \left(\left(\frac{\partial}{\partial \theta} \ell \right) \left(\frac{\partial}{\partial \theta} \ell \right)^\top \right) = \mathbb{E} \left(\frac{\partial^2}{\partial \theta^2} \ell \right) = \frac{\partial^2}{\partial \theta^2} \ell \rightarrow \text{FI}(\theta) = \frac{\partial^2}{\partial \theta^2} \ell,$$

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

$$\begin{aligned} \text{FI}(\theta) &= \mathbb{E} \left[\left(\frac{\partial}{\partial \theta} \ell(\theta) \right) \left(\frac{\partial}{\partial \theta} \ell(\theta) \right)^\top \right] \\ &= \mathbb{E} \left[\left(\frac{\partial}{\partial \theta} \ell(\theta) - \mathbb{E} \left[\frac{\partial}{\partial \theta} \ell(\theta) \right] \right) \left(\frac{\partial}{\partial \theta} \ell(\theta) - \mathbb{E} \left[\frac{\partial}{\partial \theta} \ell(\theta) \right] \right)^\top \right] \\ (31) \quad &= \text{Var} \left[\frac{\partial}{\partial \theta} \ell(\theta) \right], \end{aligned}$$

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 torej očitne.

4.2.2. *Fisherjeva zbirna metoda v logističnem modelu.* Poglejmo za trenutek nazaj v razdelek 3.4.1, natančneje k enačbam (12) in (15). Iz prejšnjih razdelkov vemo tudi, da velja

$$\text{FI}(\theta) = \mathbb{E} \left[-\frac{\partial^2}{\partial \theta^2} \ell(\theta) \right] \stackrel{(15)}{=} X^\top v(\theta) X = -\frac{\partial^2}{\partial \theta^2} \ell(\theta),$$

kjer smo seveda uporabili že dokazano informacijsko enakost. Tako vidimo, da Fisherjeva zbirna metoda in Newton–Raphsonova metoda v primeru logistične regresije res sovpadata, saj je matrika drugih odvodov ravno enaka informacijski matriki. Vse skupaj zapošemo kot

$$\begin{aligned} \hat{\theta}_{i+1} &= \hat{\theta}_i - \left(\frac{\partial^2}{\partial \theta^2} \ell(\hat{\theta}_i) \right)^{-1} \frac{\partial}{\partial \theta} \ell(\hat{\theta}_i) \\ (32) \quad &= \hat{\theta}_i + (X^\top v(\hat{\theta}_i) X)^{-1} X^\top (y - \mu(\hat{\theta}_i)). \end{aligned}$$

5. PRIMERI

5.1. **Ocenjevanje parametrov v logističnem modelu.** V praktično usmerjenem delu naloge smo v Pythonu implementirali zgoraj opisani postopek Fisherjeve zbirne metode za binomsko porazdeljene slučajne spremenljivke. Za delo smo uporabili knjižnici `NumPy` za računanje z matrikami in vektorji, reševanje sistemov enačb ter invertiranje in 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 izpeljanim enačbam v razdelkih 3.4.1 in 3.7.1, zato jih tu ne bomo ponovno navajali.

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

$$\text{logit}(p) = X\beta,$$

torej kanoničnega logističnega modela.

function LogitModel(iteracije, X, Y, β_{zacetni} , ϵ)

```

 $p = \frac{\exp(X^\top \beta_{\text{zacetni}})}{1 + \exp(X^\top \beta_{\text{zacetni}})}$ 
 $V = p(1 - p)$ 
 $\text{Score} = X^\top (Y - p)$ 
 $\text{Info} = X^\top V X$ 
Reši sistem na h: Info * h = Score
 $\beta_{\text{star}} = \beta_{\text{zacetni}}$ 
 $\beta_{\text{nov}} = \beta_{\text{star}} + h$ 
while  $i \leq \text{iteracije}$  do
  if  $\beta_{\text{nov}} - \beta_{\text{star}} \geq \epsilon$  then
     $p = \frac{\exp(X^\top \beta_{\text{nov}})}{1 + \exp(X^\top \beta_{\text{nov}})}$ 
     $V = p(1 - p)$ 
     $\text{Score} = X^\top (Y - p)$ 
     $\text{Info} = X^\top V X$ 
    Razreši na h:
     $\text{Info} * h = \text{Score}$ 
     $\beta_{\text{star}} = \beta_{\text{nov}}$ 
     $\beta_{\text{nov}} = \beta_{\text{star}} + h$ 
  else
    Dosegli smo zeleno natančnost v zadostnem številu korakov
    return  $\beta_{\text{nov}}$ 
  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 . *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

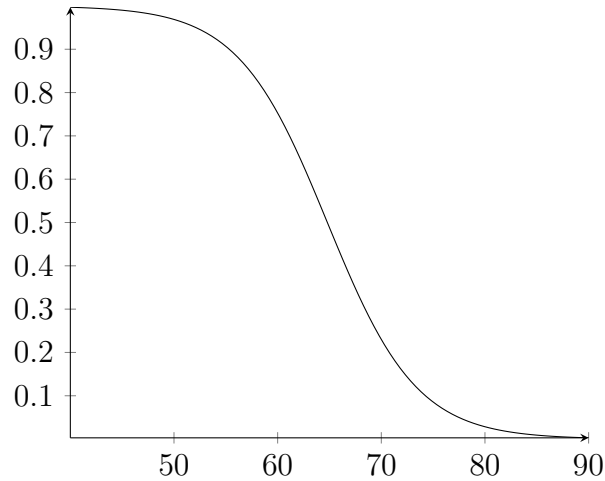
Poženimo algoritem najprej le na podatkih o temperaturi. Za začetno vrednost približka β vzemimo 0, natančnost ϵ pa si predpšemo na 0,001. Ker imamo le eno pojasnjevalno spremenljivko uporabljamo torej model oblike

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

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

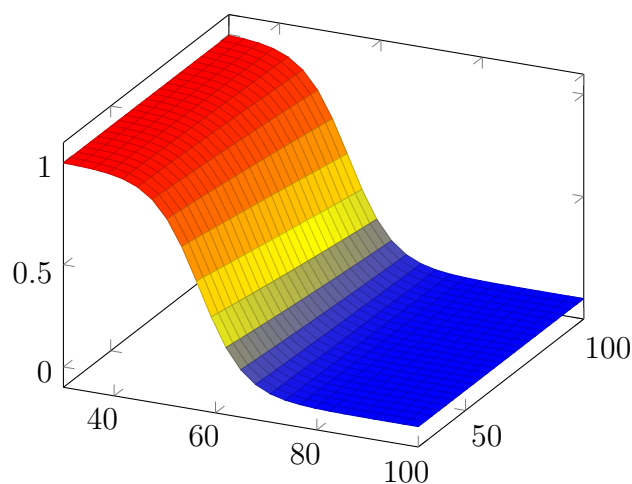


SLIKA 6. Izračunane verjetnosti z eno pojasnjevalno spremenljivko

Vidimo da je sigmoida v tem primeru obrnjena drugače kot na sliki 1 - tak rezultat nam da negativen predznak parametra β_1 . Sklepamo lahko, 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

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



SLIKA 7. 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 *koefficiente* pri Hessejevi matriki in funkciji zbira - to so le deli dejanske formule, izračunani posebej za večjo preglednost.

function ProbitModel(iteracije, X, Y, β_{zacetni} , ϵ)

```

 $p = \Phi(X^\top \beta_{\text{zacetni}})$ 
 $\text{Score}_{\text{koe}} = \frac{Y-p}{p*(1-p)} * \phi(X^\top \beta_{\text{zacetni}})$ 
 $\text{Score} = X^\top \text{Score}_{\text{koe}}$ 
 $\text{Hess}_{\text{koe}} = \frac{(Y-p)(-\phi(X^\top \beta_{\text{zacetni}})X^\top \beta_{\text{zacetni}} - \phi(X^\top \beta_{\text{zacetni}}))}{p(1-p)} - \frac{\phi(X^\top \beta_{\text{zacetni}})^2(1-2p)}{(p(1-p))^2}$ 
 $\text{Hess} = \text{diag}(\text{Hess}_{\text{koe}})$ 
 $H = X^\top \text{Hess}_{\text{koe}} X$ 
Reši sistem na h:  $H * h = \text{Score}$ 
 $\beta_{\text{star}} = \beta_{\text{zacetni}}$ 
 $\beta_{\text{nov}} = \beta_{\text{star}} - h$ 
while  $i \leq \text{iteracije}$  do
  if  $\beta_{\text{nov}} - \beta_{\text{star}} \geq \epsilon$  then
     $p = \Phi(X^\top \beta_{\text{nov}})$ 
     $\text{Score}_{\text{koe}} = \frac{Y-p}{p*(1-p)} * \phi(X^\top \beta_{\text{nov}})$ 
     $\text{Score} = X^\top \text{Score}_{\text{koe}}$ 
     $\text{Hess}_{\text{koe}} = \frac{\phi(X^\top \beta_{\text{nov}})}{p*(1-p)} * \left( \phi(X^\top \beta_{\text{nov}}) \frac{2*p*Y-p^2-Y}{p*(1-p)} - X(Y-p) \right)$ 
     $H = X^\top \text{Hess}_{\text{koe}} X$ 
    Razreši na h:
     $H * h = \text{Score}$ 
     $\beta_{\text{star}} = \beta_{\text{nov}}$ 
     $\beta_{\text{nov}} = \beta_{\text{star}} - h$ 
  else
    Dosegli smo želeno natančnost v zadostnem številu korakov
    return  $\beta_{\text{nov}}$ 
  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

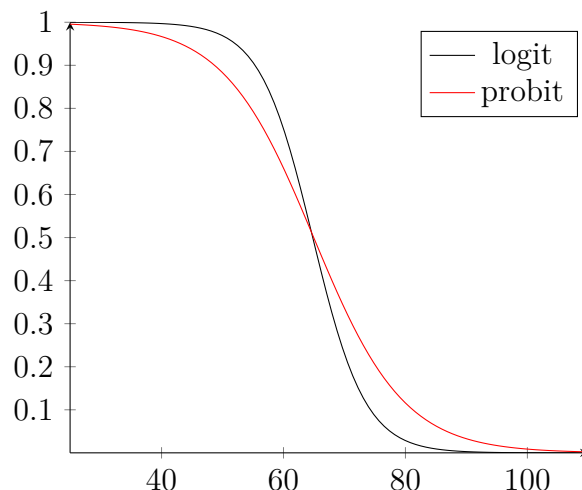
$$\frac{e^{8.77495-0.13510x}}{1 + e^{8.77495-0.13510x}}.$$

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. Oglejmo si še direktno primerjavo dobljenih rezultatov.



SLIKA 8. 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 bistveno, 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's scoring Fisherjeva zbirna metoda

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.