

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Nelineární regrese v programu R

Plzeň, 2016

Bc. Kateřina Palková

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 10. 5. 2016

Bc. Kateřina Palková

Poděkování

Děkuji vedoucí své diplomové práce paní RNDr. Blance Šedivé, Ph.D. za odborné vedení, vstřícný přístup a trpělivost při psaní této práce.

Abstrakt

Tato diplomová práce se zabývá porovnáváním různých metod odhadu parametrů nelineární regrese. Metody jsou testovány ve statistickém prostředí R na několika sadách dat. Výsledné odhady získané jednotlivými metodami jsou pak porovnávány mezi sebou z hlediska přesnosti odhadu (tj. blízkosti k teoretickému řešení), hodnoty kritériální funkce a dle výpočetního času algoritmu.

Klíčová slova: nelineární regrese, metody odhadu parametrů, metoda nejmenších čtverců, výpočetní prostředí R

Abstract

This thesis discusses comparison of several parameter estimation methods for nonlinear regression models. The methods are tested in the software environment R using several datasets. The resulting estimates of parameters are compared with each other by exactness (i.e. distance to theoretical solution), value of least squares criterion and computing time.

Keywords: nonlinear regression, parameter estimation methods, least squares method, software environment R

Obsah

1	Úvod	10
2	Program R	11
3	Nelineární regrese	14
3.1	Model	14
3.2	Metoda nejmenších čtverců	15
3.3	Konfidenční množina a profilový diagram	16
4	Metody nelineární regrese	18
4.1	Nederivační metody	18
4.1.1	Metody přímého hledání	18
4.1.2	Simplexové metody	20
4.1.3	Metody používající náhodná čísla	25
4.2	Derivační metody	29
4.2.1	Gaussovy-Newtonovy metody	31
4.2.2	Metody Marquardtova typu	32
5	Praktická část	34
5.1	Data	34
5.1.1	Puromycin data	34
5.1.2	Farmakologická data	35
5.2	Konfidenční množiny a profilové diagramy	37
5.3	Lineární aproximace	40
5.4	Porovnání jednotlivých algoritmů - simulovaná data	42
5.4.1	Výsledky	46
5.5	Porovnání jednotlivých algoritmů - reálná data	51
5.5.1	Puromycin data	51
5.5.2	Farmakologická data	54
5.5.3	Výsledky	57

6 Závěr	63
Literatura	65
A Obecný matematický základ	69
B Datové sety	71
C Přílohy na CD	75

Seznam obrázků

4.1	Hookeova-Jeevesova metoda pro dva parametry. (Čárkovaně jsou znázorněny neúspěšná posunutí. Modře jsou označená posunutí k nové iteraci.)	19
4.2	Operace reflexe (vrchol R) v Nelderově-Meadově metodě pro 2 parametry.	22
4.3	Operace expanze (vrchol E) v Nelderově-Meadově metodě pro 2 parametry.	22
4.4	Operace kontrakce v Nelderově-Meadově metodě pro 2 parametry. (Bod C_1 je vnitřní a bod C_2 je vnější kontrakce.)	23
4.5	Operace redukce (vrcholy W^* a G^*) v Nelderově-Meadově metodě pro 2 parametry.	24
4.6	Znázornění funkce $V_i(\lambda)$ v Marquardtově metodě pro dva parametry	33
5.1	Vykreslení dat s a bez puromycinu	35
5.2	Vykreslení farmakologických dat	36
5.3	Vykreslení konfidenčních množin včetně intervalových odhadů pro jednotlivé parametry θ_1 nebo θ_2 pro puromycin data. Intervalové odhady jsou znázorněny zelenou čárkovanou čarou.	37
5.4	Profilový diagram pro data bez puromycinu.	38
5.5	Profilový diagram pro data s puromycinem.	38
5.6	Vykreslení konfidenčních množin včetně intervalových odhadů pro jednotlivé parametry β nebo γ pro farmakologická data. Intervalové odhady jsou znázorněny zelenou čárkovanou čarou.	39
5.7	Profilový diagram pro farmakologická data.	39
5.8	Výsledek metody lineární aproximace pro data s puromycinem.	40

5.9	Výsledek metody lineární aproximace pro data bez průmocinu.	41
-----	---	----

Seznam tabulek

5.1	Výsledky Nelderova-Meadova simplexového algoritmu - simulovaná data (model (5.3)). Zeleně jsou označeny nejmenší hodnoty kritériální funkce ve srovnání s tabulkou 5.3.	48
5.2	Výsledky Nelderova-Meadova simplexového algoritmu - simulovaná data (model (5.4)). Zeleně jsou označeny nejmenší hodnoty kritériální funkce ve srovnání s tabulkou 5.4.	48
5.3	Výsledky - simulovaná data (model (5.3)). Zeleně je označeno několik nejmenších hodnot kritériální funkce. Žlutě je označen algoritmus a počáteční odhady, které končí chybou.	49
5.4	Výsledky - simulovaná data (model (5.4)). Zeleně je označeno několik nejmenších hodnot kritériální funkce. Žlutě jsou označeny algoritmy a počáteční odhady, které končí chybou.	50
5.5	Výsledky Nelderova-Meadova simplexového algoritmu - data s puromycinem (treated puromycin). Zeleně jsou označeny nejmenší hodnoty kritériální funkce ve srovnání s tabulkou 5.8.	59
5.6	Výsledky Nelderova-Meadova simplexového algoritmu - data bez puromycinu (untreated puromycin). Zeleně jsou označeny nejmenší hodnoty kritériální funkce ve srovnání s tabulkou 5.9.	59
5.7	Výsledky Nelderova-Meadova simplexového algoritmu - farmakologická data. Zeleně jsou označeny nejmenší hodnoty kritériální funkce ve srovnání s tabulkou 5.10.	59

5.8	Výsledky - data s puromycinem (treated puromycin). Zeleně je označeno několik nejmenších hodnot kriteriální funkce. Žlutě je označen algoritmus a počáteční odhady, které končí chybou.	60
5.9	Výsledky - data bez puromycinu (untreated puromycin). Zeleně je označeno několik nejmenších hodnot kriteriální funkce. Žlutě je označen algoritmus a počáteční odhady, které končí chybou.	61
5.10	Výsledky - farmakologická data. Zeleně je označeno několik nejmenších hodnot kriteriální funkce. Žlutě je označen algoritmus a počáteční odhady, které končí chybou.	62
B.1	Puromycin data	71
B.2	Farmakologická data	72
B.3	Generovaná data 1	73
B.4	Generovaná data 2	74

Kapitola 1

Úvod

Hlavním cílem diplomové práce je porovnání různých metod odhadů parametrů nelineární regrese, aplikování těchto metod ve výpočetním softwaru R a zhodnocení rychlosti a hlavně přesnosti nalezení odhadu regresních parametrů.

Práce je rozdělena do několika kapitol. Nejprve je popsán software R a jeho hlavní funkce. Ve třetí kapitole jsou shrnuty základní poznatky nelineárních regresních modelů a postup získávání odhadů regresních parametrů metodou nejmenších čtverců. Čtvrtá kapitola se zabývá jednotlivými odhady parametrů v nelineární regresi. Jsou zde podrobně popsány jednotlivé algoritmy včetně zastavovacích podmínek.

Další, pátá kapitola je věnována praktické části této práce. V této kapitole jsou popsány sady dat, na kterých jsou testovány všechny metody odhadů parametrů nelineární regrese. Dále je v této kapitole zmíněné nastavení zastavovacích podmínek a konstant v některých metodách. Velká část je pak určena pro výsledky jednotlivých algoritmů a porovnání mezi nimi. V poslední kapitole pak shrneme veškeré výsledky a poznatky z testování všech algoritmů.

Kapitola 2

Program R

Program R [20] je programovací jazyk a prostředí pro zpracování statistických výpočtů a grafických výstupů. Jedná se o volně šiřitelný software, který lze používat na unixových platformách i v operačních systémech Windows a MacOS. Existuje také mnoho uživatelsky příjemných rozhraní využívající R, například mnou preferované RStudio [22].

Program R obsahuje nástroje pro manipulaci s daty, analýzu dat, výpočty a grafická zobrazení. Základní součástí tohoto programovacího jazyku jsou podmínky, cykly, rekurzivní funkce, sady operátorů pro vektorové a maticové výpočty, načítání a ukládání z a do mnoha různých formátů a spousta již implementovaných funkcí. Kromě toho lze tyto základní možnosti programu vzhledem k otevřené platformě rozšiřovat o další balíčky (packages), které jsou také volně dostupné přes CRAN (Comprehensive R Archive Network) [21]. Každý balíček se nejdříve musí stáhnout a nainstalovat, a poté načíst následujícími příkazy.

```
install.packages("jméno_balíčku")  
library("jméno_balíčku")
```

Jedním z balíčků je i `datasets`, který obsahuje různé sady dat určených k výpočtům. Seznam všech těchto datových sad včetně jejich popisu lze nalézt v [23].

V programu R jsou též integrovány funkce pro řešení nelineárních regresních modelů. Nejčastěji používanou je funkce `nls()` z balíčku `stats`. Funkce `nls()` (z anglického Nonlinear Least Squares) vyu-

žívá pro své výpočty, jak už název napovídá, metodu nejmenších čtverců. Funkce počítá bodové odhady parametrů nelineárních modelů třemi různými algoritmy, které lze měnit v argumentu `algorithm` této funkce. Jako výchozí je nastaven Gaussův-Newtonův algoritmus. Další možnou hodnotou argumentu je `plinear` (Golubův-Pereyrův algoritmus) nebo `plnls` pro algoritmus NL2SOL (An Adaptive Non-linear Least-Squares Algorithm), které v této práci nejsou použity. Podrobný popis funkce lze nalézt v dokumentaci R-funkcí [24]. Existují i další funkce zaměřené na nelineární modely. Například funkce `nls2()` využívající řešení hrubou silou, vhodná obzvlášť, když funkce `nls()` selhává. Více o těchto funkcích lze nalézt v [1, 2] a dalších.

V této práci jsou pro všechny metody uvedené v kapitole Metody nelineární regrese napsány vlastní kódy. R-funkce `nls()` je použita jen pro vykreslení konfidenčních množin.

Užitečné R-funkce použité v praktické části této práce

- Grafické
 - `plot()` - vytvoření grafu a vykreslení,
 - `lines()` - dodatečné přidání funkce do grafu,
 - `points()` - dodatečné přidání bodů do grafu,
 - `segments()` - dodatečné přidání úsečky do grafu,
- Statistická analýza
 - `summary()` - spočítá základní popisné statistiky,
- Matice a maticové operace
 - `solve()` - inverzní matice,
 - `t()` - transponovaná matice,
 - `nrow()` - počet řádků v matici,
 - `ncol()` - počet sloupců v matici,
 - `colMeans()` - aritmetické průměry v jednotlivých sloupcích matice,
- Generování náhodných veličin z daného rozdělení
 - `rnorm()` - náhodná veličina z normálního rozdělení,
 - `runif()` - náhodná veličina z rovnoměrného rozdělení,

- Funkce vracející TRUE/FALSE hodnotu
`is.finite()` - je konečná hodnota,
`is.nan()` - je hodnota NaN,
`!is.nan()` - není hodnota NaN,
- Derivace funkce
`D()` - symbolický výpočet první derivace,
- Práce se soubory
`write.table()` - ukládání do souboru,
`read.table()` - načítání ze souboru.

Kapitola 3

Nelineární regrese

Podklady k následující kapitole byly čerpány z [3, 4, 5, 6], kde lze najít mnoho dalších doplnění a rozšíření.

3.1 Model

Cílem všech regresních modelů je vysvětlit variabilitu nezávislé náhodné veličiny Y závislostí její střední hodnoty na jedné nebo více nenáhodných nezávislých proměnných označovaných jako x . V dalším textu bude dodržováno značení: Y je vektor náhodných veličin a Y_j jeho prvek, x je také vektor (případně matice pro více nezávisle proměnných), ale deterministických veličin a x_i je prvek vektoru (případně řádek matice).

Nelineární regresní model uvažujeme ve tvaru

$$Y = f(x, \beta) + \varepsilon, \quad (3.1)$$

kde $f(x, \beta)$ je známá regresní funkce, $\beta = (\beta_1, \beta_2, \dots, \beta_p)$ je vektor p neznámých parametrů modelu a ε je vektor náhodných složek modelu, pro které platí $\varepsilon \sim \mathbb{N}(0, \sigma^2 I)$. Dále budeme předpokládat, že $f(x, \beta)$, $\frac{\partial}{\partial \beta_j} f(x, \beta)$ a $\frac{\partial^2}{\partial \beta_j \partial \beta_i} f(x, \beta)$ jsou spojitě pro všechny β z parametrického prostoru Ω a všechny x . Parametrický prostor Ω je množina všech přípustných parametrů β a předpokládáme, že se jedná o otevřenou konvexní podmnožinu prostoru \mathbb{R}^p . Posledním předpokladem je to, že matice prvních derivací regresní funkce typu $n \times p$, která je dána vztahem $F(\beta) = (f_j(x_i, \beta))$, má alespoň v okolí správné tj. teoretické hodnoty β^* hodnost p .

Lineární regresní model je model, ve kterém je regresní funkce $f(x, \beta)$ lineární vzhledem k parametrům β . Pro lineární regresní modely platí následující podmínka

$$g_j = \frac{\partial f(x, \beta)}{\partial \beta_j} = konst, \quad j = 1, \dots, p, \quad (3.2)$$

kde p je počet parametrů modelu. O nelineární regresní model se jedná, pokud je nejméně pro jeden parametr β_j parciální derivace g_j jeho nekonstantní funkcí.

Vzhledem k podmínce (3.2) můžeme nelineární regresní modely rozdělit na

- Neseparabilní modely

Podmínka (3.2) neplatí pro žádný parametr. Příkladem je regresní funkce

$$f(x, \beta) = e^{\beta_1 x} + e^{\beta_2 x} + e^{\beta_3 x}.$$

- Separabilní modely

Podmínka (3.2) platí pro jeden nebo více parametrů. Příkladem je funkce

$$f(x, \beta) = \beta_1 + \beta_2 \sin(\beta_3 x),$$

která je nelineární pouze vzhledem k β_3 .

- Vnitřně lineární modely

Tyto modely jsou sice nelineární, ale vhodnou transformací je lze převést na modely lineární. Například regresní funkce

$$f(x, \beta) = \frac{1}{\beta_1} x.$$

Transformací $\beta_2 = \frac{1}{\beta_1}$ převedeme tento model na lineární.

Vhodnou transformací lze též odstranit numerické či statistické problémy regrese či z neseparabilních modelů získat separabilní a naopak. Transformace totiž obvykle vede ke změně parametrického prostoru.

3.2 Metoda nejmenších čtverců

Uvažujme funkci

$$S(\beta) = \sum_{i=1}^n (Y_i - f(x_i, \beta))^2. \quad (3.3)$$

Odhad b metodou nejmenších čtverců je takový prvek z parametrického prostoru Ω , který minimalizuje $S(\beta)$.

$$b = \arg \min_{\beta \in \Omega} \left\{ \sum_{i=1}^n (Y_i - f(x_i, \beta))^2 \right\} \quad (3.4)$$

Funkce $S(\beta)$ je proto často nazývána kritériální funkcí. Pro odhad rozptylu použijeme obdobně jako u lineárního modelu

$$s^2 = \frac{S(b)}{n - p}, \quad (3.5)$$

přičemž se uvažuje, že $n > p$.

Další možnou metodou odhadu parametrů je metoda maximální věrohodnosti. Více o ní lze nalézt například v [3, 4]. Pokud považujeme ε z modelu (3.1) za náhodnou nezávislou veličinu z normálního rozdělení s nulovou střední hodnotou a konstantním rozptylem, potom je odhad parametrů b pomocí metody nejmenších čtverců shodný s metodou maximální věrohodnosti a odhady rozptylu pomocí obou metod jsou asymptoticky ekvivalentní [3, 4].

3.3 Konfidenční množina a profilový diagram

Uvažujme testování hypotézy o teoretické hodnotě $\beta^* = \beta^0$. V souvislosti s touto hypotézou budeme hledat tzv. konfidenční množinu pro daný vektor odhadů parametrů. Věrohodnostní konfidenční množina porovnává hodnotu věrohodnostní (kritériální (3.3)) funkce pro b a pro β^0 a má tvar

$$KM = \left\{ \beta : S(\beta) \leq S(b) \left(1 + \frac{p}{n - p} F_{p, n-p}(\alpha) \right) \right\}, \quad (3.6)$$

kde F je kvantil Fisherova rozdělení a β je z parametrického prostoru Ω . Více o konfidenčních množinách a jejich typech lze najít v [4, 5, 29] aj.

Označme $\tilde{b}_j(\beta)$ vektor parametrů, který minimalizuje kritériální funkci $S(\beta)$ (3.3) za podmínky, že $\beta_j = \beta$. Následně uvažujme výraz

$$\tau_j(\beta) = \frac{\sqrt{S(\tilde{b}_j(\beta)) - S(b)}}{s} \text{sign}(\beta - b_j), \quad (3.7)$$

kde s je odhad směrodatné odchylky vyjádřený ze vztahu (3.5). Za profilový diagram považujeme znázornění bodů $[\beta, \tau_j(\beta)]$ případně $[\beta, |\tau_j(\beta)|]$ v okolí bodového odhadu b_j parametru β_j . Více o profilových diagramech lze opět nalézt například v [4, 5].

Kapitola 4

Metody nelineární regrese

Vzhledem k nelineárnímu charakteru regresní funkce $f(x, \beta)$ je v případě nelineární regrese nutné hledat odhady b z (3.4) numericky. Při numerickém hledání odhadů parametrů modelu (3.1) se v nelineární regresi můžeme setkat s několika problémy výpočtu

- b nemusí obecně existovat,
- b nemusí být jednoznačně určené,
- iterační numerická metoda pro hledání parametrů nemusí konvergovat nebo bude konvergovat k nesprávné hodnotě (např. k lokálnímu namísto ke globálnímu extrému).

4.1 Nederivační metody

V následujících popisech metod bude dodržováno značení $b_j(i)$ jako odhadu j -tého parametru v i -tém iteračním kroku.

4.1.1 Metody přímého hledání

Existuje mnoho jednoduchých až heuristických postupů, které přímo hledají minimum. Mezi nejznámější metody přímého hledání patří Hookeův-Jeevesův algoritmus [3, 7, 12]. Tento algoritmus patří mezi velice intuitivní a je nenáročný i z hlediska výpočetní složitosti. Na druhou stranu nevýhodou může být rychlost respektive pomalost konvergence.

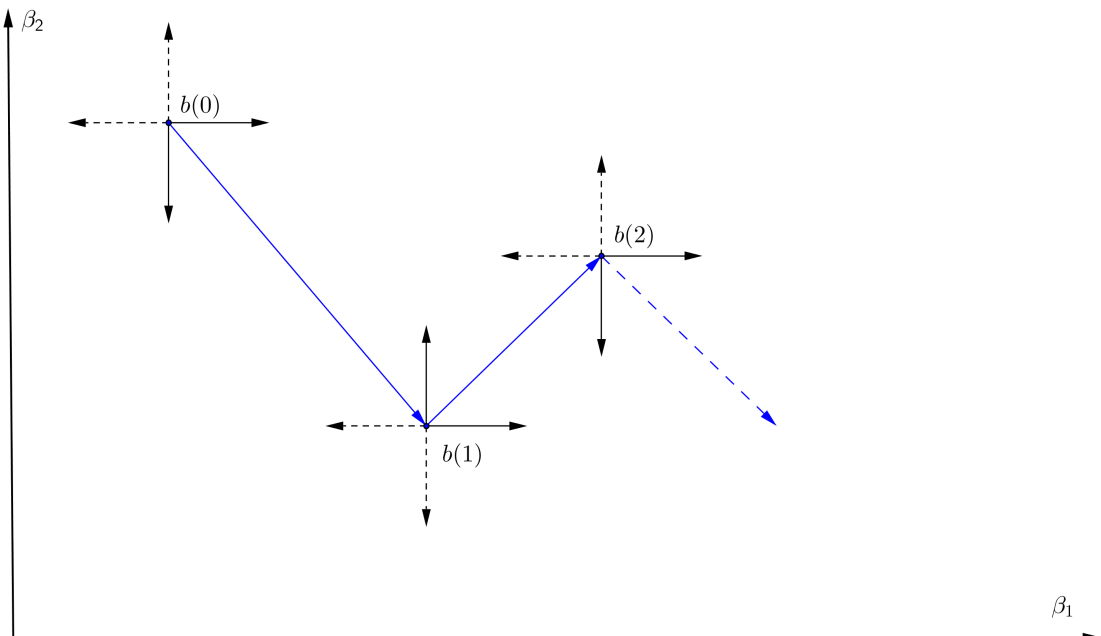
Hookeovu-Jeevesovu metodu lze popsat v několika krocích.

- 1 Určíme počáteční hodnoty všech parametrů $b_j(0)$, $j = 1, \dots, p$ a zároveň velikost krokového posunu δ_i v i -té iteraci, který se může buď postupně zmenšovat nebo být konstantní.
- 2 V každé i -té iteraci se snažíme zmenšit kriteriální funkci S (viz (3.3)) změnou $b_1(i)$ o δ_i . Takže pokud hodnota $b_1^*(i) = b_1(i) + \delta_i$, zmenší aktuální hodnotu $S((b_1^*(i), b_2(i), \dots, b_p(i)))$, tak směr určený body $b_1^*(i)$ a $b_1(i)$ je považován za směr minimalizace prvního parametru. Jinak obdobně zkusíme změnu o $-\delta_i$. Pokud žádný posun $\pm\delta_i$ nezlepší $S((b_1^*(i), b_2(i), \dots, b_p(i)))$, tak ponecháme starou hodnotu parametru (hodnotu z předchozí iterace). Stejně pokračujeme pro druhý a případně i další parametr.
- 3 Jednorozměrnou minimalizací určíme nové hodnoty parametrů $b(i+1)$ neboli

$$b(i+1) = b^*(i) + h_i(b^*(i) - b(i)),$$

kde $h_i = \min_h S(b^*(i) + h(b^*(i) - b(i)))$.

- 4 Body 2 a 3 opakujeme dokud není splněna zastavovací podmínka.



Obrázek 4.1: Hookeova-Jeevesova metoda pro dva parametry. (Čárkovaně jsou znázorněny neúspěšná posunutí. Modře jsou označena posunutí k nové iteraci.)

Schématický postup Hookeova-Jeevesova algoritmu pro dva parametry lze vidět na obrázku 4.1. Ve zdroji [12] je jako zastavovací podmínka pro zmenšující se krok δ_i použita nerovnost $|\delta_i| \leq \epsilon$. Lze však použít normu v eukleidovském smyslu mezi dvěma za sebou následujícími iteracemi $i + 1$ a i , tj. podmínku

$$\sum_{j=1}^p (b_j(i) - b_j(i + 1))^2 < \epsilon. \quad (4.1)$$

4.1.2 Simplexové metody

Simplexové metody používají k hledání optima adaptivní polyedry (simplexy). Mezi nejjednodušší a univerzálně použitelné algoritmy patří metoda navržená Nelderem a Meadem [3, 7, 25, 26].

Je-li p počet parametrů regresního modelu, tak simplex je konvexní mnohostěn určený $p + 1$ vrcholy v p -rozměrném prostoru. Pro $p = 2$ je tudíž simplexem trojúhelník, pro $p = 3$ jehlan s trojúhelníkovou podstavou, atd. K hledání minima se používá několik základních operací: reflexe, expanze, kontrakce a redukce.

Nelderův-Meadův algoritmus začíná určením výchozího simplexu. Ve zdroji [3] se výchozí simplex obecně konstruuje jako matice U rozměru $(p + 1) \times p$, ve které řádky představují souřadnice jednotlivých vrcholů. Pro regulární výchozí simplex se stejnými hranami délky t vypadá matice U následovně

$$U = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ c & a & a & \dots & a \\ a & c & a & \dots & a \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a & a & a & \dots & c \end{bmatrix}, \quad (4.2)$$

kde

$$a = \frac{t(\sqrt{p+1} - 1)}{p\sqrt{2}} \quad (4.3)$$

a

$$c = \frac{t(m-1) + \sqrt{p+1}}{p\sqrt{2}}. \quad (4.4)$$

Následující popis algoritmu a odpovídající obrázky bude pro dva parametry. Simplexem bude trojúhelník a matice U bude rozměru 3×2 .

1 Označení vrcholů

Tři vrcholy (B , G , W) označíme tak, aby platila následující nerovnost $S(B) < S(G) < S(W)$ tří kritériálních funkcí daných vztahem (3.3). Označení vrcholů B , G a W je převzato z anglické literatury, kde B značí nejlepší vrchol (best), W nejhorší vrchol (worst) a G jako dobrý vrchol (good).

2 Střed dvou nejlepších vrcholů

K dalším výpočtům se používá střed strany BG neboli

$$M = \frac{B + G}{2}.$$

3 Reflexe (Obrázek 4.2)

Reflexe je zrcadlové promítnutí vrcholu W , které lze vyjádřit ve tvaru

$$R = M + \alpha(M - W). \quad (4.5)$$

Je vhodné zachovat regularitu simplexu, proto se obvykle volí $\alpha \approx 1$ [3, 7].

4 Expanze (Obrázek 4.3)

Expanzí do bodu E rozumíme

$$E = M + \gamma(R - M), \quad (4.6)$$

kde γ je větší než 2 [3] (nebo dle některé literatury alespoň větší než 1 [7]). Expanze se používá ke zrychlení algoritmu v případě, kdy máme dobrý směr, ale minimum je možná daleko.

5 Kontrakce (Obrázek 4.4)

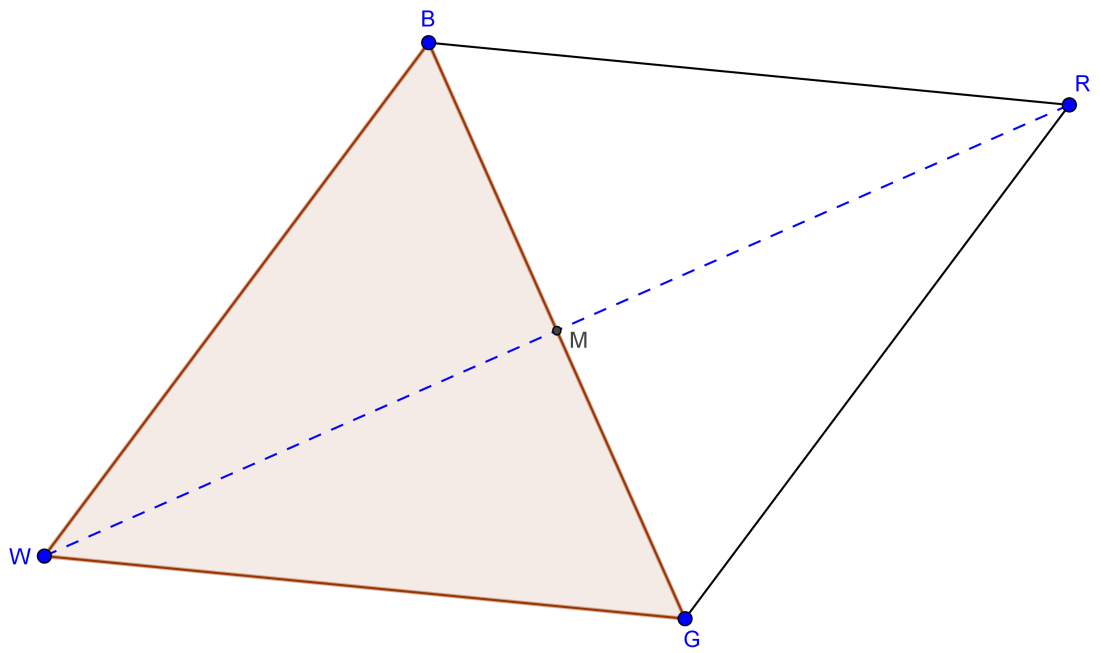
Kontrakce znamená zkrácení simplexu do bodu C . Rozlišujeme však dva druhy kontrakce: vnitřní a vnější. Vnitřní kontrakce vypadá následovně

$$C_1 = M + \mu(W - M) \quad (4.7)$$

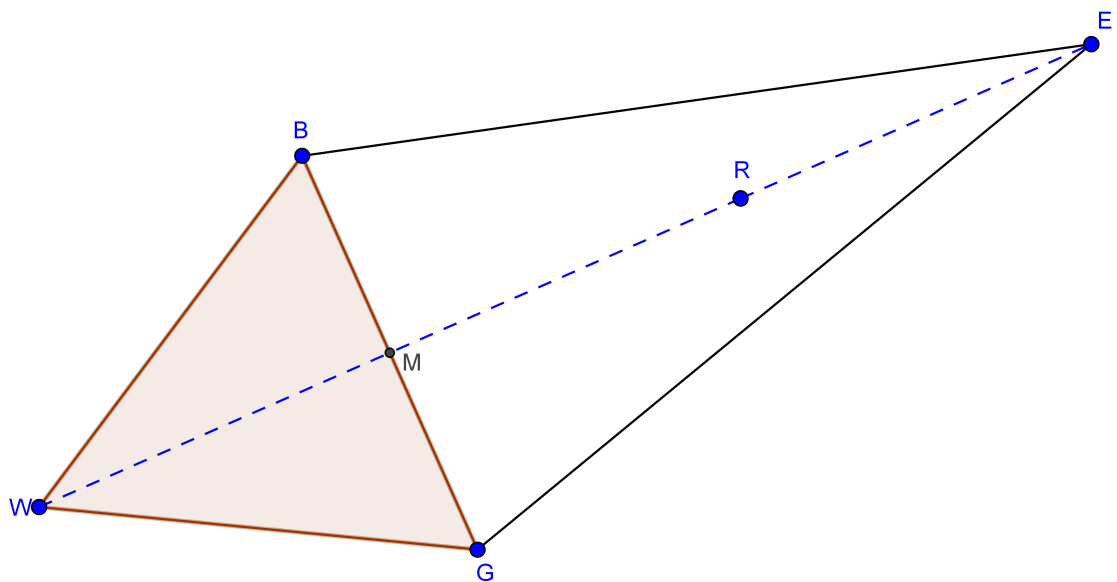
a vnější kontrakce je ve tvaru

$$C_2 = M + \mu(R - M). \quad (4.8)$$

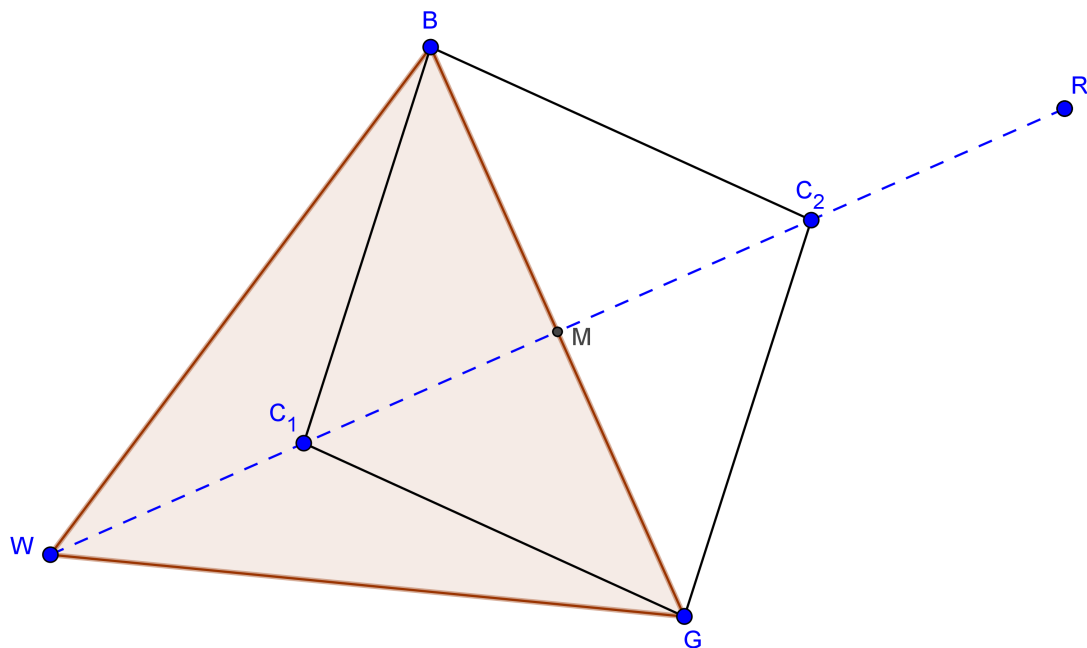
V obou případech se volí $\mu < 1$ [3, 7].



Obrázek 4.2: Operace reflexe (vrchol R) v Nelderově-Meadově metodě pro 2 parametry.



Obrázek 4.3: Operace expanze (vrchol E) v Nelderově-Meadově metodě pro 2 parametry.



Obrázek 4.4: Operace kontrakce v Nelderově-Meadově metodě pro 2 parametry. (Bod C_1 je vnitřní a bod C_2 je vnější kontrakce.)

6 Redukce (Obrázek 4.5)

Poslední operací cyklu je redukce, ve které dochází ke zkrácení simplexu kolem vrcholu B . Ostatní vrcholy jsou nahrazeny novými, pro které platí

$$G^* = B + \lambda(G - B) \quad (4.9)$$

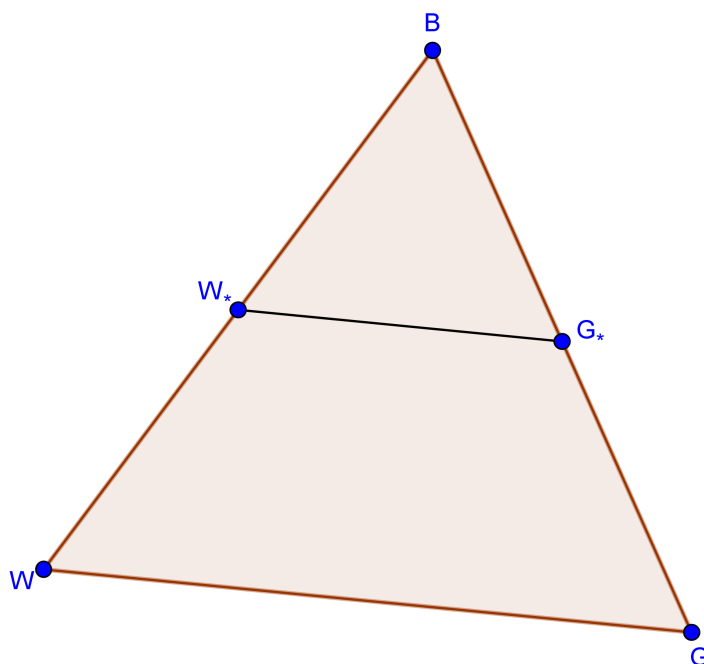
a

$$W^* = B + \lambda(W - B). \quad (4.10)$$

Parametr λ se obvykle volí roven 0,5 [3, 7], ale je třeba kontrolovat zda nedochází k nadměrnému krácení simplexu.

Stručný popis jedné iterace algoritmu je pak následující.

Pokud $S(R) < S(G)$, používáme buď reflexi nebo expanzi. Když $S(B) < S(R)$, tak vrchol W nahradíme vrcholem R , nebo pokud $S(B) \geq S(R)$ a zároveň $S(E) < S(R)$, tak vrchol W nahradíme vrcholem E a v ostatních případech vrchol W nahradíme vrcholem R . Pokud ovšem $S(R) \geq S(G)$, tak používáme buď kontrakci nebo redukci. Když $S(R) < S(W)$, tak vrchol W nahradíme vrcholem R . Když $S(C) < S(W)$, tak vrchol W nahradíme vrcholem C , jinak vrchol W nahradíme vrcholem W^* a vrchol G nahradíme vrcholem G^* .



Obrázek 4.5: Operace redukce (vrcholy W^* a G^*) v Nelderově-Meadově metodě pro 2 parametry.

Celý postup opakujeme, dokud zastavovací podmínka neukončí celý algoritmus. Pro názornost lze algoritmus vidět v Pseudokódu 1.

Nelder a Mead [18] doporučují ukončit algoritmus pokud budou platit obě následující podmínky

$$|S(W) - S(B)| < \epsilon_1 \quad (4.11)$$

a

$$\frac{1}{p+1} \sum_{j=1}^{p+1} (U_j(i) - U_j(i-1))^2 < \epsilon_2, \quad (4.12)$$

kde konstanty ϵ_1 a ϵ_2 jsou obvykle voleny 10^{-4} a 10^{-8} [3] a $U_j(i)$ označuje j -tý vrchol simplexu (j -tý řádek matice U) v i -té iteraci.

Rozšíření algoritmu na více než 2 parametry lze najít v [3, 7]. Hlavním rozdílem je odlišné pojetí výpočtu bodu M , jako těžiště všech vrcholů kromě toho nejhoršího (vrcholu W), což pro $p = 2$ představuje střed úsečky BG .

```

if  $S(R) < S(G)$  then
  | Příklad (i)
else
  | Příklad (ii)
end

```

<pre> Příklad (i) if $S(B) < S(R)$ then Nahrad' vrchol W vrcholem R else Spočti E a $S(E)$ if $S(E) < S(R)$ then Nahrad' vrchol W vrcholem E else Nahrad' vrchol W vrcholem R end end </pre>	<pre> Příklad (ii) if $S(R) < S(W)$ then Nahrad' vrchol W vrcholem R end Spočti C a $S(C)$ if $S(C) < S(W)$ then Nahrad' vrchol W vrcholem C else Spočti G^* a W^* Nahrad' vrchol W vrcholem W^* Nahrad' vrchol G vrcholem G^* end end </pre>
---	--

Pseudokód 1: Jedna iterace Nelderova-Meadova algoritmu napsaná v pseudokódu

4.1.3 Metody používající náhodná čísla

Tento typ metod je vhodný i pro složitější regresní funkce $f(x, \beta)$, při nichž ostatní metody většinou selhávají. Při své jednoduchosti jsou následující metody poměrně efektivní a nevyžadují dobré počáteční hodnoty parametrů. Náhodná respektive pseudonáhodná čísla se generují z normovaného normálního rozdělení $\mathbb{N}(0, 1)$, případně z rovnoměrného rozdělení $\mathbb{R}(0, 1)$.

Algoritmus CRS (Controlled Random Search)

Algoritmus CRS [3, 13, 14, 15] je kombinací simplexové metody a náhodného hledání. Používá se matice W rozměru $(10p + 10) \times p$, kde řádky této matice jsou náhodně vybrané body b z prostoru parametrů Ω . Z matice W pak náhodně vybíráme řádky a tvoříme z nich novou matici U rozměru $p + 1 \times p$. Postup hledání minima má několik kroků:

- 1 Nalezneme $b^B = \min S(W_i)$ a $b^W = \max S(W_i)$, přičemž hledáme mezi všemi danými kombinacemi parametrů b neboli přes všechny řádky matice W . S označuje kriteriální funkci (3.3).

2 Položíme $U_1 = b^B$. Ze zbylých řádků matice W náhodně vybereme p řádků a přeznačíme je jako $U_i, i = 2, \dots, (p + 1)$.

3 Z vrcholů $U_i, i = 1, \dots, (p + 1)$ sestavíme simplex a najdeme nový bod U_R jako reflexi vrcholu U_{p+1} . Stejně jako u Nelderovy-Meadovy metody s dosazenou konstantou $\alpha = 1$ do vztahu (4.5) platí

$$U_R = 2M - U_{p+1} = 2 \sum_{i=1}^p \frac{U_i}{p} - U_{p+1},$$

kde M je těžiště vrcholů $U_i, i = 1, \dots, p$.

4 Pokud platí $S(U_R) \geq S(b^W)$, přejdeme opět na krok 2. Jinak se místo b^W v matice W dosadí U_R .

5 Ukončíme iteraci a zkontrolujeme zastavovací podmínku. Pokud nedojde k zastavení algoritmu, opakujeme od kroku 1. Jako vhodnou zastavovací podmínku lze použít rozdíl nejhorsího a nejlepšího vrcholu simplexu neboli $|S(W) - S(B)| < \epsilon$ [15].

Algoritmus GSA (Generalized Simulation Annealing)

Algoritmus GSA [3, 16] vychází z konstrukce náhodného bodu X na povrchu p -rozměrné koule o poloměru Δr a středu $b(i)$. Nutná je opět znalost počátečního odhadu parametrů, tj. $b(0)$. Celý algoritmus GSA pak má tyto kroky

1 Nalezneme náhodný směr Z . Pro složky tohoto vektoru platí

$$Z_j = \frac{N_j}{\sqrt{\sum_{l=1}^p N_l^2}}, j = 1, \dots, p,$$

kde $N_j, j = 1, \dots, p$ jsou náhodná čísla z normovaného normálního rozdělení.

2 Určíme nový bod X vztahem

$$X = \Delta r Z + b(i).$$

Poloměr Δr se obvykle volí 0,15 [3].

3 Jestli je $\Delta S = S(X) - S(b(i)) \leq 0$ (S ze vztahu (3.3)), dosadíme

$$b(i + 1) = X$$

a i -tá iterace je u konce. Pokud ale $\Delta S > 0$, přijmeme nový bod X jako $b(i+1)$ s pravděpodobností $P_p = e^{-\varkappa \Delta S}$, kde \varkappa je kladná konstanta ovlivňující rychlost konvergence (obvykle $\varkappa = 3, 5$ [3]). V případě nepřijetí nového bodu X přecházíme na krok 1 tj. ke generování jiného náhodného směru Z .

- 4 Po přijetí nového bodu $b(i+1)$ ukončíme i -tou iteraci a zkontrolujeme zastavovací podmínku. Jako zastavovací podmínku použijeme normu v eukleidovském smyslu mezi dvěma za sebou následujícími iteracemi $i+1$ a i , tj.

$$\sum_{j=1}^p (b_j(i) - b_j(i+1))^2 < \epsilon.$$

Algoritmus ARST (Adaptive Random Search Technique)

Algoritmus ARST [3] je opět jednoduchý a poměrně dobře fungující algoritmus. Jestliže známe nějaký počáteční odhad parametrů $b(0)$, můžeme jednu iteraci algoritmus provést v těchto krocích:

- 1 Vypočteme nový bod X podle vztahu

$$X = b(i) + \frac{D(2R-1)^K}{K}, \quad (4.13)$$

kde D je velikost přírůstku (obvykle $0, 5$ [3]), R je náhodné číslo z rovnoměrného rozdělení $\mathbb{R}(0, 1)$ a K je parametr, který se přizpůsobuje úspěšnosti předchozích iterací (čím větší K , tím je rozdělení druhého sčítance ve výrazu (4.13) špičatější a jeho rozptyl je menší).

- 2 Jestliže $S(X) < S(b(i))$ (S ze vztahu (3.3)), tak dosadíme

$$b(i+1) = X.$$

Pokud ne, tak provedeme případnou změnu parametru K a pokračujeme opět krokem 1.

- 3 Po přijetí nového bodu $b(i+1)$ zkontrolujeme zastavovací podmínku a případně celý postup opakujeme. Jako zastavovací podmínku použijeme normu v eukleidovském smyslu mezi dvěma za sebou následujícími iteracemi $i+1$ a i , tj.

sebou následujícími iteracemi $i + 1$ a i , tj.

$$\sum_{j=1}^p (b_j(i) - b_j(i + 1))^2 < \epsilon.$$

Parametr K adaptujeme dle následujících pravidel:

- Na začátku algoritmu - $K = 1$.
- Po pěti úspěšných iteracích - $K = 3$.
- Po dalších patnácti úspěšných iteracích - $K = 5$.
- Po dalších deseti úspěšných iteracích - $K = 7$.
- Pokud je konvergence pomalá, tak se provádí půlení a zaokrouhlení K až maximálně na $K = 1$.

Algoritmus ARS (Adaptive Random Search)

Algoritmus ARS [3, 17] vychází z algoritmu ARST a snaží se lépe využívat adaptace. V případě, že známe počáteční odhad parametrů $b(0)$, tak určíme pět vektorů směrodatné odchylky ve tvaru

$$\sigma_1 = 0, 3l + 2|b(0)|, \quad (4.14)$$

$$\sigma_k = 0, 1^{(k-1)}\sigma_1, \quad k = 2, 3, 4, 5, \quad (4.15)$$

kde l označuje jednotkový vektor. V případě, že známe pouze nějaké rozmezí jednotlivých b_j , $j = 1, \dots, p$, lze za σ_1 použít rozdíl horní a dolní meze [17].

Poté se pro každý vektor σ_k , $k = 1, \dots, 5$ vygeneruje $100/k$ -krát bod

$$X = b(i) + \sigma_k N,$$

přičemž N je vektor nezávislých náhodných veličin z normovaného normálního rozdělení $\mathbb{N}(0, 1)$. Jestliže je $S(X) < S(b(i))$, tak dosadíme $b(i) = X$.

Do následující iterace vstupuje vektor σ_k , $k = 1, \dots, 5$, který nejvíce zmenší hodnotu $S(b(i))$. Jako odhad $b(i + 1)$ v další iteraci volíme takové X , které odpovídá minimální hodnotě $S(b(i))$.

Sami autoři této metody (Pronzato a kolektiv [17]) doporučují algoritmus ukončit v případě, kdy pro pět po sobě následujících iterací jsou nalezeny stejné hodnoty σ_k , $k = 1, \dots, 5$.

4.2 Derivační metody

Derivační metody [3, 4, 5] pro metodu nejmenších čtverců jsou iterační a jejich řešení je silně závislé na zvolení počátečních hodnot parametrů b , tj. $b(0)$. V každé iteraci se uvažuje předchozí hodnota parametrů $b(i)$, ke které se přičítá vhodný přírůstkový vektor.

$$b(i+1) = b(i) + \Delta(i) \quad (4.16)$$

Obecně hledání minima $S(b)$ (3.3) probíhá následovně.

- 1 Stanovení počátečních hodnot parametrů $b(0)$.
- 2 Nalezení vhodného směrového vektoru V_i .
- 3 Určení skaláru α_i , tak aby přírůstkový vektor $\Delta(i) = \alpha_i V_i$ byl přijatelný¹.
- 4 Ověření zastavovací podmínky.

1 Počáteční hodnoty $b(0)$

Neexistuje žádný obecný postup pro určení počátečních hodnot parametrů. Vycházet však můžeme z fyzikální či geometrické interpretace. Pokud jsou modely vnitřně lineární, lze jejich odhad parametrů metodou nejmenších čtverců pro zlinearizovaný model brát jako počáteční hodnoty pro model nelineární.

2 Směrový vektor V

Derivace $S(b)$ v bodě $\Delta = b + \alpha V$ podle skaláru α je následující

$$\frac{\partial S(b)}{\partial \alpha} = \left(\frac{\partial S(b)}{\partial b} \right)^T \frac{\partial b}{\partial \alpha} = g^T(b)V, \quad (4.17)$$

kde g je gradient v místě Δ . Pro $\alpha \rightarrow 0$ vypočítáme z rovnice (4.17) tzv. směrovou derivaci ve tvaru

$$S_D = \left. \frac{\partial S(b)}{\partial \alpha} \right|_{\alpha \rightarrow 0} = g^T V. \quad (4.18)$$

¹Za přijatelný vektor $\Delta(i)$ se obvykle považuje vektor splňující podmínku

$$S(b(i) + \Delta(i)) < S(b(i)).$$

Některé algoritmy však dovolují i rovnost či nepatrný růst $S(b(i) + \Delta(i))$ oproti $S(b(i))$.

Gradient je směr nejstrmějšího růstu, proto za podmínku přijatelnosti směrového vektoru V považujeme $g^T V < 0$. Pokud je V přijatelný, tak existuje pozitivně definitní matice R , pro kterou

$$V = -Rg. \quad (4.19)$$

Tudíž směrová derivace dosazením do vztahu (4.18)

$$S_D = -g^T Rg. \quad (4.20)$$

3 Optimální skalár α

K určení optimálního α provedeme Taylorovo rozvoj druhého řádu výrazu $S(b + \alpha V)$

$$S(b + \alpha V) = S(b) + \alpha g^T V + \frac{\alpha^2}{2} V^T H V, \quad (4.21)$$

kde H je Hessova matice definovaná v příloze (A.5). Tato rovnice je vzhledem k α přibližně kvadratická, takže lze pro stanovení optimálního skaláru α uvažovat, že $\frac{\partial S(b + \alpha V)}{\partial \alpha} = 0$. Vyjde proto

$$\alpha^* = \frac{-\frac{\partial S(b)}{\partial \alpha}}{\frac{\partial^2 S(b)}{\partial \alpha^2}} = -g^T V (V^T H V)^{-1} \quad (4.22)$$

a dosazením z rovnice (4.19) dostaneme tzv. Raleighův koeficient

$$\alpha^* = g^T Rg (g^T R^T H Rg)^{-1}. \quad (4.23)$$

Další možností hledání optimální α je jednosměrná minimalizace $S(b)$ ve směru V .

4 Zastavovací podmínka

Přírozenou podmínkou optima b je to, že gradient je nulový. Proto lze ukončit iterace pokud je norma gradientu dostatečně malá, tj.

$$\|g\|^2 = \sum_{j=1}^p g_j^2 < \epsilon. \quad (4.24)$$

4.2.1 Gaussovy-Newtonovy metody

Podrobnější popis metod zmíněných v této sekci lze nalézt v [2, 3, 8, 9, 11].

Vhodný směrový vektor může být určen pomocí rovnice (4.21) pro $\alpha = 1$. Pokud uvažujeme $\frac{\partial S(b+V)}{\partial V} = 0$, tak optimální směrový vektor $V_i = N_i$ s využitím (A.5) a (A.4) vyjde ve tvaru

$$N_i = -H^{-1}g = (J^T J + B)^{-1} J^T e, \quad (4.25)$$

kde J je Jakobiho matice (A.3), B je matice (A.6) a $e = Y - f(x, b)$ je vektor reziduí. Dosazením zpátky do vztahu (4.23) dopočteme že $\alpha^* = 1$, a tudíž směrový vektor je přímo přírůstkový vektor, $N_i = \Delta_i$. Výše uvedená metoda se nazývá Newtonovou metodou. Tato metoda často nekonverguje dostatečně rychle a navíc je nutné dopočítávat druhé derivace pro matici B ze vztahu (A.6). Proto se častěji používá Gaussova-Newtonova metoda, která spočívá v zanedbání matice B . Směrový vektor $V_i = L_i$ má pak tvar

$$L_i = (J^T J)^{-1} J^T e. \quad (4.26)$$

Dosazení aproximace $H \approx J^T J$ do vztahu (4.23) vede opět k $\alpha^* = 1$. Gaussova-Newtonova metoda poměrně dobře konverguje pokud platí alespoň jedna z následujících podmínek

- Rezidua e_i jsou malá, tzn. počáteční odhad parametrů $b(0)$ byl velmi dobrý.
- Regresní funkce $f(x, \beta)$ je téměř lineární, tj. norma Hessovy matice je malá a prvky matice jsou téměř nulové.
- Rezidua e_i mají střídavá znaménka, takže B je přibližně nulová matice.

Pokud metoda nekonverguje nebo konverguje pomalu, lze použít různé modifikace Gaussovy-Newtonovy metody, případně tzv. kvazinewtonovské metody. Tyto metody lze využít k zpřesnění matice $J^T J$, aby se blížila k Hessově matici, nebo při singularitě matice $J^T J$ a dalších numerických problémech. Více o modifikacích Gaussovy-Newtonovy metody lze nalézt například v [3, 27].

4.2.2 Metody Marquardtova typu

V metodách Marquardtova typu [3, 4, 11] je směrový vektor zvolen jako směr největšího spádu, tj. $V_i = -g$. Ze vztahu (4.19) lze vyvodit, že $R = I$, kde I je jednotková matice odpovídající velikosti. Dosazením do (4.23) dostaneme Raleighův koeficient ve tvaru

$$\alpha^* = g^T g [g^T H g]^{-1} \approx g^T g [g^T (J^T J)^{-1} g]^{-1} \quad (4.27)$$

Přírůstkový vektor je $\Delta_i = -\alpha^* g$, což odpovídá gradientním metodám. Marquardtovy metody efektivně využívají kombinace směrů Newtonovy metody N_i (4.25) nebo její aproximace L_i (4.26) společně se směrem $-g$.

Nejznámější je Marquardtova metoda [3, 4, 11], která počítá směrový vektor dle vztahu

$$V_i(\lambda) = (J^T J + \lambda D_i^T D_i)^{-1} J^T e, \quad (4.28)$$

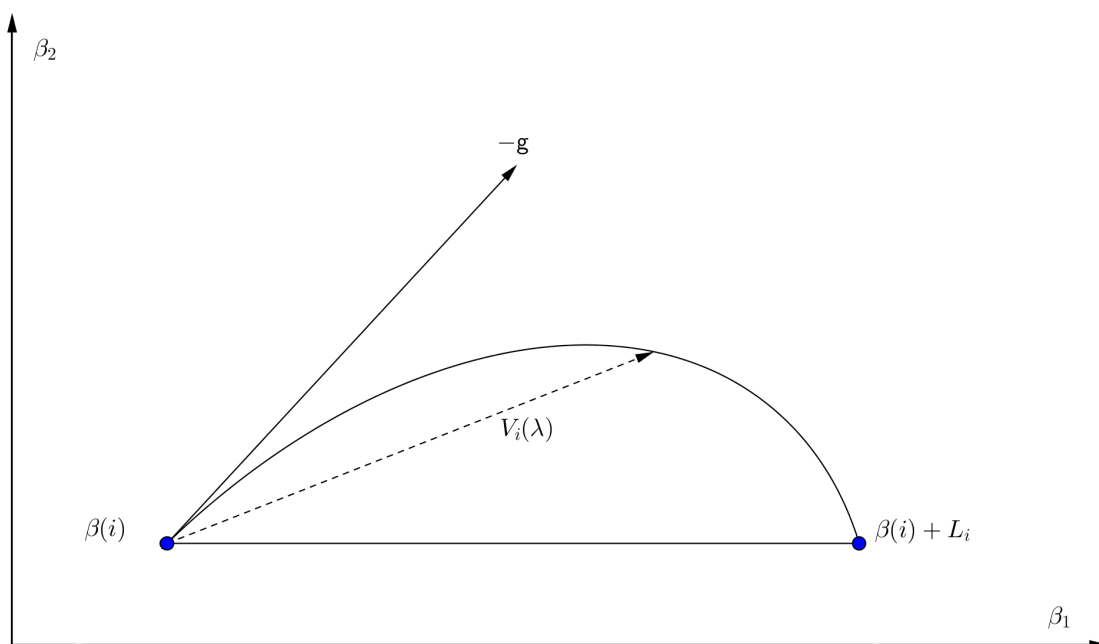
kde λ je parametr, který se mění v každé iteraci metody a D_i je diagonální matice. Diagonální prvky matice D_i se obvykle volí rovny diagonálním prvkům $(J^T J)$. Vektor $V_i(\lambda)$ má následující vlastnosti

- Norma $\|V_i(\lambda)\|$ je klesající funkcí dle λ , která konverguje k 0 pro $\lambda \rightarrow \infty$
- Kosinus úhlu mezi $V_i(\lambda)$ a $-g_i$ je rostoucí funkce.
- Kosinus úhlu mezi směrem L_i a $-g_i$ je klesající funkce.

Jednotlivé modifikace Marquardtovy metody se liší v určování parametru λ . Ideální volba parametru λ zajišťuje regulárnost matice $(J^T J + \lambda D_i^T D_i)$, krácení kroku $V_i(\lambda)$ při oddálení od směru N_i (L_i) a možnost regulace mezi těmito směry.

Volba parametru λ

- 1 Původní algoritmus
 - počáteční nastavení: $\lambda_0 = 0.01$
 - úspěšný krok: $\lambda_{i+1} = \lambda_i/10$
 - neúspěšný krok: $\lambda_{i+1} = 10\lambda_i$



Obrázek 4.6: Znázornění funkce $V_i(\lambda)$ v Marquardtově metodě pro dva parametry

2 Nashův algoritmus

počáteční nastavení: $\lambda_0 = 10^{-4}$

úspěšný krok: $\lambda_{i+1} = \max(10^{-6}, \lambda_i/10)$

neúspěšný krok: $\lambda_{i+1} = \max(10^{-6}, 4\lambda_i)$

Příčemž jako úspěšný krok se považuje krok, který zmenší hodnotu $S(b)$. Marquardtova metoda má také řadu nevýhod. Mezi největší patří nutnost v každém kroku hledat novou inverzní matici $(J^T J + \lambda D_i^T D_i)^{-1}$ kvůli změně λ .

Kapitola 5

Praktická část

V této kapitole jsou popsány všechny sady dat, nastavení počátečních podmínek algoritmů a konstant v některých algoritmech a vypsání výsledky s komentářem. Všechny následující použité datové sety je možno vidět v příloze B.

5.1 Data

5.1.1 Puromycin data

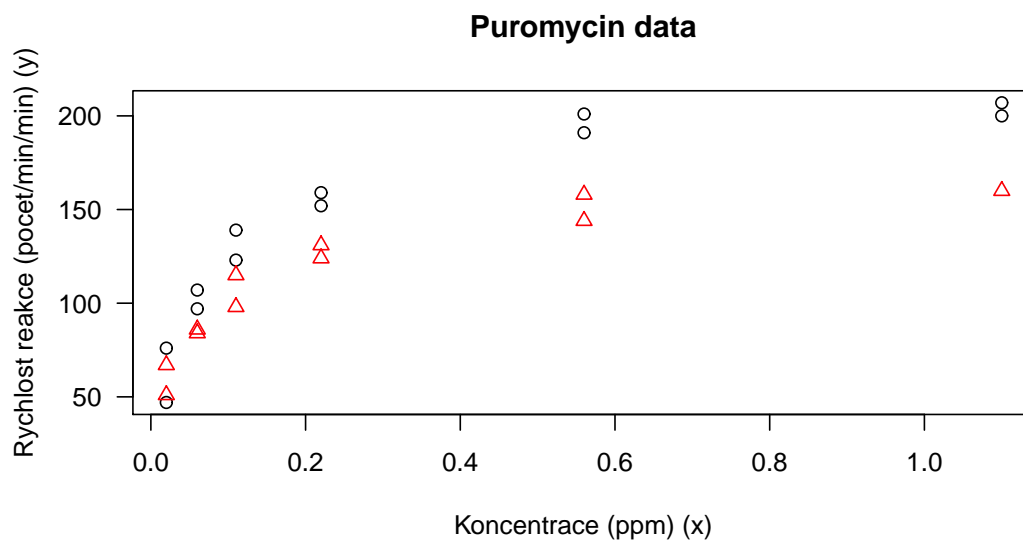
Jedná se o data, která jsou součástí R-knihovny dataset [28]. Více o těchto datech lze nalézt v [4, 5].

Data byla naměřena při chemickém experimentu prováděném Margaret Treloarovou v roce 1974 [19], při kterém je rychlost reakce rate závislá na koncentraci conc . Rychlost reakce byla měřena nepřímo pomocí počtu impulzů radioaktivního záření, proto se rychlost udává v počtu za minutu. Koncentrace je uváděna v počtu částic na jeden milion (ppm z anglického parts per million). Data byla navíc měřena za dvojnásobných podmínek: s puromycinem a bez něho. Puromycin je zde používán ke zrychlení chemické reakce.

Jako regresní funkci v tomto případě použijeme model vytvořený Leonorem Michaelisem a Maud Mentenovou [31]

$$f(x; \theta_1, \theta_2) = \frac{\theta_1 x}{\theta_2 + x}, \quad (5.1)$$

který je definovaný pro $x \geq 0$ a je lineární vzhledem k parametru θ_1 .

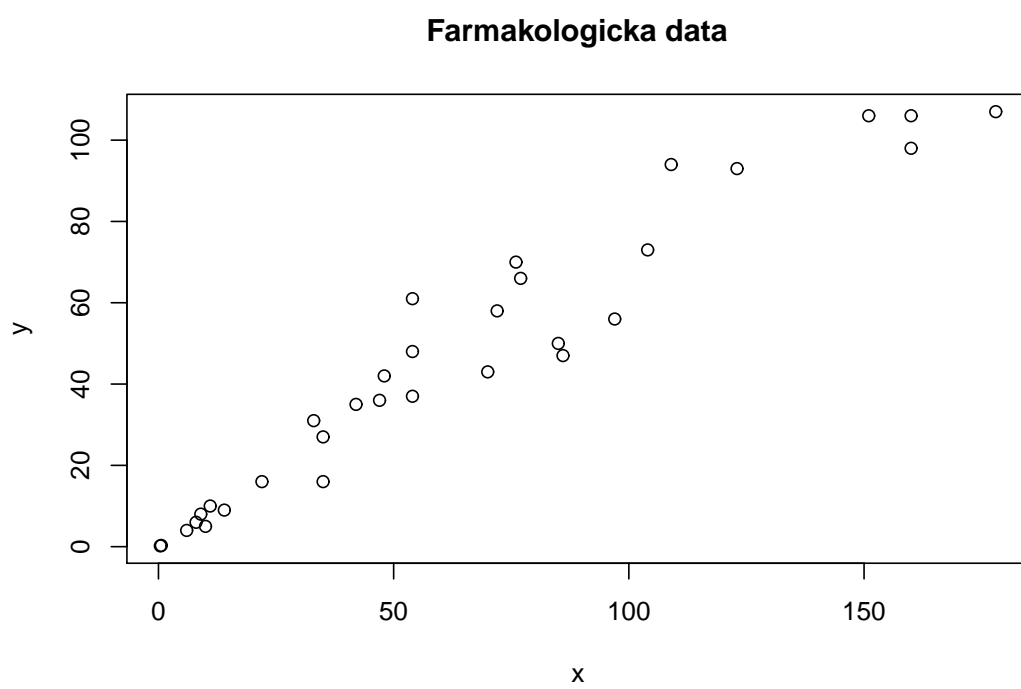


Obrázek 5.1: Vykreslení dat s a bez puromycinu

5.1.2 Farmakologická data

Tato data, která lze stáhnout na [30], jsou součástí knihy [4]. Jde o farmakologický jev zkoumaný doc. RNDr. Karlem Zvárou, CSc. a doc. Ivo Janků, DrSc. Regresní funkce je v tomto případě složitější a má tvar

$$f(x; \beta, \gamma) = \frac{1}{\gamma}(x + (625 - x)(1 - \exp(-\beta x / (625 - x)))) \quad (5.2)$$

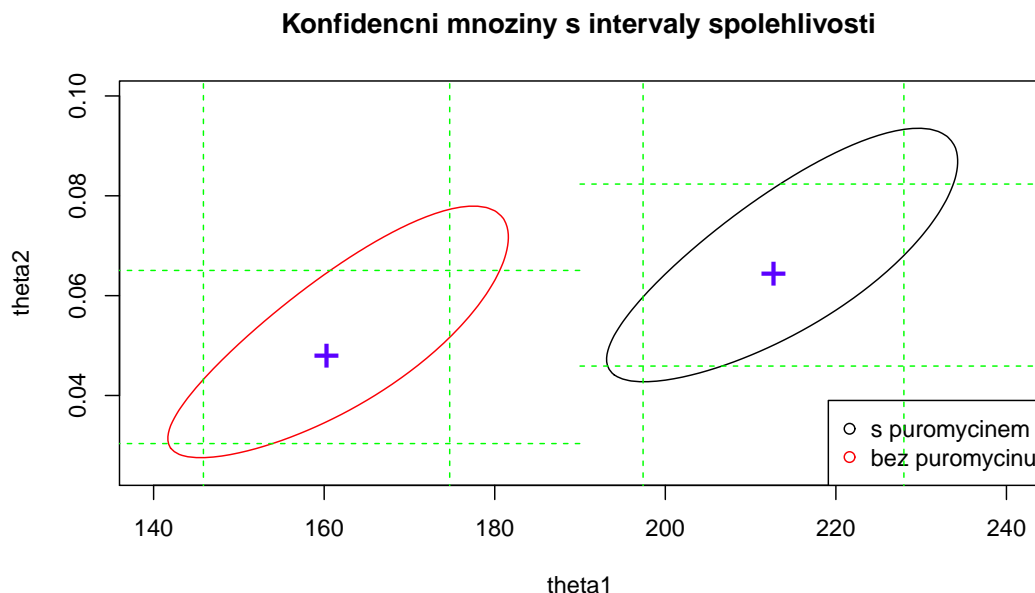


Obrázek 5.2: Vykreslení farmakologických dat

5.2 Konfidenční množiny a profilové diagramy

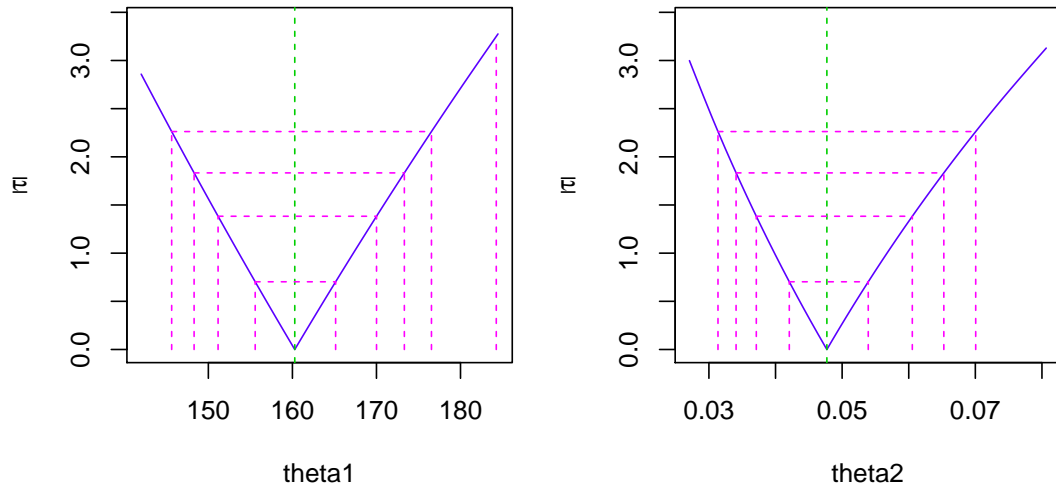
Níže na obrázcích 5.3 a 5.6 jsou vykreslené 95% konfidenční množiny (3.6), kde $\alpha = 0,05$. Konfidenční množiny pro puromycin data (obrázek 5.3) mají více eliptický tvar než konfidenční množina pro farmakologická data (obrázek 5.6), která se zdá být více zdeformovaná. Z těchto vlastností konfidenčních množin lze usuzovat, že farmakologická data jsou více nelineární a je vhodné pro ně používat pouze nelineární regresní model. Avšak pro puromycin data lze zkusit lineární aproximaci a počítat s daty jako s lineárními. Nicméně výsledky lineární aproximace, které lze vidět dále, ukazují, že tento způsob výpočtu není příliš vhodný. Ke konfidenčním množinám byly přidány ještě 95% intervalové odhady (viz (A.9)) pro jednotlivé parametry regresního modelu.

Na profilových diagramech 5.4, 5.5 a 5.7 jsou fialově znázorněné intervaly spolehlivosti se spolehlivostí popořadě 99 %, 95 %, 90 %, 80 % a 50 %. Závěr z vykreslených profilových diagramů je podobný jako u konfidenčních množin. V úloze s farmakologickými daty se více projevuje nelinearita.



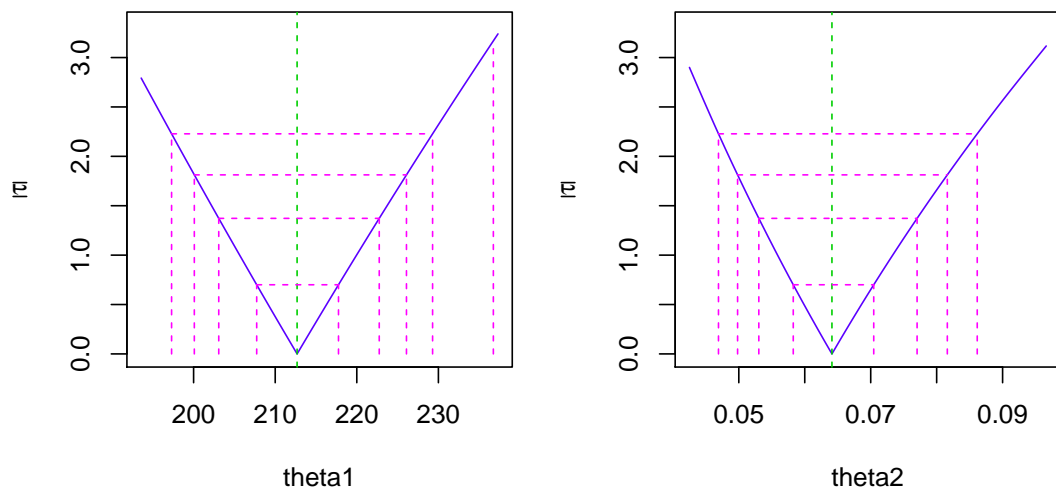
Obrázek 5.3: Vykreslení konfidenčních množin včetně intervalových odhadů pro jednotlivé parametry θ_1 nebo θ_2 pro puromycin data. Intervalové odhady jsou znázorněny zelenou čárkovanou čarou.

Data bez puromycinu



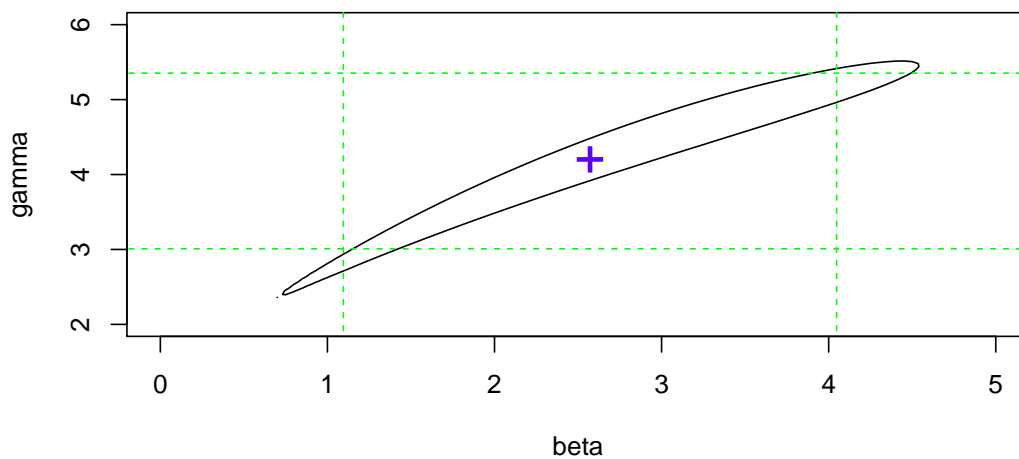
Obrázek 5.4: Profilový diagram pro data bez puromycinu.

Data s puromycinem



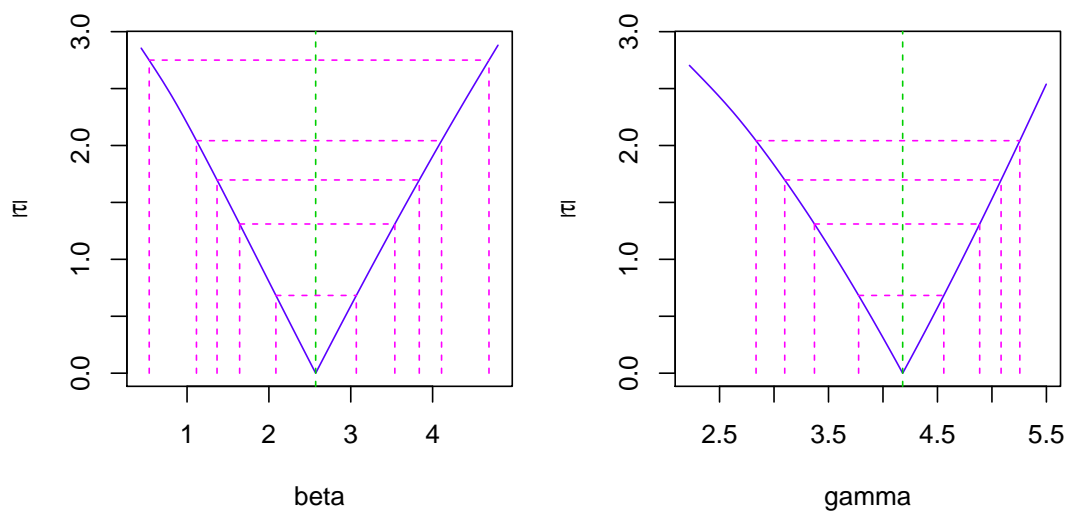
Obrázek 5.5: Profilový diagram pro data s puromycinem.

Konfidencni mnozina s intervaly spolehlivosti



Obrázek 5.6: Vykreslení konfidencních množin včetně intervalových odhadů pro jednotlivé parametry β nebo γ pro farmakologická data. Intervalové odhady jsou znázorněny zelenou čárkovanou čarou.

Farmakologicka data

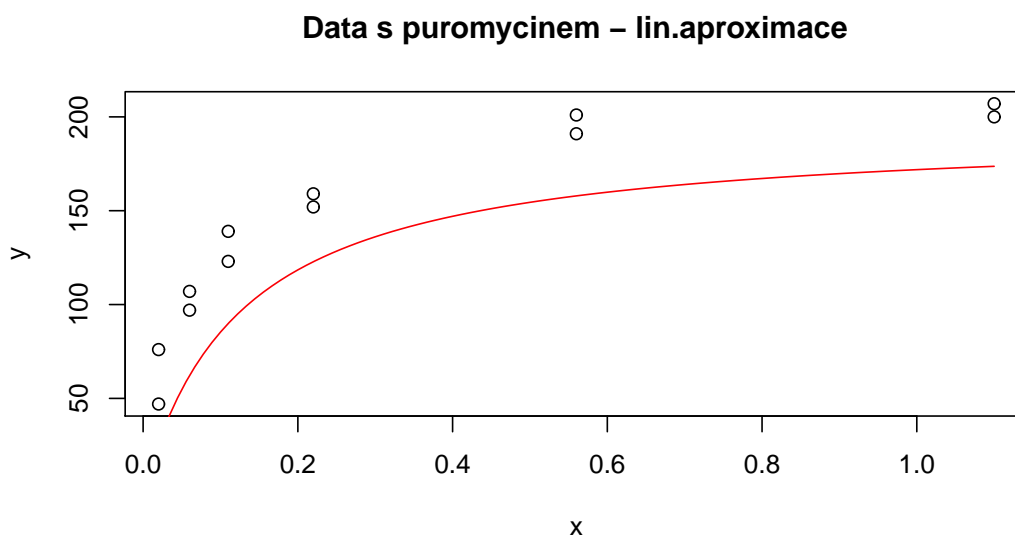


Obrázek 5.7: Profilový diagram pro farmakologická data.

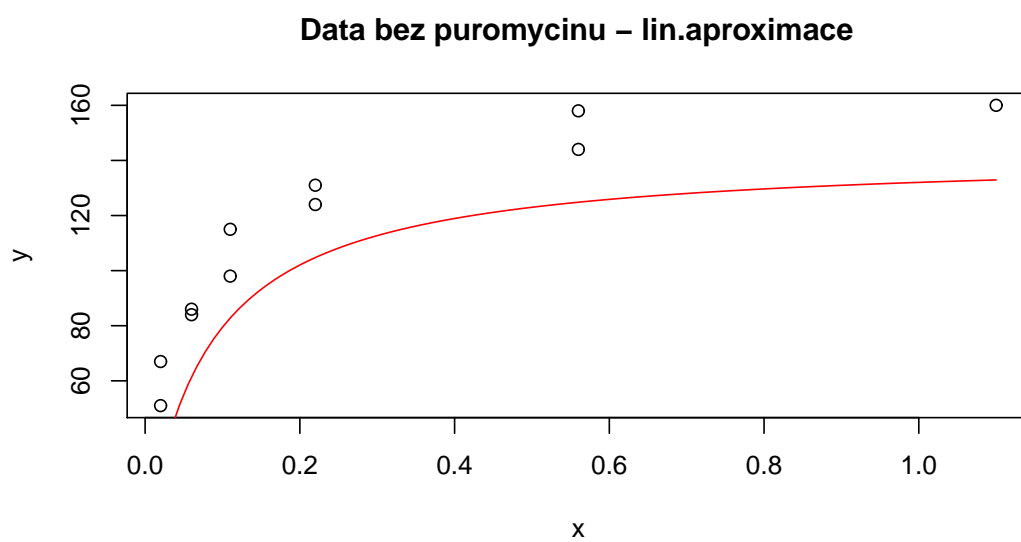
5.3 Lineární aproximace

Jak už bylo uvedeno výše, tak lineární aproximace bude provedena pouze pro puromycin data. Metoda lineární aproximace spočívá v rozepsání regresní funkce $f(x, \beta)$ pomocí Taylorova rozvoje prvního řádu (viz (A.7)), kde jako x_0 je použit průměr všech nezávisle proměnných. Poté je možné provést klasický lineární odhad regresních parametrů metodou nejmenších čtverců. Další informace o lineární regresi lze nalézt například v [3, 4, 6, 10] a mnoha dalších.

Výsledné odhady parametrů touto metodou pro data s puromycinem jsou $b_1 = 193,702$ a $b_2 = 0,126999$ a pro data bez puromycinu to jsou odhady $b_1 = 142,502$ a $b_2 = 0,0793064$. Tyto výsledky už dle pohledu na obrázky 5.8 a 5.9 nejsou příliš dobré.



Obrázek 5.8: Výsledek metody lineární aproximace pro data s puromycinem.



Obrázek 5.9: Výsledek metody lineární aproximace pro data bez puromycinu.

5.4 Porovnání jednotlivých algoritmů - simulovaná data

Vytvoříme dvě nové sady dat, které lze vidět v tabulkách B.3 a B.4 v příloze B. Použijeme k tomu regresní funkce, které jsou zmíněné v 5.1, pouze přidáme náhodnou složku s nulovou střední hodnotou a malým konstantním rozptylem. Oba regresní modely tedy vypadají následovně

$$Y = \frac{b_1 x}{b_2 + x} + \varepsilon_1, \varepsilon_1 \sim \mathbb{N}(0, 1) \quad (5.3)$$

a

$$Y = \frac{1}{b_1} (x + (625 - x)(1 - \exp(-b_2 x / (625 - x)))) + \varepsilon_2, \varepsilon_2 \sim \mathbb{N}(0, 25). \quad (5.4)$$

Jako teoretické hodnoty v modelu (5.3) jsou použity hodnoty parametrů $b_1^* = 99,5861$ a $b_2^* = 0,0345$. V modelu (5.4) jsou teoretické hodnoty $b_1^* = 6,6578$ a $b_2^* = 10,0845$.

V následující části práce budeme porovnávat metody nelineární regrese, které jsou popsány v kapitole 4. Metody porovnáme dle přesnosti odhadu parametrů, počtu iterací nutných k zastavení algoritmu zastavovací podmínkou a dle celkového času, po který algoritmus běží. Pro oba datové sady budou navíc použité dva druhy počáteční odhadů $\beta(0)$ - jeden blízký teoretickému řešení β^* a druhý vzdálenější β^* . Algoritmy metod používajících náhodná čísla byly spuštěny stokrát a výsledné odhady parametrů, výpočetní čas a počet iterací byly pak zprůměrovány.

Hookeova-Jeevesova metoda

Jako zastavovací podmínka byla použita nerovnost

$$|\delta_i| \leq 0,0001.$$

Velikost kroku $|\delta_i|$ je půlena po každé iteraci jako ve zdroji [7]. Pro data s modelem (5.3) byly jako dobré počáteční odhady parametrů zvoleny $b_1(0) = 103$ a $b_2(0) = 0,02$ a jako vzdálenější, chybné odhady $b_1(0) = 70$ a $b_2(0) = 1$. Pro data s modelem (5.4) byly zvoleny dobré odhady jako $b_1(0) = 8$ a $b_2(0) = 12$ a jako chybné $b_1(0) = 20$ a $b_2(0) = 25$.

Nelderova-Meadova simplexová metoda

Pro zastavení tohoto algoritmu musí být splněné dvě zastavovací podmínky (4.11) a (4.12)

$$|S(W) - S(B)| < 10^{-4}$$

a

$$\frac{1}{p+1} \sum_{j=1}^{p+1} (U_j(i) - U_j(i-1))^2 < 10^{-8}.$$

Hodnoty simplexových konstant ze vztahů (4.5), (4.6), (4.7), (4.8), (4.9) a (4.10) jsou následovně pevně zvolené: $\alpha = 1$, $\mu = 0,55$, $\gamma = 2,9$ a $\lambda = 0,5$.

Počáteční podmínky jsou v tomto algoritmu specifické tím, že potřebujeme celý počáteční simplex, v našem případě trojúhelník. Pro model (5.3) zvolíme tři počáteční simplex: simplex dle (4.2) s $t = 25$, simplex dle (4.2) s $t = 150$ a trojúhelník tvořený body se souřadnicemi $A_1 = [50; 1]$, $B_1 = [150; 1]$ a $C_1 = [100; 0]$. Pro model (5.4) zvolíme také tři počáteční simplex: simplex dle (4.2) s $t = 25$, simplex dle (4.2) s $t = 150$ a trojúhelník tvořený body se souřadnicemi $A_2 = [0; 5]$, $B_2 = [15; 5]$ a $C_2 = [8; 20]$. Pro simplex zadané konkrétními třemi body tak platí, že teoretické řešení β^* leží uvnitř daného trojúhelníku.

CRS

Pro metodu CRS byl jako zastavovací podmínka použit rozdíl mezi nejhorším a nejlepším vrcholem simplexu čili

$$|S(W) - S(B)| < 0,0001.$$

Navíc, aby došlo k ukončení algoritmu, bylo nutné přidat podmínku pro počet neúspěšných výběrů z matice W . Tento počet neúspěchů byl nastaven na velkou hodnotu tj. 1 000 000. V metodě CRS se používá výběr z matice W , takže všechny počáteční odhady nemohou být stejné, protože by operace reflexe nenašla žádný nový bod. Všechny možné počáteční odhady obou parametrů byly vygenerovány z rovnoměrných spojitých rozdělení na daných intervalech.

Pro data s modelem (5.3) pro dobré počáteční odhady parametru $b_1(0)$ je generováno z intervalu $[98, 108]$ a parametru $b_2(0)$ z intervalu

[0, 01; 0, 04]. Pro chybné odhady těchto dat je $b_1(0)$ generováno z intervalu [65, 75] a $b_2(0)$ z intervalu [0, 8; 1, 2]. Pro data s modelem (5.4) pro dobré počáteční odhady parametru $b_1(0)$ je generováno z intervalu [6, 10] a parametru $b_2(0)$ z intervalu [10, 14]. Pro chybné odhady těchto dat je $b_1(0)$ generováno z intervalu [18, 22] a $b_2(0)$ z intervalu [23, 27].

GSA

V metodě GSA jako zastavovací podmínka slouží nerovnost

$$\sum_{j=1}^p (b_j(i) - b_j(i+1))^2 < 0,0001.$$

Opět musela být přidána podmínka pro počet neúspěšných pokusů, která byla nastavena na hodnotu 100, což se jeví jako dostatečné omezení. Pro data s modelem (5.3) byly jako dobré počáteční odhady zvoleny hodnoty $b_1(0) = 103$ a $b_2(0) = 0,02$ a jako vzdálenější, chybné odhady $b_1(0) = 70$ a $b_2(0) = 1$. Pro data s modelem (5.4) byly zvoleny dobré odhady jako $b_1(0) = 8$ a $b_2(0) = 12$ a jako chybné $b_1(0) = 20$ a $b_2 = 25$.

Parametry \varkappa a Δr jsou volené, tak jak je doporučováno v [3] neboli $\varkappa = 3,5$ a $\Delta r = 0,15$.

ARST

V metodě ARST jako zastavovací podmínka opět slouží nerovnost

$$\sum_{j=1}^p (b_j(i) - b_j(i+1))^2 < 0,0001.$$

Pro data s modelem (5.3) byly jako dobré počáteční odhady zvoleny $b_1(0) = 103$ a $b_2(0) = 0,02$ a jako vzdálenější, chybné odhady $b_1(0) = 70$ a $b_2(0) = 1$. Pro data s modelem (5.4) byly zvoleny dobré odhady jako $b_1(0) = 8$ a $b_2(0) = 12$ a jako chybné $b_1(0) = 20$ a $b_2(0) = 25$. Parametr D je volen $D = 0,5$ [3].

ARS

Algoritmus ARS se ukončí v případě, kdy pro pět po sobě následujících iterací jsou nalezeny stejné hodnoty σ_k , $k = 1, \dots, 5$ [17]. Počáteční hodnoty jsou stejné jako v předcházejícím algoritmu. Pro

data s modelem (5.3) byly jako dobré počáteční odhady parametrů zvoleny $b_1(0) = 103$ a $b_2(0) = 0,02$ a jako vzdálenější, chybné odhady $b_1(0) = 70$ a $b_2(0) = 1$. Pro data s modelem (5.4) byly zvoleny dobré odhady jako $b_1(0) = 8$ a $b_2(0) = 12$ a jako chybné $b_1(0) = 20$ a $b_2(0) = 25$.

Gaussova-Newtonova metoda

Zastavovací podmínkou pro tuto derivační metodu je gradient blízký nule tj.

$$\|g\|^2 = \sum_{j=1}^p g_j^2 < 0,0001.$$

Pro data s modelem (5.3) byly jako dobré počáteční odhady zvoleny $b_1(0) = 103$ a $b_2(0) = 0,02$ a jako vzdálenější, chybné odhady $b_1(0) = 70$ a $b_2(0) = 1$. Pro data s modelem (5.4) byly zvoleny dobré odhady jako $b_1(0) = 8$ a $b_2(0) = 12$ a jako chybné $b_1(0) = 20$ a $b_2 = 25$. Nicméně pro vzdálené odhady algoritmus končí chybou, matice $(J^T J)$ je singulární, proto nelze spočítat její inverzní matici. Proto byl algoritmus testován ještě pro další sadu vzdálených počátečních odhadů, pro něž algoritmus nekončil chybou. Konkrétně $b_1(0) = 70$ a $b_2(0) = 0$ pro data s modelem (5.3) a $b_1(0) = 1$ a $b_2(0) = 1$ pro data s modelem (5.4).

Marquardtova metoda

Zastavovací podmínka pro gradient blízký nule

$$\|g\|^2 = \sum_{j=1}^p g_j^2 < 0,0001$$

je zde nedostačující, proto je přidána další. Konkrétně nerovnost

$$\sum_{j=1}^p (b_j(i) - b_j(i+1))^2 < 0,0001.$$

Pro data s modelem (5.3) byly jako dobré počáteční odhady parametrů zvoleny $b_1(0) = 103$ a $b_2(0) = 0,02$ a jako vzdálenější, chybné odhady parametrů $b_1(0) = 70$ a $b_2(0) = 1$. Pro data s modelem (5.4) byly zvoleny dobré odhady parametrů jako $b_1(0) = 8$ a $b_2(0) = 12$ a jako chybné $b_1(0) = 20$ a $b_2 = 25$. Ovšem algoritmus pro vzdálené

počáteční odhady končí chybou. Opět kvůli singularitě matice ($J^T J$). Je proto přidán další počáteční odhad: $b_1(0) = b_2(0) = 1$. Toto nastavení bylo použito pro oba způsoby adaptace λ uvedených v 4.2.2, tj. pro původní a Nashův algoritmus.

5.4.1 Výsledky

Výsledky lze vidět v tabulkách 5.1, 5.2, 5.3 a 5.4. U nejpřesnějších metod se odhad b_1 lišil od teoretického b_1^* v řádu desetin, v případě odhadu b_2 a teoretické hodnoty b_2^* byl rozdíl v řádu tisícín.

Hookeův-Jeevesův algoritmus

Tento algoritmus je časově velmi rychlý, počet jeho iterací je předem daný kvůli způsobu krácení kroku. Pro dobré počáteční odhady parametrů se teoretickému řešení ještě víc přiblíží. V případě modelu 5.4 s dobrými počátečními odhady najde algoritmus jeden z nejpřesnějších odhadů obou parametrů. Pro chybné počáteční odhady algoritmus nenalézá tak dobré odhady parametrů. Ovšem při opakovaném spuštění (při kterém je vždy počáteční odhad parametrů stejný jako závěrečný odhad při předcházejícím spuštění) algoritmus většinou dospěje k dobrým odhadům parametrů.

Nelderův-Meadův simplexový algoritmus

Simplexový algoritmus je nejvíce stabilním algoritmem. Zajímavé je, že algoritmus časově trvá velmi obdobně pro jakékoliv počáteční simplex. Pokud není počáteční simplex příliš malý (u trojúhelníku myšleno obsahově), tak tento algoritmus vždy našel velmi přesné odhady s nejmenší hodnotou kritériální funkce. Navíc tato metoda nepotřebuje žádné počáteční odhady, stačí ji obecný počáteční simplex (4.2) s dostatečně velkou hodnotou t .

Algoritmus CRS

Další z algoritmů, které vždy dospěly k nejpřesnějším odhadům parametrů. Nevýhodou však je největší výpočetní čas ze všech metod. CRS je pomalý i pro velmi dobré počáteční odhady parametrů.

Algoritmus GSA

Algoritmus GSA je naopak jedním z nejrychlejších algoritmů. Bohužel

už ne tak přesný. Obstojné odhady parametrů najde pouze pro dobré počáteční hodnoty. Pro vzdálené počáteční odhady je jedním z nejhorších.

Algoritmus ARST

Obdobné jako u algoritmu GSA. ARST je velice rychlý algoritmus, ze všech uvedených metod nejrychlejší. Bohužel algoritmus je nepřesný pro všechny testované sady dat a počáteční odhady.

Algoritmus ARS

Většinou velmi přesný algoritmus, ale v modelu (5.3) s chybnými počátečními odhady zastavovací podmínka ukončí algoritmus s velmi nepřesnými odhady a velmi vysokou hodnotu kritériální funkce. Jedná se ale spíše o výjimku. Časově patří k průměru mezi porovnávanými metodami.

Gaussova-Newtonova metoda

Tato metoda je nejrychlejší (v časovém smyslu i dle počtu iterací) při hledání velmi dobrých odhadů parametrů. Jedinou nevýhodou je ukončení algoritmu chybou, pokud je matice $(J^T J)$ singulární.

Marquardtova metoda

Marquardtova metoda je pomalejší a méně přesná než Gaussova-Newtonova metoda. Pro tyto dvě sady dat při vzdálených počátečních odhadech navíc také končí chybou z důvodu singularity matice $(J^T J)$.

Poč. simplex	počet iterací m	$b_1(m)$	$b_2(m)$	$S(b_1(m), b_2(m))$	čas [s]
t = 25	84	99,02887	0,03206581	28,36188	0,60
t = 150	77	99,02886	0,03206592	28,36188	0,56
3 body	57	99,02877	0,03206554	28,36188	0,52

Tabulka 5.1: Výsledky Nelderova-Meadova simplexového algoritmu - simulovaná data (model (5.3)). Zeleně jsou označeny nejmenší hodnoty kritériální funkce ve srovnání s tabulkou 5.3.

Poč. simplex	počet iterací m	$b_1(m)$	$b_2(m)$	$S(b_1(m), b_2(m))$	čas [s]
t = 25	50	6,625410	9,835022	545,8086	0,56
t = 150	58	6,625405	9,834957	545,8086	0,56
3 body	57	6,625413	9,835003	545,8086	0,52

Tabulka 5.2: Výsledky Nelderova-Meadova simplexového algoritmu - simulovaná data (model (5.4)). Zeleně jsou označeny nejmenší hodnoty kritériální funkce ve srovnání s tabulkou 5.4.

Algoritmus	$b_1(0)$	$b_2(0)$	počet iterací m	$b_1(m)$	$b_2(m)$	$S(b_1(m), b_2(m))$	čas [s]
Hookeův-Jeevesův	103 70	0.02 1	15 15	96,59468000 89,14687000	0,0254596900 0,0103630400	93,71471000 1307,44600000	0,1700 0,2200
CRS	[98; 108] [65; 75]	[0,01; 0,04] [0,8; 1,2]	378,47 469,7	99,02914908 99,01821148	0,0320682700 0,0320395700	28,36385449 28,46030202	0,8983 28,9300
GSA	103 70	0.02 1	3,79 17,89	103,39238869 73,53906000	0,0461113100 -0,0058562777	213,01555044 10994,28000000	0,0116 0,0323
ARST	103 70	0.02 1	3,8 7,57	103,02301007 69,03231000	0,0430100700 0,0323096000	312,85268025 23053,53000000	0,0119 0,0081
ARS	103 70	0.02 1	77,55 192,97	99,02865656 88,05768250	0,0320369400 -0,0544678100	28,36203694 1344,65768229	0,2168 0,6486
Gaussův-Newtonův	103 70 70	0.02 1 0	6 4 7	99,02882000 -76672557 99,02882000	0,0320657300 -3650731 0,0320657300	28,36188000 188448,9 28,36188000	0,1600 0,1400 0,1700
Marquardtův-původní	103 70	0.02 1	8 36275	96,45001000 189,15080000	0,0200003500 0,6300154000	165,06790000 14771,18000000	0,1900 340,6400
Marquardtův-Nashův	103 70	0.02 1	6 20	96,61873000 239,67150000	0,0200002200 0,9992342000	165,90530000 20054,25000000	0,1800 0,2900

Tabulka 5.3: Výsledky - simulovaná data (model (5.3)). Zeleně je označeno několik nejmenších hodnot kritériační funkce. Žlutě je označen algoritmus a počáteční odhady, které končí chybou.

Algoritmus	$b_1(0)$	$b_2(0)$	počet iterací m	$b_1(m)$	$b_2(m)$	$S(b_1(m), b_2(m))$	čas [s]
Hookeův-Jeevesův	8	12	15	6,6254680	9,8358230	545,808600	0,1700
	20	25	15	7,0857510	19,4354800	2554,898000	0,2500
CRS	[6; 10]	[10; 14]	422,97	6,6254560	9,8358530	545,809691	12,1581
	[18; 22]	[23; 27]	471,44	6,6245480	9,8304180	546,993604	24,0928
GSA	8	12	7,89	6,9264860	14,1070140	1211,959877	0,0147
	20	25	41,39	7,3857530	43,6727470	6193,633942	0,0416
ARST	8	12	7,86	6,8270930	10,8270930	904,473577	0,0080
	20	25	10,44	18,1711800	23,1711800	80227,329820	0,0075
ARS	8	12	47,5	6,6254410	9,8351890	545,809325	0,2211
	20	25	47,91	6,6252000	9,8340890	545,823940	0,2178
Gaussův-Newtonův	8	12	6	6,6254080	9,8349940	545,808600	0,1700
	20	25	15	-153415,74196	-61,2952800	1,20539e+30	0,2000
Marquardtův-původní	1	1	8	6,6254080	9,8349930	545,8086	0,14
	8	12	218	6,6528410	9,9466810	548,080000	0,5900
Marquardtův-Nashův	20	25	2	16,6379000	1198,8280000	71270,400000	0,1800
	1	1	5	0,2271048	-0,7357678	7814,474000	0,1500
Marquardtův-Nashův	8	12	2	7,9999990	12,0003000	5708,404000	0,2200
	20	25	6	19,6353100	704,6281000	87577,300000	0,1900
1	1	2	1,0000010	0,9999826	1318138,000000	0,2200	

Tabulka 5.4: Výsledky - simulovaná data (model (5.4)). Zeleně je označeno několik nejmenších hodnot kritériální funkce. Žlutě jsou označeny algoritmy a počáteční odhady, které končí chybou.

5.5 Porovnání jednotlivých algoritmů - reálná data

V další části práce budeme opět porovnávat metody nelineární regrese, které jsou popsány v kapitole 4, tentokrát ovšem na reálných datových sadách zmíněných v 5.1. Data lze vidět v tabulkách B.1 a B.2 v příloze B. Metody porovnáme dle hodnoty kritériální funkce (3.3), počtu iterací nutných k zastavení algoritmu zastavovací podmínkou a dle celkového času, po který algoritmus běží. Pro všechny datové sady budou opět použité dva druhy počáteční odhadů $\beta(0)$. Jelikož neznáme přesnou hodnotu β^* , budeme jako dobré počáteční odhady považovat ty s menší hodnotou kritériální funkce. A naopak, za chybné počáteční odhady považujeme odhady s větší hodnotou kritériální funkce.

Algoritmy metod používajících náhodná čísla byly spuštěné stokrát a výsledné odhady parametrů, výpočetní čas a počet iterací byly pak zprůměrovány.

5.5.1 Puromycin data

Tento datový set obsahuje dvě sady dat - s puromycinem (ve zdrojových kódech a v datech označené jako treated) a bez puromycinu (označené jako untreated). V modelu 5.1 se vyskytují parametry θ_1 a θ_2 , které ale v následujícím textu budou popořadě nahrazeny b_1 a b_2 , aby odpovídaly textu v kapitole 4.

Hookeova-Jeevesova metoda

Jako zastavovací podmínka byla použita nerovnost

$$|\delta_i| \leq 0,0001.$$

Velikost kroku $|\delta_i|$ je půlena po každé iteraci jako ve zdroji [7]. Pro data s puromycinem byly jako dobré počáteční odhady parametrů zvoleny $b_1(0) = 200$ a $b_2(0) = 0,06$ a jako chybné odhady $b_1(0) = 150$ a $b_2(0) = 0,5$. Pro data bez puromycinu byly zvoleny dobré odhady jako $b_1(0) = 150$ a $b_2(0) = 0,06$ a jako chybné $b_1(0) = 200$ a $b_2(0) = 0,5$.

Nelderova-Meadova simplexová metoda

Pro zastavení tohoto algoritmu musí být splněné dvě zastavovací pod-

mínky (4.11) a (4.12)

$$|S(W) - S(B)| < 10^{-4}$$

a

$$\frac{1}{p+1} \sum_{j=1}^{p+1} (U_j(i) - U_j(i-1))^2 < 10^{-8}.$$

Hodnoty simplexových konstant ze vztahů (4.5), (4.6), (4.7), (4.8), (4.9) a (4.10) jsou následovně pevně zvolené: $\alpha = 1$, $\mu = 0,55$, $\gamma = 2,9$ a $\lambda = 0,5$.

Počáteční podmínky jsou v tomto algoritmu specifické tím, že potřebujeme celý počáteční simplex, v našem případě trojúhelník. Pro puromycin data zvolíme tři počáteční simplex: simplex dle (4.2) s $t = 25$, simplex dle (4.2) s $t = 150$ a trojúhelník tvořený body se souřadnicemi $A_t = [180; 0, 01]$, $B_t = [240; 0, 01]$ a $C_t = [200; 1]$ pro data s puromycinem nebo trojúhelník tvořený body se souřadnicemi $A_u = [130; 0, 01]$, $B_u = [190; 0, 01]$ a $C_u = [160; 1]$ pro data bez puromycinu.

CRS

Pro metodu CRS byl jako zastavovací podmínka použit rozdíl mezi nejhorším a nejlepším vrcholem simplexu čili

$$|S(W) - S(B)| < 0,0001.$$

Navíc, aby došlo k ukončení algoritmu, bylo nutné přidat podmínku pro počet neúspěšných výběrů z matice W . Tento počet neúspěchů byl nastaven na velkou hodnotu tj. 1 000 000. V metodě CRS se používá výběr z matice W , takže všechny počáteční odhady nemohou být stejné, protože by operace reflexe nenašla žádný nový bod. Všechny možné počáteční odhady obou parametrů byly vygenerovány z rovnoměrného spojitého rozdělení na intervalu, jehož středem je počáteční odhad používaný například v Hookeově-Jeevesově metodě. V datech s puromycinem pro dobré počáteční odhady parametru $b_1(0)$ je generováno z intervalu $[190, 210]$ a parametru $b_2(0)$ z intervalu $[0, 04; 0, 06]$. Pro chybné odhady těchto dat je $b_1(0)$ generováno z intervalu $[140, 160]$ a $b_2(0)$ z intervalu $[0, 4; 0, 6]$. V datech bez puromycinu jsou dobré počáteční odhady $b_1(0)$ z intervalu $[140, 160]$ a chybné počáteční odhady $b_1(0)$ z intervalu $[190, 210]$. Počáteční odhady pro

$b_2(0)$ jsou v obou případech ze stejného intervalu jako v datech s puromycinem.

GSA

V metodě GSA jako zastavovací podmínka slouží nerovnost

$$\sum_{j=1}^p (b_j(i) - b_j(i+1))^2 < 0,0001.$$

Opět musela být přidána podmínka pro počet neúspěšných pokusů, která byla nastavena na hodnotu 100, což se jeví jako dostatečné omezení. Pro data s puromycinem byly jako dobré počáteční odhady zvoleny $b_1(0) = 200$ a $b_2(0) = 0,06$ a jako chybné odhady $b_1(0) = 150$ a $b_2(0) = 0,5$. Pro data bez puromycinu byly zvoleny dobré odhady jako $b_1(0) = 150$ a $b_2(0) = 0,06$ a jako chybné $b_1(0) = 200$ a $b_2 = 0,5$. Parametry \varkappa a Δr jsou volené, tak jak je doporučováno v [3] neboli $\varkappa = 3,5$ a $\Delta r = 0,15$.

ARST

V metodě ARST jako zastavovací podmínka opět slouží nerovnost

$$\sum_{j=1}^p (b_j(i) - b_j(i+1))^2 < 0,0001.$$

Pro data s puromycinem byly jako dobré počáteční odhady zvoleny $b_1(0) = 200$ a $b_2(0) = 0,06$ a jako chybné odhady $b_1(0) = 150$ a $b_2(0) = 0,5$. Pro data bez puromycinu byly zvoleny dobré odhady jako $b_1(0) = 150$ a $b_2(0) = 0,06$ a jako chybné $b_1(0) = 200$ a $b_2(0) = 0,5$. Parametr D je volen $D = 0,5$ [3].

ARS

Algoritmus ARS se ukončí v případě, kdy pro pět po sobě následujících iterací jsou nalezeny stejné hodnoty σ_k , $k = 1, \dots, 5$ [17]. Počáteční hodnoty jsou stejné jako v předcházejícím algoritmu. Pro data s puromycinem byly jako dobré počáteční odhady zvoleny $b_1(0) = 200$ a $b_2(0) = 0,06$ a jako chybné odhady $b_1(0) = 150$ a $b_2(0) = 0,5$. Pro data bez puromycinu byly zvoleny dobré odhady jako $b_1(0) = 150$ a $b_2(0) = 0,06$ a jako chybné $b_1(0) = 200$ a $b_2(0) = 0,5$.

Gaussova-Newtonova metoda

Zastavovací podmínkou pro tuto derivační metodu je gradient blízky nule tj.

$$\|g\|^2 = \sum_{j=1}^p g_j^2 < 0,0001.$$

Pro data s puromycinem byly jako dobré počáteční odhady zvoleny $b_1(0) = 200$ a $b_2(0) = 0,06$ a jako chybné odhady $b_1(0) = 150$ a $b_2(0) = 0,5$. Pro data bez puromycinu byly zvoleny dobré odhady jako $b_1(0) = 150$ a $b_2(0) = 0,06$ a jako chybné $b_1(0) = 200$ a $b_2 = 0,5$. Nicméně pro chybné odhady algoritmus končí chybou, matice $(J^T J)$ je singulární, proto nelze spočítat její inverzní matici. Proto byl algoritmus spuštěn ještě pro další sadu chybných počátečních odhadů, pro něž algoritmus nekončil chybou. Konkrétně $b_1(0) = 150$ a $b_2(0) = 0$ pro data s puromycinem a $b_1(0) = 200$ a $b_2(0) = 0$ pro data bez puromycinu.

Marquardtova metoda

Zastavovací podmínka pro gradient blízky nule

$$\|g\|^2 = \sum_{j=1}^p g_j^2 < 0,0001$$

je zde nedostačující, proto je přidána další. Konkrétně nerovnost

$$\sum_{j=1}^p (b_j(i) - b_j(i+1))^2 < 0,0001.$$

Pro data s puromycinem byly jako dobré počáteční odhady zvoleny $b_1(0) = 200$ a $b_2(0) = 0,06$ a jako chybné odhady $b_1(0) = 150$ a $b_2(0) = 0,5$. Pro data bez puromycinu byly zvoleny dobré odhady jako $b_1(0) = 150$ a $b_2(0) = 0,06$ a jako chybné $b_1(0) = 200$ a $b_2 = 0,5$. Toto nastavení bylo použito pro oba způsoby adaptace λ uvedených v 4.2.2, tj. pro původní a Nashův algoritmus.

5.5.2 Farmakologická data

V modelu 5.2 se vyskytují parametry γ a β , které ale v následujícím textu budou popořadě nahrazeny b_1 a b_2 , aby odpovídaly textu v kapitole 4.

Hookeova-Jeevesova metoda

Zastavovací podmínkou je nerovnost

$$|\delta_i| \leq 0,0001.$$

Velikost kroku $|\delta_i|$ je půlena po každé iteraci jako ve zdroji [7]. Jako dobré počáteční odhady parametrů jsou použité hodnoty $b_1(0) = 5$ a $b_2(0) = 5$. Jako chybné počáteční odhady pak $b_1(0) = 15$ a $b_2(0) = 15$.

Nelderova-Meadova simplexová metoda

Pro zastavení tohoto algoritmu musí být splněné dvě zastavovací podmínky (4.11) a (4.12)

$$|S(W) - S(B)| < 10^{-4}$$

a

$$\frac{1}{p+1} \sum_{j=1}^{p+1} (U_j(i) - U_j(i-1))^2 < 10^{-8}.$$

Hodnoty simplexových konstant ze vztahů (4.5), (4.6), (4.7), (4.8), (4.9) a (4.10) jsou následovně pevně zvolené: $\alpha = 1$, $\mu = 0,55$, $\gamma = 2,9$ a $\lambda = 0,5$.

Počáteční podmínky jsou v tomto algoritmu specifické tím, že potřebujeme celý počáteční simplex, v našem případě trojúhelník. Pro data zvolíme dva počáteční simplexu: simplex dle (4.2) s $t = 25$, simplex dle (4.2) s $t = 150$.

CRS

Pro metodu CRS byl jako zastavovací podmínka použit rozdíl mezi nejhorším a nejlepším vrcholem simplexu čili

$$|S(W) - S(B)| < 0,0001.$$

Navíc, aby došlo k ukončení algoritmu, bylo nutné přidat podmínku pro počet neúspěšných výběrů z matice W . Tento počet neúspěchů byl nastaven na 1 000 000. Všechny možné počáteční odhady obou parametrů byly vygenerovány z rovnoměrného spojitého rozdělení na intervalu, jehož středem je počáteční odhad používaný například v Hookeově-Jeevesově metodě. Dobré počáteční odhady parametrů $b_1(0)$

$b_1(0)$ a $b_2(0)$ jsou z intervalu $[3, 7]$ a naopak chybné parametry $b_1(0)$ a $b_2(0)$ jsou generovány z intervalu $[13, 17]$.

GSA

V metodě GSA jako zastavovací podmínka slouží nerovnost

$$\sum_{j=1}^p (b_j(i) - b_j(i+1))^2 < 0,0001.$$

Opět musela být přidána podmínka pro počet neúspěšných pokusů, která byla nastavena na hodnotu 100. Dobré počáteční odhady parametrů jsou $b_1(0) = b_2(0) = 5$ a naopak chybné odhady parametrů jsou $b_1(0) = b_2(0) = 15$.

Parametry \varkappa a Δr jsou opět volené, tak jak je doporučováno v [3] neboli $\varkappa = 3, 5$ a $\Delta r = 0, 15$.

ARST

V metodě ARST jako zastavovací podmínka opět slouží nerovnost

$$\sum_{j=1}^p (b_j(i) - b_j(i+1))^2 < 0,0001.$$

Dobré počáteční odhady parametrů jsou opět $b_1(0) = b_2(0) = 5$ a naopak chybné jsou $b_1(0) = b_2(0) = 15$. Parametr D je volen $D = 0, 5$ [3].

ARS

Algoritmus ARS se ukončí v případě, kdy pro pět po sobě následujících iterací jsou nalezeny stejné hodnoty σ_k , $k = 1, \dots, 5$ [17]. Jako dobré počáteční odhady parametrů jsou zvoleny $b_1(0) = b_2(0) = 5$ a jako chybné počáteční hodnoty $b_1(0) = b_2(0) = 15$.

Gaussova-Newtonova metoda

Jako zastavovací podmínka je použita

$$\|g\|^2 = \sum_{j=1}^p g_j^2 < 0,0001.$$

Pro data byly jako dobré počáteční odhady zvoleny $b_1(0) = 5$ a $b_2(0) = 5$ a jako chybné odhady $b_1(0) = 15$ a $b_2(0) = 15$. Algoritmus s chybnými počáteční odhady opět skončí chybou, protože

matice $(J^T J)$ je singulární. Jako jiné nedobré odhady byly zvoleny $b_1(0) = -5$ a $b_2(0) = -5$, pro tyto odhady už algoritmus proběhne v pořádku.

Marquardtova metoda

Zastavovací podmínka pro gradient blízky nule

$$\|g\|^2 = \sum_{j=1}^p g_j^2 < 0,0001$$

je zde nedostačující, proto je přidána další. Konkrétně nerovnost

$$\sum_{j=1}^p (b_j(i) - b_j(i+1))^2 < 0,0001.$$

Jako dobré počáteční odhady jsou opět voleny $b_1(0) = 5$ a $b_2(0) = 5$ a jako chybné odhady $b_1(0) = 15$ a $b_2(0) = 15$.

Toto nastavení bylo použito pro oba způsoby adaptace λ uvedených v 4.2.2, tj. pro původní a Nashův algoritmus.

5.5.3 Výsledky

Jak lze vidět z kompletních výsledků v tabulkách 5.5, 5.6, 5.7, 5.8, 5.9 a 5.10 algoritmy CRS, ARS, Gaussův-Newtonův a Nelderův-Meadův simplexový algoritmus vždy skončili s nejlepšími odhady parametrů z hlediska hodnoty kritériální funkce. Časově nejrychlejší z nich je Gaussova-Newtonova metoda, nicméně u ní je časté (hlavně u ne příliš dobrých počátečních odhadů), že skončí chybou kvůli singularitě matice $(J^T J)$.

Hookeův-Jeevesův algoritmus

V zásadě platí to samé, co je psáno o tomto algoritmu v podsekcí 5.4.1. Hlavně v případě dobrých počátečních odhadů parametrů najde odhady b , jejichž hodnota kritériální funkce patří k lepším v porovnání s ostatními algoritmy.

Nelderův-Meadův simplexový algoritmus

Opět o algoritmu platí to samé, co v podsekcí 5.4.1. Nejstabilnější algoritmus, který při ne příliš malém simplexu dospěje k nejmenším

nalezeným hodnotám kritériální funkce.

Algoritmus CRS

Na rozdíl od testování na simulovaných datech algoritmus CRS skončí vždy s odhadem parametrů, jenž mají nejmenší hodnoty kritériální funkce ve srovnání s ostatními metodami. Nevýhodou však je největší výpočetní čas ze všech metod. CRS je pomalý pro všechny vyzkoušené počáteční odhady parametrů.

Algoritmus GSA

Algoritmus GSA je naopak jedním z nejrychlejších algoritmů. Z hlediska hodnoty kritériální funkce ovšem patří k těm nejhorším.

Algoritmus ARST

Obdobné jako u algoritmu GSA. ARST je velice rychlý algoritmus, ze všech uvedených metod nejrychlejší. Bohužel algoritmus patří dle hodnoty kritériální funkce k těm nejhorším.

Algoritmus ARS

Algoritmus ARS patří mezi nejlepší. Vždy našel odhady s nejmenšími hodnotami kritériální funkce. Časově patří k průměru mezi porovnávanými metodami.

Gaussova-Newtonova metoda

Tato metoda je nejrychlejší (v časovém smyslu i dle počtu iterací) při hledání odhadů parametrů s nejmenšími hodnotami kritériální funkce. Jedinou nevýhodou je ukončení algoritmu chybou, pokud je matice $(J^T J)$ singulární.

Marquardtova metoda

Marquardtova metoda je méně náchylná k ukončení chybou než Gaussova-Newtonova metoda. Je však pomalejší (v časovém smyslu i dle počtu iterací) než zmíněná Gaussova-Newtonova metoda. V porovnání hodnot kritériální funkce je na tom také lépe Gaussova-Newtonova metoda. Nelze ani říct, který z přístupů volby parametru λ je lepší, zda původní či Nashův. Vždy záleží na použitých datech.

Poč. simplex	počet iterací m	$b_1(m)$	$b_2(m)$	$S(b_1(m), b_2(m))$	čas [s]
t = 25	113	212,6837	0,06412124	1195,449	0,58
t = 150	84	212,6837	0,06412126	1195,449	0,63
3 body	57	212,6838	0,06412134	1195,449	0,55

Tabulka 5.5: Výsledky Nelderova-Meadova simplexového algoritmu - data s puromycinem (treated puromycin). Zeleně jsou označeny nejmenší hodnoty kritériální funkce ve srovnání s tabulkou 5.8.

Poč. simplex	počet iterací m	$b_1(m)$	$b_2(m)$	$S(b_1(m), b_2(m))$	čas [s]
t = 25	89	160,2801	0,04770827	859,6043	1,00
t = 150	73	160,2800	0,04770811	859,6043	0,65
3 body	61	160,2801	0,04770822	859,6043	0,56

Tabulka 5.6: Výsledky Nelderova-Meadova simplexového algoritmu - data bez puromycinu (untreated puromycin). Zeleně jsou označeny nejmenší hodnoty kritériální funkce ve srovnání s tabulkou 5.9.

Poč. simplex	počet iterací m	$b_1(m)$	$b_2(m)$	$S(b_1(m), b_2(m))$	čas [s]
t = 25	50	4,180829	2,571400	2061,549	0,53
t = 150	64	4,180797	2,571368	2061,549	0,55

Tabulka 5.7: Výsledky Nelderova-Meadova simplexového algoritmu - farmakologická data. Zeleně jsou označeny nejmenší hodnoty kritériální funkce ve srovnání s tabulkou 5.10.

Algoritmus	$b_1(0)$	$b_2(0)$	počet iterací m	$b_1(m)$	$b_2(m)$	$S(b_1(m), b_2(m))$	čas [s]
Hookeův-Jeevesův	200	0,06	15	210,0299000	0,06159721	1212,084000	0,2100
	150	0,5	15	181,8053000	0,03896301	3768,855000	0,1600
CRS	[190; 210]	[0,04; 0,08]	399,23	212,6845002	0,06412210	1195,449337	2,9951
	[140; 160]	[0,4; 0,6]	482,46	212,6879000	0,06412182	1195,499000	22,5229
GSA	200	0,06	3,95	200,4489000	0,05360341	1592,909000	0,0140
	150	0,5	10,08	151,8399000	0,02212247	12782,210000	0,0189
ARST	200	0,06	2,48	199,9951000	0,05505247	1648,827000	0,0080
	150	0,5	5,84	149,5315000	0,03153130	16706,230000	0,0509
ARS	200	0,06	64,42	212,6818000	0,06411844	1195,449000	0,1542
	150	0,5	155,35	212,6836000	0,06412102	1195,449000	0,3916
Gaussův-Newtonův	200	0,06	8	212,6837000	0,06412128	1195,449000	0,1400
	150	0,5	5	-1,625176e+10	2,969089e+08	366222,2	0,1100
Marquardtův-původní	150	0	12	212,6837000	0,06412128	1195,449000	0,1900
	200	0,06	13	210,0072000	0,06000000	1223,681000	0,1900
Marquardtův-Nashův	150	0,5	398683	255,0369000	0,14807330	6241,486000	37914,0200
	200	0,06	13	210,0072000	0,06000000	1223,681000	0,1900
	150	0,5	22	368,8179000	0,49995750	29979,490000	0,2900

Tabulka 5.8: Výsledky - data s puromycinem (treated puromycin). Zeleně je označeno několik nejmenších hodnot kritériální funkce. Žlutě je označen algoritmus a počáteční odhady, které končí chybou.

Algoritmus	$b_1(0)$	$b_2(0)$	počet iterací m	$b_{-1}(m)$	$b_{-2}(m)$	$S(b_1(m), b_2(m))$	čas [s]
Hookeův-Jeevesův	150	0,06	15	167,63070000	0,05532461	965,602400000	0,1500
	200	0,5	15	226,92940000	0,14419350	6743,095000000	0,1600
CRS	[140; 160]	[0,04; 0,08]	391,6	160,28092576	0,04770748	859,605266100	3,6769
	[190; 210]	[0,4; 0,6]	470,72	160,27097840	0,04769960	859,617586600	17,7250
GSA	150	0,06	4,69	150,57400000	0,03953554	1096,520000000	0,0140
	200	0,5	7,37	201,35539820	0,10010180	3514,676998200	0,0134
ARST	150	0,06	3,26	149,98110000	0,04112676	1328,367000000	0,0106
	200	0,5	5,34	199,59865860	0,09865860	3731,666641400	0,0108
ARS	150	0,06	83,94	160,28034804	0,04770853	859,604361470	0,2052
	200	0,5	160,44	160,27986603	0,04770786	859,606666545	0,3914
Gaussův-Newtonův	150	0,06	10	160,28000000	0,04770818	859,604300000	0,1400
	200	0,5	6	-116221708168	-1275508565	84856,99	0,1500
Marquardtův-původní	150	0,06	13	160,28000000	0,04770818	859,604300000	0,1900
	200	0,5	19	167,75770000	0,05999992	1028,574000000	0,2400
Marquardtův-Nashův	150	0,06	13	312,26970000	0,49996920	20603,080000000	0,2700
	200	0,5	19	167,75770000	0,05999992	1028,574000000	0,2500
	200	0,5	22	312,53710000	0,49998270	20603,560000000	0,2800

Tabulka 5.9: Výsledky - data bez puromycinu (untreated puromycin). Zeleně je označeno několik nejmenších hodnot kritériální funkce. Žlutě je označen algoritmus a počáteční odhady, které končí chybou.

Algoritmus	$b_1(0)$	$b_2(0)$	počet iterací m	$b_1(m)$	$b_2(m)$	$S(b_1(m), b_2(m))$	čas [s]
Hookeův-Jeevesův	5	5	15	4,7028840	3,279551	2126,135000	0,2200
	15	15	15	6,8654470	7,769005	4477,849000	0,2400
CRS	[3; 7]	[3; 7]	415,29	4,1802390	2,570647	2061,555105	17,7107
	[13; 17]	[13; 17]	429,02	4,1494090	2,546252	2062,464324	31,9134
GSA	5	5	5,46	5,7182630	4,950737	2730,063942	0,0122
	15	15	19,71	9,8042770	23,002223	13959,037311	0,0237
ARST	5	5	7,49	5,9215290	5,921529	3393,207627	0,0089
	15	15	9,99	13,3160600	13,316060	22151,789130	0,0081
ARS	5	5	786,06	4,0682300	2,474860	2061,097470	3,8020
	15	15	478,5	4,0303910	2,442346	2060,953717	2,3293
Gaussův-Newtonův	5	5	6	4,1808120	2,571380	2061,549000	0,1800
	15	15	24	10569595	-47,47544	49593390	0,2000
Marquardtův-původní	-5	-5	12	4,1808122	2,571380	2061,549000	0,1400
	5	5	27	4,5995560	3,104265	2101,332000	0,2700
Marquardtův-Nashův	15	15	65	4,6895690	3,224343	2121,081000	0,3500
	5	5	2	5,0000010	4,999993	4665,594000	0,1800
	15	15	2	14,999990	15,000110	26891,230000	0,1700

Tabulka 5.10: Výsledky - farmakologická data. Zeleně je označeno několik nejmenších hodnot kritériální funkce. Žlutě je označen algoritmus a počáteční odhady, které končí chybou.

Kapitola 6

Závěr

Práce byla zaměřená na porovnání metod odhadů parametrů nelineární regrese. Mezi nejlepší v testování patřily algoritmy CRS a ARS, Gaussova-Newtonova metoda a Nelderova-Meadova simplexová metoda. Pokud byla známá teoretická hodnota β^* , tak tyto algoritmy se jí vždy nejvíce přiblížily. V případech, ve kterých teoretická hodnota β^* známá nebyla, zmíněné algoritmy našly odhady s nejmenšími hodnotami kriteriální funkce (3.3).

Jako nejvíce stabilní se ukázal být Nelderův-Meadův simplexový algoritmus. Tento algoritmus vždy našel odhady parametrů, které byly ve srovnání s ostatními metodami nejbližší známé teoretické hodnotě β^* nebo jejichž hodnota kriteriální funkce byla nejmenší v případě neznámé β^* . Navíc nemusíme znát žádné dobré počáteční odhady parametrů, vystačíme si s dostatečně velkým počátečním simplexem.

Nejrychlejším algoritmem, který nalézá velmi dobré odhady ve smyslu hodnoty kriteriální funkce či vzdálenosti od známé teoretické hodnoty β^* , je Gaussova-Newtonova metoda. Má však poměrně velkou nevýhodu. Zvláště pro špatné počáteční odhady tento algoritmus často končí chybou z důvodu singularity matice $(J^T J)$ a tudíž neexistence inverzní matice $(J^T J)^{-1}$.

Algoritmus CRS je další metodou, která dává velmi dobré odhady regresních parametrů. Opět chápeme velmi dobré odhady ve smyslu hodnoty kriteriální funkce či vzdálenosti od známé teoretické hodnoty β^* . Jedinou nevýhodou je rychlost respektive spíše pomalost konvergence ke konečnému odhadu parametru.

Posledním z algoritmů, které vrací velmi dobré odhady (ve stejném smyslu jako výše), je algoritmus ARS. Avšak pro simulovaná data s modelem (5.3) a počátečních odhadech $b_1(0) = 70$ a $b_2(0) = 1$ konvergoval k jinému než ke globálnímu řešení. V ostatních datových sadách i pro jiné počáteční podmínky fungoval algoritmus ARS v pořádku a našel velmi dobré odhady parametrů.

Všechny ostatní testované algoritmy nebyly už tak přesné. Hlavně v případě použití chybných počátečních hodnot metody konvergovaly pomalu a hlavně k chybným hodnotám, které nebyly globálním extrémem.

Literatura

- [1] HUET, S. Statistical tools for nonlinear regression: a practical guide with S-PLUS and R examples. 2nd ed. New York: Springer, c2004. ISBN 0387400818.
- [2] RITZ, Christian a Jens C. STREIBIG. Nonlinear regression with R. New York, N.Y.: Springer, c2008. ISBN 978-0-387-09615-5.
- [3] MELOUN, Milan a Jiří MILITKÝ. Statistická analýza experimentálních dat. Vyd. 2., upr. a rozš. Praha: Academia, 2004. ISBN 80200-1254-0.
- [4] ZVÁRA, Karel. Regrese. Praha: Matfyzpress, 2008. ISBN 978-80-7378-041-8.
- [5] BATES, Douglas M. a Donald G. WATTS. Nonlinear regression analysis and its applications. New York: Wiley, c1988. ISBN 0471816434.
- [6] REIF, Jiří. Metody matematické statistiky. Plzeň: Západočeská univerzita, 2000. ISBN 80-7082-593-6.
- [7] NASH, John C. a Mary WALKER-SMITH. Nonlinear parameter estimation: an integrated system in BASIC. [rev. ed.]. Ottawa, Ontario, Canada: Nash Information Services, 1995. ISBN 0-88769-009-2.
- [8] SEBER, G. A. F. a C. J. WILD. Nonlinear regression. Hoboken, N.J.: Wiley-Interscience, c2003. Wiley series in probability and statistics. ISBN 0-471-47135-6.
- [9] SCHMIDT, Rüdiger. Advances in nonlinear parameter optimization. New York: Springer-Verlag, 1982. ISBN 0387113967.

- [10] ANDĚL, Jiří. Základy matematické statistiky. Vyd. 3. Praha: Matfyzpress, 2011. ISBN 978-80-7378-162-0.
- [11] MADSEN, K. , H. B. NIELSEN a O. TINGLEFF. Methods for Non-Linear Least Squares Problems. Vyd. 2. Informatics and Mathematical Modelling, Technical University of Denmark, 2004. Dostupné také z: http://www.ltu.se/cms_fs/1.51590!/nonlinear_least_squares.pdf
- [12] STANIMIROVIC, Predrag S., Milan B. TASIC a Miroslav RISTIC. Symbolic implementation of the Hooke-Jeeves method. Yugoslav journal of operations research. 1999, 9(2), 285-300. ISSN 0354-0243. Dostupné také z: https://www.researchgate.net/publication/288478592_Symbolic_implementation_of_the_Hooke-Jeeves_method
- [13] PRICE, W. L. Global optimization by controlled random search. Journal of Optimization Theory and Applications. 1983, 40(3), 333-348. DOI: 10.1007/BF00933504. ISSN 0022-3239. Dostupné také z: <http://link.springer.com/10.1007/BF00933504>
- [14] PRICE, W. L. A controlled random search procedure for global optimisation. The Computer Journal. 1977, 20(4), 367-370. DOI: 10.1093/comjnl/20.4.367. ISSN 0010-4620. Dostupné také z: <http://comjnl.oupjournals.org/cgi/doi/10.1093/comjnl/20.4.367>
- [15] KAELO, P. a M. M. ALI. Some Variants of the Controlled Random Search Algorithm for Global Optimization. Journal of Optimization Theory and Applications. 2006, 130(2), 253-264. DOI: 10.1007/s10957-006-9101-0. ISSN 0022-3239. Dostupné také z: <http://link.springer.com/10.1007/s10957-006-9101-0>
- [16] TSALLIS, Constantino a Daniel A. STARIOLO. Generalized simulated annealing. Physica A: Statistical Mechanics and its Applications. 1996, 233(1-2), 395-406. DOI: 10.1016/S0378-4371(96)00271-3. ISSN 03784371. Dostupné také z: <http://linkinghub.elsevier.com/retrieve/pii/S0378437196002713>

- [17] PRONZATO, Luc, Eric WALTER, Alain VENOT a Jean-Francois LEBRUCHEC. A general-purpose global optimizer: Implimentation and applications. *Mathematics and Computers in Simulation*. 1984, 26(5), 412-422. DOI: 10.1016/0378-4754(84)90105-8. ISSN 03784754. Dostupné také z: <http://linkinghub.elsevier.com/retrieve/pii/0378475484901058>
- [18] RYAN, P. Barry., Richard L. BARR a H. David. TODD. Simplex techniques for nonlinear optimization. *Analytical Chemistry*. 1980, 52(9), 1460-1467. DOI: 10.1021/ac50059a019. ISSN 0003-2700. Dostupné také z: <http://pubs.acs.org/doi/abs/10.1021/ac50059a019>
- [19] TRELOAR, Margaret, Jennifer M. STURGESS a Mario A. MOSCARELLO. An Effect of Puromycin on Galactosyltransferase of Golgi-rich Fractions from Rat Liver. *Journal of Biological Chemistry*. 1974, 249(20), 6628-6632. ISSN 0021-9258. Dostupné také z: <http://www.jbc.org/content/249/20/6628.full.pdf>
- [20] The R Project for Statistical Computing. R. [online]. 2016 [cit. 2016-05-01]. Dostupné z: <https://www.r-project.org>
- [21] Contributed Packages. CRAN. [online]. 2016 [cit. 2016-05-01]. Dostupné z: <https://cran.r-project.org/web/packages/>
- [22] Open source and enterprise-ready professional software for R. RStudio. [online]. c2016 [cit. 2016-05-01]. Dostupné z: <https://www.rstudio.com>
- [23] The R Datasets Package. R. [online]. 2016 [cit. 2016-05-01]. Dostupné z: <https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/00Index.html>
- [24] Nonlinear Least Squares. R. [online]. 2016 [cit. 2016-05-01]. Dostupné z: <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/nls.html>
- [25] GAVIN, Henri P. The Nelder-Mead Algorithm in Two Dimensions. [online] 2016 [cit. 2016-05-01]. Dostupné z: <http://people.duke.edu/~hpgavin/cee201/Nelder-Mead-2D.pdf>

- [26] Nelder-Mead algorithm. Scholarpedia. [online] 2011 [cit. 2016-05-01]. Dostupné z: http://www.scholarpedia.org/article/Nelder-Mead_algorithm
- [27] FRALEY, Christina. Algorithms for nonlinear least-squares problems. [online] Stanford University, Dept. of Operations Research, Systems Optimization Laboratory, 1988 [cit. 2016-05-01]. Dostupné z: <http://www.dtic.mil/dtic/tr/fulltext/u2/a196071.pdf>
- [28] Reaction Velocity of an Enzymatic Reaction. R. [online] 2016 [cit. 2016-05-01]. Dostupné z: <https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/Puromycin.html>
- [29] ZVÁRA, Karel. Křivost v nelineární regresi. [online] 1992 [cit. 2016-05-01]. Dostupné z: http://www.statspol.cz/robust/1992_zvara_92.pdf
- [30] Zvára, Karel. Regrese. MATFYZPRESS. [online]. 2008 [cit. 2016-05-01]. Dostupné z: <http://www.karlin.mff.cuni.cz/~zvara/regrese/kniha/kniha08.html>
- [31] Michaelis–Menten kinetics. Wikipedia: the free encyclopedia. [online]. 2001- [cit. 2016-05-06]. Dostupné z: https://en.wikipedia.org/wiki/Michaelis\T1\textendashMenten_kinetics

Příloha A

Obecný matematický základ

Informace v této příloze byly čerpány z [3, 6].

Gradient

Vektor g je gradientem funkce $S(\beta)$, jestliže pro jeho složky platí

$$g_k = \frac{\partial S(\beta)}{\partial \beta_k}, \quad k = 1, \dots, p \quad (\text{A.1})$$

Hessova matice

Hessova matice je čtvercová (rozměru $p \times p$) symetrická matice druhých derivací funkce $S(\beta)$ podle parametrů. Její složky vypadají následovně

$$H_{ij} = \frac{\partial^2 S(\beta)}{\partial \beta_i \partial \beta_j}, \quad i, j = 1, \dots, p \quad (\text{A.2})$$

Jakobiho matice

Jakobiho matice je matice o rozměrech $n \times p$, jejíž prvky jsou první derivací regresní funkce dle jednotlivých parametrů v jednotlivých bodech, neboli

$$J_{ij} = \frac{\partial f(x_i, \beta)}{\partial \beta_k}, \quad i = 1, \dots, n, \quad j = 1, \dots, p \quad (\text{A.3})$$

Pomocí Jakobiho matice lze vypočítat gradient jako

$$g = -2J^T e, \quad (\text{A.4})$$

kde e je vektor reziduí s prvky $e_i = Y_i - f(x_i, \beta)$, nebo Hessovu matici jako

$$H = 2[J^T J + B] \quad (\text{A.5})$$

$$B_{jk} = \sum_{i=1}^n e_i \frac{\partial^2 f(x_i, \beta)}{\partial \beta_j \partial \beta_k}, \quad j, k = 1, \dots, p \quad (\text{A.6})$$

Taylorův rozvoj

Užitím Taylorova rozvoje lze aproximovat funkci $f(x, \beta)$ v okolí bodu β_0 vztahem

$$f(x_i, \beta) = f(x_i, \beta_0) + J_i(\beta - \beta_0) + \frac{1}{2}(\beta - \beta_0)^T G(\beta - \beta_0), \quad (\text{A.7})$$

kde

$$G_{jk} = \frac{\partial^2 f(x_i, \beta)}{\partial \beta_j \partial \beta_k}, \quad j, k = 1, \dots, p \quad (\text{A.8})$$

a J_i označuje i -tý řádek Jakobiho matice.

Intervalový odhad střední hodnoty μ při neznámém rozptylu

Oboustranný, $100(1 - \alpha)\%$ intervalový odhad je popsán vztahem

$$1 - \alpha = P \left(\bar{x} - t_{1-\frac{\alpha}{2}}(n-1) \frac{s}{\sqrt{n}} < \mu < \bar{x} + t_{1-\frac{\alpha}{2}}(n-1) \frac{s}{\sqrt{n}} \right), \quad (\text{A.9})$$

kde α je hladina významnosti neboli chyba 1. druhu, $t_{1-\frac{\alpha}{2}}(n-1)$ je kvantil studentova rozdělení při $n-1$ stupních volnosti a s je výběrová směrodatná odchylka daná vztahem

$$s = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}. \quad (\text{A.10})$$

Příloha B

Datové sety

	conc	rate	state
1	0.02	76	treated
2	0.02	47	treated
3	0.06	97	treated
4	0.06	107	treated
5	0.11	123	treated
6	0.11	139	treated
7	0.22	159	treated
8	0.22	152	treated
9	0.56	191	treated
10	0.56	201	treated
11	1.1	207	treated
12	1.1	200	treated
13	0.02	67	untreated
14	0.02	51	untreated
15	0.06	84	untreated
16	0.06	86	untreated
17	0.11	98	untreated
18	0.11	115	untreated
19	0.22	131	untreated
20	0.22	124	untreated
21	0.56	144	untreated
22	0.56	158	untreated
23	1.1	160	untreated

Tabulka B.1: Puromycin data

	x	y
1	178	107
2	160	98
3	160	106
4	123	93
5	151	106
6	104	73
7	109	94
8	97	56
9	85	50
10	86	47
11	77	66
12	76	70
13	72	58
14	70	43
15	54	48
16	54	61
17	54	37
18	48	42
19	47	36
20	42	35
21	35	16
22	35	27
23	33	31
24	22	16
25	14	9
26	11	10
27	10	5
28	9	8
29	8	6
30	6	4
31	0.6	0.3
32	0.4	0.2

Tabulka B.2: Farmakologická data

x	y
0.0535510423271917	60.56485465074
1.12526201475323	96.6226267177824
0.161190590578201	82.0380083429913
0.697417227265518	94.8701059075091
0.653938961947942	94.5952452946618
0.163491646970506	82.2325806556266
0.335122427981975	90.2893047942098
0.749105725157214	95.1916945378413
1.00498297371194	96.2879896333961
1.07198201409553	96.4956214807122
0.0295032995756948	45.8941973704057
0.391225872389111	91.5230879097766
0.648990728598065	94.5638890834673
0.363335685613309	90.9543351457637
0.957898492814554	96.1253011736366
0.740420179938292	95.1333874985666
0.270903301326325	88.3289663653044
1.17047317515663	96.7314551618535
1.01586071129683	96.3122687896828
1.12634481862497	96.6264702307599
0.463708311464358	92.697036514389
0.71222860252785	94.9808358583008
0.875467910766485	95.8065643369571
0.0566598864902277	61.9092377122864
0.0619415702588856	63.9632871529835
0.758080827029142	95.2443489383585
0.627589699882269	94.3724322266751
0.0985181065698853	73.7506786917015
0.948352471661055	96.1020775332927
1.08112880773905	96.5115888744178

Tabulka B.3: Generovaná data 1

x	y
70.7799895785283	70.7398907220059
221.732163762441	93.5200292062403
51.4515606698114	59.8762917529422
164.293312479742	91.3406330004776
1.5539729250595	4.21786099298084
36.3720979695208	47.5587657721726
207.636354209855	92.1632885910984
7.10062616760843	11.3626652004336
49.3228839272633	55.2448419592428
178.544077274622	92.2009874460078
82.5916270527523	76.5850089272326
5.3335667331703	7.88273248832999
147.82906878111	91.105125676149
226.518132735509	91.272621583698
5.31898519373499	9.35424300705434
218.683349125087	92.4601881371577
26.0358180638868	35.7258724321884
213.558662090218	94.0228905846492
206.589685262414	93.5619340124641
178.664747071918	92.3078697847156
248.575675993226	92.9577447738726
74.1923863489646	73.0805710318815
94.0160495920572	81.2825668689538
6.88112029992044	11.2625420610614
22.6324306980241	33.7774569237818
165.803251686739	91.726359384545
106.234216315905	83.2071612278314
78.2547705958132	73.9512952412059
174.488717837958	92.4444242135173
190.703113266034	95.2417458550582

Tabulka B.4: Generovaná data 2

Příloha C

Přílohy na CD

- Diplomová práce
- Datové sady
- Zdrojové kódy (R)
- Kompletní výsledky testování
- Poster